IBM

z/OS Parallel Sysplex Configuration Overview

- An update of the Parallel Sysplex Configuration Volume 1
- High-level design concepts for Parallel Sysplex
- The workloads in Parallel Sysplex

Pierre Cassier, Keith George
Frank Kyne, Bruno Lahousse
Sylvie Lemariey, Christian Matthys
Masaya Nakagawa, Jean-Jacques Noguera
Dominique Richard, Philippe Richard
Pascal Tillard, Steve Wall

Redbooks

ibm.com/redbooks

International Technical Support Organization

z/OS Parallel Sysplex Configuration Overview

September 2006

SG24-6485-00

Note: Before using this information and the product it supports, read the information in “Notices” on page vii.

First Edition (z/OS Parallel Sysplex)

This edition applies to IBM Parallel Sysplex technology used with operating systems z/OS (program number 5694-A01) or OS/390 (program number 5647-A01.)

Contents

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:
This information contains sample application programs in source language, which illustrates programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. You may copy, modify, and distribute these sample programs in any form without payment to IBM for the purposes of developing, using, marketing, or distributing application programs conforming to IBM's application programming interfaces.

Trademarks

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

Advanced Function Printing™
Advanced Peer-to-Peer Networking®
AFP™
AIX 5L™
AIX®
BatchPipes®
CICS/ESA®
CICS®
CICSPlex®
CUA®
DataJoiner®
DataPropagator™
Domino®
DB2 Connect™
DB2®
DFSMShsm™
DRDA®
DS6000™
DS8000™
Enterprise Storage Server®
Extended Services®
ESCON®
eServer™
FlashCopy®
FICON®
Geographically Dispersed Parallel Sysplex™
GDPS®
Hiperspace™
HyperSwap™
Informix®
IBM®
IMS™
IMS/ESA®
Language Environment®
Lotus®
MQSeries®
MVS™
MVS/ESA™
MVS/SP™
Net.Commerce™
Net.Data®
NetView®
OS/2®
OS/390®
Parallel Sysplex®
Processor Resource/Systems Manager™
PR/SM™
PROFS®
QMF™
Rational®
Redbooks™
Redbooks (logo) ™
Resource Link™
RACF®
RAMAC®
RETAIN®
RMF™
S/390®
Sysplex Timer®
System z™
System z9™
SAA®
SNAP/SHOT®
SQL/DS™
Tivoli Management Environment®
Tivoli®
TotalStorage®
TME®
Virtualization Engine™
VisualAge®
VM/ESA®
VSE/ESA™
VTAM®
WebSphere®
z/Architecture™
z/OS®
z/VM®
zSeries®
z9™

The following terms are trademarks of other companies:

Enterprise JavaBeans, EJB, IPX, Java, Java Naming and Directory Interface, JavaBeans, JDBC, JVM, J2EE, ONC, RSM, Solaris, Sun, Sun Microsystems, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, and service names may be trademarks or service marks of others.

Preface

This IBM® Redbook is an update to the first volume of a previous set of three Parallel Sysplex® Configuration Redbooks™. Most of the information in the second and third volumes is either available elsewhere, or else it is no longer required.

This redbook will provide you with the information you require to understand what *is* a Parallel Sysplex. With an understanding of the basics, it then goes on to describe how these components are used to enable the two fundamental capabilities of Parallel Sysplex: namely, the ability to concurrently update a database from two or more database managers (thereby removing single points of failure), and dynamic workload balancing.

The redbook then discusses how the Parallel Sysplex-exploiting products enable you to start delivering continuous application availability, a growing requirement for most IBM System z™ clients. Typical client workloads are discussed, and their exploitation of Parallel Sysplex is described in clear terms.

This redbook is an excellent starting point for those involved in designing and configuring a Parallel Sysplex. It should also be used by those that already have a Parallel Sysplex when further exploitation is being considered. The redbook refers throughout to other relevant publications that cover specific areas in more detail.

This book is an update to the following set of books:

- *OS/390 Parallel Sysplex Configuration Volume 1: Overview*, SG24-5637
- *OS/390 Parallel Sysplex Configuration Volume 2: Cookbook*, SG24-5638
- *OS/390 Parallel Sysplex Configuration Volume 3: Connectivity*, SG24-5639

The team that wrote this redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization (ITSO), Poughkeepsie Center (NY, USA) and in the Products and Solutions Support Center (PSSC), in Montpellier, France. The primary authors of this version were:

Keith George	IBM UK
Masaya Nakagawa	IBM Japan

The following people were part of the project:

Pierre Cassier	PSSC**,** IBM France
Frank Kyne	ITSO, Poughkeepsie, NY, IBM USA
Bruno Lahousse	PSSC, IBM France
Sylvie Lemariey	PSSC**,** IBM France
Jean-Jacques Noguera	PSSC**,** IBM France
Noel Richard	PSSC**,** IBM France
Philippe Richard	PSSC**,** IBM France
Pascal Tillard	PSSC**,** IBM France

Steve Wall PSSC, IBM France

The project was managed by

Christian Matthys ITSO project leader in the PSSC, IBM France

Also, thanks to the following people for reviewing these publications, providing material and offering invaluable advice and guidance. Their contributions have made this book eminently more accurate and more readable than it otherwise might have been.

Shoichi Ashigai IBM Japan

Gary King IBM USA

David Raften IBM USA

David H Surman IBM USA

Ray Whiffin Software AG

Steve Zehner IBM USA

To everyone else who contributed, our sincere thanks.

Comments welcome

Your comments are important to us!

We want our redbooks to be as helpful as possible. Please send us your comments about this or other redbooks in one of the following ways:

- Fax the evaluation form found in IBM Redbooks evaluation on page --- to the fax number shown on the form.
- Use the online evaluation form found at `http://www.redbooks.ibm.com/`
- Send your comments in an Internet note to `redbook@us.ibm.com`

Become a published author

Join us for a two- to six-week residency program! Help write an IBM Redbook dealing with specific products or solutions, while getting hands-on experience with leading-edge technologies. You'll team with IBM technical professionals, Business Partners, and clients.

Your efforts will help increase product acceptance and client satisfaction. As a bonus, you'll develop a network of contacts in IBM development labs, and increase your productivity and marketability.

Find out more about the residency program, browse the residency index, and apply online at:

`ibm.com/redbooks/residencies.html`

Comments welcome

Your comments are important to us!

We want our Redbooks to be as helpful as possible. Send us your comments about this or other Redbooks in one of the following ways:

- Use the online **Contact us** review redbook form found at:

  `ibm.com/redbooks`

- Send your comments in an e-mail to:

  `redbook@us.ibm.com`

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYJ Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

1

Introduction to the configuration of a Parallel Sysplex

This redbook is designed as a cookbook. It contains a series of *recipes* you may use to design your new Parallel Sysplex or enhance your existing Parallel Sysplex. Which *recipes* you use and in what order you use them depends on what you want to achieve: The book is *not* intended to be read sequentially. It should not be your only source of information: it is intended to consolidate other sources of information at a suitable level of detail.

For a list of the sources used, refer to:

- *Recommended Sources of further information* at the beginning of each chapter.
- The IBM Techdocs Web site at:

  `http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs`

  We refer many times to documents available from this excellent site, which should be one of your favorites. It covers all IBM platforms.
- The IBM Publications Web site. Most IBM publications can be downloaded from here:

  `http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi`
- The IBM Redbooks Web site. We refer to many other Redbooks, all of which can be downloaded from here.
- The packages and books listed in “Related publications” on page 301.

Recommended sources of further information: The following sources provide support for the information in this chapter:

- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS Parallel Sysplex Overview: An Introduction to Data Sharing & Parallelism*, SA22-7661
- *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061
- *ABCs of z/OS System Programming Volume 5*, SG24-6985
- The IBM Parallel Sysplex Home Page:

  http://www.ibm.com/servers/eserver/zseries/pso
- The Parallel Sysplex Education and Training, available on the following Web site:

  http://www.ibm.com/servers/eserver/zseries/pso/education.html

Welcome to the System z Parallel Sysplex Home Page!:

There is an IBM Web site dedicated to Parallel Sysplex. The site is intended to provide up-to-date Parallel Sysplex-related information, and to act as a hub from which you can be directed to other relevant sites. To make the use of this site easier, you should add the address of the following Web site to your Favorites in your Web browser:

`http://www.ibm.com/servers/eserver/zseries/pso/`

On this Home page you will find information about:

- News and events related to Parallel Sysplex.
- IBM Parallel Sysplex-related service offerings. This includes information about the Geographically Dispersed Parallel Sysplex™ offering.
- New tools, such as the *IBM Health checker for z/OS® and Sysplex*.

At the time of writing, this Web site was organized into the following sections:

- White papers pages. All white papers of interest to the Parallel Sysplex are available in PDF format.
- Coupling Facility pages. These are links for more information related to CF configurations, CFLEVEL, and structure Duplexing.
- Availability pages. These are short documents that discuss how z/OS features can be used to improve the availability of your Parallel Sysplex.
- Tools and Wizards pages, to get the latest news about some z/OS Features or additional tools. For example, two Web-based assistants and two tools are available to ease your migration to a Parallel Sysplex environment:
  - The IBM eServer™ zSeries® Parallel Sysplex Customization Wizard is for system programmers planning to migrate to a Parallel Sysplex environment for the first time, as well as for those who want to verify their Parallel Sysplex setup.
  - The IBM Coupling Facility Structure Sizer simplifies the task of estimating the amount of storage required by the coupling facility structures used in your installation.
  - To help in testing recovery from coupling facility problems, the INJERROR tool injects an error into a CF structure to simulate damage to the structure.
  - XISOLATE helps you maintain performance critical data sets on isolated DASD subsystems. With XISOLATE, you can simplify the task of ensuring that this critical SMS-managed data resides on DASD subsystems that are physically separated from others in the sysplex.
- Product pages. Announcement information for latest IBM eServer zSeries processors and major software products.
- Education pages. These contain information about Parallel Sysplex education offerings (workshops, extract of IBM course training, and so on) and tools like *Parallel Sysplex Trainer Environment (PSTE)*.

1.1 Terminology

To avoid confusion, you should read the following list of terms as used throughout this book. The terms that we use may be different than you have seen elsewhere, but they are used consistently in this book:

- CF rather than Coupling Facility partition.
- CF link rather than coupling link, ISC link, or Coupling Facility channel.
- CFCC rather than Coupling Facility Control Code.
- CP rather than CPU, engine, or processor.
- CPC rather than CEC, processor, model, computer, server, or machine.
- DASD rather than disk.
- External link identifies any CF link that connects between different CPCs. This includes ISC links and ICB (copper) links.
- Failure-Independent CF identifies any CF that resides in a CPC that does not contain any other images (CF or z/OS) from the same sysplex.
- Fiber link rather than ISC or real link. Contrast fiber links to ICs (internal) and ICBs (copper) CF links.
- GBP rather than group buffer pool.
- IC rather than Internal Coupling Link.
- ICB rather than Integrated Cluster Bus or Cluster Bus Link.
- ICF rather than Internal Coupling Facility.
- Image rather than z/OS image.
- LP rather than PR/SM™ LPAR partition, logical partition, or LPAR.
- MVS™, OS/390®, and z/OS all mean z/OS.
- Parallel Sysplex rather than Parallel Sysplex cluster.
- RACF® and z/OS Security Server are used in the same context.
- Standalone CF rather than external CF or 2064 Model 100, 2066 Model OCF, or 2084 Model 300.
- zSeries Business Partners, rather than solution developers, business partners, or independent software vendors (ISVs).
- Storage rather than memory or RAM.
- STP is Server Time Protocol.
- Sysplex rather than complex or systems complex.
- z/OS Communications Server, and Communications Server for z/OS are used synonymously.
- VTAM® is the SNA component of the z/OS Communications Server.

For more information about the terms and acronyms used in this book, refer to “Glossary” on page 279 and “Abbreviations and acronyms” on page 269.

1.2 The purpose of this book

This book has been written to help you configure a Parallel Sysplex. The emphasis in this book is on *high level configuration design*. By this, we mean that you:

- Order the right hardware and software.
- Decide how it will work. For example:
  - Will all subsystems run on every image?
  - How will my subsystems work in normal operation?
  - How will I operate and manage the subsystems/sysplex?
  - What happens if a component fails?
  - How will systems management functions work across the sysplex?

The book is not designed to help you justify the use of a Parallel Sysplex, nor to implement it (install it and make it work). It is designed with two purposes in mind:

- For new users of Parallel Sysplex.

  To help you make the *initial design* decisions so that implementation goes smoothly and nothing is overlooked.
- For existing users of Parallel Sysplex.

  To provide information regarding enhancing your Parallel Sysplex by utilizing the latest functions and facilities of new hardware and software.

Initially, you will probably only be interested in approximate sizings. Later, you will need to be sure that the hardware and software ordered is both correct and complete. If you have not thoroughly considered the operational issues listed in this redbook, you may find that you have to make alterations at a later date.

You can expect to make several iterations through this redbook, at different levels of detail, as your Parallel Sysplex evolves from an idea to a firm decision.

This redbook brings together new information and information that is already available, but scattered among many different sources. It contains the latest information based on the experience at the time of writing. As Parallel Sysplex continues to evolve, you should always check the latest information.

This redbook contains information about the environments that are capable of being configured now. These are primarily the DB2®, IMS™ TM, IMS DB, CICS®, CICS/VSAM RLS, and WebSphere® Application Server transaction processing and data sharing environments. The book also contains information about other exploiters, including such z/OS functions as JES, RACF, VTAM, TCP/IP, Sysplex Distributor, BatchPipes®, system logger, shared tape, Enhanced Catalog Sharing, and Global Resource Serialization (GRS).

Recommendation to check background information: The content of this redbook is based on many sources. For a deeper understanding of the background information, always check the detailed information.

1.3 Base hardware levels for this redbook

The minimum hardware level required for all the facilities described in this redbook is the IBM 2064 CPC (z900). Many, but not all, of the facilities will work on previous levels of CPCs. Mixed levels of hardware may cause some functions of a Parallel Sysplex to work differently.

Information about the older levels of hardware (9021s, 9121s, 9674s, and 9672s) was largely removed to make the document more readable. Many clients have moved to 2064 and later generations of zSeries CPCs, so it was felt that the old information was no longer required.

More information regarding facilities available on previous levels of CPCs can be found in Table 2 in Chapter 1, “Introduction to the Configuration of a Parallel Sysplex”, of *OS/390 Parallel Sysplex Configuration, Volume 1: Overview*, SG24-5637. For those clients that are still using these older CPC levels, all three volumes of the previous level of this redbook are still orderable. The names and order numbers are:

- *OS/390 Parallel Sysplex Configuration, Volume 1: Overview*, SG24-5637
- *OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook*, SG24-5638
- *OS/390 Parallel Sysplex Configuration, Volume 3: Connectivity*, SG24-5639

9672 considerations

This redbook is aimed at IBM eServer zSeries and IBM System z9™ 109 clients. There are significant limitations on 9672 in these environments, in particular:

- A z990 LPAR cannot coexist with a G4 LPAR in a Parallel Sysplex (whether running CFCC or the operating system).
- A z9 LPAR cannot use a G5 or G6 CF.
- The highest level of z/OS that supports G5/6 9672 is z/OS 1.5, which goes out of support in March 2007.

1.4 Main reasons for Parallel Sysplex

The main reasons for moving to a Parallel Sysplex are briefly discussed here. They are covered more thoroughly in *z/OS Parallel Sysplex Overview: An Introduction to Data Sharing & Parallelism*, SA22-7661.

For a discussion of the key design points for Parallel Sysplex, refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

1.4.1 Continuous application availability

When you have a single copy of any system component, hardware, software, or data, you are inevitably exposed to system outages because of either failure of the component or because of planned changes to the component that require it to be taken offline.

One of the goals of Parallel Sysplex is to eliminate the impact that scheduled outages have on application availability, and minimize the effects of an unscheduled outage by allowing work to continue executing on the remaining systems in the sysplex. This requires, among other things, that the system be designed for redundancy within the Parallel Sysplex. Applications must be capable of running across multiple systems, with access to the data being possible from at least two systems. If at least two instances of a resource exist, your applications can continue to run even if one of the resources fails.

It is important to remember that Parallel Sysplex only provides the facilities for continuous availability. Parallel Sysplex on its own will not eliminate scheduled or unscheduled application outages; the application itself must also be designed for continuous availability. Sharing data is only one part of this design.

A classic example is where IMS databases are made unavailable to online users to allow for batch updates. These batch update programs could be re-written as Batch Message Processing (BMP) programs that can perform updates while still having the databases available for online users.

Another example is that data in a DB2 database may be unavailable while it is being reorganized (and this will occur no matter how many data sharing members you have). DB2 for z/OS reduces the impact of database reorgs by the provision of the online reorg utility. This utility allows full access to the data for *most* of the time that the reorg is taking place. The latest versions of DB2 allow some concurrent schema changes (see *DB2 UDB for z/OS: Design Guidelines for High Performance and Availability*, SG24-7134).

Similar considerations apply to other IBM subsystems.

More discussion on continuous application availability is found in Chapter 3, “Continuous availability in Parallel Sysplex” on page 99 and on the *Availability Pages* via the Parallel Sysplex home page, or directly from:

`http://www.ibm.com/servers/eserver/zseries/pso/availability.html`

1.4.2 Workload balancing

Without workload balancing, installations with multiple CPCs have often had to upgrade one CPC to provide more capacity, while another CPC may have had spare capacity. The alternative was to redistribute work manually, which is time-consuming and can only handle short-term imbalances. Manual intervention also has the potential to introduce errors and could not be used to dynamically and constantly balance workload across the installed CPCs.

With a Parallel Sysplex, the basic framework exists for workload balancing. There is a small cost for doing workload balancing in a Parallel Sysplex, but the cost by and large is not related to the number of systems, and the workload balancing advantages are significant. Many subsystems exploit this framework to redistribute work across systems, thus allowing the systems to be run at higher utilizations, and allowing spare capacity anywhere in the Parallel Sysplex to be used to satisfy the demands of the workload.

All the major IBM subsystems (CICS, DB2, IMS, MQSeries®, TSO, VTAM, TCP/IP, and JES) contain support for Parallel Sysplex workload balancing. More discussion on subsystem exploitation of workload balancing can be found in Chapter 4, “Workloads in Parallel Sysplex” on page 157.

1.4.3 Nondisruptive addition of scalable CPC capacity

If a part of the workload cannot be split between multiple images, Parallel Sysplex may offer a solution to add additional capacity: for example, it may allow multiple instances of the application to share the data across multiple images. Some of the capabilities are discussed here.

Just-In-Time nondisruptive growth

With a Parallel Sysplex, multiple options exist for providing more capacity. One option is to upgrade or replace existing CPCs (vertical growth). Parallel Sysplex and the cloning facilities of z/OS allow new CPCs to be added alongside existing CPCs (horizontal growth). There is also the hybrid option of both vertical and horizontal growth.

Which option you choose will depend on several factors, including:

- Availability requirements
- Whether there is an upgrade path
- Whether there is a larger CPC available
- CP speed considerations
- Restrictions imposed by existing configuration
- Cost - both hardware and software

Vertical growth

This is the traditional option. Vertical upgrades within a processor range have normally been nondisruptive at the application level even on a single processor since the introduction of the Capacity Upgrade on Demand (CUoD) feature several years ago on G5, and since then it has become the norm to define reserved processors in LPAR image profiles. An LPAR can then be given an extra LP by varying one of the reserved processors online.

With z990 and later processors additional options have become available such as Customer Initiated Upgrade (CIU) and On-off Capacity on Demand (OOCoD). More information about these options is available at the IBM eServer zSeries Web site:

`http://www.ibm.com/systems/z/`

Sysplex also allows nondisruptive vertical growth between processor ranges. The CPC being upgraded is taken out of the Parallel Sysplex and upgraded (for example, from a z990 to a z9 109), while continuing to run the workload on the remaining CPCs in the Parallel Sysplex. The upgraded CPC can then be reintroduced to the Parallel Sysplex when the upgrade is complete.

Horizontal growth

At times it will be more suitable to add a new processor rather than upgrading an existing one. The additional CPC can probably be installed during normal operation, without time pressure or risk to the existing service. Connection to the existing Parallel Sysplex can often be achieved *nondisruptively*, though there are some situations in which nondisruptive connection is not possible. Testing can then be performed as required. When the new CPC has been proven to everyone's satisfaction, work can then be gradually migrated onto it, with less critical work migrated first to allow further validation.

Adding additional capacity by horizontal growth in a Parallel Sysplex will not immediately benefit a single-threaded application if it is unable to share data between multiple instances. However, other work in the system can be moved or directed away from the system running the single-threaded application.

Scalability

With Parallel Sysplex, it is possible to have an application that can run without modification on the smallest IBM z890 CPC or on a Parallel Sysplex of multiple IBM z9 109s. The difference in capacity is many orders of magnitude. Careful design of the application would be required to ensure that an application could scale over such a range of capacity. Nevertheless, the Parallel Sysplex infrastructure of hardware and software can and does scale over that range and clients have found good scalability of applications.

Scalability in a Parallel Sysplex is not subject to the same *drop off* in benefit from adding more images as a tightly coupled multiprocessing (MP) CPC is when more CPs are added. As more images are added to the Parallel Sysplex, you achieve *almost linear growth*. This is shown graphically in Figure 1-1.

Over time, other constraints have emerged in application architectures when running a single large instance, for example, constraints on virtual storage or on logging capacity of a single DB2 instance, or some unforeseen application related constraint. Parallel Sysplex offers a solution to many of these sorts of problems without rewriting the applications or re-partitioning the databases or waiting for a release of DB2 (for example) that fixes the problem. It does this by providing several application instances, each of which has its own less constrained copy of the constrained resource.

Some information about performance and scalability in a Parallel Sysplex is found in *System/390 Parallel Sysplex Performance*, SG24-4356.

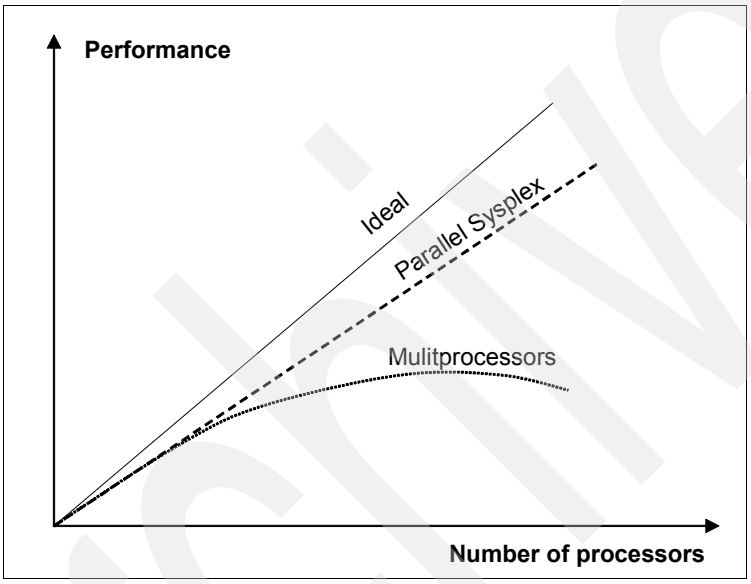


Figure 1-1 Scalability in Parallel Sysplex

1.4.4 Reduced total cost of computing

Implementing a Parallel Sysplex may be sometimes less expensive than upgrading a system. Some cost aspects of the Parallel Sysplex are discussed in this section.

Application portfolio investment protection

Existing applications can often exploit the availability benefits of the Parallel Sysplex environment with few or no changes. The most significant common change that may be required is that you may want to alter the frequency of commits or checkpoints to reduce the contention for shared resources.

The majority of changes needed for applications to exploit a data sharing or dynamic workload balancing environment are implemented in the IBM subsystems, such as CICS, DB2, IMS, MQSeries, and VTAM.

In essence, this means that Parallel Sysplex provides compatibility such that your existing applications will continue to run.

Software pricing in Parallel Sysplex

The ability to aggregate the MSUs of the CPCs in your Parallel Sysplex provides the possibility of reduced software license charges in a Parallel Sysplex. Options such as Workload License Charges (WLC) with subcapacity pricing and Select Application Licence Charge should be investigated to calculate the benefit.

For information about IBM software licenses in the Parallel Sysplex, refer to the IBM eServer zSeries sysplex software pricing Web site:

http://www.ibm.com/servers/eserver/zseries/swprice/sysplex

There are a number of specialist companies that offer contract review and negotiation services. Your IBM representative can provide a list of some these companies.

Single system image

Parallel Sysplex can potentially provide a logical single system image to users, applications, and the network. In addition, Parallel Sysplex provides the ability to have a single point of control for your systems operations staff.

Single systems image is discussed further in Chapter 2, “High-level design concepts for Parallel Sysplex” on page 13.

1.5 Value of Parallel Sysplex: a summary

The value of Parallel Sysplex is best described by its ability to deliver the functions that meet the ever-increasing requirements of today's businesses. These requirements include:

- Continuous availability of applications
- Reduction or elimination of planned application outages
- Scalability to virtually unlimited capacity to meet the high transaction volumes and response times of today and tomorrow
- Investment protection of existing applications by providing functions that allow these applications to operate in the e-business environment without a complete rewrite.
- A secure environment for existing and e-business transactions
- A development environment that provides the tools and languages to develop new applications for today and tomorrow
- A platform for server consolidation, to reduce the cost and complexity of having a large server farm to manage
- A simple growth path delivered with low incremental and total cost of computing

Parallel Sysplex is ideally suited to today's environment and is continually being developed to meet new and changing requirements.

1.6 The distinction between Base and Parallel Sysplex

Parallel Sysplex evolved from Base Sysplex. A general description of each follows.

Note: Because of the benefits of aggregated software charging and the rules that have to be followed in order to qualify for it, in most cases we expect that Base Sysplexes are also a Parallel Sysplex.

1.6.1 Base Sysplex

In September 1990, IBM introduced the SYStems comPLEX, or sysplex, to help solve the difficulties of managing multiple z/OS systems. This established the groundwork for simplified multisystem management through the Cross-System Coupling Facility (XCF) component of z/OS. XCF services allow authorized applications on one system to communicate with applications on the same system or on other systems. In a Base Sysplex, connectivity and communication between images is provided by channel-to-channel (CTC) links. The couple data set, which is shared between all of the images, holds control information and provides a mechanism for monitoring the status of the images. When more than one CPC is involved, a Sysplex Timer® synchronizes the time on all systems.

XCF communication is done only through dedicated XCF CTCs.

Base Sysplex definition: A Base Sysplex is that set of systems that share a sysplex couple dataset and all of which have the same sysplex name.

1.6.2 Parallel Sysplex

Base Sysplex laid the foundations for communications between subsystems on the participating z/OS images, but these were insufficient to provide the speed and integrity necessary for data sharing. To provide this capability, the Coupling Facility (CF) was introduced and was implemented in a Logical Partition (LP) of a CPC.

The use of the CF by subsystems, such as IMS, DB2, and CICS/VSAM RLS, ensures the integrity and consistency of data throughout the sysplex. The capability of linking many systems and providing multisystem data sharing makes the sysplex platform ideal for parallel processing, particularly for online transaction processing (OLTP) and decision support.

Parallel Sysplex definition: A Parallel Sysplex is that set of systems within a sysplex that all have access to the same one or more CFs. While a basic sysplex is an actual entity, with a defined name (the sysplex name), a Parallel Sysplex is more conceptual. There is no member or couple dataset anywhere that contains a name for the Parallel Sysplex, or a list of the systems it contains. Rather, it is the super-set of a number of other plexes (RACFplex, VTAMplex, and so on) that all share the same CF or set of CFs. For more information, refer to 2.2.8, “What different 'plexes are there?” on page 28.

In short, a Parallel Sysplex builds on the Base Sysplex capability, and allows you to increase the number of CPCs and images that can directly share work. The CF allows high performance and multisystem data sharing across all the systems. In addition, workloads can be dynamically balanced across systems with the help of workload management functions. All major IBM subsystems exploit Parallel Sysplex and covering that form is the purpose of the rest of the redbook.

2

High-level design concepts for Parallel Sysplex

In this chapter, we look into some of the considerations that have to be taken into account when designing your Parallel Sysplex configuration.

Recommended sources of further information: The following sources provide support for the information in this chapter:

- *Batch Processing in a Parallel Sysplex*, SG24-5329
- *CICS Transaction Server for z/OS V3.1 Installation Guide,* GC34-6426
- *DB2 UDB for z/OS V8 Data Sharing: Planning and Administration,* SC18-7417
- *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908
- *IMS/ESA V6 Parallel Sysplex Migration Planning Guide for IMS TM and DBCTL*, SG24-5461
- *z/OS MVS Initialization and Tuning Reference,* SA22-7592
- *z/OS V1R7.0 MVS Planning: Global Resource Serialization,* SA22-7600
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS V1R1.0 Parallel Sysplex Application Migration,* SA22-7662
- *z/OS V1R5.0 Parallel Sysplex Test Report,* SA22-7663
- *S/390 Parallel Sysplex: Resource Sharing*, SG24-5666
- *SNA in a Parallel Sysplex Environment*, SG24-2113
- *TCP/IP in a Sysplex*, SG24-5235
- *Using VTAM Generic Resources with IMS*, SG24-5487

2.1 Deciding if Parallel Sysplex is right for you

When Parallel Sysplex was first announced, it was primarily seen as being of interest only to very large clients. Also, it was at a time when availability requirements were not as challenging as they are today. However, as time has passed, requirements have changed, and Parallel Sysplex technology has seen many improvements. As a result, Parallel Sysplex is now a technology that applies to all clients. In this section, we discuss the things to consider when deciding if Parallel Sysplex is the right technology for your company:

- Do you have application availability requirements that are proving difficult to achieve? One survey found that outage costs across 10 industries ranged from $25,000 (US) an hour up to $6,500,000 (US) an hour–and this was before the advent of e-business! If you plan to use Parallel Sysplex to improve your availability, you need to review all of your planned and unplanned outages over the past year, and identify which ones could have been avoided in a Parallel Sysplex environment. Are you already configured for maximum availability? There is less benefit in moving to Parallel Sysplex if you are going to leave other single points of failure in your configuration.
- Maybe you wish to move in order to get better utilization from your installed CPCs; in this case, you need to review your workloads to decide which ones you can split over the various systems, and then consider the following questions:
  - Will the splitting of these workloads have a significant enough effect to balance out your CPC utilizations?
  - Can the workloads be split?
  - Do most of the products required by these workloads support data-sharing and workload balancing?
- Or maybe your company is getting serious about e-business, and you want to know which is the best platform to host these new applications. There are a number of requirements if you wish to provide a successful e-business application:
  - The application must be available 24 hours a day, 7 days a week: remember that your clients could now be in any country in the world, and will expect your application to be available at any time that suits *them*.
  - You must be able to provide consistent, acceptable, response times, and react to abrupt changes in the rate of requests.
  - You want your clients to be able to see current data, not data as it existed twelve hours ago. This means the e-business application has to have access to the live data, not a point-in-time copy.
  - You need to be able to fully utilize all the installed MIPS, while at the same time being able to protect selected critical applications.
- Most e-business applications either access host-resident data, or front-end existing host-based applications. Parallel Sysplex data sharing helps ensure that the data and applications are continuously available, and the Parallel Sysplex workload balancing features help ensure that incoming requests get routed to the Web server that is most able to provide the required service levels.
- Perhaps your aim is for simplified operations and systems management. This is likely to be of interest to installations that already have a number of systems and are struggling with controlling them in a consistent manner.
- For some installations, the aim may be to reduce software costs. Depending on your current configuration, implementing a Parallel Sysplex could have a significant impact on your software licence costs, depending on your product mix and current licence agreements.

- The use of data mining applications has become common, but companies sometimes find that queries can run for unacceptably long times. Parallel Sysplex, together with WLM and DB2, can help reduce these run times by an order of magnitude, while still protecting the response and turnaround times of existing production work.

Whatever your reason for considering Parallel Sysplex, there will be some associated cost. It is said that there is no such thing as a *free lunch*. To get from where you are, to the point of having a Parallel Sysplex up and running, is going to require an investment in both time and hardware. If you have done sufficient investigation, you should by now have an idea of the savings and additional business value that a Parallel Sysplex brings you.

There is a white paper, *Value of Resource Sharing*, available on the Parallel Sysplex home page:

http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gf225115.pdf

and a redbook, *S/390 Parallel Sysplex: Resource Sharing*, SG24-5666, that discusses the use of Parallel Sysplex for resource sharing.

VM/ESA® can play an important role in Parallel Sysplex environments. For those installations that have not yet implemented a Parallel Sysplex, VM provides the ability to define and test a Parallel Sysplex configuration without requiring any specialized Parallel Sysplex hardware (CFs and CF links). For those clients that have already implemented Parallel Sysplex, there is still a potential role for VM. VM provides an ideal training environment for Parallel Sysplex, both for operators and system programmers, and allows destructive testing to be carried out without risk to the production environment.

Refer to Chapter 4, “Workloads in Parallel Sysplex” on page 157 for more information about subsystem exploitation of Parallel Sysplex. In particular, note that:

- CICS was the first IBM transaction subsystem to deliver support for Parallel Sysplex. If you currently use CICS with one or more of IMS/DB, DB2, or VSAM, then you are in a strong position to derive benefits from Parallel Sysplex, possibly with few or no application changes. Each release of CICS since CICS/ESA® 4.1 has provided additional support for Parallel Sysplex, and reduced the number of issues that can constrain your ability to do workload balancing.
- DB2 supports VTAM generic resources and database sharing. DB2 and Parallel Sysplex together provide the ability to manage very large databases: this is sometimes called a *Teraplex*. You can find more information about DB2 Parallel Sysplex support in general in *DB2 UDB for z/OS V8 Data Sharing: Planning and Administration,* SC18-7417, and in the redbook *DB2 UDB for z/OS Version 8 Performance Topics*, SG24-6465.
- IMS provides support for both VTAM generic resources and shared message queues, as well as support for sharing all of its database types (except MSDBs, which should be migrated to VSO DEDBs). See *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908.
- MQSeries for z/OS 5.2 and higher has the ability to place the message queues in a CF structure. Queue-sharing also leads to easy system management of MQSeries resources, and you can broadcast MQSeries commands to all queue managers in a queue sharing group.

2.1.1 S/390 partners in development software

Several products from S/390 Partners in Development[1] also support and exploit Parallel Sysplex. The following database managers have announced exploitation of Parallel Sysplex:

Adabas Cluster Services

In 1996, Software AG's leading DBMS product, Adabas, took advantage of the IBM Parallel Sysplex clustering technology to provide users with 24x7 support. Using the Parallel Sysplex technology, Adabas provides continuous availability of the DBMS by distributing the database load across multiple z/OS images. Subsequent to its initial release, the Parallel Sysplex technology has made significant technological advancements. Current support for Parallel Sysplex is provided by the add-on product ADABAS Cluster Services.

Foremost with the new Adaplex is its multi-update support–any of the 1 to 32 Adaplex nuclei (that is, instances), running in the 1 to 32 z/OS images, will have the capability to update the DBMS. This advancement alone greatly reduces networking traffic across the images by ensuring that all DBMS requests have an affinity to the Adaplex nucleus local to that user. Furthermore, with multi-update support, the possibility of an outage will be greatly reduced because:

- Loss of an update nucleus is no longer a single point of failure.
- z/OS ARM support all but eliminates down-time of a nucleus.
- Sysplex System Managed Rebuild functionality ensures continuous availability of the DBMS during planned outages.
- Adabas's enhanced online Recovery mechanism automates the recovery of a failed nucleus or user in a timely manner.

Each of the multiple instances of the Adabas nuclei will use the Coupling Facility structures to maintain integrity and consistency of the shared data. Adaplex will ensure serialization of the individual Adabas instances across each of the local buffer pools using an optimistic serialization protocol. The actual serialization is achieved via a buffer validity test, a buffer invalidate process, and Coupling Facility conditional write function–a unique feature of the Parallel Sysplex coupling technology. Additionally, the Sysplex XES services will be used for optimistic locking to resolve any block contention during concurrent transactions and DBMS activities.

Through closer integration with the Parallel Sysplex Technology, for example, use of XCF facilities for enhanced communication services, support of Dynamic Transaction Routing, and other strategic enhancements, it is anticipated that there will be further performance improvements.

As for the DBMS itself, the following are some of the features inherent of the DBMS itself:

- Parallel compression/decompression
- Parallel format buffer translation
- Parallel sorting, retrieval, and searching
- Command/Protection Logging
- Universal Encoding Support
- Online Utilities

Note: Software AG kindly provided more information you can find in Appendix A, “Software AG’s Adabas in a Parallel Sysplex environment” on page 263.

[1] More information about S/390 partners in development on: http://www.ibm.com/systems/z/solutions/isv/

For more information about Adabas Cluster Services refer to Software AG or see the Web site:

http://www.softwareag.com/Corporate/products/adabas/add_ons/cluster.asp

Datacom and IDMS

CA-Datacom and IDMS, from Computer Associate exploit the current mainframe architecture. Hardware and software capabilities are exploited.

For more information about Datacom, contact Computer Associates or refer to:

http://www3.ca.com/solutions/Solution.aspx?ID=2899

For more information about IDMS, contact Computer Associates or refer to:

http://www3.ca.com/solutions/SubSolution.aspx?ID=2903

S/390 Partners in Development information

If you are using a third-party database manager or transaction manager, check with the vendor for Parallel Sysplex support and SW license information.

You may also check the list of S/390 Partners in Development products that tolerate and exploit Parallel Sysplex under the *Applications by S/390 Technology* heading at the following Web site:

http://www.ibm.com/systems/z/solutions/isv/

2.2 Parallel Sysplex high-level design

Do not skip this step and dive straight into configuring and sizing the hardware. *There are important considerations here that will affect the configuration.* It will not take long to review this section.

2.2.1 Coupling facilities

Confusion has arisen in the past due to loose usage of terminology:

- A CF is an LPAR that runs CFCC, and the LPAR may have one or more engines that may be dedicated or shared.
- An ICF is a processor that can only run CFCC, and a given processor may have only ICFs or it may have normal MVS CPs (on which CFCC could run in an LPAR) and ICFs (and possibly other speciality engines, such as IFLs). The important point is that an ICF does not contribute to the software msu rating of the machine (because it cannot run z/OS).

However, when people say ICF, they frequently mean a CF LPAR running on ICF processors in a CPC that also runs z/OS, in contrast to a stand-alone CF, which is a CPC running only CF LPARs (normally on ICF engines).

It is fairly unusual for CFs to run on other than ICFs these days, because the alternative is to use normal MVS CPs. Hardware and software pricing terms and conditions (both IBM and ISV, but more particularly ISV software terms and conditions) usually make this unattractive.

All CFs within the same family of servers can run the same CFCC code and CFCC always runs within an LPAR. So when positioning CF configurations, one should be careful not to assume there is a fundamental functional difference, but rather one based on other considerations, such as how failure conditions are handled.

2.2.2 How many Parallel Sysplexes do I need?

The number of Parallel Sysplexes needed in an installation will vary. For most installations, however, two Parallel Sysplexes is likely to be the norm, with maybe a third monoplex systems programming sandpit. This should cover both the current users' *production* environments and systems programmers' *testing* environments.

For some installations, a need will exist for additional Parallel Sysplexes because of availability, technical, or business reasons. Application development will normally be part of the production sysplex. As in the past, when new releases of software are brought in, these new releases will be put on the development images first.

For companies offering Facilities Management (outsourcing providers) services, for example, there might be specific business reasons why they would not want to include all their workloads in the same Parallel Sysplex.

Some of the specific things to consider when deciding how many Parallel Sysplexes to have are:

- Ease of operation and management

  It is easier to operate and manage a single large Parallel Sysplex than many smaller ones. If improved availability is one of your goals, this should not be overlooked - more outages are caused by human error than by hardware failures.

  Assuming that you have a test sysplex, which we recommend, the test sysplex should be used as learning environment to get familiar with the aspects of managing a sysplex and also as a place to test commands and procedures before they are used on the production system.

- Cost

  In nearly all cases, the cost of hardware and software will increase as the number of Parallel Sysplexes increases. Remember that CF partitions cannot be shared between different sysplexes, and that each Parallel Sysplex will need its own dedicated links between z/OS images and CFs that are not in the same CPC. It is possible to share ICF CPs between several CF LPs; however, remember that each CF LP must have its own dedicated CPC storage.

- Protecting your production systems

  There is a certain amount of risk associated with putting development systems into the same Parallel Sysplex as your production systems. The amount of this risk will vary from installation to installation. If your developers have a talent for bringing the development system down every other day, then it may be wiser to isolate them from the production systems. On the other hand, if you have a properly configured test Parallel Sysplex, and have thorough test suites, then the risk should be negligible.

  If you have not implemented a thorough test suite, then multi-system outages or problems may be encountered in the production environment rather than on the test systems. Some installations would therefore prefer to thoroughly test new software releases that have sysplex-wide effects (such as JES, XCF, and so on) in a less availability-critical environment. In this situation, you may prefer to keep the development environment in a sysplex of its own.

  The system programmer test sysplex should always be kept separate from the production sysplex. The use of VM is one way of providing a self-contained environment where the configuration can be changed with just a few commands, and mistakes cannot impact a system outside the VM environment.

- The scope of work for batch, CICS, and IMS transactions, shared data, physical location and shared DASD

  If there are systems that share nothing with other systems, maybe for business or security reasons, it may make more sense to place those systems in separate Parallel Sysplexes.
- Applications with affinities to an LPAR (older, non data sharing) have a requirement to maintain service level objectives as though they were on a separate machine. Typically, these applications cannot survive the scheduling of rolling IPLs and maintenance that data sharing applications can. Therefore, some clients do not mix data sharing applications with those that cannot tolerate the associated operational flexibility that data sharing applications bring.
- Some "applications" have a need to run active in more than one sysplex or site to avoid outages that are sysplex or site-wide in scope. These applications have their own data synchronization processes.
- GDPS® requires the control image(s) to be in a separate sysplex.

All of these factors, and more, have to be considered when trying to decide which systems should be in or out of a given Parallel Sysplex.

Which environments should be Parallel Sysplexes?

Normally, a logical basis for the sysplex boundary is the JES2 MAS or the JES3 complex. In some instances, the CPCs that share DASD and tape devices become the logical basis for the sysplex. If you have complexes with different naming conventions, different job scheduler, or different security systems, there might be a need to map these into separate Parallel Sysplexes. Always evaluate the test environment and the software test/migration philosophy that is currently used.

2.2.3 Software coexistence considerations

A key feature of Parallel Sysplex is the ability to perform rolling upgrades, both for the operating system and for subsystems, in order to provide continuous application availability, even though an operating system instance or subsystem instance has been taken down. The rules are on a product by product basis, but in all cases provide N, N+1 coexistence between releases.

z/OS

To provide the maximum flexibility for introducing new levels of software into the Parallel Sysplex, IBM has provided coexistence support for up to four consecutive releases of z/OS. In any practical client situation, we would expect to see, at most, two different releases of z/OS in use in a single Parallel Sysplex and that only for a relatively short migration period; only IBM testing is likely to see four consecutive releases. For example, it is possible for z/OS 1.4 and z/OS 1.7 to coexist in the same Parallel Sysplex.

IBM's current policy is to provide maintenance (service) for each release of z/OS and z/OS.e for three years following their general availability (GA) date. However, service on a particular release might be extended beyond the normal three-year period (as in the case of z/OS 1.4). Prior to withdrawing service for any version or release of z/OS or z/OS.e, IBM intends to provide at least 12 months notice.

From z/OS 1.6 onwards, z/OS is on an annual release cycle with releases in September each year, so release N will be going out of service just as release N+3 becomes available. Starting with z/OS 1.6 and z/OSe 1.6, the coexistence, fallback, and migration policy are aligned with the service policy, that is, supported releases can coexist in a sysplex. Clearly then, with an annual release cycle and three year support for a release, future clients will need to plan to

move from Release n to Release n+1 or n+2 of z/OS unless service extensions for particular releases are announced.

For a description of z/OS coexistence considerations, refer to *z/OS and z/OS.e V1R7.0 Planning for Installation,* GA22-7504. This includes a list of the minimum supported levels of IBM products.

Support for z/OS 1.5 (and z/OS 1.4) is due to expire 31st March 2007, at which point no supported level of z/OS will be available for the 9672.

Note: For the subsystems, the coexistence considerations are different. In the case of different product versions, running multiple versions concurrently over a protracted period is likely to incur significant IBM software cost.

DB2

For DB2, you can normally have two consecutive versions in a Parallel Sysplex data sharing group, and in general migration from Version N to version N+1 is required. One exception has been the ability to migrate from DB2 V5 to DB2 V7 directly. You should plan to migrate from V7 to V8.

IMS

For IMS, you can have up to three consecutive releases of IMS within a data sharing group at the same time, as long as all the releases are still supported.

CICS

The current CICS position is that any supported CICS release may coexist in a CICSplex with any other supported release; however, certain functions may be limited by the release level of the members of the CICSplex.

However, some restrictions exist when running a mixed release data sharing group that may reduce availability or operability. This support is intended for migration purposes only. Plan to maintain this situation for as short a period of time as possible and make sure that a fallback plan is in place before beginning actual implementation. Review the relevant subsystem Release Guides for more information.

2.2.4 Combined Parallel Sysplex test and production environments

Figure 2-1 on page 21 shows a possible configuration that includes two Parallel Sysplexes, one aimed for *production* (non-shaded) and one aimed for *test* (shaded). The production Parallel Sysplex is in this case using stand-alone CFs, (that is, the CPCs running the production CFs have no non-CF LPARS on them).

CFCC Enhanced Patch Apply (z990/z890 and IBM System z9)

With the CFCC Enhanced Patch Apply, you can perform a disruptive install of new CFCC code on a *test* CF and run it, while a *production* CF image in the same CEC remains at the base CFCC code level. Then, when the test CF is successful, then new CFCC code can be installed on the production CF. Both installs can be done without a Power On Reset (POR) of the CEC.

At the time of writing, if you are doing data sharing, we recommend that you use failure-independent CFs in the production Parallel Sysplex. If you are doing resource sharing, then you may opt for one or both of the stand-alone CFs being replaced by a CF running on

an ICF on one of the z/OS production CPCs, either failure-independent or non-failure-independent. In case of a production Parallel Sysplex with two ICFs, then these two ICFs would be placed on separate CPCs.

Figure 2-1 and Figure 2-2 show combined Parallel Sysplex test and production environments using ICFs.

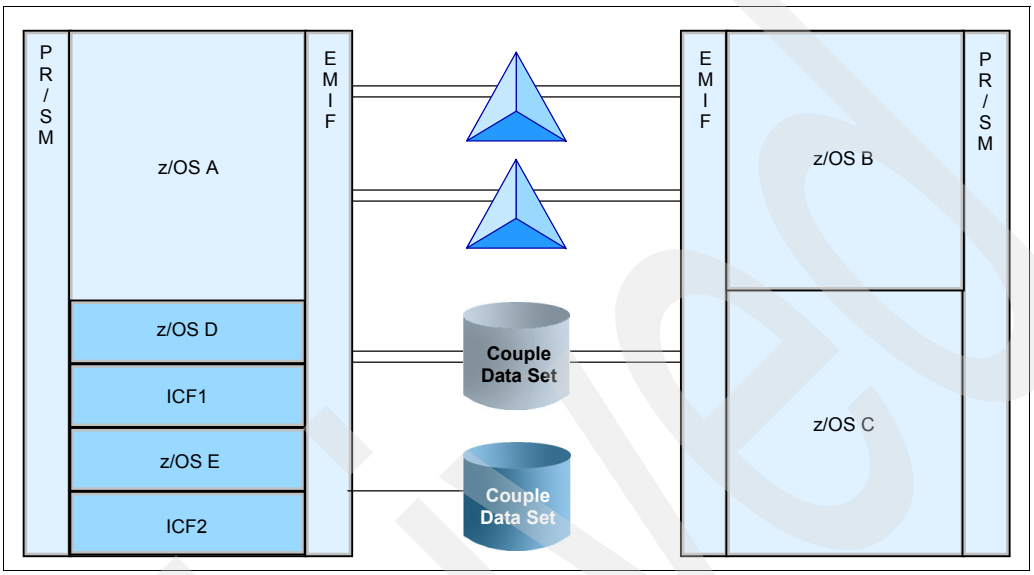


Figure 2-1 Combined environments for data sharing

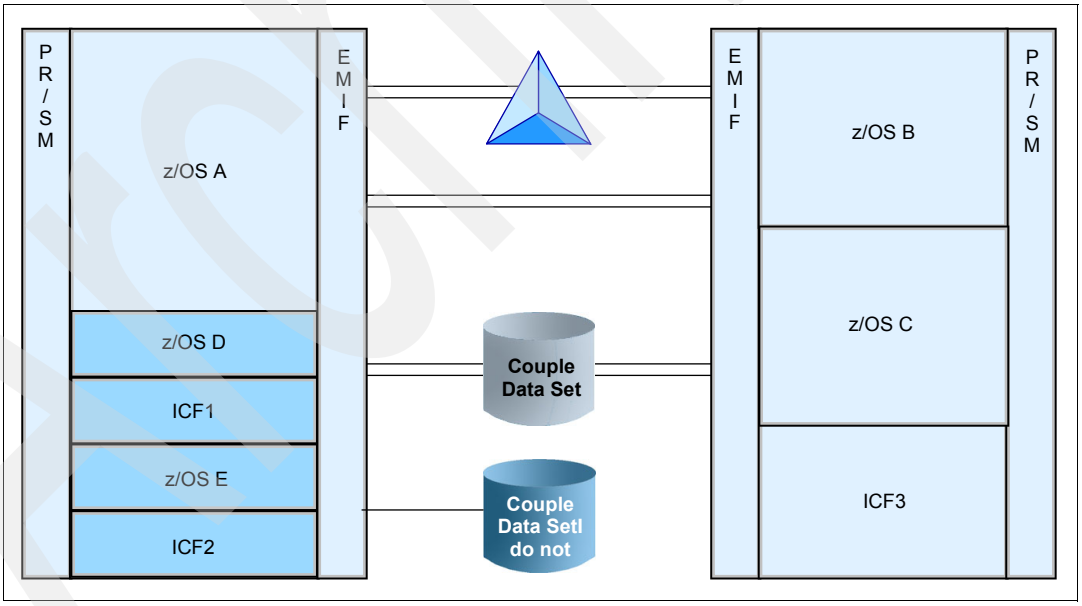


Figure 2-2 Combined environments for data sharing (GBP Duplexing)

Figure 2-3 shows the combined environments for resource sharing.

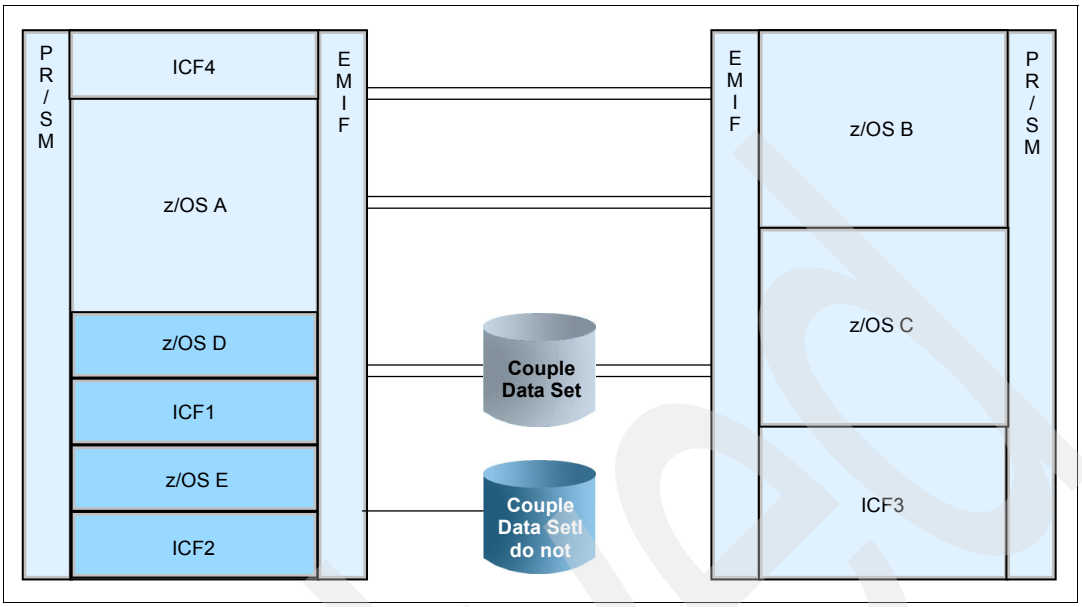


Figure 2-3 Combined environments for resource sharing

Note: Two LPs in the same CPC, but in different sysplexes, require separate physical CF links to attach to their respective CFs. (This is not shown in any of the figures.)

However, if two LPs in the same sysplex and in the same CPC attach to the same CF, then they can share physical CF links to make that attachment. This is because CF Sender (CFS) links can be shared between operating system LPs using EMIF, but CF Receiver (CFR) links cannot be shared between CF LPs.

2.2.5 Pre-production environment options

It is imperative that you have a test sysplex (or pre-production or operations proving sysplex) that is separate and distinct from the production sysplex available for certain types of testing, including operator training. For example, systems programmers must have the capability of testing new versions of software in a Parallel Sysplex environment. For more discussion on testing, refer to 4.8, “Test considerations in Parallel Sysplex” on page 258.

Recommendation for test environment in Parallel Sysplex: The current recommendation is that a test Parallel Sysplex should be separate and distinct from the production sysplex to ensure that any failures do not propagate to the production environment. This environment should also be as close in configuration to the production environment as is possible; however, the same capacity is not necessarily required. Remember that the more the test environment matches the production workload, including stress testing, the higher the probability that potential problems will be discovered before they reach the production environment.

The fundamental questions are how many z/OS and how many CF LPARs should there be in the test environment and on which processors should they be placed? Answers to these questions are likely to depend upon cost, history, client specific preferences, and the function being tested and the perceived risk. In quite a few cases, there will not be a lot of choice

about the second question because the client has only one or two CPCs and cannot afford another for testing.

Since true sysplex high availability (that is, to provide availability against the failure of a CPC or of a z/OS or CFCC image) requires at least two z/OS images and two CF LPARs, this will often be the configuration of choice for the test environment, and these LPARs could all be on one CPC.

Another option is the use of z/VM® for testing. As VM does not support external CF links, VM guests cannot currently participate in a multi-CPC Parallel Sysplex. A VM guest can, however, participate in a Parallel Sysplex within the one CPC. Enhancements to VM and selected CPCs provide the ability to run CFCC in a virtual machine (a CFVM), and VM will emulate the CF links between the CFVM and the z/OS guests. This gives you the ability to run multiple sysplexes. Information about VM's support for Parallel Sysplex, and specifically how VM can assist in Parallel Sysplex testing, is available on the Web at:

`http://www.vm.ibm.com/os390/`

Let us discuss the different Parallel Sysplex configurations and see what level of testing can be accomplished in each, as shown in Figure 2-4 on page 24:

- Two or more z/OS systems with two or more CFs

  This is the full baseline configuration, which is similar to the production environment. With this configuration, there are no issues, since all recovery scenarios and Parallel Sysplex options can be tested.

- Two or more z/OS systems with only one CF

  This allows full testing in normal operation, but does not allow testing of recovery procedures (for example, a structure rebuild because of CF or connectivity failures from an image).

- One z/OS system and one CF

  This CF does not test key data sharing functions. It may allow familiarity to be gained with CF operation, but full testing would still be required in another environment.

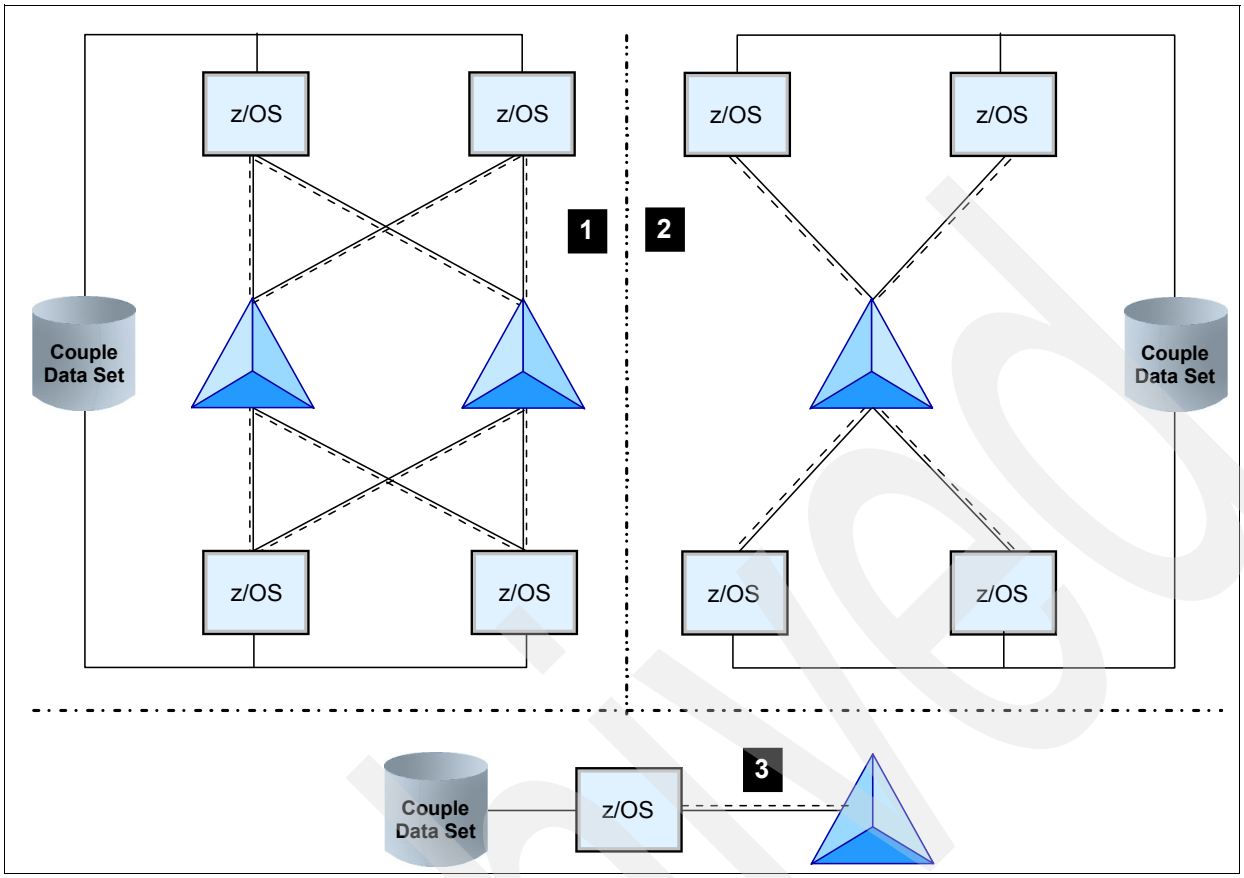


Figure 2-4 Examples of a test Parallel Sysplex: Configurations with External CFs

2.2.6 DASD sharing

In a multisystem environment, access to shared DASD needs serialization to maintain data integrity. Global Resource Serialization (GRS) is the component that manages resource serialization within and across systems. GRS has been available since MVS/SP™ V1.3.

Management of cross-system resources requires a fast communication method between each GRS subsystem. Prior to the announcement of OS/390 R2, the only way GRS communicated was to transfer a buffer called the *RSA message* around a ring architecture. This ring architecture was actually implemented through CTC or XCF links between each system that wished to share resources.

Since OS/390 R2, GRS has implemented a method to communicate called *GRS star*. GRS star uses a lock structure in the CF to store the status of shared resources. GRS still supports the ring architecture option; however, the GRS star architecture is the recommended option. The GRS Ring and Star Comparison (IBM Benchmark: *System/390 Parallel Sysplex Performance*, SG24-4356), shown in Table 2-1 on page 25, was done in 1998.

Table 2-1 GRS characteristics

GRS characteristic	Ring topology	Star topology
ENQ response time (ms)	10+ (increases with the number of systems).	<.04 (stable).
Real storage (frames)	1000+ (increases with the number of systems).	<1500+ (stable).
System ITR	0.9.	1.0.
Availability and recovery	Ring reconstruction is complex and may take a long time. It is a function of the number of systems in the ring.	No operator action required if another system fails.
Note: These numbers (rounded) are extracted from *S/390 Parallel Sysplex Performance*, SG24-4356. They apply to a 4-ways GRS complex running OS/390 R2.		

Since then, CF technology has improved greatly, but CTC performance has not, and so the numbers would be far better now for GRS Star.

To summarize:

- ENQ response time is significantly better in a star topology, even for sysplexes with as few as two systems. For batch jobs that process a large number of files, the reduced ENQ and DEQ times can have a significant impact on the elapsed time of the jobs.
- ITR may improve, mainly due to the removal of RSA message-passing overhead.
- Availability and recovery are better in star topology due to the way a lock structure is accessed.

GRS Star or Ring in a Parallel Sysplex?: The improvements brought by the star topology (speed, scalability, and recovery capabilities) make it the recommended choice for all Parallel Sysplexes. However, ring topology implementation may be dictated by specific installation needs (see “DASD sharing outside Parallel Sysplex” on page 26.

In a Parallel Sysplex, the following rules apply:

- GRS must be active in each system of the Parallel Sysplex.
- All systems in a Parallel Sysplex must be part of the same GRS complex.
- Systems in the sysplex, but not belonging to the Parallel Sysplex, must also be in the same GRS complex as those in the Parallel Sysplex. (In this situation, you obviously cannot use GRS star.)
- While the use of the GRS ring topology allows systems outside the sysplex to be part of the GRS complex, the use of the GRS star topology requires that the GRS complex exactly matches the Parallel Sysplex.
- If systems outside the sysplex are part of the GRS ring complex, they cannot be part of another sysplex.

DASD sharing outside Parallel Sysplex

If there is a need to share DASD outside the Parallel Sysplex or across Parallel Sysplexes, hardware reserve/release *must* be used to serialize z/OS access to these devices. Figure 2-5 depicts two Parallel Sysplexes and a stand-alone z/OS sharing DASD. DASD in the middle of the picture can be shared by all systems if hardware reserve/release is used to serialize the access.

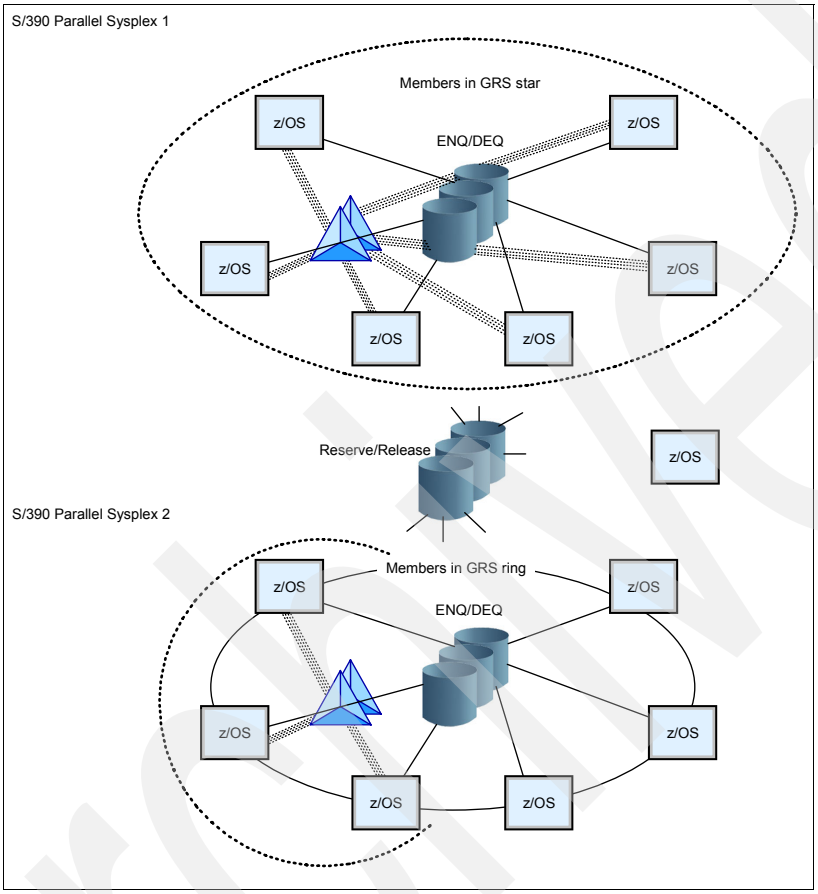


Figure 2-5 Sharing DASD between multiple sysplexes or GRS complexes

Sharing DASD across sysplexes requires a strict naming convention. It is the responsibility of the function or program that is writing on shared DASD to issue the RESERVE macro.

How to control hardware reserve or release requests: ISGGREX0 exit routine (member ISGGREXS of SYS1.SAMPLIB) can be used to ensure that all access to shared DASD outside the GRS complex is serialized with hardware reserve/release. We recommend activating this exit if you plan to share data across GRS complexes. It should be noted that this exit is not meant as a general purpose way of sharing DASD between sysplexes, and it has a number of restrictions.

Note: Because the multisystem data set serialization component of Multi Image Manager (MIM) from Computer Associates (CA) does not maintain a knowledge of sysplex, nor does it function as a ring, it does not have the same restrictions as GRS. Using this product, you are able to share DASD across multiple sysplexes without any integrity exposure and without having to depend on programs/functions issuing a reserve.

If you wish to use such a product, you should specify GRSRNL= EXCLUDE in your IEASYSxx member. This tells GRS that all ENQ, RESERVE, and DEQ macro requests with a scope of SYSTEMS are treated as though they had been found in the SYSTEMS exclusion RNL. Their scope is changed to SYSTEM and they are processed locally.

If an ENQ is issued with RNL= NO specified, the exclusion is bypassed, and the scope of SYSTEMS is retained. RNL= NO is used for special resources that must be processed by GRS, even if an equivalent product is installed.

If you already have MIM installed, it is possible to run MIM alongside GRS. You may use GRS for all global serialization except for those data sets you want to share between sysplexes. Include these data sets in the GRS exclusion list, and define them in the MIM inclusion list. This will be easier if you have strict naming conventions for these types of data sets.

There are, however, a number of ENQueues that are issued with a scope of SYSTEMS that are meant to provide serialization within a sysplex. Using a product such as MIM can cause *false enqueue contention* when a resource that is being serialized on a system in one sysplex is passed to a system that resides within another sysplex. The following are examples of this type of ENQueue (all of which are issued with RNL= NO specified):

- SYSZRACF (data sharing mode)
- SYSZRAC2
- SYSZLOGR
- SYSZAPPC
- SYSZDAE
- SYSZMCS

For more information about these topics, refer to:

- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS V1R7.0 MVS Planning: Global Resource Serialization,* SA22-7600

There is a tool called the *ENQ/DEQ/RESERVE Monitor* to help monitor the number of ENQs and who is issuing them. This tool is shipped in SYS1.SAMPLIB in z/OS R3 and follow-on releases and is documented in Chapter 3, “Using the ENQ/RESERVE/DEQ monitor tool“, of *z/OS V1R7.0 MVS Planning: Global Resource Serialization,* SA22-7600. IBM has accepted client requirements to be able to share DASD between different sysplexes, and, at the time of writing, is investigating possible solutions that might be delivered in a future release of z/OS.

2.2.7 System symmetry when configuring a Parallel Sysplex

We recommend that systems in a Parallel Sysplex are configured *symmetrically*. For more information about this, refer to *OS/390 V2R5.0 Parallel Sysplex Systems Management,* GC28-1861.

Symmetry, in the context of a Parallel Sysplex discussion, refers to replicating or cloning the hardware and software configurations across the different physical CPCs in the Parallel Sysplex. That is, an application that is going to take advantage of parallel processing might have identical instances running on all images in the Parallel Sysplex. The hardware and software supporting these applications should also be configured identically (or as close to identical as possible) on most or all of the systems in the Parallel Sysplex, to reduce the amount of work required to define and support the environment.

This does not mean that every CPC must have the same amount of storage and the same number or type of CPs; rather, the *connectivity* should be symmetrical (for example, connections to devices, CFs, CTC, and so on). A device should also have the same device number on every z/OS system.

The concept of symmetry allows new systems to be easily introduced, and permits automatic workload distribution in the event of failure or when an individual system is scheduled for maintenance. Symmetry also significantly reduces the amount of work required by the systems programmer in setting up the environment. Systems programmers and operations personnel using the following will find it easier to operate a Parallel Sysplex where the concept of symmetry has been implemented to a large degree, as in the following:

- Consistent device numbering
- A single IODF, containing the definitions for all CPCs and devices in the installation
- Good naming conventions
- System symbols
- Single Point Of Control (SPOC)/Single System Image (SSI) support in z/OS and subsystems (for example, using the enhanced ROUTE command)

These make planning, systems management, recovery, and many other aspects much simpler. A move toward Parallel Sysplex is a move toward an environment where any workload should be able to run anywhere, for availability and workload balancing. Asymmetry can often complicate planning, implementation, operation, and recovery.

There will, however, be some instances where asymmetric configurations may exist. At times this may even be desirable. For example, if you have a Parallel Sysplex environment that includes an application requiring a specific hardware resource, you may consider having that resource only on one (or a subset) of the CPCs in the Parallel Sysplex. An example is Advanced Function Printing™ (AFP™) printers connecting to certain systems.

Sub capacity WLC and system symmetry

It may be necessary to optimize software charging by separating, say, CICS/DB2 applications from IBMS DB/DC applications into separate LPARs to reduce license charges. The savings have to be balanced against the extra work involved and resource in setting this up.

2.2.8 What different 'plexes are there?

Within the Parallel Sysplex environment, there are many different 'plexes referred to. In this section, we give recommendations on how each of them relate to one another. Further, there is a brief overview of what constitutes each 'plex.

We would *strongly* recommend that, to the extent possible in your installation, you should try to have all the 'plexes line up with either the basic sysplex or the Parallel Sysplex, as is appropriate. If this is not done, the resulting confusion about which systems are in or out of which 'plex is nearly certain to lead to problems, especially in recovery situations. Having a single logical image both for units of work and for operations and management will lead to simpler, easier-to-understand operations, and also make systems management *far* easier.

A case where you might need not to follow this advice is where systems are being merged or for outsourcers who want to have separate ‘plexes for their clients; see *Merging Systems into a Sysplex*, SG24-6818, where this issue is discussed in detail.

The following lists some of the many of the 'plexes commonly referred to today and we discuss these in outline:

- Sysplex
- JESplex
- GRSplex
- RACFplex
- SMSplex
- HSMplex
- BCSplex
- OAMplex
- CICSplex
- VTAMplex
- Tapeplex

Note: These terms are not always used in the same way in standard IBM documentation.

You may wonder why we have not listed DB2plexes or IMSplexes, or even VSAMRLSplexes or MQSeriesplexes. The reason is that these 'plexes are usually referred to as data sharing groups rather than as 'plexes. Also, they tend to be much more flexible and often contain just a subset of the systems in the Parallel Sysplex, and therefore do not have many of the attributes associated with the 'plexes we describe below.

In the following sections, we define what is meant by each of the 'plexes listed above, and show each one's relationship to Parallel Sysplex, and in certain cases to other 'plexes as well.

Sysplex This describes the set of one or more systems that is given the same XCF sysplex name, and in which the authorized programs in the systems can use XCF services. All systems in a sysplex must be connected to a shared sysplex couple data set. An XCFLOCAL sysplex does not allow a sysplex couple data set. None, one, some, or all of the systems can have CF connectivity. The sysplex is the basis for our relationships in the discussion that follows.

Parallel Sysplex This is the set of systems within a sysplex that all have connectivity to the same CF. The recommended configuration is for *all* systems in the sysplex to be connected to *all* the CFs in that sysplex. If only a subset of your systems is connected to a given CF, that subset is (throughout this redbook) referred to as a Parallel Sysplex. In some documents, you may also find this subset with CF connectivity being referred to as a *CF Subplex*. Using the latter definition of a Parallel Sysplex, you may have more than one Parallel Sysplex within a sysplex if all systems are not connected to each CF. This is *not* a recommended setup.

Note: Not all IBM literature agrees on a unique definition of what constitutes a Parallel Sysplex. For the purpose of this redbook and the discussion that follows, we will use the definition provided in 1.6.2, “Parallel Sysplex” on page 11. Other documents mention that a Parallel Sysplex is composed of *two* or more systems, while we say that it can consist of just one system and a connected CF. Still other documents mention that a Parallel Sysplex may comprise systems that do *not* have CF connectivity - but this most certainly does not meet our definition.

You will also see the following terms used in IBM literature to describe different Parallel Sysplexes:

- Resource sharing Parallel Sysplex

  This describes the non-data sharing implementation. S/390 and zSeries resource sharing provides the following functionality:

  - XCF Signaling: Providing multi-system signaling
  - GRS Star: Multi-system resource serialization
  - JES Checkpointing: Multi-system checkpointing
  - Shared Tape: Multi-system tape sharing
  - Merged Operations Log: Multisystem log
  - Merged LOGREC: Multisystem log
  - RACF: Sysplex data sharing
  - Shared Catalog: Multi-system shared catalogs
  - WLM: Multi-system enclave support
  - LPAR Cluster: Intelligent Resource Director (IRD)

Note: There is no stand-alone CF requirement for a Resource Sharing Parallel Sysplex. A Resource Sharing Parallel Sysplex applies to both multi-CPC and single CPC environments.

- Data sharing Parallel Sysplex

  This describes the extension into data sharing, and covers enhancements that are made over time into areas such as high availability and disaster recovery.

A Parallel Sysplex example is shown Figure 2-6 on page 31. This example is used as the base for the additional figures in the 'plex discussion.

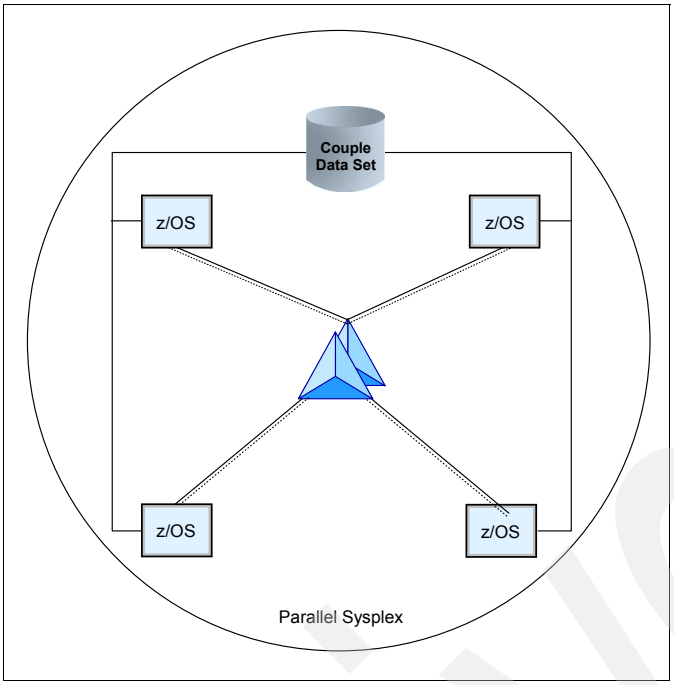


Figure 2-6 Parallel Sysplex configuration

Before you read the discussion about how the sysplex, Parallel Sysplex, and all the other 'plexes match, note that it is normally recommended that you *match as many of your 'plexes as possible*. However, you may have valid business or technical reasons for not following the recommendation initially. An example may be that, in a migration phase, you have CPCs that are not yet equipped with CF links. Images on these CPCs may be in the same Base Sysplex as the systems in the Parallel Sysplex. However, as you move these systems into the Parallel Sysplex, you should plan on lining them up with all the other systems already in the Parallel Sysplex. In the following section, we look at some pros and cons of certain configurations and provide recommendations where possible.

2.2.9 How do 'plexes relate to each other?

In order to simplify the operation and management costs in a multisystem environment, it is critical that the relationship between all the different components of that environment are thought out in advance. The redbook *Parallel Sysplex - Managing Software for Availability*, SG24-5451 contains a discussion about the different 'plexes and how they relate to each other, especially from a naming conventions perspective.

In this section, we discuss the capabilities, restrictions, and relationships between the following 'plexes:

- JESplex on Figure 2-7 on page 33.

  For the JESplex, a distinction is made between:

  - A data sharing JESplex, where checkpointing is implemented using the CF.
  - A non-data sharing JESplex.
- GRSplex in Figure 2-8 on page 35.

  For the GRSplex, a distinction is made between:

  - A GRSplex based on star topology.
  - A GRSplex based on ring topology.

- RACFplex in Figure 2-9 on page 37.

  For the RACFplex, a distinction is made between:

  – A data sharing RACFplex, where the RACF database is buffered in the CF.

  – A non-data sharing RACFplex.

  Further, a distinction is made between whether:

  – Sysplex communication is enabled.

  – Sysplex communication is not enabled.
- SMSplex in Figure 2-10 on page 39.
- HSMplex in Figure 2-11 on page 41.
- BCSplex in Figure 2-12 on page 43.
- OAMplex in Figure 2-12 on page 43.
- CICSplex in Figure 2-12 on page 43.
- GDPS is discussed further in 3.9, “GDPS: The e-business availability solution” on page 139.
- LPARplex(IRD) is described in detail in *z/OS Intelligent Resource Director*, SG24-5952.

For a summary table with key information about how 'plexes relate, including recommended configurations, refer to Figure 2-15 on page 49.

JESplex

Definition: This describes the job entry subsystem configuration sharing the same spool and checkpoint data sets. For JES2 installations, this is the multi-access spool (MAS) environment; for JES3 installations, this is the JES3 global and locals configuration. *It is recommended that there be only one MAS in a sysplex, and that the MAS matches the sysplex boundary*. The same is true for a JES3 global and locals configuration. There are a number of installations that run more than one JES2 MAS in a sysplex, or more than one JES3 complex in a sysplex, for any number of reasons.

Since JES2 and JES3 use XCF, the JES MAS or complex must be within a sysplex; they cannot span sysplexes. If you want to run both JES2 and JES3 in the same sysplex, this can be done. Here it is unlikely that either the JES2 or JES3 'plex will match the sysplex. Figure 2-7 on page 33 shows recommended and valid configurations for the JESplex.

The top configuration shows the recommended configuration where the JESplex matches the Parallel Sysplex. The JESplex should also match the GRSplex, SMSplex, BCSplex, RACFplex and possibly some of the other 'plexes discussed in the following sections.

The center configuration in JESplex Configurations shows that several JESplexes can coexist in the same Parallel Sysplex. Several JES2plexes, JES3plexes, and a combination of these can coexist in a Parallel Sysplex. JES2plexes may even overlap (have systems in common with) JES3plexes.

Note: Figure 2-7 is not meant to imply that a JESplex must use the CF. However, there are certain advantages to JES2 using the CF for checkpointing.

The bottom configuration illustrates that systems outside the Parallel Sysplex, but within the same sysplex, can belong to a JESplex. Therefore, if the sysplex is larger than the Parallel Sysplex, the JESplex can also be larger than the Parallel Sysplex (as shown in Figure 2-7).

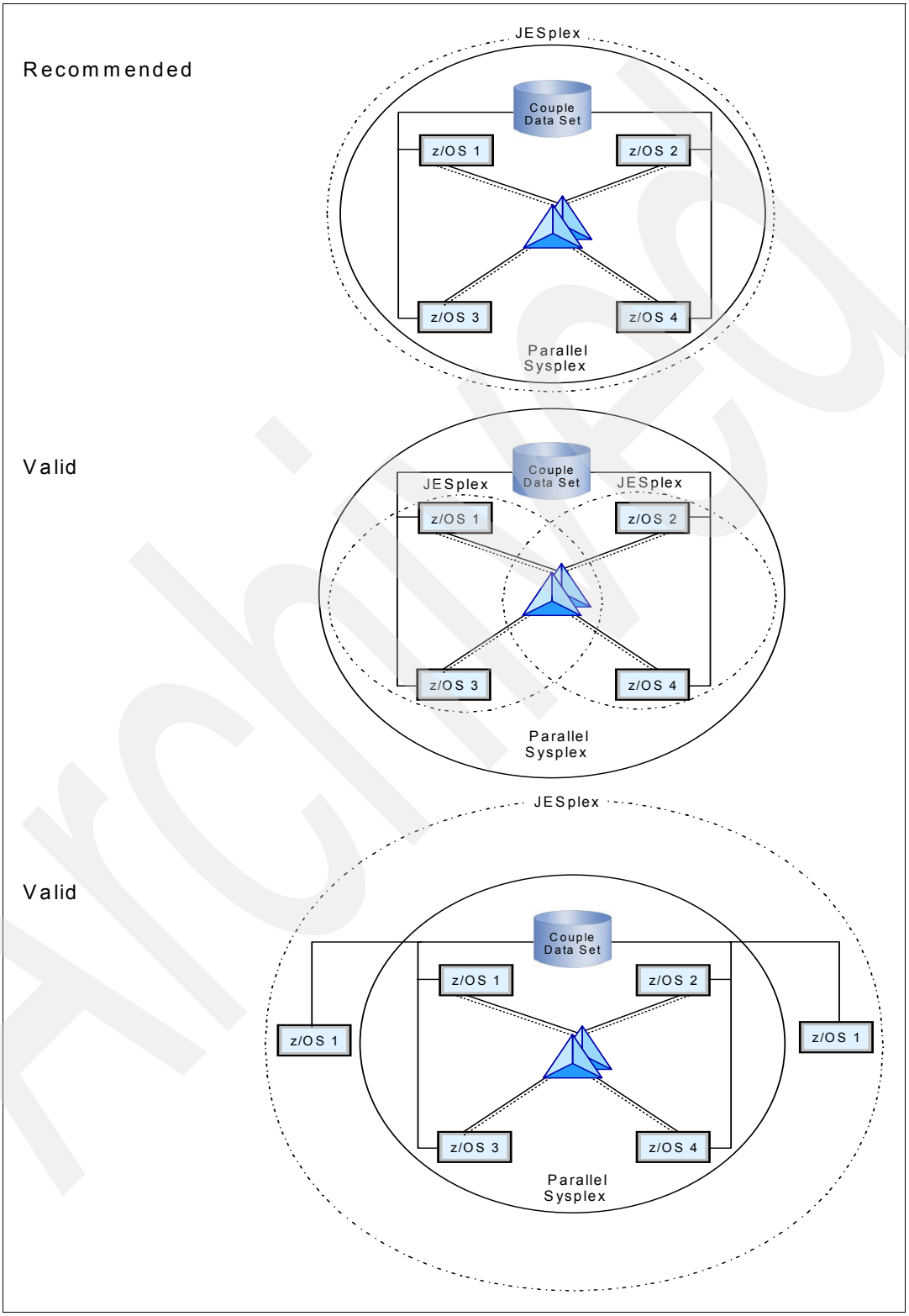


Figure 2-7 JESplex configurations

GRSplex

Definition: GRSplex describes one or more z/OS systems that use global resource serialization to serialize access to shared resources (such as data sets on shared DASD volumes). A GRSplex describes all systems that are part of either a GRS ring or GRS star configuration.

In a GRS ring configuration, systems outside a sysplex may share DASD with those systems in the sysplex. It is possible for a GRSplex based on ring topology to be larger than the sysplex, but additional systems cannot belong to another sysplex. This is also called a *mixed GRS complex*. In a mixed GRS complex, systems within the sysplex would automatically use XCF to communicate (regardless of what is specified in the GRSCNF member), while GRS would need its own dedicated CTCs to communicate with any systems outside the sysplex.

With the GRS star topology, however, the GRSplex *must* match the Parallel Sysplex. Also, every system in a Parallel Sysplex must be a member of the same GRSplex and the Parallel Sysplex must match the sysplex. In other words, in a GRS star topology, the GRSplex, the Parallel Sysplex, and the sysplex must match completely.

No matter whether GRSplexes are based on star or ring topology, device serialization *between* multiple Parallel Sysplexes must be achieved by the reserve/release mechanism.

Figure 2-8 on page 35 shows recommended, valid, and not valid configurations for the GRSplex.

The top configuration shows the recommended configuration where the GRSplex matches the Parallel Sysplex.

The center configuration shows that the GRSplex (based on GRS ring topology) may include systems outside the sysplex. If your GRSplex is larger than the sysplex, there are some operational aspects to be considered:

- Lack of automatic GRS ring rebuild in a recovery situation
- No automatic adjustment of the residency time value (RESMIL)
- No dynamic changes to the Resource Name Lists (RNLs)

Note: The figure is not meant to imply that a GRSplex must use the CF. However, a GRSplex based on the star topology must use the CF and, therefore, cannot include systems outside the Parallel Sysplex.

The bottom configuration illustrates that several GRSplexes cannot coexist in the same Parallel Sysplex.

For more information related to the GRSplex configuration in the Parallel Sysplex environment, refer to *OS/390 V2R10.0 MVS Planning: Global Resource Serialization,* GC28-1759. Also refer to 2.2.6, “DASD sharing” on page 24 for a discussion on MIM and GRS considerations.

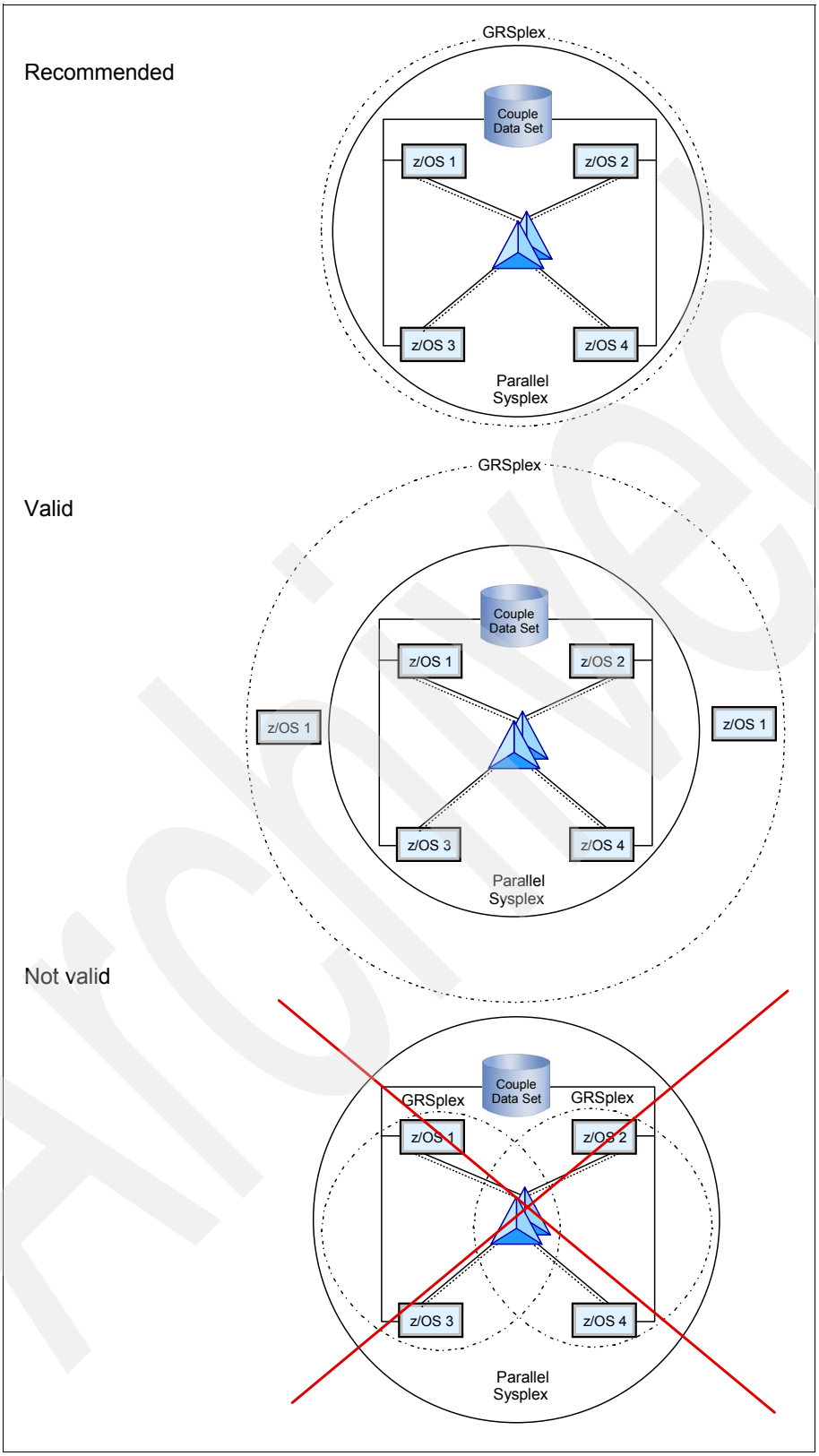


Figure 2-8 GRSplex configurations

RACFplex

Definition: This describes the systems that share the same RACF database. If the systems sharing the RACF database are all members of the same sysplex, RACF can be enabled for RACF sysplex communication. This implies RACF will use GLOBAL ENQ instead of reserve/release for serialization. In addition, commands are propagated to all systems, and updates made on one system are available to all.

If the systems sharing the RACF database are all members of the same Parallel Sysplex, RACF can also be enabled for sysplex data sharing. This implies that RACF will exploit the CF. There can be only one RACF data sharing group within a Parallel Sysplex. The installation cannot alter the RACF XCF group name (IRRXCF00), so it is not possible to have more than one RACF sysplex data sharing group in the sysplex. *It is recommended that you make the RACFplex match the Parallel Sysplex and enable it for sysplex data sharing.*

When the RACFplex matches the Parallel Sysplex, RACF can be configured with all systems sharing the same RACF database and one of the following:

- All systems not enabled for RACF sysplex communication (in this case, you should not do RACF sysplex data sharing)
- All systems enabled for RACF sysplex communication
- All systems enabled for RACF sysplex communication and RACF sysplex data sharing

After RACF sysplex communication has been enabled on all systems, RACF sysplex data sharing mode can be enabled or disabled. When sysplex communication is enabled, the second and subsequent system IPLed into the sysplex enters the RACF mode that currently exists, regardless of the data sharing bit setting in the Data Set Name Table (DSNT). After IPL, the RVARY command can be used to switch from sysplex data sharing to non-data sharing mode and vice versa.

You can share the RACF database with systems outside the Parallel Sysplex boundary. However, in that case, there are operational aspects to be considered:

- RACF sysplex data sharing cannot be enabled.
- Reserve/release is used for serialization.
- If the system is outside the sysplex, then it should not be part of another sysplex.
- Commands will not be propagated to systems outside the sysplex boundary.

You may have more than one RACFplex in the Parallel Sysplex, but only one of the RACFplexes can be enabled for RACF sysplex data sharing. The other RACFplexes will not be able to use the CF for buffering the RACF database.

If you want to exploit RACF sysplex data sharing, the participating systems all have to be within the Parallel Sysplex. RACFplex Configurations in Figure 2-9 on page 37 shows recommended, valid, and not valid RACF configurations. For more information about RACFplexes, refer to *z/OS V1R7.0 Security Server RACF Security Administrator's Guide,* SA22-7683*,* and the redbook *S/390 Parallel Sysplex: Resource Sharing*, SG24-5666.

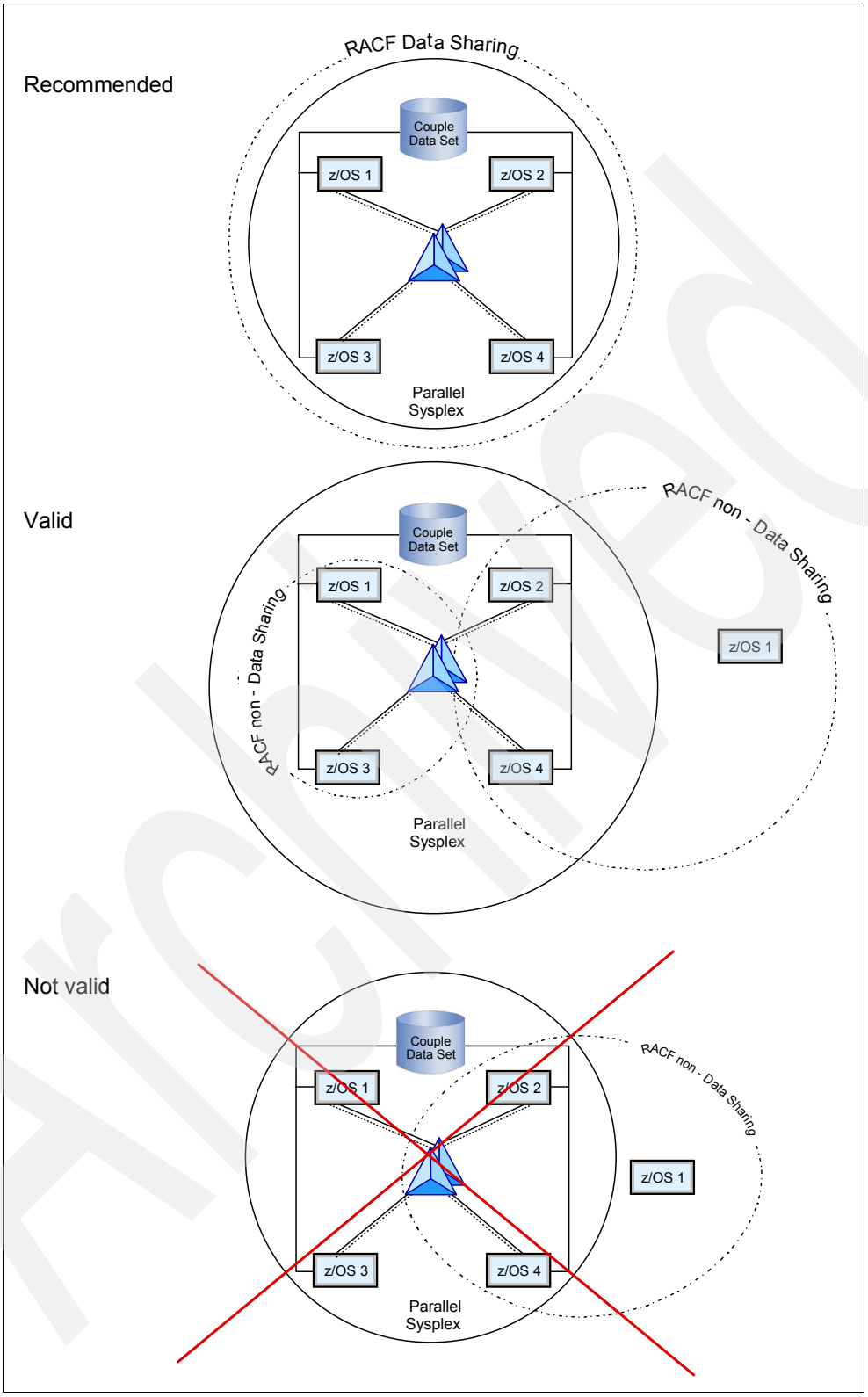


Figure 2-9 RACFplex configurations

SMSplex

Definition: This describes those systems that use the same SMS base configuration stored in the active control data set (ACDS), and the same communications data set (COMMDS). While it is possible for the SMSplex to include systems outside the sysplex, *it is recommended that, if possible, the SMS complex, or SMSplex, should match the sysplex.*

SMSplex Configurations in Figure 2-10 on page 39 shows recommended and valid configurations for the SMSplex. The top configuration shows the recommended configuration where the SMSplex matches the Parallel Sysplex. The SMSplex should also match the JESplex, GRSplex, RACFplex, and possibly some of the other 'plexes discussed.

The center configuration in SMSplex Configurations shows that the SMSplex may include systems outside the Parallel Sysplex. If your SMSplex is larger than the Parallel Sysplex, then it is likely that your SMSplex matches the sysplex and the GRSplex. The SMSplex may span several sysplexes and Parallel Sysplexes, in which case it is not possible to match the GRSplex. In this configuration, care should be taken with respect to the performance of the catalogs, ACDS, and COMMDS data sets.

The bottom configuration illustrates that the SMSplexes may be smaller than the Parallel Sysplex. Several SMSplexes may coexist in the Parallel Sysplex.

The Enhanced Catalog Sharing (ECS) protocol, introduced with DFSMS/MVS V1.5, uses a CF cache structure to hold change information for shared catalogs. This eliminates catalog-related I/O to the VVDS, resulting in better performance for both user and master catalog requests. All systems sharing a catalog that is being used in ECS mode *must* have connectivity to the same CF, and *must* be in the same GRSplex. So, if you are using ECS, the middle configuration in SMSplex configurations on Figure 2-10 on page 39 would not be valid.

SMS uses XCF communication for a number of components, including:

- OAM: This is described in the OAMplex section.
- DFSMShsm™: This is described in the HSMplex section.
- VSAM/RLS: If you are using the VSAM Record Level Sharing support for the DFSMShsm CDS, then all participating systems must have connectivity to the CF and therefore be in the same Parallel Sysplex.
- PDSE Sharing: If you are sharing PDSEs across systems, then XCF services are used to communicate status information between systems.

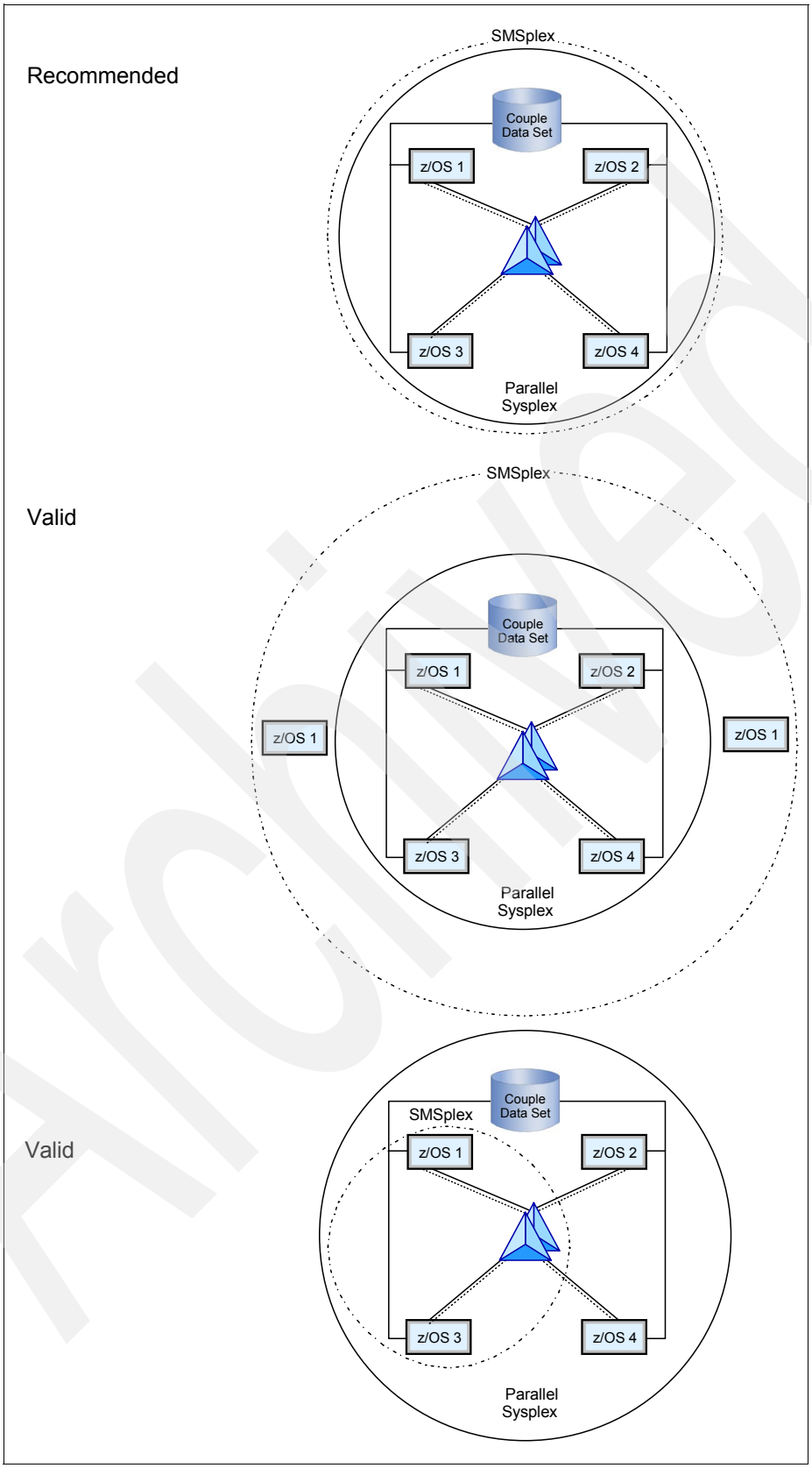


Figure 2-10 SMSplex configurations

HSMplex

Definition: Closely related to the SMSplex is the HSMplex, which is the set of DFSMShsm systems that share the same HSM journal and control data sets. There can be multiple HSMplexes in the sysplex, in which case the SETSYS keyword of the ARCCMDxx member of SYS1.PARMLIB must be used to specify a plexname suffix, The default plexname is ARCPLEX0 with suffix PLEX0. See Chapter 12, *DFSMShsm in a Sysplex Environment*, *z/OS V1R7.0 DFSMShsm Implementation and Customization Guide,* SC35-0418.

DFSMS 1.5 introduced three new Parallel Sysplex-related facilities:

- Secondary Host Promotion

  Secondary Host Promotion provides the ability for a HSM to take over unique responsibilities from another HSM Primary or Secondary Space Management (SSM) host should that host, or the system it is running on, fail. This ability is controlled by the HSM SETSYS PRIMARYHOST and SETSYS SSM commands, and uses XCF services to communicate the status of the Primary and SSM hosts to the other HSMs in the HSMplex. In order for a host to be eligible to take over, it must be in the same HSMplex as the failed host, as defined by the new PLEXNAME parameter. There is one DFSMShsm XCF group per HSMplex. The XCF group name is the HSMplex name. There is one XCF group member for each DFSMShsm host in the HSMplex.

- Single GRSplex

  The next new function is known as *Single GRSplex* support. Prior to DFSMS 1.5, when HSM executed a given function, it would issue an ENQ against a fixed name – this is to protect the integrity of the control data sets. The problem with this mechanism is that if you have two independent HSMplexes within the same GRSplex, you would get false contention if the two HSMplexes tried to execute the same function (but against a different set of DASD) at the same time.

  Single GRSplex gives you the option of changing HSM so that it issues a unique ENQ, utilizing the data set name of the associated HSM CDS. This effectively creates an HSMplex, but the scope of this HSMplex is based on the data set names of the CDSs used by the member HSMs. To enable this new support, you must specify a new parameter, RNAMEDSN = Y, in the HSM startup JCL.

- Large CDS support (this was retrofit to DFSMS 1.4 via APAR OW33226)

  The third function, DFSMShsm, has support for the use of RLS for its own Control Data Sets. The MCDS, BCDS, and OCDS can be optionally be defined as EA VSAM data sets, which greatly increases the capacity beyond 16 GB for the MCDS and BCDS, and beyond 4 GB for the OCDS. All the HSM systems accessing a given set of CDSs must have access to the same CF. The HSMplex must be the same as, or smaller than, the RLSplex.

All of these impact the systems that constitute the HSMplex.

One additional thing to consider in relation to the scope of a HSMplex is if you use the MVS Logger. When you are using Logger, all systems that are connected to a LOGR structure get notified when any of the logstreams within that structure exceed the user-specified threshold. Any of those systems are eligible to off load the logstream data to an offload data set on DASD. Therefore, all the systems that are connected to a given LOGR structure should be in the same HSMplex. Figure 2-11 on page 41 shows recommended and valid configurations for the HSMplex.

The valid configuration with members not in the sysplex would require GRS ring (to support members not in the sysplex).

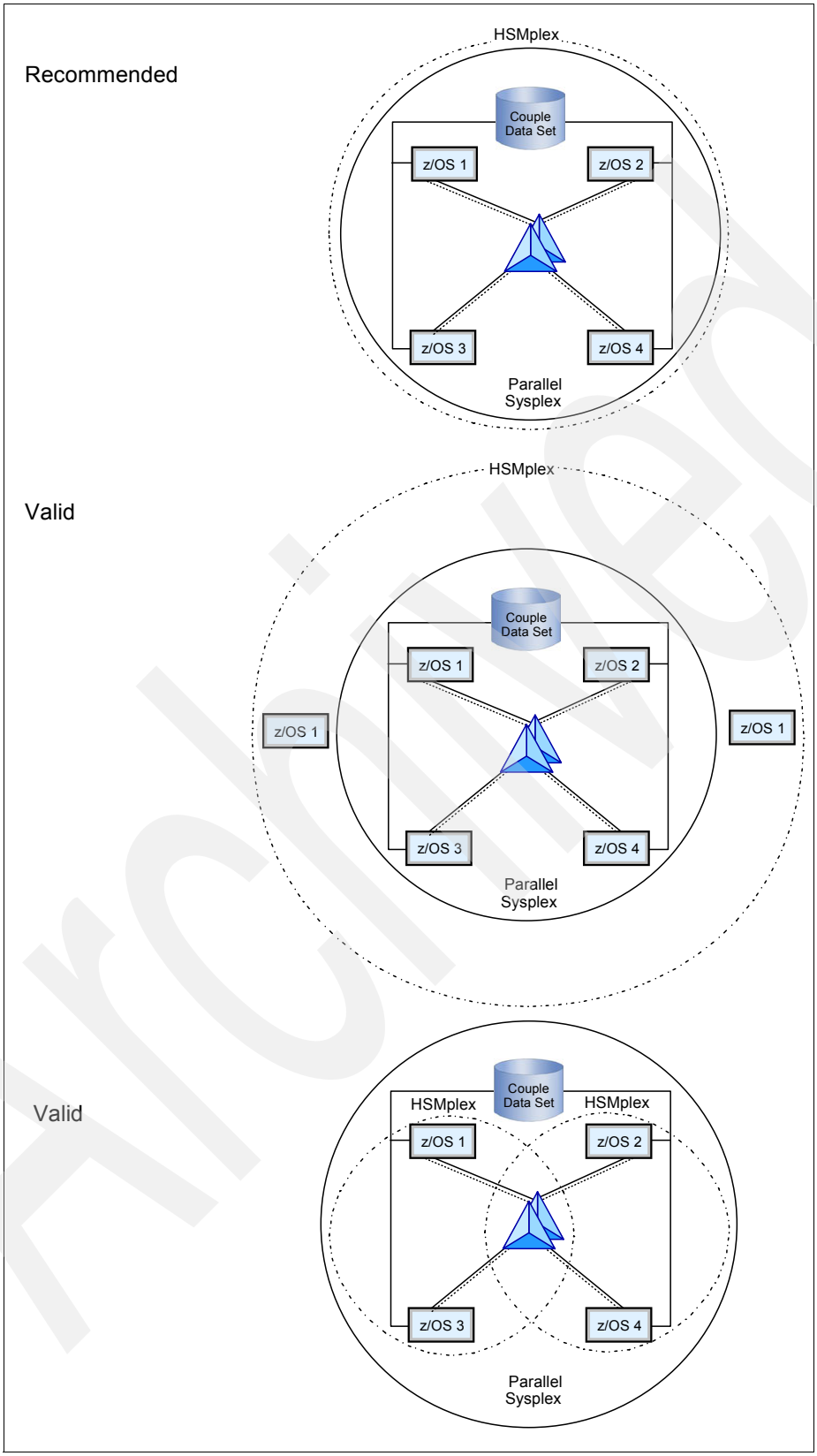


Figure 2-11 HSMplex configurations

BCSplex

Definition: The BCSplex is the set of systems sharing a set of catalogs in Enhanced Catalog Sharing (ECS) mode. Once a catalog is open in ECS mode, it cannot be concurrently opened by a system that does not have ECS support, or that does not have access to the ECS structure in the CF. A new ENQ is used to indicate the fact that a catalog is open in ECS mode, so all the systems sharing a catalog that *could potentially* be opened in ECS mode *must* be in the same GRSplex. Failure to adhere to this restriction will almost definitely result in a damaged catalog.

ECS mode is supported on DFSMS 1.5 and higher systems.

BCSplex configurations in Figure 2-12 on page 43 shows both valid and invalid configurations for the BCSplex. The top configuration is valid because all the systems in the BCSplex are within the Parallel Sysplex, and thus have access to the ECS structure in the CF. The middle configuration is also valid. As long as *all* the systems accessing any catalog in the sysplex (that is eligible to be opened in ECS mode) have access to the same ECS CF structure, that is a valid configuration. As systems 1 and 3 both have access to the same CF, that is a valid configuration. The bottom configuration is not valid because the systems outside the Parallel Sysplex cannot access the ECS structure in the CF and therefore cannot open a catalog that is open in ECS mode by a system within the Parallel Sysplex.

For more information, refer to the redbook *Enhanced Catalog Sharing and Management*, SG24-5594.

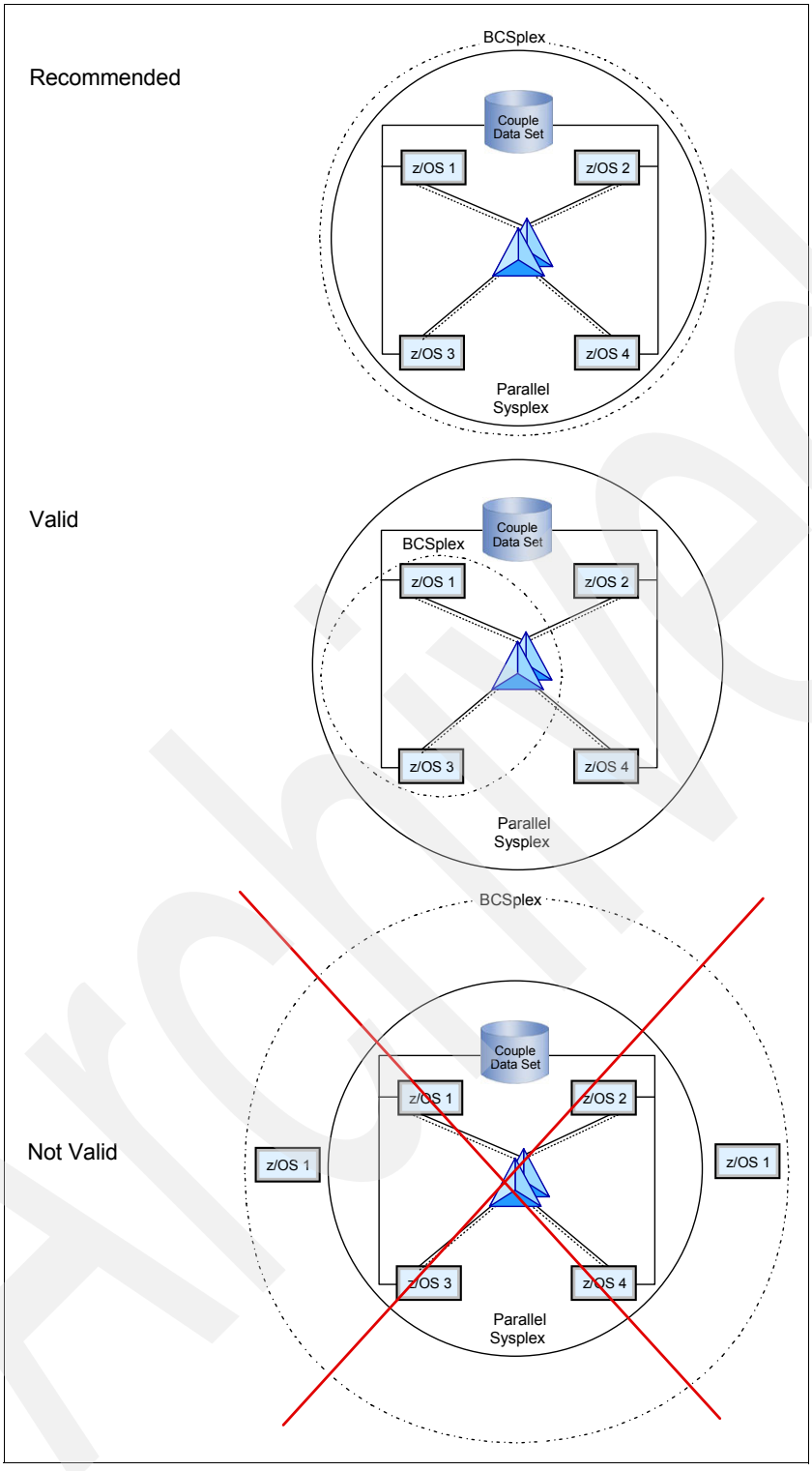


Figure 2-12 BCSplex configurations

OAMplex

Definition: DFSMS 1.5 introduced the ability for multiple OAM instances, on multiple z/OS images, to have access to shared OAM objects. The OAM instances must all be in XCF mode, in the same XCF group, and the associated DB2 subsystems must all be in the same data sharing group. An OAMplex, therefore, is the set of OAM instances that have access to the shared OAM objects.

OAMplex Configurations on Figure 2-13 on page 45 shows both valid and invalid configurations for the OAMplex. The top two configurations are valid because all the systems in the OAMplex are within the Parallel Sysplex, and thus have access to the DB2 structures in the CF. The bottom configuration is not valid because the systems outside the Parallel Sysplex cannot be in the DB2 data sharing group, and therefore cannot be in the OAMplex. Even if they are in the same sysplex (and thus can communicate using XCF), they still cannot be in the OAMplex; they must also be in the same DB2 data sharing group as all the other systems in the OAMplex.

For more information, refer to *z/OS V1R7.0 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426.

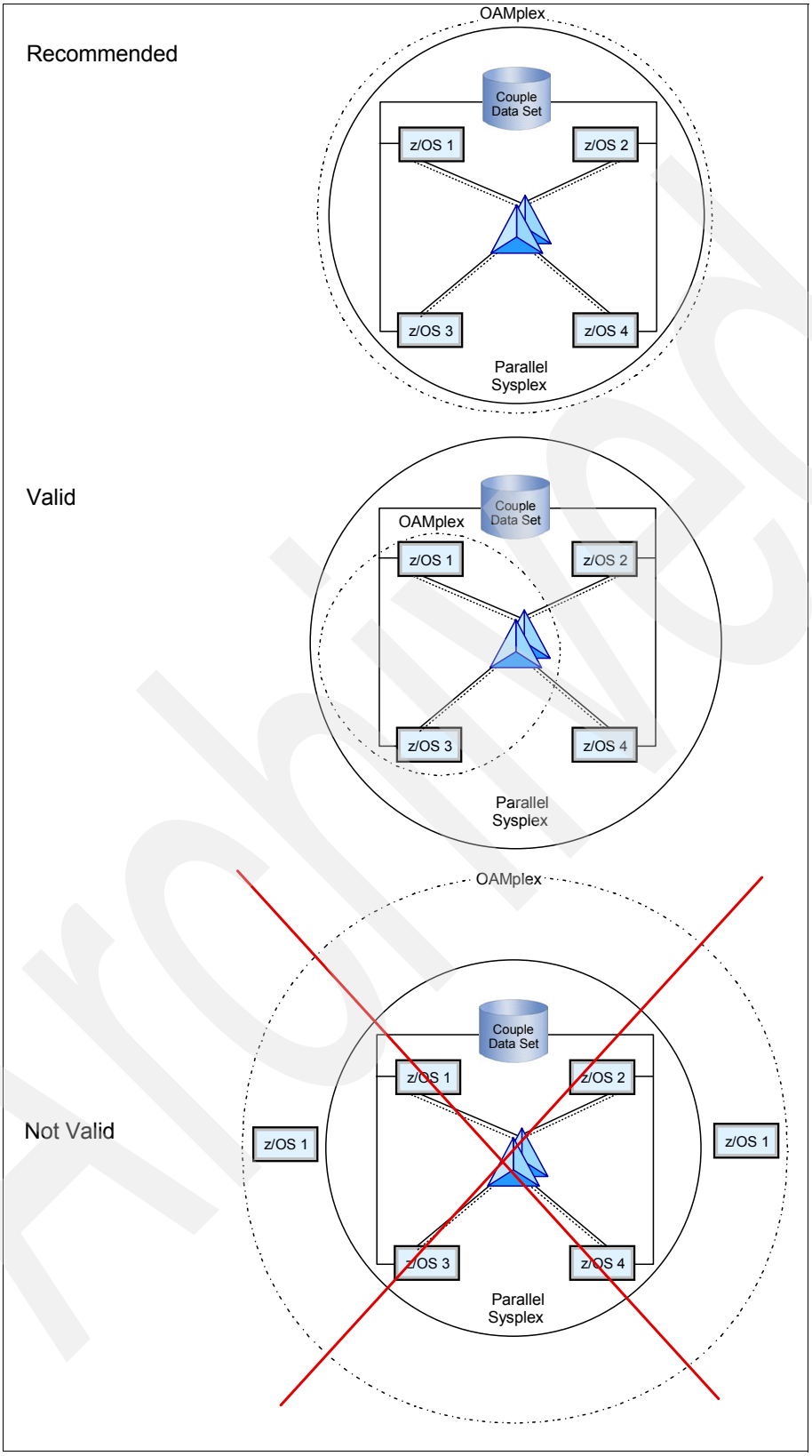


Figure 2-13 OAMplex configurations

CICSplex

Definition: There are a number of different configurations that can be termed a CICSplex. Some of these are:

- The set of CICS regions that share a common VTAM generic resource name
- The set of CICS regions that share non-recoverable temporary storage in a CF
- The set of CICS regions that have access to shared data tables in a CF
- The set of CICS regions that all access the same shared data (IMS, DB2, or VSAM)
- The set of CICS regions that all write to the same logger structure in a CF
- The set of CICS regions that use XCF for MRO communication

A CICSPlex® SM is a term for a group of CICS regions that it is to monitor and control as a logical entity.

Of these, the CICSplex as defined to CICSPlex SM can include CICS systems outside the sysplex, even on non-S/390 platforms, and therefore will be excluded from the following discussion.

If you plan to use XCF for MRO communication, then all the participating CICS regions need to be in the same Base Sysplex.

For the other four CICSplexes, all systems in each CICSplex need to communicate with the same CF. To make recovery easier, it is *highly* recommended that all four 'plexes line up with each other and with the Parallel Sysplex. CICSplex Configurations in Figure 2-14 on page 47 shows both recommended and valid configurations for the CICSplex.

Note: The second figure is only valid for the CICSPlex SM type of CICSplex; all the others need to at least be in the same Base Sysplex.

Refer to *CICS Transaction Server for z/OS V3.1 Installation Guide,* GC34-6426 for a much more detailed explanation of the role of the CF in a CICS TS environment.

CICS TS R2 removed the requirement for CF for CICS logging and journals, using the DASD Logger. However, this configuration is only valid in a single system environment. It does *not* support data sharing environments.

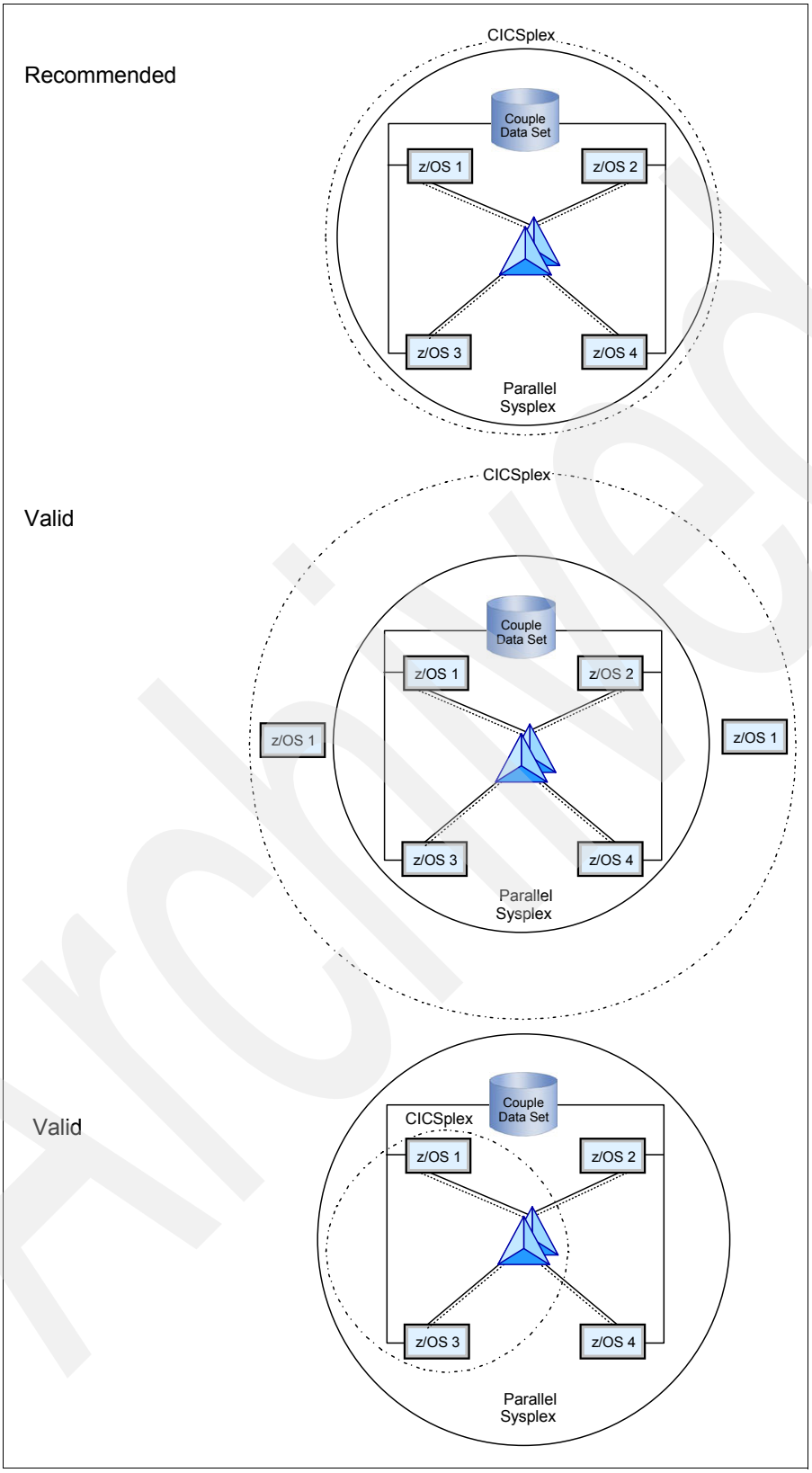


Figure 2-14 CICSplex configurations

Tapeplex

Definition: Prior to z/OS 1.2, automatic tape switching used a CF structure called IEFAUTOS. This was replaced by ATS Star in z/OS 1.2 with better manageability and availability characteristics. ATS star uses GRS and XCF services to maintain serialization when allocating tapes. We strongly recommend using GRS star when using ATS star.

2.2.10 'Plex summary

Figure 2-15 on page 49 summarizes the information about the various 'plexes. The table shows that the recommendation is almost always to match the 'plex with the Parallel Sysplex.

The columns in the table indicate:

- 'plex = Parallel Sysplex = sysplex

  The 'plex matches the Parallel Sysplex and all systems are connected to each CF. In other words, the same number of images belong to the 'plex, the Parallel Sysplex, and the sysplex.

- CF required

  The 'plex requires a CF. Certain 'plexes have a mandatory CF requirement.

- Data sharing using a CF

  The 'plex uses the CF for data sharing among members of a data sharing group.

- < Parallel Sysplex

  The scope of the 'plex is smaller than the Parallel Sysplex. In other words, fewer systems may belong to the 'plex than to the Parallel Sysplex.

- > Parallel Sysplex

  The scope of the 'plex is greater than the Parallel Sysplex. In other words, a larger number of systems may belong to the 'plex than to the Parallel Sysplex.

  Care should be exercised at this point. In certain cases, systems outside the sysplex cannot be part of other sysplexes, or other restrictions may apply.

- > 1 per Parallel Sysplex

  Several 'plexes can coexist in the Parallel Sysplex. More than one 'plex may exist concurrently within the Parallel Sysplex. In certain cases, the 'plexes may partly overlap (for example, a system may belong to two JESplexes in that it may be in both a JES2plex and a JES3plex).

- > 1 Parallel Sysplex per 'plex

  Several Parallel Sysplexes can coexist in the 'plex. Or, put the other way around, the 'plex may comprise more than one Parallel Sysplex. Having a yes in this column does not always have an identical meaning. For a more complete discussion on this, refer to the last column in Figure 2-15 on page 49, which contains a pointer to a discussion pertaining to particular 'plexes.

- Non-z/OS extension

  z/OS can provide a single point of control for non-mainframe instances of the application.

'plex	'plex=Parallel Sysplex=sysplex	CF required	< Parallel Sysplex	> Parallel Sysplex	> 1 per Parallel Sysplex	> 1 Parallel Sysplex per 'plex	Non z/OS Extension
Sysplex	✔ ✔			✔		✔ (1)	
JESplex (chkpt in CF)	✔ ✔	✔	✔ (2)		✔		
JESplex (chkpt not in CF)	✔		✔	✔ (3)	✔	✔	
Pipeplex	✔ ✔ (6)	✔	✔		✔		
Batchplex	✔ ✔ (6)		✔	✔	✔		
GRSplex (star)	✔ ✔ ✔	✔					
GRSplex (ring)	✔			✔ (4)			
RACFplex (sysplex data sharing)	✔ ✔	✔	✔ (5)				
RACFplex (non sysplex sharing)	✔		✔				
SMSplex	✔	✔ (9)	✔	✔	✔	✔	
HSMplex (using VSAM/RLS CDS)	✔ ✔	✔	✔		✔ (8)		
HSMplex (non-data sharing)	✔		✔	✔	✔ (8)	✔	
BCSplex	✔ ✔ ✔	✔	✔				
OAMplex	✔ ✔	✔	✔		✔		
WLMplex	✔ ✔		✔	✔			
RMFplex	✔ ✔		✔	✔			
OPCplex	✔		✔	✔	✔ (7)	✔	✔
CICSplex	✔	✔ (10)	✔	✔	✔	✔	✔
VTAMplex(GR)	✔ ✔	✔	✔		✔		
VTAMplex (MNPS)	✔ ✔	✔	✔		✔		

Figure 2-15 'Plex summary

Comments related to Figure 2-15:

- Entries with:
  - Single checkmark indicate a *possible* configuration. Potential drawbacks are discussed in the section.
  - Double checkmarks indicate a *recommended* configuration.
  - Triple checkmark indicate a *mandatory* configuration.
- (1): This configuration is not recommended.

- (2): This configuration is possible if the JES2 MAS does not contain all the systems in the Parallel Sysplex.
- (3): For pre-V5 JES2 and JES3 releases, there is no restriction on which systems are in the JESplex. For JES2 and JES3 V5 and above, the JESplex can be greater than the Parallel Sysplex, but it cannot extend beyond the Base Sysplex.
- (4) Systems outside the Parallel Sysplex may not be part of other sysplexes.
- (5): If only a subset of the systems in the Parallel Sysplex are sharing the RACF database.
- (6): Batchplex must match the Pipeplex.
- (7): For the OPC standby function to work, all controllers must be in the same sysplex.
- (8): This option is available stating with DFSMS/MVS V1.5.
- (9): If using ECS, all sharing systems must be in the same Parallel Sysplex and same GRSplex.
- (10): CF required if using temporary storage, data tables, or named counter in CF support.

2.3 Dynamic workload balancing in Parallel Sysplex

With parallel processing, the workload can be distributed and balanced across a subset, or across all the CPCs, in the Parallel Sysplex. Dynamic workload balancing aims to run each arriving transaction in the *best* system. IBM's two major transaction management systems, IMS and CICS, due to their different structures, implement workload balancing in different fashions, as the following describes:

- IMS offers workload balancing by implementing algorithms that make it possible to *pull* messages from a sysplex-wide shared message queue. In IMS, you can also perform workload balancing using VTAM GR. Using VTAM GR is complementary to using shared queues. Generic resources distribute user logons across multiple IMS subsystems, while shared queues distributes the task of executing the transactions across multiple message processing regions associated with multiple IMS subsystems.

  IMS support for the z/OS Workload Manager (WLM) helps z/OS balance the workload in accordance with business objectives.
- CICS implements workload management by *pushing* transactions to different AORs. These AORs are in the same or on different images.

  In CICS, there are two aspects to workload management:

  – Workload separation: for example, using TORs, AORs, and FORs
  – Workload balancing: distributing log on requests and transactions across more than one CICS region

  Both may be in operation within a CICSplex. CICS may use three workload balancing techniques:

  – Goal mode algorithm, using CP/SM
  – Shortest queue algorithm, using CP/SM
  – Dynamic transaction routing program (either supplied by CICSPlex SM or by your installation)

  CICS TS can share CICS non-recoverable temporary storage sysplex-wide. When the temporary storage queues are shared by all images in the Parallel Sysplex, there are fewer affinities tying a transaction to a particular CICS region.
- MQSeries supports placement of Message Queues within a CF structure, thus allowing servers on multiple z/OS images to process messages from a single queue.

- z/OS WLM can dynamically manage jobs that are goal-oriented by distributing them to multiple systems in a sysplex, reducing operational demands and improving their total response time.
- z/OS implements workload balancing for TSO/E using generic resources.
- DB2 provides automatic work balancing through its sysplex query parallelism function. DB2 also has WLM-managed Stored Procedures. This facility allows you to let WLM manage the number of stored procedure address spaces, based on the importance of the stored procedure and the available capacity on the system.

  DB2 can distribute Distributed Data Facilities (DDF) requests received from a gateway across servers on different images, using information from WLM workload manager about CPC capacity available to those images. DB2 uses VTAM generic resources to route DRDA® session initialization requests across to DB2 subsystems in the data sharing group. Another helpful aspect of workload balancing is availability.

  If a z/OS failure occurs, only a portion of the logged-on users will be impacted, and subsequent transactions can be routed to another z/OS system that is unaffected by the failure.

2.4 CF architecture

Parallel Sysplex exploits CF technology. CF architecture uses hardware, specialized licensed internal code (LIC), and enhanced z/OS and subsystem code. All these elements are part of the Parallel Sysplex configuration. This section outlines the CF architecture.

The CF consists of hardware and specialized microcode (control code) that provides services for the systems in a sysplex. These services include common storage and messaging in support of data integrity and systems management by high speed processing of signals on links to and from the systems in the sysplex. The CF specialized microcode, called Coupling Facility Control Code (CFCC), runs in an LP in a stand-alone CF or a CPC. Areas of CF storage are allocated for the specific use of CF exploiters. These areas are called *structures*. There are three types of structures:

- *Lock*: For serialization of data with high granularity. Locks are, for example, used by IRLM for IMS DB and DB2 databases, by CICS for VSAM RLS, and by GRS star for managing global resource allocation.
- *Cache*: For storing data and maintaining local buffer pool coherency information. Caches are, for example, used by DFSMS for catalog sharing, RACF databases, DB2 databases, VSAM and OSAM databases for IMS, and by CICS/VSAM RLS. Caches contain both directory entries and optional data entries.
- *List*: For shared queues and shared status information. Lists are, for example, used by VTAM generic resources, VTAM Multi-Node Persistent Sessions, IMS shared message queues, system logger, the JES2 checkpoint data set, tape drive sharing, CICS temporary storage, and XCF group members for signaling.

There is also an area in the storage of the CF called *dump space*. This is used by an SVC dump routine to quickly capture serialized structure control information. After a dump, the dump space is copied into z/OS CPC storage and then to the SVC dump data set. The dump space can contain data for several structures.

For detailed information about Coupling Facility Configuration including links, refer to *Coupling Facility Configuration Options,* GF22-5042 on the Web site at:

`http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gf225042.pdf`

CF link types

CF links provide high bandwidth communication with copper or fiber optic cables between the CF and the connected CPCs. Currently, there are several types of links available:

- Internal Coupling Channel (IC): Has the highest bandwidth, which depends on the processor model. The IC emulates a CF link between sysplex images within the same CPC.
- Integrated Cluster Bus (ICB) copper cable, maximum distance of 7m between CPCs (cables is up to 10 m):
  - ICB (ICB), also known as ICB2, is not supported by zSeries or IBM System z9.
  - ICB-3 1 GBps on z900 (and z990/z9 to attach to z900).
  - ICB-4 2 GBps, different cards and a different connector to ICB3 (ICB3 to ICB4 upgrade is a hardware change) on z990 and IBM System z9.
- ISC fiber cable

  ISC-3 single mode only, 2 Gbps, maximum distance up to 10 km.

  Note: There is an RPQ available to extend the use of single-mode fiber to 20 km at 1 Gbps, RPQ 8P2197. This RPQ provides a physically different card to an ISC3 card. This requires the chargeable RPQ 8P2263 as a prerequisite, not necessarily on the same CPC. (See the RPQ manual.)

CFCC characteristics

- CFCC runs only under LPAR, or in a Coupling Facility Virtual Machine (CFVM) under VM. When running in an LP, the CPs serving the CF LP can be shared or dedicated.
- CFCC is loaded from the support element (SE) hard disk. When running in a CFVM, VM issues a call to the zSeries processor when the CF machine is logged on and uploads the CFCC into the CF virtual machine and IPLs it at LP activation. LPs are listed in IOCP along with their corresponding LP number. CF LP definitions with respect to storage, number, and type of CPs are defined independent of HCD or IOCP. CF link definitions are done in HCD or IOCP.
- The major CFCC functions are:
  - Storage management
  - Dispatcher (with MP support)
  - Support for CF links
  - Console services (HMC)
  - Trace, logout, and recovery functions
  - *Model* code that provides the list, cache, and lock structure support
- CFCC is not *interrupt-driven*. For example:
  - There are no inbound CF interrupts (the CF receiver link does not generate interrupts to CF logical CPs).
  - There are no outbound interrupts. The completion of asynchronous CF operations is detected by z/OS using a polling technique.

CFCC code is usually in a continuous loop (known as an *active wait*) searching for work.

Dynamic CF Dispatching (DCFD)

DCFD causes the dispatching algorithms to enable CFCC to share CPs in LPAR mode with other active workloads and only take CPC resources when needed. When there is no work to do, a CFCC partition using DCFD takes less than 2% of the shared CPs. For more information about Dynamic CF Dispatching, see *Parallel Sysplex Performance: Dynamic ICF Dispatching,* TD102670 in IBM Techdocs at:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD102670

2.4.1 Synchronous and asynchronous CF requests

Refer to the WSC Flash 10159, *z/OS Performance: New algorithm for synchronous to asynchronous conversion of CF requests with z/OS 1.2* for further details on this section. It can be found at:

ftp://ftp.software.ibm.com/software/mktsupport/techdocs/heuristic3.pdf

The CF is accessed through a privileged instruction issued by an z/OS component called cross system extended services (XES). The instruction refers to a subchannel. For a comprehensive discussion of the XES application interface, refer to *z/OS MVS Programming: Sysplex Services Guide,* SA22-7817 on the Web site at:

http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IEA2I630/CCONTENTS

The instruction is executed in *synchronous* or *asynchronous* mode. These modes are described as follows:

- Synchronous

  The CP in the image waits for the completion of the request. There are two types of synchronous request:

  - Synchronous Immediate
  - Synchronous Non-Immediate

  z/OS sometimes converts Synchronous Non-Immediate requests to Asynchronous requests; that is, it effectively modifies the XES macro issued by the requester. This is reported by RMF™ as *changed* requests.

  Figure 2-16 shows the conceptual flow for a synchronous request, such as a lock request to IRLM, issued by DB2 on behalf of an SQL user.

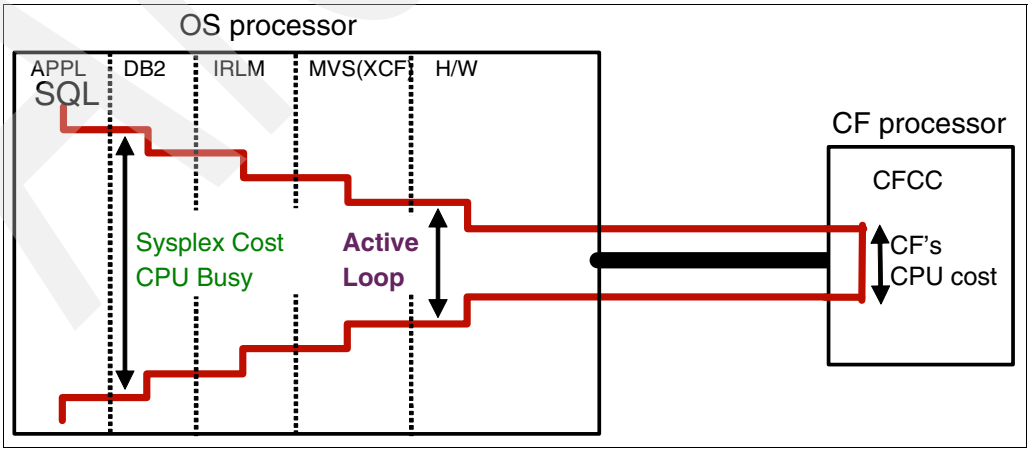


Figure 2-16 Flow in the case where the request for IRLM lock structure as sync operation

- Asynchronous

  The XES issuer issues the request, but does not wait for completion of the request. XES will either return a return code to the requester, who may continue processing other work in parallel with the execution of the operation at the CF, or XES will suspend the requester. XES recognizes the completion of the asynchronous request through a dispatcher polling mechanism and notifies the requester of the completion through the requested mechanism. The completion of the request can be communicated through vector bits in the hardware system area (HSA) of the CPC where the image is running.

  Figure 2-17 shows the conceptual flow for an asynchronous request, such as a batch unlock, issued by DB2 on behalf of an SQL user.

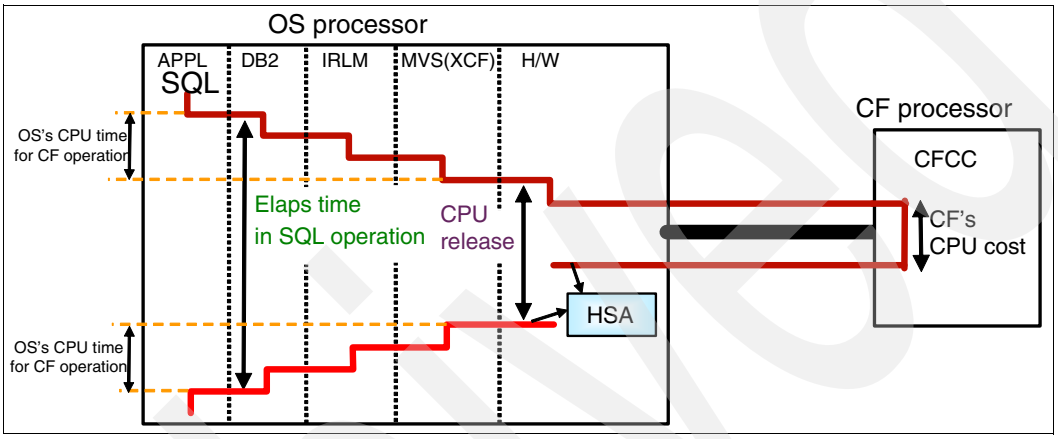


Figure 2-17 Flow in the case where the request for IRLM lock structures are changed into async operation

This is all from the point of view of the application. The idea, of course, is that synchronous requests are short and so it is not worth doing anything else until the request completes. However, the CF might be up to 100 km, which would increase the wait for a synchronous request by about 1000 microseconds, or the CF might be some much slower technology than the sender (for example, say, a z800 CF with a z9 sender). This would be a big waste of resources.

The operating system (that is, the XES subcomponent of z/OS or OS/390) itself has always changed some requests from synchronous to asynchronous. z/OS 1.2 introduced a new algorithm for determining whether or not it is more efficient to issue a command to the coupling facility synchronously or asynchronously, based on the configuration and workload. What is happening is a trade-off favoring improved host CPU capacity over a generally unnoticeable elongation of response time (as viewed by the user).

This algorithm is known as the heuristic sync/async conversion algorithm. This function monitors CF service times for all (LIST, LOCK, and CACHE) synchronous request types to a specific CF, also taking into account the amount of data transfer requested on the operation, and other request-specific operands that significantly influence the service time for the request. It compares these observed service times to thresholds to determine which operations would be more efficiently executed asynchronously. Different thresholds are used for simplex and duplex requests and for lock and non-lock requests. All thresholds are normalized based on the processor speed of the sending processor. The algorithm and thresholds are not externally adjustable.

Consider the following example, using figures from *System-Managed CF Structure Duplexing*, GM13-0103, found on the Web at:

http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130103.pdf

The host software time for a lock request on a z900-108 is 12 microseconds and the host hardware cost (that is, the time in the active loop) to a local z800 CF is 28 microseconds. So the total CPU cost of a synchronous lock request is 40 microseconds for a local CF. Suppose the CF is moved 10 km away, then each 1 km adds about 10 microseconds to response time and so synchronous response time would go up to about 140 ms. Prior to sync/async conversion, this would have been the actual CPU cost too. z/OS would now convert this request to an async request, which has a CPU cost of around 77 microseconds, and a response time of at least 177 microseconds (because of distance), let us say, 300 microseconds on average. If a typical transaction does 10 lock operations and lock response time increases from 140 microseconds to 300 microseconds, then transaction response time would elongate by 10 x. 000160=.0016 seconds and not likely to be noticed by the user. However, a batch job issuing 10 million requests would see elapsed time increase by 10,000,000*.000160=1600 seconds. However, the CPU time saved by the conversion for the batch job would be 10,000,000*(140-77)=630 seconds.

As shown in the example, conversion will not save the whole CPU time, which would have been used up in the synchronous operations active loop, but the longer the loop would have been, the more the savings, and also the more important the savings.

In a test, conducted using two z990 machines (2084-314) and two external CF machines (2084-304) at a distance of 1.5 km, *all* requests (including IRLM requests) to the CFs were asynchronous.

The new conversion algorithm has led to a significant change in the reporting of activity to some structures in the Coupling Facility RMF report. RMF does not report converted requests as changed; they are reported as asynchronous. The CHNGD field reports only the non-immediate requests that were changed because of a subchannel busy condition. The CHNGD count thus continues to be useful as an indicator of a shortage of subchannel resources. The RMF spreadsheet reporter now includes support to estimate the CF link subchannel busy condition.

Conversion also occurs if z/OS is running on shared CPs on the same CPC as a CF with shared non-ICF CPs. PR/SM may perform an *under the covers* conversion to asynchronous for both Immediate and Non-Immediate requests. z/OS is not aware of this conversion and it is not reported by RMF. The reason for the conversion is to prevent a potential deadlock from occurring. If z/OS can determine that such a deadlock will not happen, it uses a parameter on the XES macro to suppress the conversion by PR/SM. (For example, the z/OS system may be issuing the request to an external CF.)

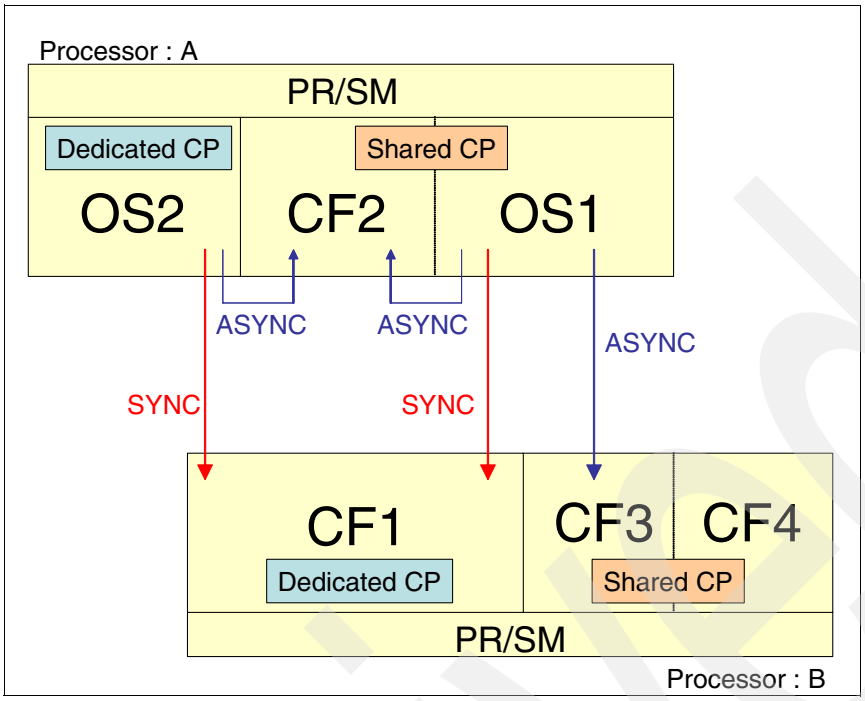


Figure 2-18 Sample configuration where requests are changed from a hardware environment

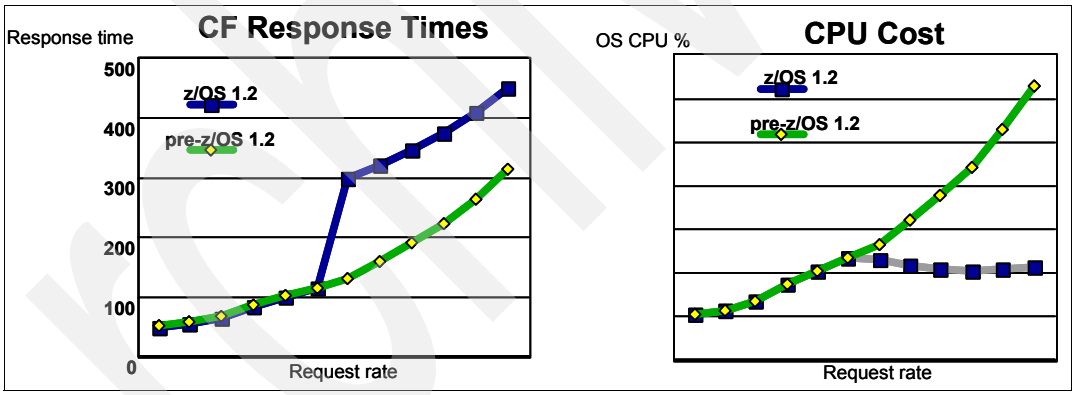


Figure 2-19 Effect on response time and z/OS CPU% of heuristic algorithm

As described so far, you might think that the heuristic algorithm is pretty static. Indeed, all requests to distant CFs will be converted, but response times to a local CF scan vary; for example, the CF may be handling a lot of requests from another operating system, or the load from a single operating system may grow, causing the familiar elongation of response times in busy systems. In that case, conversion will happen at a high load and will not at lower load levels.

Figure 2-19 illustrates two things. First, on z/OS 1.2 and higher, that as load increases, causing response time to increase, at a certain point synchronous requests are converted to asynchronous and request response time jumps. With earlier releases of the operating system, response time grew steady and did not suffer a sudden increase. Second, that on z/OS 1.2 and higher, the cost of a synchronous operation increased with increasing load, but at a certain point, it stopped increasing with load and flattened. On earlier releases of the operating system, the CPU cost of a synchronous request continued to grow as load increased.

2.4.2 XCF communications

Now we take a look at XCF and XES and what considerations have to be made when configuring a Parallel Sysplex.

XCF and XES consideration

XCF allows three methods for passing messages between members of XCF groups:

- Memory-to-memory: Between members of the same XCF group in a single image
- Channel-to-channel (CTC) connections: Between members of the same group on different images
- CF list structure: Between members of the same group on different images

The method chosen is an internal decision made by XCF based upon the target of the message and the available signalling paths. In this discussion, we only examine CTC and CF structure options. This is because they are the methods *most* relevant for consideration when configuring the Parallel Sysplex.

To ensure acceptable performance, it is vital that the CFs have sufficient capacity in terms of storage, links, and processing power. The CF configurations are set using the coupling facility resource management (CFRM) policy definitions. One of the considerations is the structure size. For details on the CFRM policy, refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

To create your own policies, we recommend that you use the Parallel Sysplex Configuration Assistant, available on the Web at:

`http://www.ibm.com/servers/eserver/zseries/zos/wizards/parallel/plexv1r1/`

The use of the tool to define the CF structures and policies is described in the redbook *S/390 Parallel Sysplex: Resource Sharing*, SG24-5666.

XCF signalling path considerations

Before we move on to the advantages of CF structures versus CTCs for signalling, look over the following terms, which are XCF-related terminology.

XCF group

A group is the set of related members defined to XCF by a multisystem application in which members of the group can communicate (send and receive data) between z/OS systems with other members of the same group. A group can span one or more of the systems in a sysplex and represents a complete logical entity to XCF.

Multisystem application

A multisystem application is a program that has various functions distributed across z/OS systems in a multisystem environment. Examples of multisystem applications are:

- CICS
- Global resource serialization (GRS)
- Resource Measurement Facility (RMF)
- Workload manager (WLM)

You can set up a multisystem application as more than one group, but the logical entity for XCF is the group.

Member

A member is a specific function (one or more routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group. However, a member is not a particular task and is not a particular routine. The member concept applies to all authorized routines running in the address space in which the member was defined. The entire address space could act as that member. All tasks and SRBs in that address space can request services on behalf of the member. Members of XCF groups are unique within the sysplex. Figure 2-20 shows the association between groups and members.

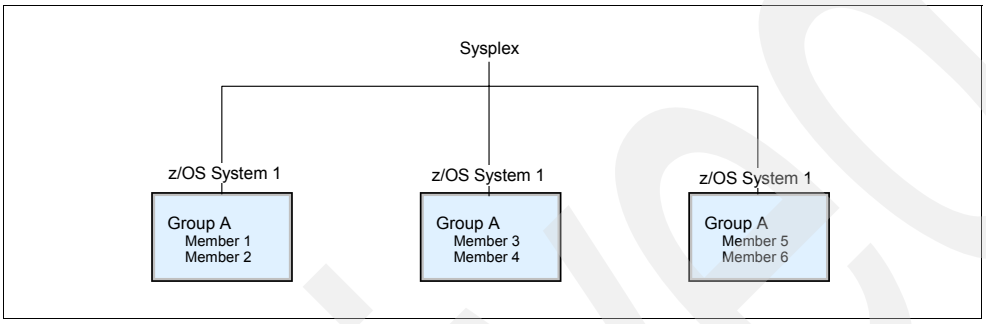


Figure 2-20 Groups and members

The two types of signalling paths of interest are:

- CTC connections
- CF list structures

When planning the signalling paths for XCF to use, remember to ensure that there is *redundancy* in whatever you configure.

For example, if you use CTCs for signalling, then you should define at least two paths into each system in the Parallel Sysplex and two paths out. This ensures that if one path should fail, there will always be an alternative path available.

When using CF list structures exclusively for XCF signalling (no CTC connectivity between systems exists), at least two signalling structures should be defined and they should be allocated in at least two CFs.

Recommendation for CF signalling structures: If two CF signaling structures are used and placed in two CFs, then you should define at least two paths into each system in the Parallel Sysplex and two paths out. Note that this refers to paths rather than links. Multiple paths can, and should, be defined on each physical link so that if you lose a link, you will still have a path-in and a path-out available on the other link. As a consequence, connectivity is maintained in case of problems with one structure or one link.

XCF supports the structure rebuild function of XES, but it will take some time if there is no alternative signalling connectivity (either CTCs or a second CF structure) available. XCF will use, as a last resort, the sysplex couple data set to accomplish the rebuild process synchronization. The XCF-internal signals used to coordinate the rebuild process generate heavy traffic to the couple data set. As a result, normal user-specified signals (for example, an instance of CICS sending a signal to another instance of CICS) may be delayed during rebuild of the last or only signalling structure in the sysplex.

Another good reason for having multiple signalling paths, either CTC or structure, is for *throughput* considerations. If XCF has multiple paths, it will not try to put a request onto an already busy path. Alternatively, you can decide to route messages for specific XCF groups along specific paths. This can be useful if one particular group is known to have specific signalling requirements, such as, it always uses long messages. The systems programmer could assign XCF groups to *transport classes* and then associate the transport class with a specific signalling path or paths.

For detailed information, refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Recommendation about XCF message sizes and signalling paths: When using XCF for the first time, assign all signalling paths to the default transport class. Then examine RMF XCF Activity reports to determine if a larger message buffer than that of the default transport class is needed, based on the percentage of messages that are too big for the default class length. This is indicated in the %BIG column.

If a message buffer is too *small* for a particular message request, the request incurs additional overhead because XCF must format a larger message buffer. If a message buffer is too *large* for a particular message request, main storage is wasted. XCF adjusts the size of the buffers in a particular transport class on a system-by-system basis based on the actual message traffic. Overhead is indicated in the %OVR column.

For example, in a given RMF interval, if 50,000 messages are sent from one system to another and the %SML=0, %FIT=90, %BIG=10 and %OVR=95, this means that 5000 messages were too big to fit in the existing message buffer and that 95% of the 5000 (big) messages incurred overhead from XCF buffer tuning functions.

If the amount of overhead from XCF buffer tuning is unacceptable, consider defining another transport class for the larger messages and assigning a signalling path or paths to that transport class to handle the larger message traffic. The following three transport classes work well in many cases and are, at the time of writing, recommended by the z/OS integration test team:

```
CLASSDEF CLASS(DEFAULT) CLASSLEN(20412) MAXMSG(750) GROUP(UNDESIG)
CLASSDEF CLASS(DEF8K) CLASSLEN(8124) MAXMSG(1000) GROUP(UNDESIG)
CLASSDEF CLASS(DEFSMALL) CLASSLEN(956) MAXMSG(750) GROUP(UNDESIG)
```

Note: It is generally not necessary to eliminate all buffer expansion and overhead when tuning XCF transport classes.

There are a few known common causes of large (64 KB) signals. WLM will occasionally send them. GRS GQSCAN processing can send them, and RMF Sysplex Dataserver processing can send them as well.

If the signalling paths are via XCF structures, then there are no means to figure out who is sending them. If you are using CTCs, then a GTF CCW trace may reveal some information. When you specify the value for CLASSLEN, subtract 68 bytes for XCF control information. XCF will round to the next 4 K increment.

Defining XCF transport classes with GROUP(UNDESIG) allows XCF to select which transport class (and their assigned signalling paths) messages should be transferred over based on message size.

The right number of signalling paths is indicated by the column BUSY in the RMF XCF Activity Report. If the busy count rises, the number of paths or the message buffer space for the outbound path should be increased. Experience shows that if all paths are made available to any transport class that needs them, the workload is spread evenly, and workload spikes are handled well.

Assigning groups such as GRS or RMF to a specific transport class is *not recommended* unless there is a proven need to do so.

Generally speaking, it is acceptable to mix the different transport classes on a single link. However, if you have tens of thousands of XCF messages per second, there may be a benefit to dedicating a transport class to a link, and have separate links for PATHIN and PATHOUT. Also, if you have very large XCF messages, you will see a greater benefit from the use of high-bandwidth links (such as IC links) than if your messages are small.

What type of signalling path should I use?

There are three possible options for the signalling between systems in a Parallel Sysplex:

- CTCs only
- CF structures only
- A combination of CTCs and CF structures

IBM used to recommend a mixture of CTCs and CF structures. With current technology, CF structures perform significantly better, are much easier to define, manage and have better recoverability and so are the normal recommendation.

For a more detailed discussion of this topic, refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

2.5 Data integrity and buffer pool consistency considerations

Data integrity is an important issue. First, we discuss data integrity prior to Parallel Sysplex (both in one system and in a multisystem environment). Then we discuss data integrity in a Parallel Sysplex.

2.5.1 Data integrity before Parallel Sysplex

When only one z/OS system has access to database records, z/OS data management products are able to address data integrity for thousands of active users. They apply a *database manager locking* technique to ensure data is valid for multiple tasks that may be accessing the same data. They use in-storage buffering to improve transaction performance, holding frequently used data and recently used data in buffer pools to avoid the overhead of reading data from DASD whenever possible.

In a multisystem environment, it is more complex for the systems to do locking and buffering than in a single system environment. When two or more systems need to share data, communication between those systems is essential to ensure that data is consistent and that each system has access to the data. One way that this communication can occur is through *message passing*.

Let us assume two images (S1 and S2) share the same database, as pictured in Figure 2-21 on page 61. When we start, the same record, ABC, is stored in each local buffer pool in each image.

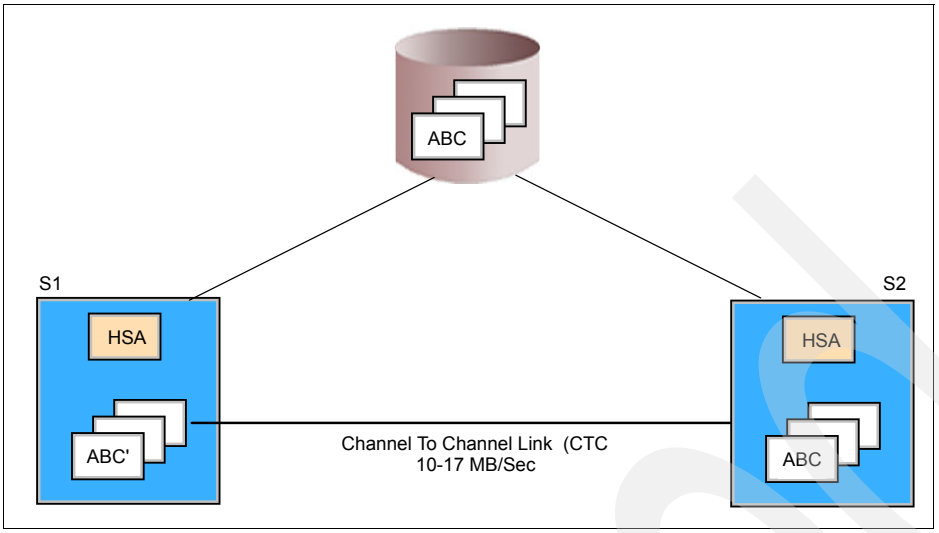


Figure 2-21 Multisystem data sharing

Let us assume that S1 needs to update the ABC record. S1 sends a message to obtain a lock across a channel-to-channel (CTC) link to S2, requesting access to the record. If S2 is not using the record, it sends a message back to confirm that S1 can have the access it requires. Consider two things about what was just described:

- Both S1 and S2 applications had to stop what they were doing to communicate with each other.
- If there are three or more systems, this scheme becomes even more complex. The *cost* of managing the sharing in this method is *directly proportional* to the number of systems involved.

This problem is even more complicated than just described. Suppose that S2 wants to read the ABC data and that the most recent copy of the record ABC is not yet written on the DASD file that both systems are sharing. It might be in an internal buffer of S1 (record ABC'). In this case, S2 must first be informed that the copy on the disk is not valid. S1 must then write it to DASD before S2 can have access to it.

The mechanism that tells S2 that the copy of the data in its buffer pool is the most recent one is called *maintaining buffer pool coherency*.

Some database managers offered such mechanisms before the Parallel Sysplex became available. For example, IMS DB provided two-way data sharing through IRLM (using VTAM CTCs for communication).

2.5.2 Data integrity in Parallel Sysplex

Parallel Sysplex provides, among other functions, the ability to manage database integrity (locking and buffering) for up to 32 images in a much simpler way than in a non-Parallel Sysplex multisystem configuration. We need to allow the sharing of data among many systems without causing the problems described in the prior example. Data must be shared in such a way that adding systems does not increase the complexity or overhead, and still preserves data integrity across updates. To solve this complex problem, a sharing technique is needed. This is implemented through the CF technology that manages the communication and serialization necessary for sharing of data. The CF function works with structures in the CFs that help manage the serialization of access to data and the coherency of the data.

z/OS provides a set of XES services that enable subsystems and authorized applications to use CF structures. These services provide buffer pool coherency and locking serialization, which allow data sharing to occur between many systems. For an example of how this works, look at Figure 2-22, which shows an example of how cache structures are used when two or more systems are sharing data.

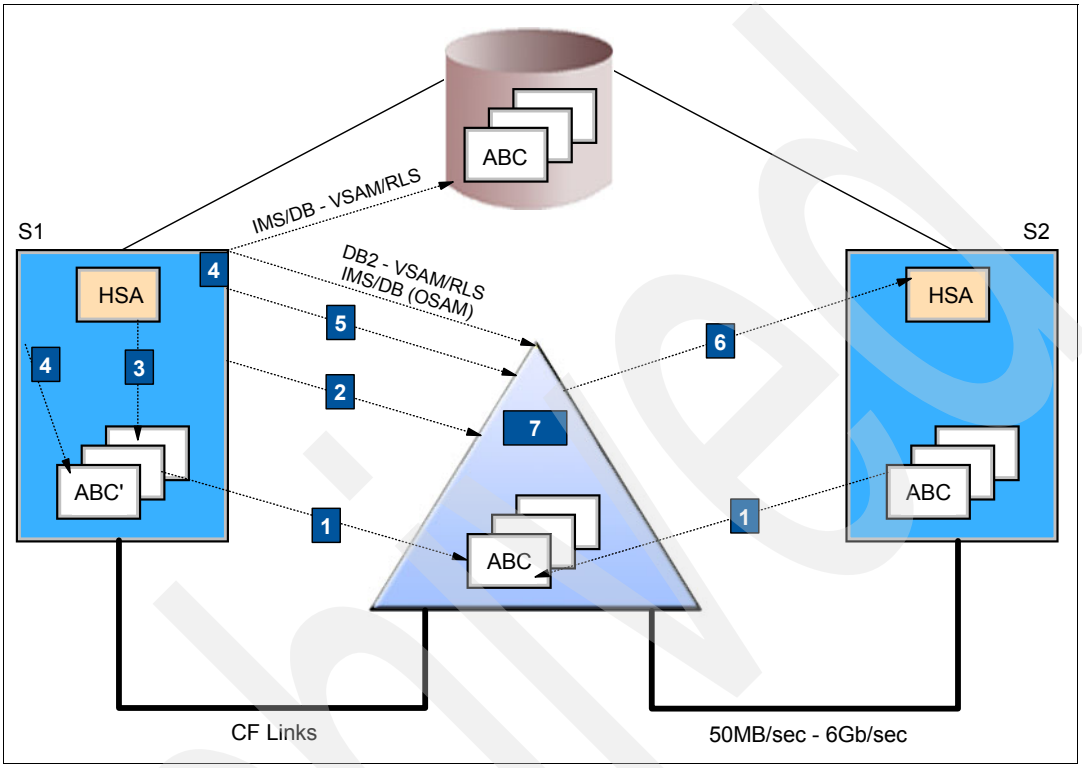


Figure 2-22 Multisystem data sharing in Parallel Sysplex

Two systems are sharing database records. Looking at the local buffer pool for each system, you can see some records (including the ABC record). Assume both systems have read record ABC in the past, and a valid current copy exists in each buffer pool:

1. Both systems have registered their interest in record ABC with the CF. Now an application in S1 (on the left) needs to update this ABC record to ABC'.
2. The DBM in S1 invokes an z/OS service that calls the CF to obtain an exclusive lock for the update. Depending on the subsystem implementation, the lock is obtained for one or more records. Assume that the lock is granted.
3. Now S1 looks in its HSA vector bit table associated with the buffer to see if the record in the buffer is a valid copy.
4. Assume it is valid, so S1 does not have to go to DASD or CF to obtain a copy.
5. S1 changes the record to ABC'. It is changed in S1's local buffer, and depending on the subsystem protocol, it is written to:
   - A cache structure of the CFs.
   - An example is IMS DB (OSAM using a store-in technique) or DB2.
   - On a per-tablespace basis.

DB2 provide several options for how data is cached in the structure:

- All data.
- Changed data only.
- Part of the data (space map pages for LOBs).

For duplexed DB2 cache structures, the data is also written to the secondary structure, but asynchronously.

– A cache structure in the CF *and* to DASD. Examples are VSAM/RLS and IMS DB V5 (OSAM) using a store-through technique.

– DASD only. An example is IMS DB V5 (VSAM) using a directory-only technique or DB2 using the *no cache* option.

6. A signal is sent by the database manager to the CF to show that this record has been updated. The CF has a list, stored in a cache structure directory, of every system in the Parallel Sysplex that has a copy of the record ABC, which has now been changed to ABC'. A directory entry is used by the CF to determine where to send cross-invalidation signals when a page of data is changed or when that directory entry must be reused.
7. Without interrupting any of the other systems in the Parallel Sysplex, the CF invalidates all of the appropriate local buffers by changing the bit setting in the HSA vector associated with the record to show that the record ABC in the local buffer is down level. This operation is called cross-invalidation. Buffer invalidation may also occur under other circumstances, such as when a contention situation is resolved through directory reclaim.
8. At this point, the serialization on the data block is released when the update operation is complete. This is done with the use of global buffer directories, placed in the CF, that keep the names of the systems that have an interest in the data that reside in the CF. This is sometimes referred as the unlock operation.

In a store-in operation, the database manager intermittently initiates a cast-out process that off-loads the data from the CF to the DASD.

The next time the images involved in data sharing need record ABC, they know they must get a fresh copy (apart from S1, which still has a valid copy in its local buffer). The systems are not interrupted during the buffer invalidation. When the lock is freed, all the systems correctly reflect the status of the buffer contents.

If S1 does not have a valid copy in the local buffer (as was assumed in step 3), it must read the record from either the cache structure in the CF or from DASD. The CF only sends messages to systems that have a registered interest in the data, and it does not generate an interrupt to tell each system that the buffer is now invalid.

The example described above details the steps necessary every time a record is referenced. Note that the shared locking also takes place if a system reads a record without ever doing an update. This guarantees that no other systems update the record at the same time. The example discussed here can easily be expanded to more than two systems.

A summarization of the event sequence needed to access a database with integrity in a Parallel Sysplex includes:

1. An element, or list of elements, is locked before it is updated.
2. An element is written to DASD or the CF no later than sync point or commit time.
3. Buffer invalidation occurs after the record is written.
4. Locks are released during sync point or commit processing, and after all writes and buffer invalidations have occurred. The locks may be released in batches.

2.5.3 Locking in Parallel Sysplex

An important concept when designing a locking structure is the *granularity* of the database element to be serialized. Making this element very small (such as a record or a row) reduces lock contention, but increases the number of locks needed. Making the element bigger (such as a set of tables) causes the opposite effect. There are options in IMS DB, CICS/VSAM RLS, and DB2 where the installation can define this granularity. Refer to the following sections for more information:

- 4.3.2, “IMS DB data sharing” on page 236
- 4.3.3, “CICS/VSAM record level sharing considerations” on page 240
- 4.3.1, “DB2 data sharing considerations” on page 228

To avoid massive database lock structures in the CF, a *hashing technique* is used. Every element in the database to be locked has an identifier (ID). This ID is hashed, and the hash result is an index into the lock structure. Because there are more database elements than lock entries, there is a probability of *synonyms*. A synonym is when two or more distinct elements point to the same lock entry. When this situation happens, all the subsystems from different images that are using this lock structure need to perform lock resource negotiation. After this negotiation, they decide if the contention was *false* (just a synonym) or *real* (two subsystems trying to get access to the same lock).

Note that the lock requester may or may not be suspended until it is determined that there is no real lock contention on the resource. For example, the requester may be processing other work in parallel while XES makes the determination. Contention can be a performance problem for workloads that have a high level of sharing (locking) activity.

Total contention is the sum of false and real contention and should be kept low.

Recommendation for locking contention: Total locking contention should be kept to less than 1.0% of the total number of requests. If possible, try to keep false contention to less than 50 percent of total global lock contention. If total global lock contention is a very low value, it might not be as important to reduce false contention.

Real contention

Real contention is a characteristic of the workload (which is the set of locks that are obtained, lock hold time, and the nature of workload concurrency), and is not easy to reduce by tuning the Parallel Sysplex configuration. It is tuned by tuning the workload itself. For example, you can try to avoid running long batch jobs concurrent with online work, avoid running concurrent batch jobs that access the same data, increase the frequency of checkpoints, accept *dirty reads*, and so forth.

Real contention is also reduced by controlling the degree of data sharing. CP/SM and WLR may, for example, help to make sure that transactions needing certain data are run on certain systems.

False contention

False contention has to do with the hashing algorithm that the subsystem is using, and the size of the lock table in the lock structure. It is true that increasing the size of the lock table in the lock structure reduces false contention. Carrying this to extremes, by making the lock table arbitrarily large, one can make the false contention become as small as one wants. This has practical limits, since, for IRLM, the only way to increase the size of the lock table is to increase the size of the lock structure as a whole. To minimize the effect of false contention, there are three options:

- Increase the lock structure size in the CF.

- Decrease the granularity of locking, or in other words, reduce the number of locks. However, by decreasing the granularity of locking, you increase the probability of real contention at the expense of the false contention. Therefore, this option might not always be considered a solution.
- Decrease the number of users connecting to the lock structure, thus allowing more lock entries within a given structure size.

2.6 System-Managed CF Structure Duplexing

For more information about CF Duplexing function. refer to the following technical documents.

- *System-Managed CF Structure Duplexing*, GM13-0103:
  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130103.pdf
- *System-Managed CF Structure Duplexing Implementation Summary,* GM13-0540:
  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130540.pdf
- *Coupling Facility Configuration Options,* GF22-5042:
  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gf225042.pdf
- Use the self-assessment questionnaire available on Resource Link™ at:
  http://www.ibm.com/servers/resourcelink

Parallel Sysplex technology can provide many benefits to the zSeries environment, including high availability, workload balancing, scalable growth, reduced cost of computing, ease of use, and investment protection of current applications.

System-Managed Coupling Facility Structure Duplexing (referred to as *CF Duplexing* within this redbook) enhances this by providing a robust general purpose, hardware assisted, easy-to-exploit mechanism for duplexing CF structure data. It provides a robust recovery mechanism for failures such as loss of a single structure or CF, or loss of connectivity to a single CF, through rapid failover to the other structure instance of the duplex pair with low exploitation cost.

CF Duplexing is designed to provide high availability in failure scenarios via the redundancy of duplexing. The benefits of CF Duplexing are:

- Availability: Faster recovery of structures by having the data already in the second CF.
- Manageability and Usability: A consistent procedure to set up and manage structure recovery across multiple exploiters
- Enablement: System-Managed Duplexing of structures that have no user-managed rebuild capability (for example, CICS Temp Storage and WebSphereMQ Shared Queues)
- Configuration (Failure-Isolation) and Cost Benefits: Having a duplex copy of a structure removes the requirement for failure-independence. It enables the use of non-stand-alone CFs (for example, ICFs) for all resource sharing and data sharing environments.

IBM does not recommend CF duplexing in all cases, nor should it be used for all structures that support it. A cost/benefit case should be looked at first. There is a performance cost because more work is done, z/OS needs to send two requests instead of one, and there is additional CF-CF communication.

2.6.1 Which structures should be duplexed?

There are varied reasons for CF duplexing, but the following structures are likely candidates for duplexing:

- Structures that do not support user-managed rebuild for recovery purposes, for example, CICS temporary storage structures.
- Logger structures that currently use DASD for staging datasets can benefit greatly from CF Duplexing.

Other structures may be duplexed for other reasons; structures that require failure-isolation from exploiting z/OS images can be located in ICFs on participating z/OS CPCs if CF Duplexing is used.

Figure 2-23 shows two stand-alone CFs with dedicated engines, while Figure 2-24 shows two ICFs with dedicated engines.

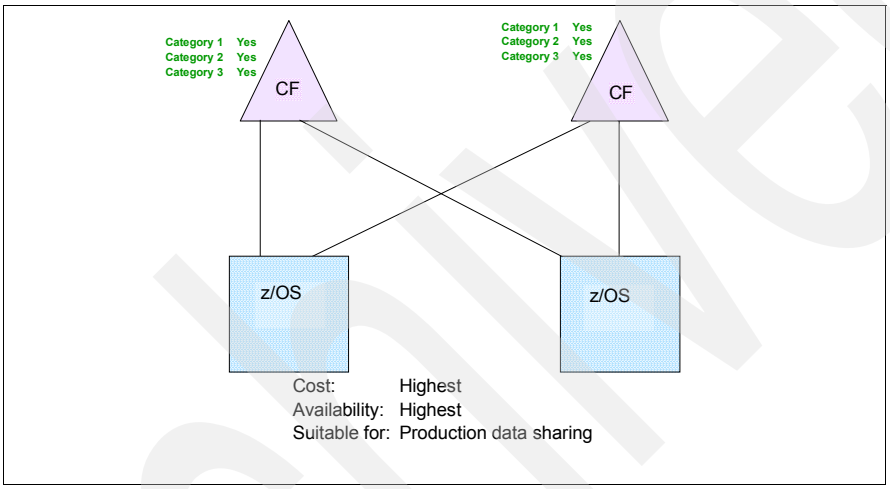


Figure 2-23 Two stand-alone CFs with dedicated engines

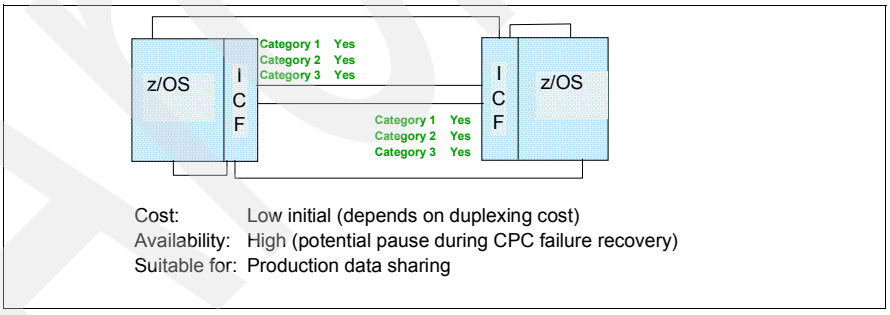


Figure 2-24 Two ICFs with dedicated engines

Before CF Duplexing was supported, z/OS and OS/390 provided several recovery mechanisms covering hardware failure scenarios (including CFs, CF links, and processors). Each method offered its own availability advantages and disadvantages:

- No recovery
  - ICICS temporary storage queues
  - MQseries Shared Queue (non-persistent messages)

- DASD backup
  - Data in a directory only cache
  - System Loggers use of staging datasets to maintain a second copy of the logstream from the time it is written to the CF until the time it is offloaded.
- User-managed rebuild (introduced in MVS/ESA SP5)

  Usually these structures require failure isolation separating the structure and its servers into separate CPCs. An example is the IRLM lock structure.
- User-managed duplexing (introduced in OS/390 R3 APAR OW28640)

  This was used by DB2 GBP.
- System-managed rebuild (introduced in OS/390 R8)

  This provides only a planned reconfiguration capability.

Before CF Duplexing, the recommendation for data sharing sysplexes was to have at least one and preferably two stand-alone CFs. The stand-alone CF contains all structures that require failure isolation. A second CF, on an ICF on one of the z/OS CPCs, can be used to contain the CF structures not requiring failure independence. That CF is also used as backup in case of failure of the stand-alone CF. This was and still is viewed as a robust configuration for application enabled datasharing.

Figure 2-25 depicts a recovery scenario for IRLM lock structures assuming a stand-alone CF is lost. Because the connectors survive, they can rebuild the lock structure in the backup CF. Similarly, if one of the z/OS processors is lost, then the application can be restarted, since the lock information is intact.

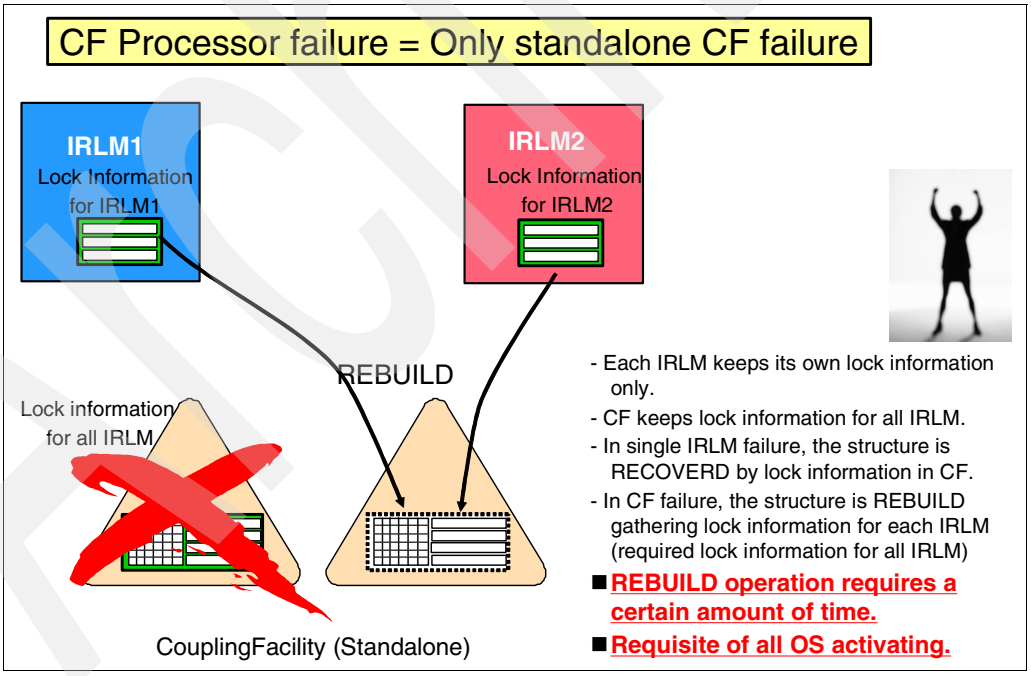


Figure 2-25 Recovery scenario by rebuilding of IRLM lock structure in CF failure

Figure 2-26 on page 68 illustrates the failure of a CPC that hosts both a z/OS image with IRLM and a CF LPAR containing the lock structure. When the CPC fails, the lock structure cannot be rebuilt in the surviving CF because both the lock structure and one of the connectors has been lost. A database recovery will be required.

From this, we can see that a high availability sysplex must have at least three CPCs (two CPCs for operating system images and one stand-alone CF), or use CF duplexing.

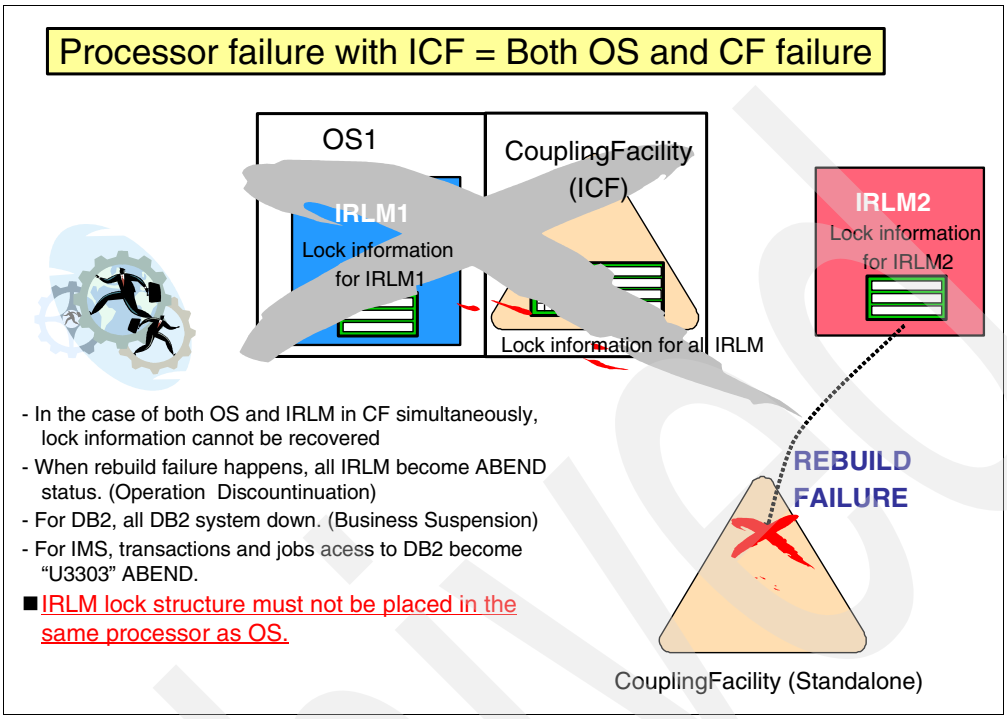


Figure 2-26 RLM lock structure in processor failure (OS and ICF with lock structure failure)

2.6.2 What is System-Managed CF structure duplexing?

With CF Duplexing, two instances of the structures exist, one on each of the two CFs. This eliminates the single point of failure when a data sharing structure is on the same server as one of its connectors.

Figure 2-27 on page 69 depicts how a request to a System Managed duplexed structure is processed. *User-managed* duplexing of DB2 Group Buffer Pools does not operate in this way.

1. A request is sent to XES from the application or the subsystem that is connected to a duplexed structure. The exploiter does not need to know if the structure is duplexed or not.
2. XES sends requests 2a and 2b separately to the primary and secondary structures in the CFs. XES will make both either synchronous or asynchronous.
3. Before the request is processed by the CFs, a synchronization point is taken between the two CFs, 3a and 3b.

> **Note:** In User-managed duplexing mode, the request to the secondary structure is always asynchronous.

4. The request is then be executed by each CF.
5. After the request is performed, a synchronization point is taken between the CFs again.
6. The result of the request is returned from each CF to XES. These, 6a and 6b, are checked for consistency.
7. Finally, the result is returned to the Exploiter.

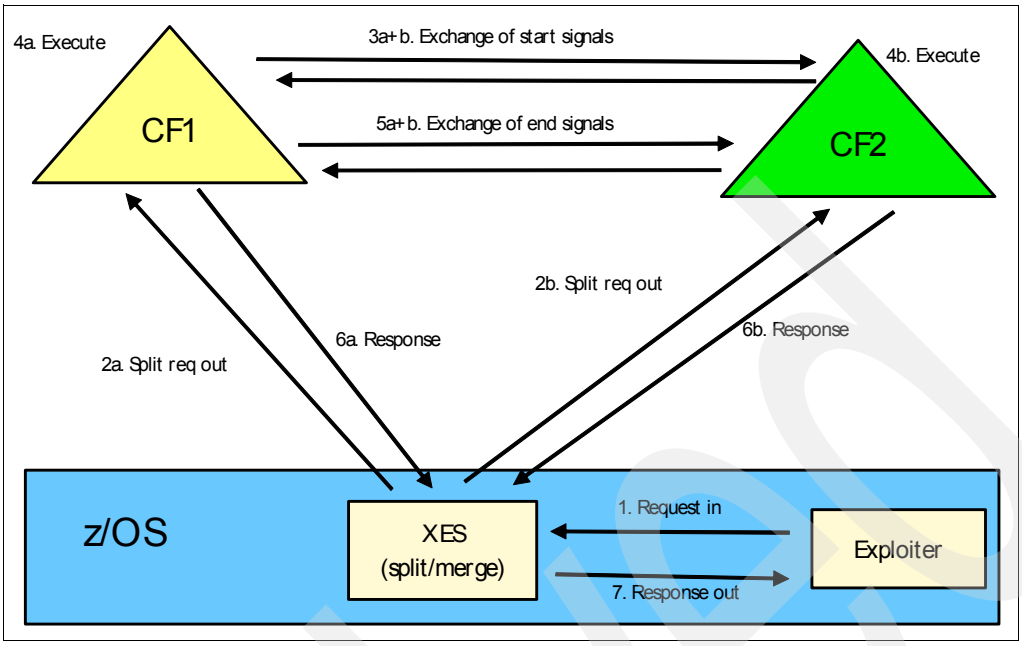


Figure 2-27 Request process for CF duplexing structures

2.6.3 Hardware and software requirements for CF duplexing

Here we discuss the hardware and software requirements for CF duplexing.

Key prerequisites

- Any of the following IBM eServer systems and internal or external coupling facilities:
  - Model 9672 G5 or G6 with Driver 26 at the current service level and CFCC Level 11. In practice, the use of these is not recommended.
  - Any zSeries or IBM System z9 processor.
  - CF to CF connectivity via coupling links.
- Additional prerequisites are listed in the technical paper *System-Managed CF Structure Duplexing*, GM13-0103.
- z/VM V3.1 or above (if you want to use System-Managed CF Structure Duplexing under z/VM).

CFCC levels

Recommended Coupling Facility Levels (CFLEVEL) to support CF duplexing are:

- CFLEVEL 14 for IBM System z9, z890, and z990.

  Aside from functional benefits, the CF CPU consumption is significantly lower on CFCC14 than CFCC 13.
- CFLEVEL13 for z800 and z900.

CF links

Preparations for CF duplexing include the requirement to connect coupling facilities to each other using coupling links. The required connectivity is bi-directional with a sender and receiver channel attached to each CF for each remote CF connection.

- For peer-mode links (ICP, CFP, and CBP), a single channel provides both the sender and receiver capabilities, that is, only one physical link is required between each pair of CFs.
- For compatibility-mode links, two links are required to establish a connection with a sender and receiver channel located in each CF.

Note: For redundancy, two peer-mode links or four compatibility-mode links are required (some of these may already be present).

Subsystems supporting System-Managed duplexing

Important: All relevant APARs listed in the CFDUPLEXING PSP bucket are required.

All supported versions of z/OS, DB2, CICS, IMS, and MQ support system-managed duplexing.

2.6.4 Which structures should be duplexed?

Figure 2-28 on page 71 shows the structures supported by User-Managed and system managed CF Duplexing. For more information, refer to *System-Managed CF Structure Duplexing*, GM13-0103. And refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625, *Parallel Sysplex Application Considerations*, SG24-6523.

The obvious candidates are those that do not support User Managed Rebuild, and to a lesser extent, those requiring failure isolation. The discussion below is based on *System-Managed CF Structure Duplexing*, GM13-0103.

Subsystem, Product or Function	Structure	Structure Type	User Managed Rebuild	System Managed Duplexing
Allocation	Shared Tape	LIST	YES	NO
Batchpipes	Multi-system pipes	LIST	YES	YES
Catalog	Enhanced Catalog Sharing	LIST	YES	NO
CICS	DFHLOG - Logger	LIST	**FAIL-ISOL**	**YES**
CICS	DFHSHUNT - Logger	LIST	**FAIL-ISOL**	**YES**
CICS	FWD Recovery - Logger	LIST	**FAIL-ISOL**	**YES**
CICS	Temp Storage	LIST	**NO**	**YES**
CICS	Shared Data Tables	LIST	**NO**	**YES**
CICS	Named Counter	LIST	**NO**	**YES**
DB2	SCA	LIST	**FAIL-ISOL**	**YES**
DB2	GBP	CACHE	YES	NO* Supports User Managed Duplexing
DB2	IRLMLOCK	LOCK	**FAIL-ISOL**	**YES**
GRS	Star	LIST	YES	NO
IMS	IRLMLOCK	LOCK	**FAIL-ISOL**	**YES**
IMS	VSO	CACHE	**FAIL-ISOL**	**YES**
IMS	OSAM	CACHE	YES	YES
IMS	VSAM	CACHE	YES	NO
IMS	CQS	LIST	YES	YES
IMS	CQS Logger	LIST	**FAIL-ISOL**	**YES**
IMS	CQS Logger (EMH)	LIST	**FAIL-ISOL**	**YES**
JES2	Checkpoint	LIST	**NO**	**YES**
MQSeries	Shared Queues	LIST	**NO**	**YES**
z/OS Operlog	Logger	LIST	**FAIL-ISOL**	**YES**
z/OS Logrec	Logger	LIST	**FAIL-ISOL**	**YES**
RACF	Shared DB	CACHE	YES	NO
RRS	Logger	LIST	**FAIL-ISOL**	**YES**
DFSMS	HSM Common Recall Queue	LIST	**NO**	**YES**
DFSMS	RLS Cashe	CACHE	YES	NO
DFSMS	RLS Lock - IGWLOCK00	LOCK	**FAIL-ISOL**	**YES**
VTAM	Generic Resource	LIST	**FAIL-ISOL**	**YES**
VTAM	Multi-Node Persistent Sessions (MNPS)	LIST	**FAIL-ISOL**	**YES**
WLM	IRD	LIST	**NO**	**YES**
WLM	Enclaves	LIST	**NO**	**YES**
XCF	Signaling	LIST	YES	NO

Figure 2-28 Recovery support for structures

To achieve high availability, *strongly recommended* structures for CF duplexing are:

- MQ Shared Message Queueing and CICS Temporary Storage/Shared Data Tables/Name Counter Server

  These structures are *not* supported for user-managed rebuild (these structures are highlighted in red in Figure 2-28). So high availability requires CF duplexing function to protect against a CF failure.

On the other hand, the following structures can be considered as *good* because of the improvement in failure recovery service level:

- IMS Common Queue Server (CQS)

  In view of the recovery time for a structure failure, these structures benefit from CF Duplexing function (rather than recovering from logs in DASD datasets).

  Recovery of the SMQ structure is similar to a database recovery process. It involves:

  – Reallocating the structure
  – Restoring the structure from the DASD image copy in the structure checkpoint data set
  – Applying changes from the System Logger structure

  Recovery of SMQ structures requires the user to take periodic *structure checkpoints* to DASD. Activity to the structure is quiesced during this time, which may be several seconds, depending on the size of the structure. For more detailed information, refer to *System-Managed CF Structure Duplexing*, GM13-0103. This form of recovery could take some time, and that is why CF duplexing is preferable.

- Common Logger structures between systems (RRS, CICS User journal, and IMS CQS Logger)

  These structures use staging datasets for availability but performance will worsen dramatically if the structure fails. So for applications (subsystems) using logger, you should decide whether you want these structures duplexed or not.

  Note: In a GDPS/PPRC multi-site environment, System Managed Duplexing should not be used to duplex CFs in different sites. The distance effects are highly undesirable and CF structure data is not preserved if a GDPS site failover occurs.

- VTAM Generic Resources and VTAM Multi-Node Persistent Sessions (MNPS)

  The cost of duplexing for these is low, as they are not frequently updated and they can be rebuilt quickly on their own. But if there were a lot of other structures being rebuilt at the same time, it might take a couple of minutes for the rebuild. During this time, while VTAM Generic Resources is unavailable, users would be unable to log off or establish a session even if a specific name is used. LU requests are queued until the rebuild completes.

For the following structures, the CF Duplexing function is supported, but is considered to be optional in general:

- Logger structures that are not shared between systems (CICS DFHLOG, DFHSHUNT)

  The structures need only to be failure isolated.

- System Logger structures (OPERLOG, LOGREC)

  These structures can use staging datasets, and user applications can continue if a structure failure occurs.

2.6.5 System-Managed CF duplexing overheads

In a sysplex environment with duplexed structures, simplex read operations will cost the same, since z/OS will send a message to only one CF as before. However, the cost of write and update operations and any type of operation that modifies a structure will increase compared with simplex structures, because there are two CF requests, two request completions, and the results have to be reconciled (this process is shown in Figure 2-27 on page 69).

Additional coupling facility storage resources and processor capacity, as well as additional z/OS to CF link capacity, may need to be added to support CF Duplexing, because two structure instances will be allocated for an extended period of time and for each duplexed coupling facility operation, two operations will be driven, one for each structure.

Potential configuration/setup cost:

- CF storage resources for duplexed structure instances
- CF-to-CF links
- System-to-CF links
- System and CF processor resources

Coupling efficiency cost:

- Duplexed command service time *deltas* resulting from the duplexing protocols
- z/OS (XES) pathlength deltas resulting from splitting/merging duplexed commands
- Impact varies with request rate and read/write ratio of workload

The overheads of CF Duplexing can include the following:

- Increased z/OS CPU utilization

  For those operations that update the structure:

  – Instead of paying the software cost to send and receive one CF request, two requests are being sent and two responses received with the results reconciled.

  – Additional pathlength is required to overlap the sending of the requests.

  – For synchronous requests, the host has to wait until both requests complete. This has the effect of increasing the synchronous response time, directly affecting host CPU utilization. Some requests may be converted to asynchronous (new function in z/OS V1R2) to reduce the impact.

- Increased coupling facility CPU utilization

  For those operations that update the structure:

  a. The coupling facility containing the original (*old*, or *primary*) structure continues to process requests as when running in simplex mode.

  b. The coupling facility containing the new structure now has to process requests that update the duplexed structure. The impact of processing these requests should have already been planned for (as *white space*) to handle a CF rebuild situation in a simplex environment.

  c. Additional CF usage for both CFs is incurred to handle the CF-to-CF communication to coordinate the updates. This communication is done to enable both images of a structure to remain synchronized with each other.

- Increased coupling facility link/subchannel utilization

  For those operations that update the structure:

  a. There will be additional traffic on the links due to the additional requests to the new (or *secondary*) structure.

  b. The CF to CF communication requires CF links.

  c. Since the z/OS to CF response times increase due to the CF to CF communication, the z/OS CF link subchannel utilization will increase.

  d. The increased response could result in requests that would be synchronous if they were simplex being issued asynchronously when duplexed, further increasing subchannel utilization.

Tips:

- Some of the additional resource requirements may be satisfied by excess capacity already in place for high availability and failover capability.
- After the failure of a duplexed CF environment, performance may improve because the structures operate in simplex mode until the failed component is restored.

2.6.6 z/OS and CF CPU cost of duplexing

Table 2-2 shows duplexing costs relative to the cost for a simplex structure. UM GBps denote DB2 GBps. SM List and SM Lock are System Managed List and System Managed Lock.

Table 2-2 Duplexing costs

Structure type	Host CPU busy	CF CPU busy	CF link subchannel busy	Percent update
UM GBps	2x	2x	2x	1%-100% avg 20%
SM Lock	4x	5x	8x	100%
SM List	3x	4x	6x	near 100%
Asynchronous operations	2.5	2.5		

Capacity Planning for hardware, CFs processors, and CF links

Note: For detailed information about capacity planning and examples, refer to the white paper *Coupling Facility Configuration Options,* GF22-5042.

CPU capacity of z/OS (CPC)

Duplexing will increase z/OS CPU usage. This is very dependent on the number (rate) of requests to the duplexed structure, the percentage of update requests, and the structure type being duplexed. For example, in the case of duplexing a DB2 data sharing primary production application, this OS capacity could increase about 5-10% in total z/OS CPU compared with the simplex case.

You may think that this increase in z/OS CPU is lower than expected. Remember, DB2 GBps do not use CF duplexing and so the additional overhead is mainly for duplexing the lock structure. *System-Managed CF Structure Duplexing*, GM13-0103 contains a worked example where DB2 datasharing overhead is estimated to increase from 10% to 19% when migrating to CF duplexing.

Figure 2-29 on page 75 gives rough estimates of the effects that the various technologies have on the z/OS capacity cost of coupling. The z/OS host technology is listed across the top and the CF technology at the side. The table gives an estimate (for example) of 9% for a z990 host connected to a z990 CF by an ICB link. Note that in practice the overhead on this table should be capped at 15%, because of the sync to async heuristic algorithm. We have not done this to allow you to scale up and down more easily.

The chart is based on nine CF operations/MIP and other access rates would scale accordingly. The actual current CF access rate/MIP for each CF in an installation is easily calculated from the RMF Structure Activity Report for that CF and the CPU Activity Report for

the LPAR. The values are then linearly scaled and totaled. It will often be the case that though the technology is all the same, the link types are different (IC and ISC, for example).

Host CF	G3	G4	G5	G6	z800	z900 1xx	z900 2xx	z890	z990
C04-SM	10%	11%	16%	19%	21%	22%	25%	---	---
C05-HL	9%	10%	14%	16%	18%	19%	22%	26%	30%
R06-HL	9%	9%	12%	14%	16%	17%	19%	22%	26%
R06-ICB	---	---	9%	10%	---	13%	14%	17%	20%
G5/6-IC	---	---	8%	8%	---	---	---	---	---
z800 ISC	9%	9%	11%	12%	11%	12%	13%	15%	18%
z800 ICB/IC	---	---	---	---	9%	10%	11%	12%	14%
z900 ISC	8%	9%	11%	12%	10%	11%	12%	14%	16%
z900 ICB/IC	---	---	8%	9%	8%	9%	10%	11%	12%
z890 ISC	7%	8%	8%	9%	9%	10%	11%	13%	15%
z890 ICB/IC	---	---	8%	8%	7%	8%	8%	9%	10%
z990 ISC	7%	8%	8%	9%	9%	10%	11%	13%	14%
z990 ICB/IC	---	---	8%	8%	7%	8%	8%	9%	9%

Figure 2-29 Effects on host processor of CF processor and CF links (in simplex environment)

z/OS storage

This size will not change much whether the structures are duplexed or not, assuming the structure has at least two CFs in its candidate list.

CF CPU capacity

In both simplex and duplexed environments, the best performance occurs when the coupling facility is less than 50% CPU busy. At utilizations higher than this, significant queuing within the CF can occur, elongating response time and increasing z/OS CPU usage for synchronous activity.

With this guideline in mind, some installations may have already planned spare CF CPU capacity (*white space*) for recovery situations by a user-managed rebuild operation. But the CPU capacity for white space that would have been needed during a failure scenario is already being used on an ongoing basis with CF Duplexing.

CF CPU requirements are much higher compared with simplex structures. This CPU cost varies with the kinds of structures, but it can use about 2x to 5x times more CF CPU for duplex compared to simplex requests. After deciding structure placement in each CF, this can be estimated based on the expected number (rates) of requests and service time to the structures. If the simplex structures exist, then the duplexing cost can be estimated using figures from RMF.

CF link capacity

When moving to duplex structures, the processor, CF, and CF links may need to be upgraded. The new response times you can expect are largely a factor of the CF type and CF Link type. Figure 2-30 gives an indication of relative times for different configurations.

CF and Link type	Lock request	Cache/List request with 4K data transfer
z890-ISC3-z890	3.33	6.88
z890-ICB3-z890	1.50	2.50
z890-ICB4-z890	1.42	1.75
z990-ISC3-z990	3.17	6.38
z990-ICB2-G5	3.00	5.75
z990-ICB3-z900	2.00	3.50
z990-ICB4-z990	1.33	1.75
z990-IC-z990	1.00	1.00

Figure 2-30 Relative response times for the different configurations

Storage size of CF

In a sysplex environment with simplex structures, this size is estimated to be able to rebuild all the structures into one CF. In the case of duplexing structures, capacity planning is the same as the simplex environment, as in Figure 2-31.

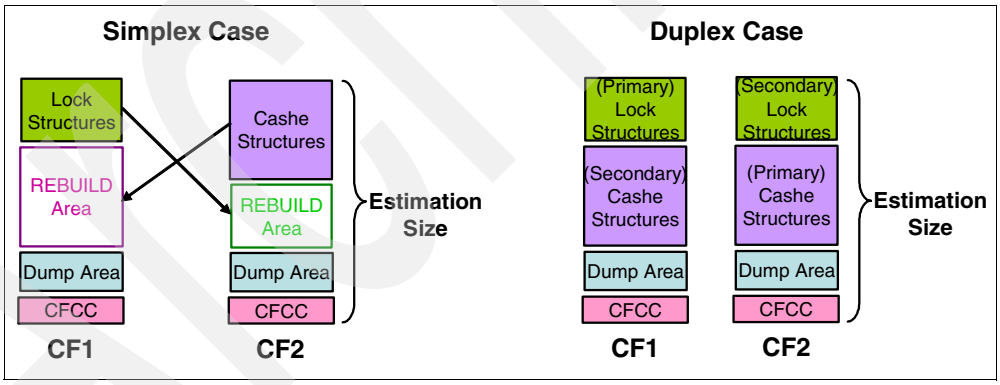


Figure 2-31 Estimation for storage size in the side of CFs

Using the *CFSizer tool,* the storage size can be estimated for each structure, depending on the CF level. For more information about the CFSizer tool, refer to:

http://www.ibm.com/servers/eserver/zseries/cfsizer/index.html

Note: By applying APAR OA03993 XCF/XES services in z/OS V1R4 or later, you can displayed detailed CF structure object utilization information using the D XCF,STR command output IXC360I.

Although CFSizer tool can be used in the case of newly built sysplex environment, this function may be assisted to estimate the storage size of each structure in coupling facility before processor models and CFLEVEL are migrated.

CF links between LPARs and processors (z/OS-to-CF, CF-to-CF)

- CF links cable type

  There are three kinds of CF links (IC, ICB, and ISC). The following are the characteristics of each link type:

  - Internal Coupling Channel (IC): An internal link connected between sysplex images within the same CPC. IC links have improved coupling performance over ICBs and coupling facility channels.
  - Integrated Cluster Bus (ICB): An external link that uses a self-timed interface (STI) cable. It has a higher speed than fiber cable link (ISC3), but there is a restriction that the maximum distance between CPCs is 7 m (cables is up to 10 m).
  - ISC fibre cable: The distance between processors can be up to 10 Km (20 Km by RPQ).
  - We can summarize this by saying that ICs and ICBs are particularly beneficial for CF duplexing.

Figure 2-32 gives an overview of CF links connectivity.

Connectivity Options	G5/G6 ISC	z800/z900/ z890/z990 ISC-3	G5/G6 ICB	z900/z990 ICB-2	z800/z900 ICB-3	z890/z990 ICB-3	z890/z990 ICB-4
G5/G6 ISC	1 Gbit/sec Compat Mode	1 Gbit/sec Compat Mode	N/A	N/A	N/A	N/A	N/A
z800/z900/ z890/z990 ISC-3	1 Gbit/sec Compat Mode	2 Gbits/sec Peer Mode	N/A	N/A	N/A	N/A	N/A
G5/G6 ICB	N/A	N/A	333MBytes/sec Compat Mode	333MBytes/sec Compat Mode	N/A	N/A	N/A
z900/z990 ICB-2	N/A	N/A	333MBytes/sec Compat Mode	Not Supported	N/A	N/A	N/A
z800/z900 ICB-3	N/A	N/A	N/A	N/A	1 GByte/sec Peer Mode	1 GByte/sec Peer Mode	N/A
z890/z990 ICB-3	N/A	N/A	N/A	N/A	1 GByte/sec Peer Mode	Not Preffered	N/A
z890/z990 ICB-4	N/A	N/A	N/A	N/A	N/A	N/A	2 GByte/sec Peer Mode

Figure 2-32 Information about CF links connectivity

- CF links channel mode

  IC channels are virtual attachments and, as such, require no real hardware. However, they do require CHPID numbers and they do need to be defined in the IOCP. IC channels will have a channel path type of ICP (Internal Coupling Peer). In this Peer mode, each subchannel can be used as both sender and receiver. We recommend that all IC links are shared by each OS and CF. Figure 2-33 is a sample definition for Internal CF links.

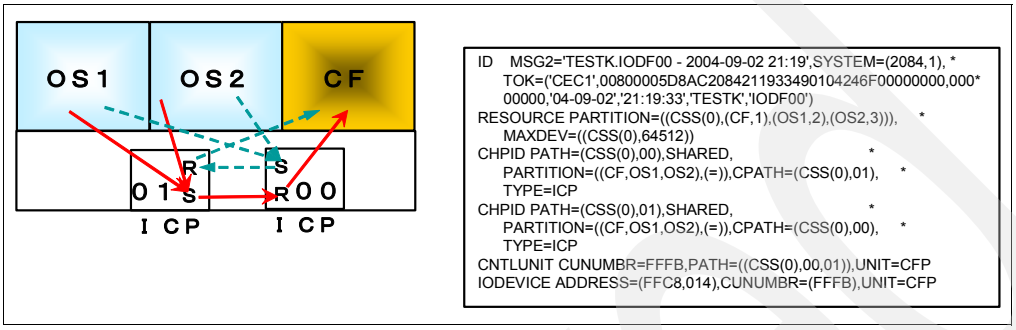


Figure 2-33 Sample definition for internal CF links (IC)

Note: The sharing between LPARs for CF links can be defined only z/OS-to-CF and z/OS-to-z/OS. Sharing between CFs is impossible.

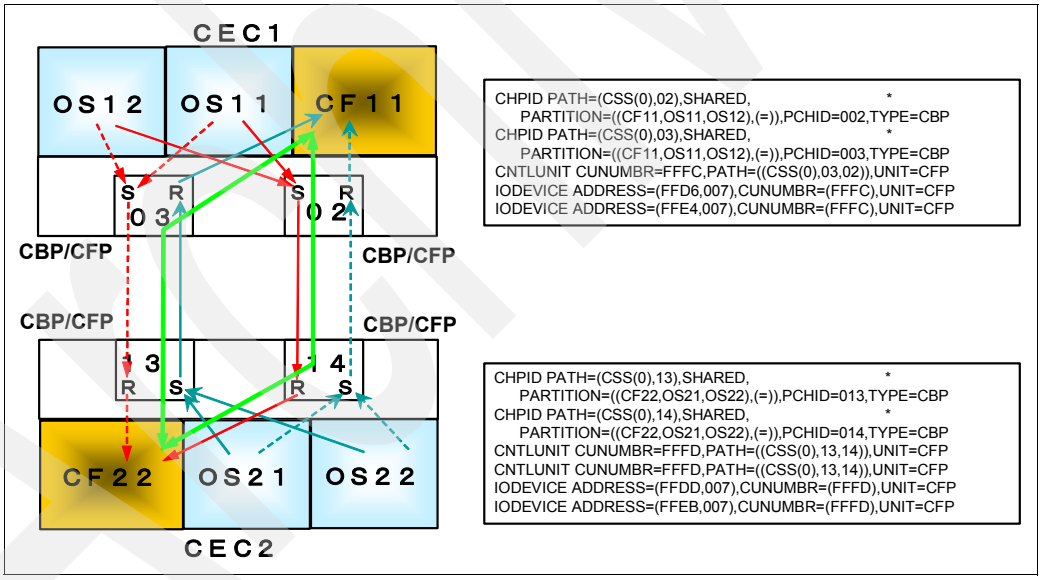


Figure 2-34 Sample definition for external CF links (ICB and ISC)

- Number of CF links

  The number of CF links should be estimated separately for z/OS to CF and CF-to-CF.

  The number of links required for z/OS-to-CF depends on the number of subchannels that satisfy the access to the structures in each CF. In CF Duplexing function, you have to take into consideration the access to secondary (or *new*) structures, because all writes to the duplexed primary structure will be replicated to the secondary. The number of subchannels, and the number (rates) of requests for structures in a CF determines the kind and number of CF link that should be chosen.

CF-to-CF links provide connectivity between a pair of coupling facilities in which a duplexed pair of structure instances are to be allocated. The link connectivity between the coupling facilities must be bidirectional (not shared between LPARs), and there should be more than one link providing CF-to-CF connectivity in each direction for highest availability. The number required for CF-to-CF depends on the numbers of subchannels that satisfy the processing to structures in each CF. This number of subchannels in CF-to-CF can be estimated, including the proportion of duplexing for all structures in CFs.

The latest level of the RMF Spreadsheet reporter now reports on subchannel utilization.

Note: For capacity planning a new tool, zCP3000 has superseded the *S/390 parallel Sysplex Quick-Sizer (SPSSZR)* tool produced by the Washington System Center (WSC). zCP3000 can project the overall hardware requirements, including:

- Number of z/Architecture™ systems required for the workload
- Processor utilization
- Storage requirement
- Number of Coupling Facilities
- Coupling Facility utilization
- Number of Coupling Facility links
- Average Coupling Facility link utilization
- Processor link utilization
- Processor link mode
- CPU service/response time

zCP3000 supports CF Duplexing and the new XCF algorithms.

2.6.7 Implementation and customization

This section discusses the implementation and the customization of the CF Duplexing function:

1. First, the CF to CF links have to be defined through HCD.

2. Select “CF/OS: Coupling facility or operation system” to CF LPAR, as in Figure 2-35.

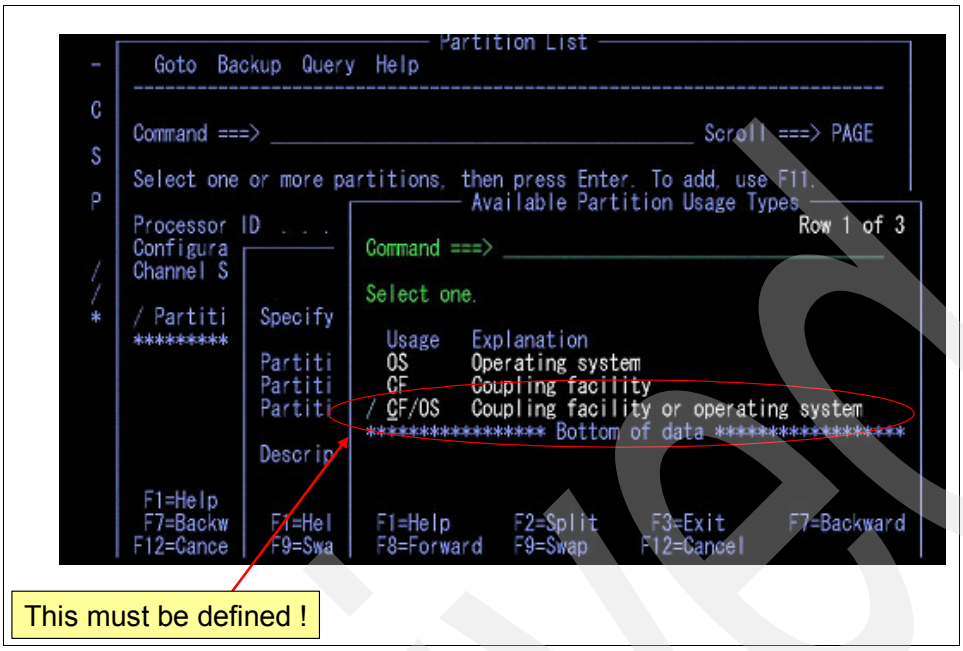


Figure 2-35 Definition of CF-to CF links with HCD panel

3. Define a CFRM policy to support duplexing and activate it.

   Figure 2-36 on page 81 shows the new item *ITEM (SMDUPLEX)*. Switch to the CFRM couple dataset by using the SETXCF ACOUPLE and SETXCF PCOUPLE commands.

4. Optionally, if you want to duplex the logger structures, you have to format the LOGR couple dataset to add SMDUPLEX and switch to it. Then, start the LOGR policy with the LOGGERDUPLEX parameter for each logger structure, as in Figure 2-36 on page 81. Two options are available:

   – LOGGERDUPLEX(COND): The system logger will first offload the log data from the structure, then begin duplexing any new log data written to the log stream:

     • For log streams defined with STG_DUPLEX=NO, the logger will begin duplexing data to local buffers.

     • For log streams defined with STG_DUPLEX=YES, the logger will begin duplexing data to staging data sets.

   – LOGGERDUPLEX(UNCOND): The system logger will continue to duplex the log data. However, for log streams defined with STG_DUPLEX=YES, system logger will begin duplexing data to staging data sets, if they were not already in use.

   For detailed information about LOGGER structures, refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

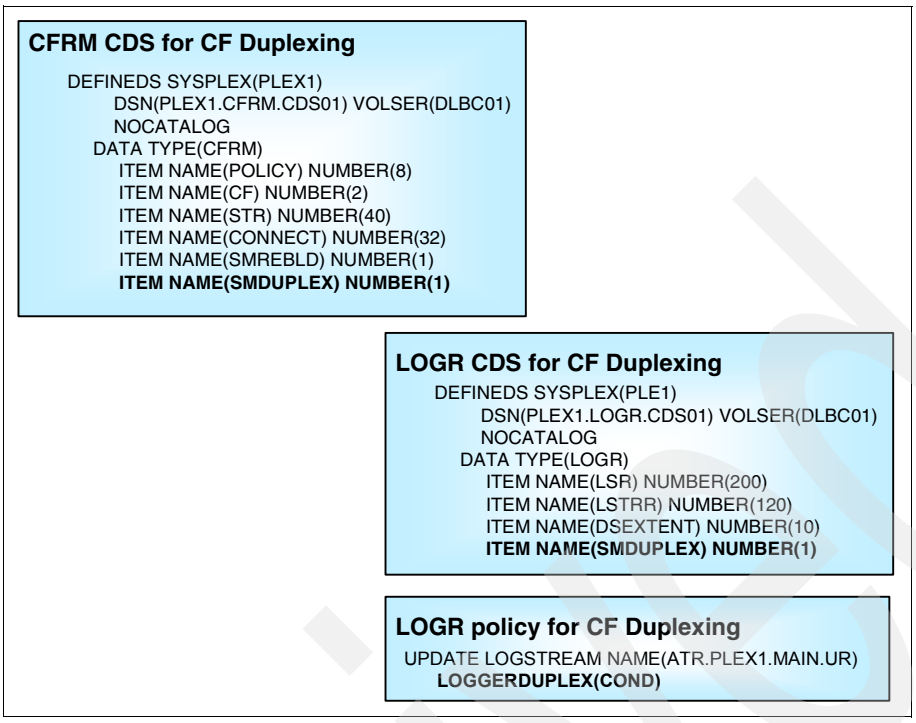


Figure 2-36 CFRM couple dataset and LOGR couple dataset, policy definition sample, or CF duplexing

5. To make use of the CF Duplexing function, the CFRM policy has to be defined with the DUPLEX keyword for each structure, as in Figure 2-37. The CFRM policy is loaded into the CFRM Couple Dataset and then started.

   The DUPLEX parameter offers three options: ENABLED | ALLOWED | DISABLED. You must define it as ALLOWED or ENABLED for the duplexed structure.

   – The structures with the ENABLED parameter are starting the duplexing as soon as they are requested and allocated.

   – The structures with the ALLOWED parameter cannot start duplexing without the input The initial definition can be determined by the operation procedure of the usual system IPL. But about CFRM policy activated in a failure, you must take into considerations.

   In summary:

   – DUPLEX(ENABLED): z/OS will automatically attempt to start and maintain duplexing for the structure at all times.

   – DUPLEX(ALLOWED): Duplexing may be manually started or stopped for the structure, but z/OS will not start duplexing automatically.

   – DUPLEX(DISABLED): Duplexing is not allowed for the structure.

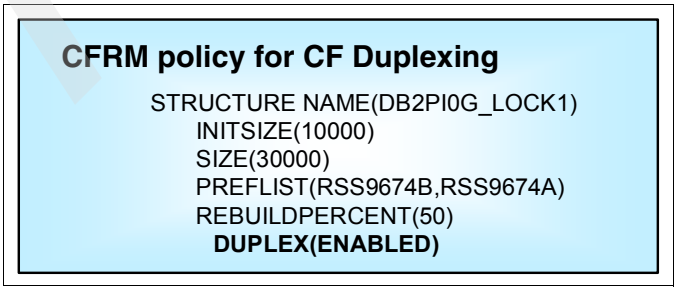


Figure 2-37 CFRM policy definition sample for CF duplexing

For detail information about the parameter in each policy for CF duplexing, refer to *z/OS MVS Setting Up a Sysplex*, SA22-7625.

2.6.8 Operational considerations

This section discusses the operational aspects of the CF duplexing.

Start, stop, and check the operation of CF duplexing

Before CF Duplexing for the structures is activated, the D CF command provides information about the CF-to-CF connectivity and links. Check whether CF-to-CF links are connected or not in REMOTELY CONNECTED COUPLING FACILITIES, as in Figure 2-38.

```
D CF

IXL150I  10.09.56  DISPLAY CF 137
COUPLING FACILITY 002084.IBM.83.0000000FD8AC
                  PARTITION: 07  CPCID: 00
                  CONTROL UNIT ID: FFFD
NAMED CEC1CF7

......


REMOTELY CONNECTED COUPLING FACILITIES
       CFNAME          COUPLING FACILITY
       --------        --------------------------
       CEC1CF8         002084.IBM.83.0000000FD8AC
                       PARTITION: 08  CPCID: 00

                       CHPIDS ON CEC1CF7 CONNECTED TO REMOTE FACILITY
                       RECEIVER:   CHPID    TYPE
                                   16       CFP

                       SENDER:     CHPID    TYPE
                                   16       CFP
```

*) This is output sample in test system environment, which both ICFs are activated at the same processor.

Figure 2-38 Display CF command output sample

The CF Duplexing function for a structure can be started and stopped by commands. There are two ways to start duplexing:

- Activate a new CFRM policy with DUPLEX(ENABLED) for the structure. If the *old* structure is currently allocated, then z/OS will automatically initiate the process to establish duplexing as soon as you activate the policy. If the structure is not currently allocated, then the duplexing process will be initiated automatically when the structure is allocated. This method attempts to reestablish duplexing automatically in the case of a failure, and also will periodically attempt to establish duplexing for structures that, for whatever reason, were not previously duplexed in accordance with the CFRM policy specification.
- Activate a new CFRM policy with DUPLEX(ALLOWED) for the structure. This method allows the structures to be duplexed; however, the duplexing must be initiated by a command: the system will not automatically duplex the structure.

Duplexing may then be manually started via the SETXCF START, REBUILD, DUPLEX command or the IXLREBLD STARTDUPLEX programming interface. This method also requires that duplexing be manually reestablished in the event of a failure. After these commands were exceed, each structure's information can be checked by the D XCF,STR,STRNAME=abc (or D XCF,STR) command, as in Figure 2-39 on page 84. When duplexing for the structure starts, both *old* and *new* structure information is shown.

Duplexing may be manually stopped via the SETXCF STOP,REBUILD,DUPLEX command or via the IXLREBLD STOPDUPLEX programming interface. When you need to stop duplexing structures, you must first decide which is to remain as the surviving simplex structure. The SETXCF STOP,REBUILD,DUPLEX,STRNAME=abc,KEEP=OLD|NEW command offers multiple parameters:

- KEEP=NEW specifies that processing should switch to the new structure and means to remain the secondary structure.
- KEEP=OLD specifies that processing should fall back to the old structure and means to remain the primary structure.

And when you stop duplexing for the structures in a specific CF, the SETXCF STOP,REBUILD,DUPLEX,CFNAME=xyz can be used too. For information about each system command of CF Duplexing, including SETXCF, refer to *z/OS V1R7.0 MVS System Commands,* SA22-7627.

```
D XCF,STR,STRNAME=DB8G_LOCK1

IXC360I  10.47.57  DISPLAY XCF 743
STRNAME: DB8G_LOCK1
 STATUS: REASON SPECIFIED WITH REBUILD START:
           POLICY-INITIATED
         DUPLEXING REBUILD
           METHOD        : SYSTEM-MANAGED
             AUTO VERSION: BBEA54DA A0D5EDCA
         REBUILD PHASE: DUPLEX ESTABLISHED
 TYPE: LOCK
 POLICY INFORMATION:
  POLICY SIZE    : 100000 K
  POLICY INITSIZE: 88000 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: N/A
  DUPLEX         : ENABLED
  PREFERENCE LIST: CEC1CF7  CEC1CF8
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY

 DUPLEXING REBUILD NEW STRUCTURE
 -------------------------------
  ALLOCATION TIME: 10/04/2004 10:25:26
  CFNAME         : CEC1CF8
  COUPLING FACILITY: 002084.IBM.83.0000000FD8AC
                     PARTITION: 08   CPCID: 00
  ACTUAL SIZE    : 88064 K
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: BBEA54DB 3C65DD0C
  LOGICAL  VERSION: BBEA54DA 65F1728C
  SYSTEM-MANAGED PROCESS LEVEL: 8
  XCF GRPNAME    : IXCLO00E
  DISPOSITION    : KEEP
  ACCESS TIME    : 0
  NUMBER OF RECORD DATA LISTS PER CONNECTION: 16
  MAX CONNECTIONS: 7
  # CONNECTIONS  : 2

 DUPLEXING REBUILD OLD STRUCTURE
 -------------------------------
  ALLOCATION TIME: 10/04/2004 10:25:25
  CFNAME         : CEC1CF7
  COUPLING FACILITY: 002084.IBM.83.0000000FD8AC
                     PARTITION: 07   CPCID: 00
  ACTUAL SIZE    : 88064 K
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: BBEA54DA 65F1728C
  LOGICAL  VERSION: BBEA54DA 65F1728C
  SYSTEM-MANAGED PROCESS LEVEL: 8
  XCF GRPNAME    : IXCLO00E
  ACCESS TIME    : 0
  NUMBER OF RECORD DATA LISTS PER CONNECTION: 16
  MAX CONNECTIONS: 7
  # CONNECTIONS  : 2

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 DB8GIRLM$IR8A001 01 00010011 CS05     DB8AIRLM 0033 ACTIVE NEW,OLD
 DB8GIRLM$IR8B002 02 0002000A CS06     DB8BIRLM 0030 ACTIVE NEW,OLD
```

Figure 2-39 Display XCF,STR,STRNAME command output sample

Recovery operation in a failure

Note: The SETXCF START,REALLOCATE command is supported by the APAR OA03481 for z/OS V1R4 or later version. It is used to rebuild and allocate all structures at the same time according to the CFRM activated policy. This command is very convenient. See 2.8, “Reallocate function” on page 96 for more information.

In Figure 2-40, we assume that the sysplex environment is working with two processors, which have CF(A) and OS(1) in one processor, and CF(B) and OS(2) in another processor. Some structures are duplexed with CFRM policy of DUPLEX(ENABLED) in this sysplex environment.

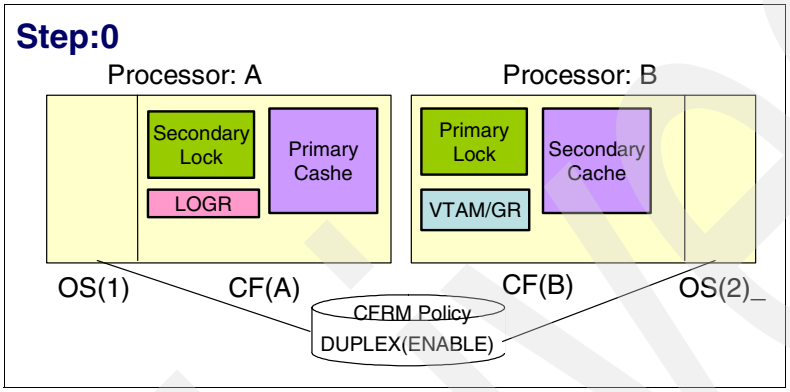


Figure 2-40 A sample system and structures environment (before a processor failure)

Suppose CPC A fails for some reason, and that CF(A) is re-activated after problem determination (Figure 2-41). Then, if the CFRM policy for the structures is defined as DUPLEX(ENABLED), duplexing will start as soon as a request is accessed to the target structure. While the duplex instance is being built, the access to each structure pauses. This may cause further performance problems.

Figure 2-41 is the recommended recovery action in this situation.

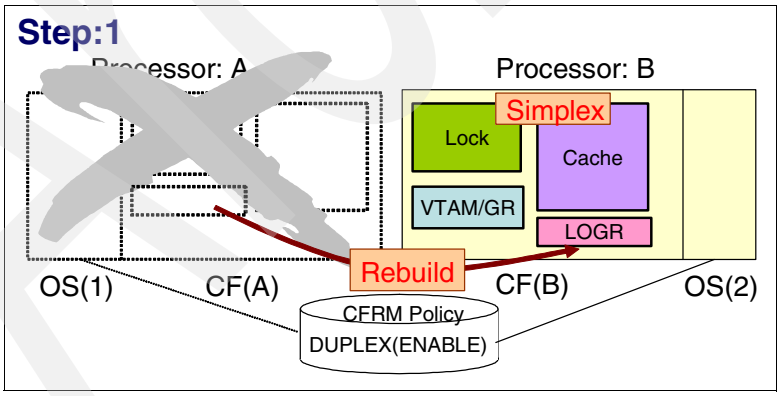


Figure 2-41 A sample system and structures environment (immediately after a processor failure)

1. After CF(A) is down, all structures rebuild and work in CF(B) (see Figure 2-42).

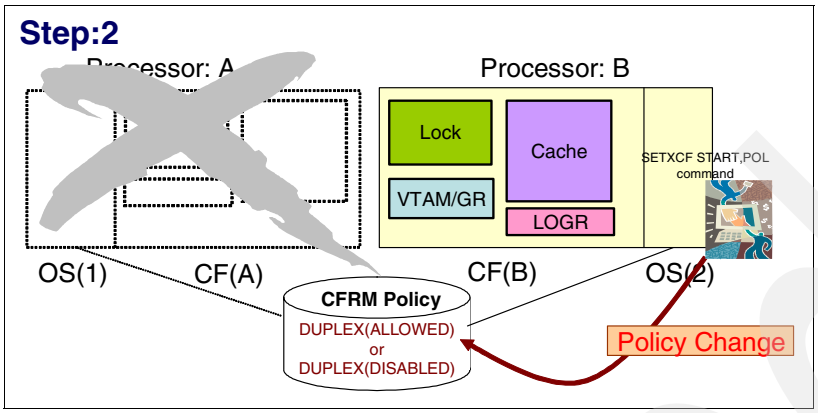


Figure 2-42 A sample system and structures environment (after changing and starting CFRM policy)

2. Before activating CF(A), change the CFRM policy to DUPLEX(ALLOWED) or DUPLEX(DISABLED) (see Figure 2-43).

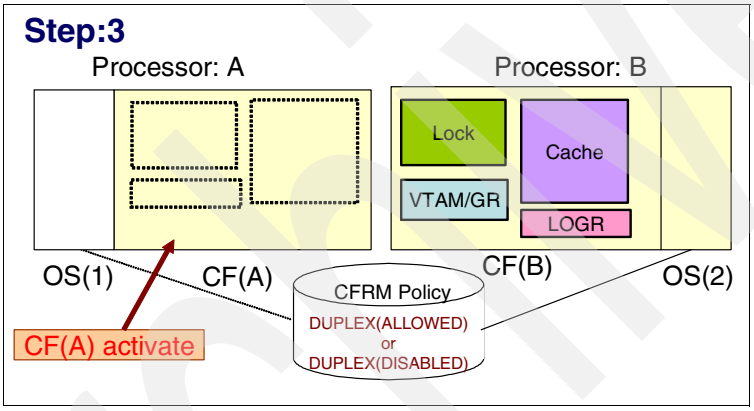


Figure 2-43 A sample system and structures environment (after recovery for a processor failure)

3. Activate CF(A). Duplexing does not start because of the CFRM policy (see Figure 2-44).

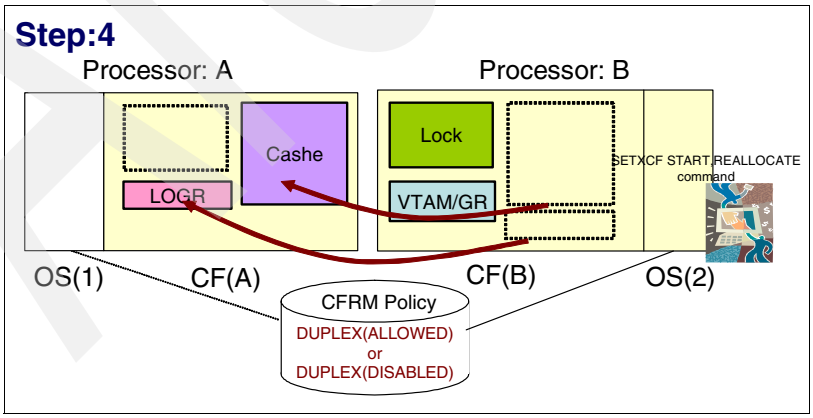


Figure 2-44 A sample system and structures environment (after rebuild operation for several structures)

4. Return the structures that should be in CF(A) by issuing SETXCF START,REALLOCATE (see Figure 2-45 on page 87).

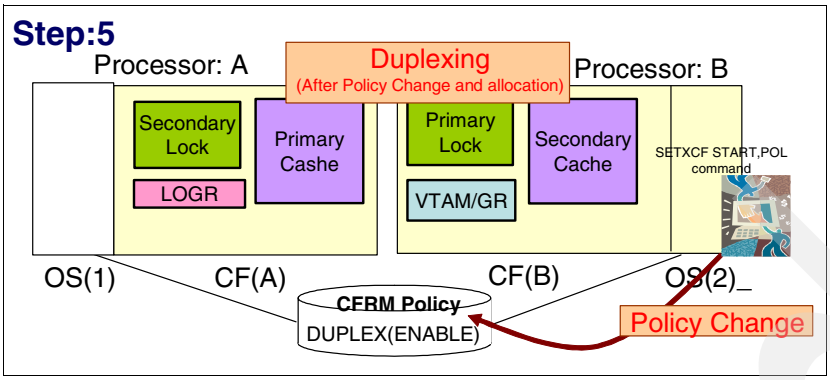


Figure 2-45 A sample system and structures environment (after returning to Duplexing for CFRM policy)

5. Change CFRM policy to the normal definition (with DUPLEX(ENABLED) and start that policy. Then duplexing will start (see Figure 2-46).

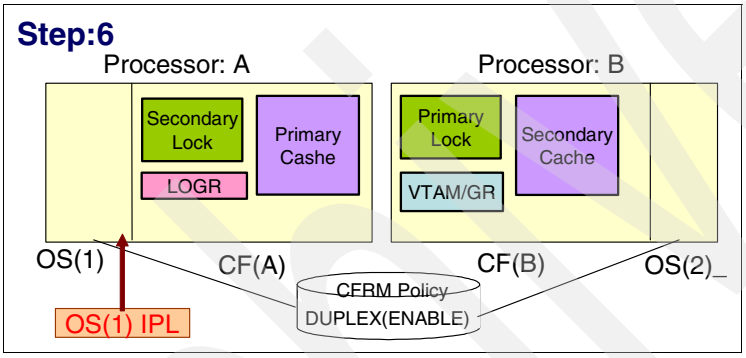


Figure 2-46 A sample system and structures environment (Final, after activating for a failed system)

6. Finally, IPL the OS(1) system.

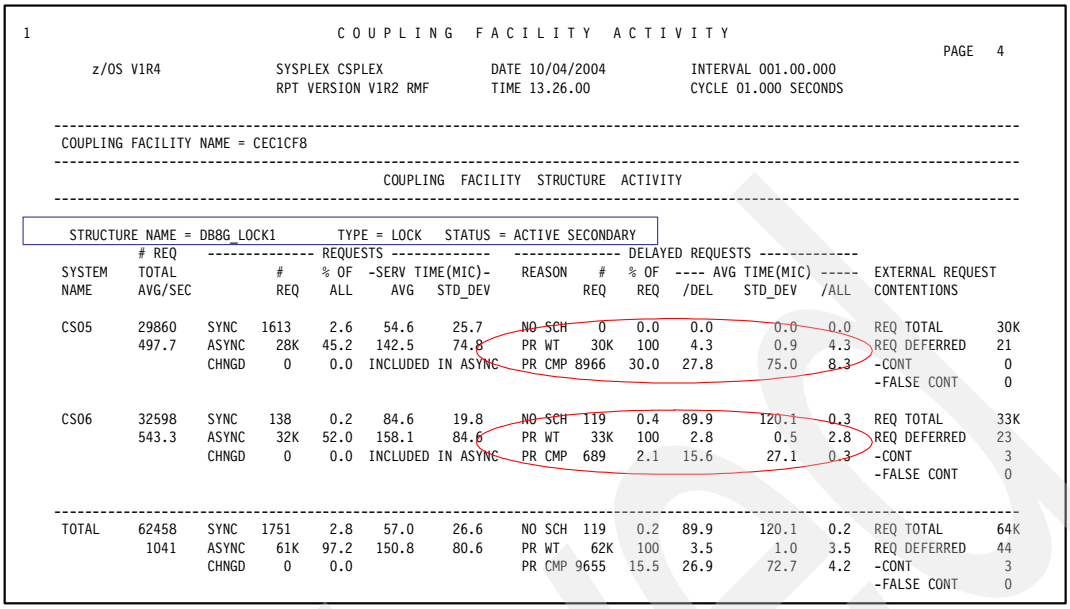
2.6.9 Monitoring considerations

RMF provides support to monitor and evaluate the actual CF structure placement and performance for duplexed structures.

New report for duplexing structure in RMF

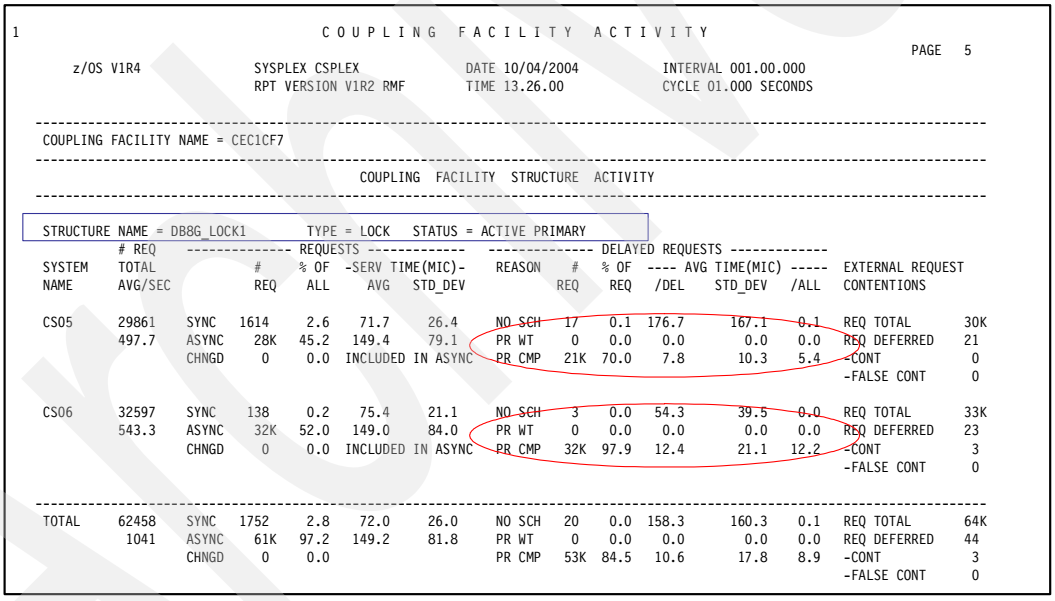
Figure 2-47 on page 88, Figure 2-48 on page 88, and Figure 2-49 on page 89 are examples of the COUPLING FACILITY STRUCTURE ACTIVITY RMF Post Processor report.

Figure 2-47 on page 88 and Figure 2-48 on page 88 show an example of the separate reports for primary and secondary structures. Two new parameters are shown in DELAYED REQUESTS for duplexing function.



```
1                                C O U P L I N G   F A C I L I T Y   A C T I V I T Y
                                                                                                                    PAGE    4
        z/OS V1R4                SYSPLEX CSPLEX             DATE 10/04/2004          INTERVAL 001.00.000
                                 RPT VERSION V1R2 RMF       TIME 13.26.00            CYCLE 01.000 SECONDS

 ------------------------------------------------------------------------------------------------------------------------------------
 COUPLING FACILITY NAME = CEC1CF8
 ------------------------------------------------------------------------------------------------------------------------------------
                                                COUPLING  FACILITY  STRUCTURE  ACTIVITY
 ------------------------------------------------------------------------------------------------------------------------------------

   STRUCTURE NAME = DB8G_LOCK1        TYPE = LOCK    STATUS = ACTIVE SECONDARY
            # REQ     ------------- REQUESTS -------------   ------------- DELAYED REQUESTS -------------
 SYSTEM     TOTAL              #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----  EXTERNAL REQUEST
 NAME       AVG/SEC          REQ     ALL     AVG   STD_DEV             REQ    REQ   /DEL    STD_DEV   /ALL   CONTENTIONS

 CSO5       29860    SYNC   1613     2.6    54.6     25.7    NO SCH     0    0.0    0.0        0.0    0.0   REQ TOTAL        30K
            497.7    ASYNC   28K    45.2   142.5     74.8    PR WT    30K    100    4.3        0.9    4.3   REQ DEFERRED     21
                     CHNGD    0     0.0  INCLUDED IN ASYNC   PR CMP 8966   30.0   27.8       75.0    8.3   -CONT             0
                                                                                                           -FALSE CONT       0

 CSO6       32598    SYNC    138     0.2    84.6     19.8    NO SCH   119    0.4   89.9      120.1    0.3   REQ TOTAL        33K
            543.3    ASYNC    32K    52.0   158.1     84.6    PR WT    33K    100    2.8        0.5    2.8   REQ DEFERRED     23
                     CHNGD    0     0.0  INCLUDED IN ASYNC   PR CMP   689    2.1   15.6       27.1    0.3   -CONT             3
                                                                                                           -FALSE CONT       0

 ------------------------------------------------------------------------------------------------------------------------------------
 TOTAL      62458    SYNC   1751     2.8    57.0     26.6    NO SCH   119    0.2   89.9      120.1    0.2   REQ TOTAL        64K
             1041    ASYNC    61K    97.2   150.8     80.6    PR WT    62K    100    3.5        1.0    3.5   REQ DEFERRED     44
                     CHNGD    0     0.0                      PR CMP  9655   15.5   26.9       72.7    4.2   -CONT             3
                                                                                                           -FALSE CONT       0
```

Figure 2-47 RMF “Coupling Facility Activity” report sample 1



```
1                                C O U P L I N G   F A C I L I T Y   A C T I V I T Y
                                                                                                                    PAGE    5
        z/OS V1R4                SYSPLEX CSPLEX             DATE 10/04/2004          INTERVAL 001.00.000
                                 RPT VERSION V1R2 RMF       TIME 13.26.00            CYCLE 01.000 SECONDS

 ------------------------------------------------------------------------------------------------------------------------------------
 COUPLING FACILITY NAME = CEC1CF7
 ------------------------------------------------------------------------------------------------------------------------------------
                                                COUPLING  FACILITY  STRUCTURE  ACTIVITY
 ------------------------------------------------------------------------------------------------------------------------------------

   STRUCTURE NAME = DB8G_LOCK1        TYPE = LOCK    STATUS = ACTIVE PRIMARY
            # REQ     ------------- REQUESTS -------------   ------------- DELAYED REQUESTS -------------
 SYSTEM     TOTAL              #     % OF  -SERV TIME(MIC)-    REASON    #   % OF  ---- AVG TIME(MIC) -----  EXTERNAL REQUEST
 NAME       AVG/SEC          REQ     ALL     AVG   STD_DEV             REQ    REQ   /DEL    STD_DEV   /ALL   CONTENTIONS

 CSO5       29861    SYNC   1614     2.6    71.7     26.4    NO SCH    17    0.1  176.7      167.1    0.1   REQ TOTAL        30K
            497.7    ASYNC   28K    45.2   149.4     79.1    PR WT      0    0.0    0.0        0.0    0.0   REQ DEFERRED     21
                     CHNGD    0     0.0  INCLUDED IN ASYNC   PR CMP   21K   70.0    7.8       10.3    5.4   -CONT             0
                                                                                                           -FALSE CONT       0

 CSO6       32597    SYNC    138     0.2    75.4     21.1    NO SCH     3    0.0   54.3       39.5    0.0   REQ TOTAL        33K
            543.3    ASYNC    32K    52.0   149.0     84.0    PR WT      0    0.0    0.0        0.0    0.0   REQ DEFERRED     23
                     CHNGD    0     0.0  INCLUDED IN ASYNC   PR CMP   32K   97.9   12.4       21.1   12.2   -CONT             3
                                                                                                           -FALSE CONT       0

 ------------------------------------------------------------------------------------------------------------------------------------
 TOTAL      62458    SYNC   1752     2.8    72.0     26.0    NO SCH    20    0.0  158.3      160.3    0.1   REQ TOTAL        64K
             1041    ASYNC    61K    97.2   149.2     81.8    PR WT      0    0.0    0.0        0.0    0.0   REQ DEFERRED     44
                     CHNGD    0     0.0                      PR CMP   53K   84.5   10.6       17.8    8.9   -CONT             3
                                                                                                           -FALSE CONT       0
```

Figure 2-48 RMF “Coupling Facility Activity” report sample 2

- PR WRT: The amount of time that the system was holding one subchannel while waiting to get the other subchannel, to launch the duplexed operation.
- PR CMP: One of the two duplexed operations has completed, but the completed subchannel remains unavailable for use until the other operation completes.

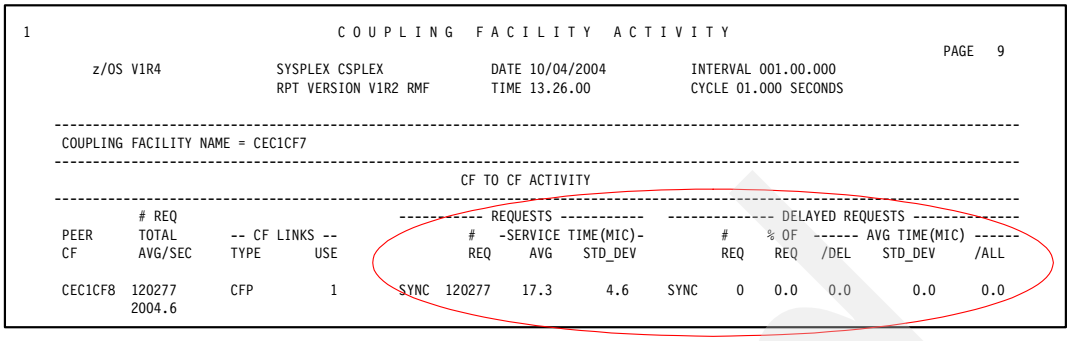


Figure 2-49 RMF "Coupling Facility Activity" report sample 3

- CF TO CF ACTIVITY: The new CF-to-CF Activity section in Figure 2-49 shows a summary of basic counts for duplexing related operations only. In addition, the new CF-to-CF section contains the information about the specific CF link types. This enhancement, the display CF link type information, has also been made in the existing subchannel activity section.

Performance consideration

In a sysplex environment with CF Duplexing structures, the following fields should be monitored

- Delay status for service time in duplexed structures
- Proportion of sync and async operations in structures
- Utiltization of subchannel in CF links

For detailed information about RMF reports, refer to *z/OS Resource Measurement Facility Report Analysis,* SC33-7991 on the Web at:

http://publibz.boulder.ibm.com/epubs/pdf/erbzra41.pdf

In a sysplex environment with CF Duplexing structures, you should monitor "DELAYED REQUESTS" for each structure and CF to CF in the RMF "Coupling Facility Activity" report.

For example, in the RMF Coupling Facility Structure Activity report for each structure shown in Figure 2-47 on page 88, look at average service time and request counts for sync and async in REQUESTS. The average delay time (/ALL in "DELAYED REQUESTS") should be about 10 to 20% or less of average service time (AVG in "REQUESTS"). When the reported value in "DELAYED REQUESTS" is larger, you should consider taking some action such as increasing the number of links.

The RMF CF to CF Activity report in Figure 2-49 should be looked at in a similar fashion. The request counts between CFs shown in this report will be about 2 to 2.5 times greater than the requests to a duplexing structure. It is desirable that there are no delays for any requests, but we recommend keeping the average delay time (/ALL in "DELAYED REQUESTS") to about 10% or less of average service time (AVG in "REQUESTS").

Note: The RMF Spreadsheet Reporter Trend Report tool can monitor the Coupling Facility. It is available from:

http://www.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/rmftools.htm

Additional documentation about CF duplexing

Clients interested in deploying System-Managed CF Structure Duplexing in their test, development, or production Parallel Sysplex should read the technical white paper *System-Managed CF Structure Duplexing*, GM13-0103, and analyze their Parallel Sysplex environments to understand the performance and other considerations of using the function. *System-Managed CF Structure Duplexing*, GM13-0103 is available at:

http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130103.pdf

Additional information about Coupling Facility Structure Duplexing can be found from the following sources:

- *z/OS and z/OS.e V1R7.0 Planning for Installation,* GA22-7504
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- http://www.ibm.com/servers/eserver/zseries/pso
- http://wwwibm.com/servers/eserver/zseries/zos/bkserv
- *Coupling Facility Configuration Options,* GF22-5042
- CFLevel info:

  http://www.ibm.com/servers/eserver/zseries/pso/cftable.html
- CF Sizer:

  http://www.ibm.com/servers/eserver/zseries/cfsizer/
- Parallel Sysplex Sizer (see your IBM rep)

2.7 CFSizer

There are many products that use the CF to support improved sharing of data, message passing, and locking. For each CF exploiter, starting points for structure sizes are given along with some rationale for the size. In every instance, the size of the structure will depend on some installation-specific entity that will have to be estimated.

Some of the values are easier to estimate than others. Some of the estimated values are precise, and others potentially have large variances. For a more accurate estimate, use the CF Sizer tool available on the Internet at:

http://www-1.ibm.com/servers/eserver/zseries/cfsizer/

For the most accurate sizings, you will need to use the structures in your environment and monitor the actual structure storage usage. It is safer to overallocate structures if you are unsure. The structure-full monitoring support added in OS/390 V2R9 will help by reporting if a structure exceeds a specified usage of any of its storage.

CFSizer is a web-based application that will estimate structure sizes based on the latest CFLEVEL for the IBM products that exploit the coupling facility (see Figure 2-49 on page 89).

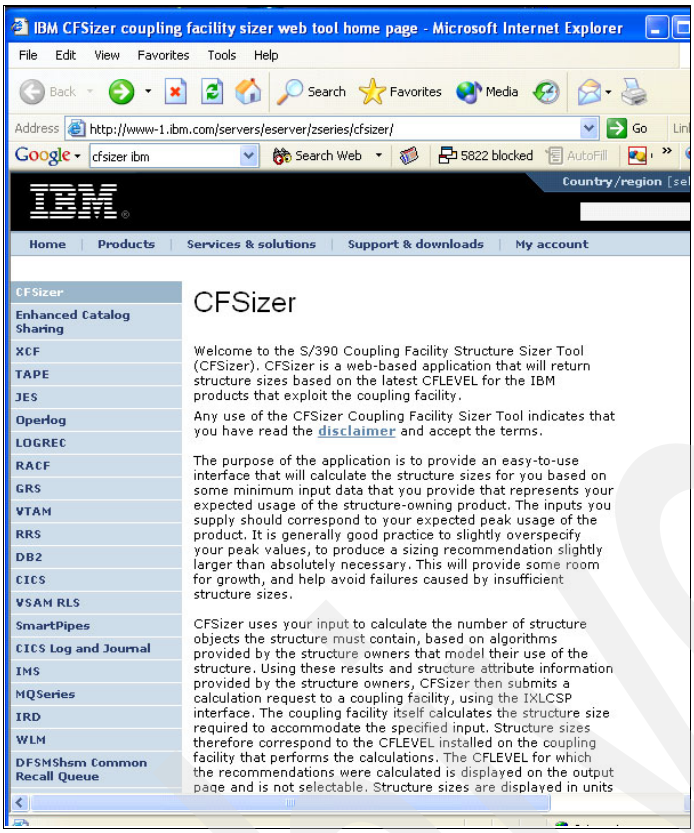


Figure 2-50 CF sizer Web site welcome page

The purpose of the application is to provide an easy-to-use interface that will calculate the structure sizes for you based on some minimum input data that you provide that represents your expected usage of the structure-owning product. The inputs you supply should correspond to your expected peak usage of the product. It is a generally good practice to slightly overspecify your peak values, to produce a sizing recommendation slightly larger than absolutely necessary. This will provide some room for growth, and help avoid failures caused by insufficient structure sizes.

CFSizer uses your input to calculate the number of structure objects the structure must contain, based on algorithms provided by the structure owners that model their use of the structure

The CFLEVEL for which the recommendations were calculated is displayed on the output page and is not selectable. Structure sizes are displayed in units of 1 KB, as they would be entered in your CFRM policy.

To size a structure, click one of the product links in the left side navigation bar (for example, XCF, Tape, JES, and so on) to access that product page. From that product page, select one or more structures to size by clicking the product's check box.

See the example in Figure 2-51 for the VSAM RLS structure.

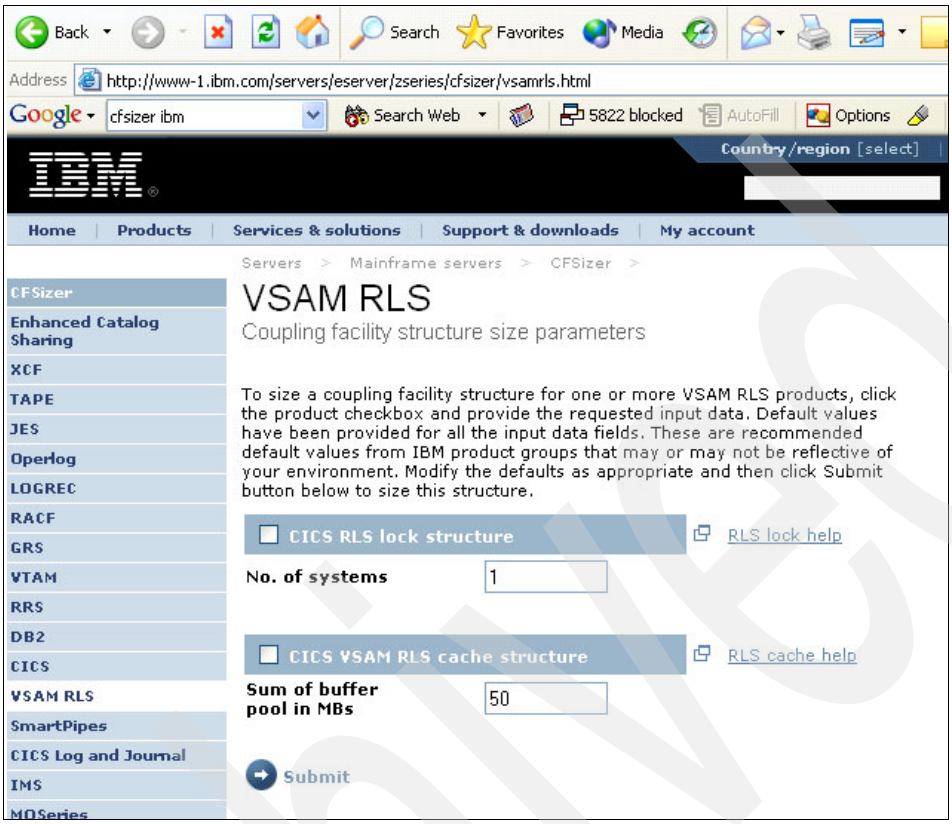


Figure 2-51 Sizing example with VSAM RMS structure

In this window, you select which structure you want the Sizer to help you calculate the size of by selecting it (with the corresponding check boxes), you fill in the empty boxes according to the information provided by your subsystem supporting personnel. If you need further help, or have no idea what to type in, you may ask for additional help by clicking the **Help** link.

For product help, click the **Help** link provided on each product page. A help window will open that will provide some additional details about the product and the input fields.

After providing the required information, click the **Submit** button, and the CF sizer will make the calculations, and present you with the results in a new browser window as in Figure 2-52 on page 93.

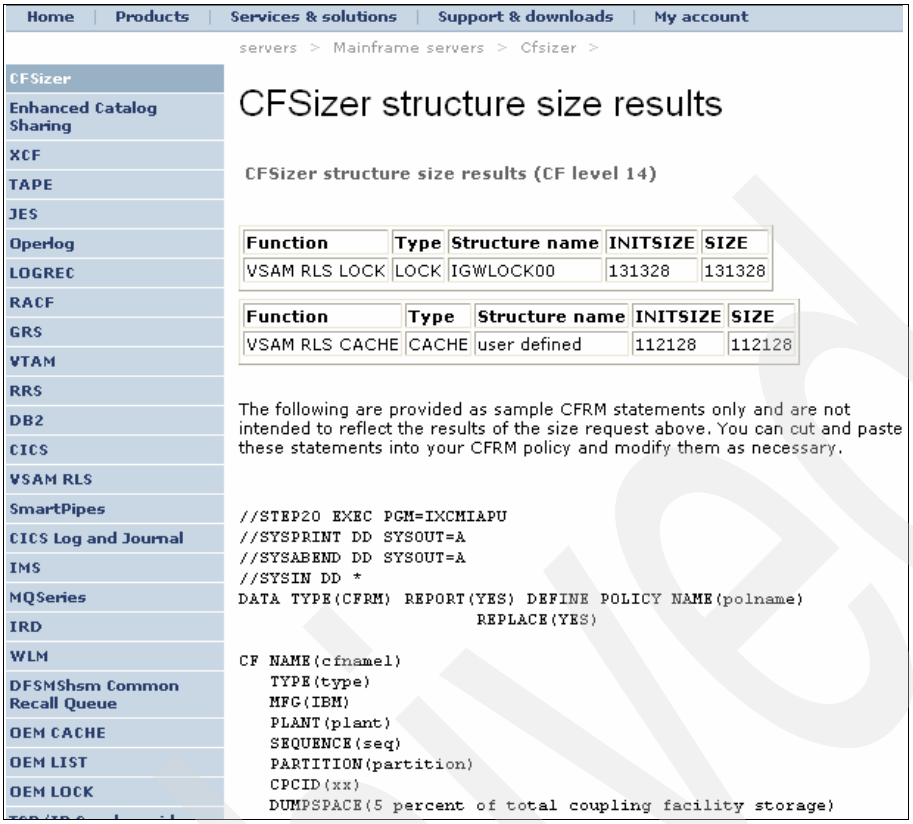


Figure 2-52 CF sizer structure size calculation output

The output of the sizer not only contains a summary of the SIZE and INITSIZE parameters required to allocate the structure, but it also contains a sample JCL to run the IXCMIAPU utility to create or update the CFRM policy. You can easily use it as a template and do a cut or paste in your z/OS TSO/ISPF edit environment.

Important: If you are upgrading your CFCC microcode on one or more CFs, and your existing structures appear to be adequately sized, you may be able to use the current structure sizes to calculate the sizes required by the new CFLEVEL. IBM recommends that you periodically recalculate your structure sizes through CFSizer, based on expected usage and structure attributes.

2.7.1 Sizer utility

The *sizer utility program* is available for download as a zipped package containing the following files:

sizer.obj	Sizer utility executable (binary, FB 80).
runsize.jcl	JCL to invoke the utility, as a started task (binary, FB 80).
linksize.jcl	Link edit JCL (binary, FB 80). Update this JCL to install the utility in an authorized library.

The Sizer Utility Program.doc Utility documentation (Microsoft® Word document) can be download from the CFSizer Web site at the following address:

http://www-1.ibm.com/servers/eserver/zseries/cfsizer/altsize.html

This utility is provided *as is* and is not supported by IBM. It will size currently allocated structures for each of the online /accessible CFs in a client's CFRM policy.

The utility is useful in an upgrade scenario as follows:

1. Move all structures out of the first CF to be upgraded.
2. Upgrade the CF to the desired CFLEVEL.
3. Run the Sizer utility with all structures allocated in a CF at the original CFLEVEL.
4. Using the utility output, update the CFRM policy to reflect the sizes required at the higher CFLEVEL.
5. Upgrade the rest of the CFs and distribute the structures as desired.

When run, the SIZER utility will inspect all of the CF structure instances that are currently allocated, in all the CFs that are accessible from the system where the utility is run, analyze, and report on their detailed CF structure attributes (in CF architecture terms), and then determine and report on the structure size for the structures in several different ways:

- The CFRM policy size-related definitions or defaults for the structure.
- The current, actual allocated structure sizing information.
- The calculated sizing information for a structure having identical attributes to the allocated structure instance, if it were to be allocated in each of the other CFs in the installation that are accessible to the system where the utility is run. In effect, this is a *what-if* calculation to determine what the sizing requirements would be for this structure if it were to be allocated in each of the other CFs in the configuration, at whatever CFLEVEL those CFs may be.

The output of the SIZER utility may be useful to you in determining how to make CFRM policy structure size definition changes (SIZE, INITSIZE, and MINSIZE parameters) to accommodate CF structure size growth between CFLEVELs. For additional general information about CF structure sizing and requirements, see the CF Sizer Web site at:

`http://www.ibm.com/servers/eserver/zseries/cfsizer/`

However, note that in order for the SIZER utility to be useful for this purpose, you must already have provided a CF at the new *uplevel* CFLEVEL in your installation, and it must be configured and active in your sysplex, with connectivity to the system on which the SIZER utility is being run.

Note that if more than one active instance of a particular structure is allocated at the time that the SIZER utility is run (for example, because the structure is duplexed, or is in the process of being rebuilt), each of the structure instances will be reported on separately.

The SIZER utility generates a report with information about every currently-allocated structure, in every CF in your installation that is accessible to the system where the utility is run.

Having thus determined and reported on the current allocation attributes for a particular allocated structure instance, the second part of the report for each structure shows the applicable policy/actual/calculated structure sizing information. This is displayed in a tabular format, with the columns of the table shown in Example 2-1 on page 95 (note that the size figures are all shown in units of KB):

- CFNAME: The name of the CF for which sizing information is being presented.
- The first row (with a pseudo-CFNAME of POLICY) contains the structure sizing information from the CFRM policy, rather than from any allocated or calculated structure instance.

- The second row (with a pseudo-CFNAME of CURRENT) shows the actual allocation sizing information for the structure as it is currently allocated, in the CF where it is currently allocated.
- The third through nth rows show the calculated sizing information for the structure, as though it were to be allocated with identical structure attributes (as shown in the first part of the report) to the current instance, for each accessible CF in the configuration.

Example 2-1 Sizer utility sample output

```
09.45.25 JOB12849  SIZER: PROCESSING STARTED - VERSION 0.04 - 06/30/03
 09.45.26 JOB12849  SIZER: ------------------------------------------------------
 09.45.26 JOB12849  SIZER: STRUCTURE MP1IMSVSAM
 09.45.26 JOB12849  SIZER:
 09.45.26 JOB12849  SIZER:      CACHE STRUCTURE
 09.45.26 JOB12849  SIZER:      NAMECLASS...         NO    ADJUNCT.....         NO
 09.45.26 JOB12849  SIZER:      UDF ORDER...         NO
 09.45.26 JOB12849  SIZER:      ELEMCHAR....          0    MAXELEMNUM..          0
 09.45.26 JOB12849  SIZER:      COCLASSES...          1    STGCLASSES..          1
 09.45.26 JOB12849  SIZER:      ENTRIES.....     22,646    ELEMENTS....          0
 09.45.26 JOB12849  SIZER:
 09.45.26 JOB12849  SIZER:   CFNAME       MINSIZE    STRSIZE    MAXSIZE  MINCTLSTG
 09.45.26 JOB12849  SIZER:   --------  ---------- ---------- ---------- ----------
 09.45.26 JOB12849  SIZER:   POLICY             0          0      6,556        N/A
 09.45.26 JOB12849  SIZER:   CURRENT          512      6,656      6,656        N/A
 09.45.26 JOB12849  SIZER:   CF2A             512      6,656      6,656      6,656
 09.45.26 JOB12849  SIZER:   CF2B             512      6,656      6,656      6,656
 09.45.26 JOB12849  SIZER: ------------------------------------------------------
 09.45.26 JOB12849  SIZER: STRUCTURE IMS_EMHQ_STR
 09.45.26 JOB12849  SIZER:
 09.45.26 JOB12849  SIZER:      LIST STRUCTURE
 09.45.26 JOB12849  SIZER:      KEY/NAME.... ENTRY KEYS    ADJUNCT.....         YES
 09.45.26 JOB12849  SIZER:      ELEMCHAR....          1    MAXELEMNUM..         120
 09.45.26 JOB12849  SIZER:      LIST CNT....        192    LOCK ENTS...         256
 09.45.26 JOB12849  SIZER:      ENTRIES.....      6,914    ELEMENTS....      13,631
 09.45.26 JOB12849  SIZER:      EMCS........     17,912
 09.45.26 JOB12849  SIZER:
 09.45.26 JOB12849  SIZER:   CFNAME       MINSIZE    STRSIZE    MAXSIZE  MINCTLSTG
 09.45.26 JOB12849  SIZER:   --------  ---------- ---------- ---------- ----------
 09.45.26 JOB12849  SIZER:   POLICY        11,028     14,708     20,480        N/A
 09.45.26 JOB12849  SIZER:   CURRENT        5,376     14,848     20,480        N/A
 09.45.26 JOB12849  SIZER:   CF2A          10,496     15,360     20,480     15,360
 09.45.26 JOB12849  SIZER:   CF2B          10,496     15,360     20,480     15,360
 09.45.26 JOB12849  SIZER: ------------------------------------------------------
 09.45.26 JOB12849  SIZER: STRUCTURE IGWLOCK00
 09.45.26 JOB12849  SIZER:
 09.45.26 JOB12849  SIZER:      LOCK STRUCTURE
 09.45.26 JOB12849  SIZER:      RDATA ENTS..     67,392    NUMUSERS....         23
 09.45.26 JOB12849  SIZER:      LOCK ENTS...  2,097,152
 09.45.26 JOB12849  SIZER:


 09.45.26 JOB12849  SIZER:   CFNAME       MINSIZE    STRSIZE    MAXSIZE  MINCTLSTG
 09.45.26 JOB12849  SIZER:   --------  ---------- ---------- ---------- ----------
 09.45.26 JOB12849  SIZER:   POLICY             0     20,788     40,192        N/A
 09.45.26 JOB12849  SIZER:   CURRENT       11,520     25,856     40,192        N/A
 09.45.26 JOB12849  SIZER:   CF2A          12,032     26,112     40,192     26,112
 09.45.26 JOB12849  SIZER:   CF2B          12,032     26,112     40,192     26,112
 09.45.26 JOB12849  SIZER: ------------------------------------------------------
```

The reported information in each of these rows is described in more detail in the accompanying .doc file that comes with the utility.

Note: This list of CFs will include the CF where the structure currently resides, and thus with the SIZER output in hand, one can compare the actual allocation sizing information of the structure in its current CF (as shown in the CURRENT row) to the calculated sizing information for that structure in the very same CF.

In general, these actual and calculated values will be close, but you should be aware that it is not at all uncommon for them to differ somewhat, perhaps even to differ very substantially, from one another. The main reason for these actual-versus-calculated discrepancies has to do with differences in the structure allocation history: the CURRENT structure may have been rebuilt, altered, and so on, since it was initially allocated, while the calculated structure sizes show the sizing information for a hypothetical newly-allocated structure in the indicated CF. This difference in history may result in different allocations of CF-internal control areas within the structure, which may in turn result in the different sizing results visible in the SIZER output.

2.8 Reallocate function

After a CF structure is allocated, the system makes no attempt to optimize the placement of that structure with respect to the installation's wishes (as expressed in the active CFRM policy). There are many reasons why structures, even if they are initially placed in their client-desired locations, can and do move around and over time get into sub-optimal locations from the clients standpoint. Specifically, there was no easy means for restoring structures to their desired location, and especially in cases where structures support user-managed or system-managed duplexing rebuild and the installation has three or more CFs; the use of existing commands to restore structures to their desired locations is quite a cumbersome process.

This problem is addressed by providing a new XCF function introduced with z/OS 1.4 (HBB7707), controlled by enhanced SETXCF operator commands. The function is called the REALLOCATE process, with REALLOCATE as the new parameter added to the SETXCF START and SETXCF STOP commands.

The REALLOCATE process uses existing XCF structure allocation algorithms to recognize the need to relocate structure instances by comparing the current location with the location selected by allocation criteria using either the active or pending CFRM policy. When the locations differ or a policy change is pending, the REALLOCATE process uses the structure rebuild process to accomplish the needed adjustments.

Structure rebuild processing supports:

- User-managed rebuild
- User-managed duplexing rebuild
- System-managed rebuild
- System-managed duplexing rebuild

The operator commands are:

- SETXCF START, REALLOCATE
- SETXCF STOP, REALLOCATE

The REALLOCATE process is a new XCF function that simplifies all Parallel Sysplex Coupling Facility (CF) maintenance procedures. While it provides ease-of-use benefits for all Parallel Sysplex clients, the most significant advantage is for those with any of these conditions:

- More than two CFs
- Duplexed structures, such as DB2 Group Buffer Pools
- Installations wanting structures to reside in specific CFs
- Configurations with CFs having different characteristics

2.8.1 The problem

When CF maintenance is performed, at least one CF is emptied. That process usually occurs by rebuilding all the structures into an alternate CF. If the Coupling Facility Resource Management (CFRM) policy preference list specifies CF1, CF2, and CF3 for a duplexed structure and CF1 is emptied prior to maintenance, the structure then resides in CF2 and CF3. When CF1 is brought back on line following maintenance, nothing changes, because the structure is already duplexed.

XCF/CFRM is not aware of the desire to have the primary/*old* structure in the *most preferred CF* (CF1), and the secondary/*new* structure in the *second most preferred CF* (CF2) in the preference list. After duplexed structures have been moved, XCF/CFRM does not attempt to restore them to their *most preferred* CF locations.

Every time a new structure instance gets allocated, XCF allocation criteria are used to place it in the *most preferred* CF in the preference list that does not already contain an active instance of the structure. It is as satisfactory to have the primary and secondary in *backwards* order as it is in the *forwards* order.

Most clients place structures in specific CFs for certain considerations, such as load balancing. In the example above, it might be adequate for CF3 to contain an instance of the structure in order to maintain duplexing during CF maintenance, but not acceptable to keep it there indefinitely. The characteristics of DB2's duplexed Group Buffer Pools (GBP) are that the secondary has far less activity than the primary GBP, and thus there is a specific CF desired for each copy, again for load balancing.

Following CF maintenance operations the desired structure placement can be difficult to achieve, often requiring manual operator commands on a structure by structure basis. For installations with a large number of duplexed structures, the task to *reposition* them is onerous for clients' staffs.

Furthermore, the current method of emptying a CF through the operator command SETXCF START,REBUILD,CF=cf_name,LOC=OTHER has two disadvantages:

- All rebuilds occur at the same time, resulting in contention for the CFRM Couple Data Set. This contention elongates the rebuild process for all of the affected structures, making the rebuilds more disruptive to ongoing work.
- The IXC structures do not participate in that process, but must be rebuilt via manual commands on a structure by structure basis.

2.8.2 The solution

Initiate the REALLOCATE process by issuing the SETXCF START,REALLOCATE operator command to perform any of the following actions:

- Move structures out of a CF following a CFRM policy change that deletes/changes that CF (for example, in preparation for a CF upgrade).

- Move structures back into a CF following a CFRM policy change that adds/restores the CF (for example, following a CF upgrade/add).
- Clean up pending CFRM policy changes that may have accumulated for whatever reason, even in the absence of any need for structure *relocation* per se.
- Clean up simplex or duplexed structures that were allocated in or moved into the *wrong* CFs, for whatever reason (for example, the *right* CF was inaccessible at the time of allocation).
- Clean up duplexed structures that have primary and secondary *reversed* due to a prior condition that resulted in having duplexing stopped with KEEP=NEW and the structure reduplexed REALLOCATE minimizes CRFM Couple Data Set contention through its serial processing and automatically relocates the IXC* structures (they do not require special handling).

For more information, refer to:

- z/OS Integration Test site:

  http://www-1.ibm.com/servers/eserver/zseries/zos/integtst/

  The JUNE 2004 Parallel Sysplex Test Report contains detailed examples from system logs showing usage of the enhanced SETXCF commands supporting the REALLOCATE process.
- *z/OS V1R7.0 MVS System Commands,* SA22-7627
- Look at the special flash paper available on the TECHDOC Web site at:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10285

3

Continuous availability in Parallel Sysplex

Continuous availability is one of the major advantages of migrating to a Parallel Sysplex. This chapter discusses how some of the configuration choices affect system availability in a Parallel Sysplex.

Parallel Sysplex aims to provide continuous availability at the application level to users.

Recommended Sources of Further Information: The following sources provide support for the information in this chapter:

- *DB2 on MVS Platform: Data Sharing Recovery*, SG24-2218
- *DB2 UDB for OS/390 and Continuous Availability*, SG24-5486
- *DB2 UDB for OS/390 V6 Data Sharing: Planning and Administration,* SC26-9007
- *DB2 UDB for OS/390 V6 Release Planning Guide,* SC26-9013
- *DB2 Universal Database™ Server for OS/390 V7 What's New?,* GC26-9017
- *MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*, SG24-4037
- *IBM System z9 and @server zSeries Connectivity Handbook*, SG24-5444
- IBM Global Services High Availability Services, available on the Web at: http://www.ibm.com/services/tsm/Implementing
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries*, SG24-5680
- *IMS/ESA Sysplex Data Sharing: An Implementation Case Study*, SG24-4831
- *IMS/ESA Data Sharing in a Parallel Sysplex*, SG24-4303
- *IMS/ESA Multiple Systems Coupling in a Parallel Sysplex*, SG24-4750
- *IMS/ESA Sysplex Data Sharing: An Implementation Case Study*, SG24-4831
- *IMS V9 Release Planning Guide,* GC17-7831
- OS/390 Integration Test Site at: http://www.s390.ibm.com/os390/support/os390tst
- *OS/390 V2R8.0 Parallel Sysplex Hardware and Software Migration,* GC28-1862
- *OS/390 Parallel Sysplex Test Report,* GC28-1963
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- Parallel Sysplex Availability Checklist available on the Web at: http://www.ibm.com/servers/eserver/zseries/library/whitepapers/pdf/availchk_parsys.pdf
- *Parallel Sysplex Configuration Planning for Availability*, WSC Flash 9829
- *Parallel Sysplex - Managing Software for Availability*, SG24-5451
- *OS/390 Parallel Sysplex Recovery,* GA22-7286
- *System/390 MVS Parallel Sysplex Continuous Availability Presentation Guide*, SG24-4502
- *System/390 MVS Parallel Sysplex Continuous Availability SE Guide*, SG24-4503
- *SFM Functions and the Impact of OW30814 for Parallel Sysplex*, WSC Flash 9825
- *SNA in a Parallel Sysplex Environment*, SG24-2113
- *TCP/IP in a Sysplex*, SG24-5235
- *XCF Service Recommendations to Maximize Sysplex Availability*, WSC Flash 9838

3.1 Why availability is important

There are real costs associated with system outages. In business terms, an outage costs real dollars and can seriously impact the ability to respond to changing business needs.

A *planned outage* is inconvenient at best. Some companies simply cannot tolerate any unplanned down time, and so will schedule a planned outage for a time (for example, 3 AM Sunday morning) when the fewest users are affected. This means that necessary system work is postponed until a time when the applications are not available. It also means that system programmers must work inconvenient hours, and help may be unavailable if something goes wrong.

An *unplanned outage* can, of course, happen during the busiest time of the day and therefore have a serious impact. Also, because of the panic nature of such an outage, the potential exists for additional harm to be caused by errors that are introduced as people rush to get the system restored.

Availability definitions: Levels of availability are defined as follows:

- *High availability*: A system that delivers an acceptable or agreed level of service during scheduled periods.
- *Continuous operation*: A system that operates 7 days a week, 24 hours a day, with no scheduled outages.
- *Continuous availability:* A system that delivers an acceptable or agreed level of service 7 days a week, 24 hours a day.
- *Continuous availability* is the combination of high availability and continuous operation.

Source: *System/390 MVS Parallel Sysplex Continuous Availability Presentation Guide*, SG24-4502

For many reasons, a continuously available system is becoming a requirement today. In the past, you most likely concentrated on making each part of a computer system as fail-safe as possible and as fault-tolerant as the technology would allow. Now it is recognized that it is impossible to achieve true continuous availability without the careful *management of redundancy.*

3.1.1 Parallel Sysplex is designed to allow management of redundancy

One of the advantages of Parallel Sysplex is that it allows a configuration of multiple redundant parts, each of which is doing real work. In the past, we had the concept of one system acting as backup for another, doing low priority work or essentially very little work unless the primary system failed. This is costly and difficult to manage.

With Parallel Sysplex, many systems are actively doing work and are peers to each other. The implementation of this capability requires multiple access paths through the network, multiple application processing regions and multiple database managers sharing common data, with mechanisms for workload balancing.

In case of a failure, or a need to do preventive maintenance on one of the systems, the remaining systems can assume the extra work without interruption, though users connected through the network to a failed or removed system (for example, a system containing a CICS TOR) will need to reconnect. No intervention is required by the remaining systems.

Several levels of redundancy are provided in Parallel Sysplex. In addition to *system redundancy*, there can also be redundancy at, for example, the subsystem level, such as having several AORs per z/OS image.

The old adage says that you should not put all your eggs in one basket. With one very large single system, that is what you are doing. This is binary availability, so either all of it runs, or none of it runs.

If, instead, the processing power is broken up into a number of smaller pieces, much of the total capacity remains if one of these pieces break. For this reason, many installations today run multiple stand-alone systems. The workload is partitioned so that a single system outage will not affect the entire installation, although one of the workloads will not run. To manage installations in this manner, the systems programmer must manage several systems, Sysres packs, master catalogs, and parmlib members, all of which are different.

Parallel Sysplex allows the system programmer to manage several copies of one single system image. Each of the systems in Parallel Sysplex can be a clone of the others, sharing master catalogs, Sysres packs, and parmlib members. In addition, because of the fact that each individual system can access the data equally, if one system is lost, the work is shifted to another system in the complex and continues to run.

Parallel Sysplex- the instant solution?: You should not be under the illusion that implementing Parallel Sysplex alone will immediately produce continuous availability. The configuration of Parallel Sysplex has to conform to the same requirements for addressing continuous availability as in a non-sysplex environment. For example, you must have:

- Adequate redundancy of critical hardware and software
- No single points of failure
- Well-defined procedures/practices for:
  - Change management
  - Problem management
  - Operations management

Thus, the design concept of Parallel Sysplex provides a platform that enables continuous availability to be achieved in a way that is not possible in a non-sysplex environment, but it cannot be achieved without considering of the basic rules of configuring for continuous availability.

Also, the full exploitation of data sharing and workload balancing available through the Parallel Sysplex, and the level of software that will support these functions, will ultimately help provide continuous availability.

When continuous availability is discussed, it is often centered on hardware reliability and redundancy. Much emphasis is put on ensuring that single points of failure are eliminated in channel configurations, DASD configurations, environmental support, and so on. These are important, but far short of the whole picture.

A survey was done to determine what activities were being carried out in installations during planned outages. The numbers quoted are not necessarily definitive, but do serve to highlight a point. The percentages show the amount of time spent on each activity:

- Database backups: 52%
- Software maintenance: 13%
- Network: 10%
- Application: 8%
- Hardware maintenance: 8%
- Other: 9% (for example, environmental maintenance)

The key message from this survey is that during the time allocated for planned outages, nearly 75% of the activity is directed at *software-related issues*. Therefore, while the hardware element is important when planning for continuous availability, equal if not more consideration must be given to the software element.

Recommendation: We recommend that an analysis of all outages that occurred over the previous 12-24 months be carried out and compared with an estimate of what would have happened in Parallel Sysplex. In many cases, the analysis may need to be qualified by statements such as: *for systems that have implemented data sharing*, or *provided transaction affinities have been resolved*. In addition, we recommend that a formal availability review be done for hardware, software, and the environment, for both failures and changes. Parallel Sysplex with data sharing and load balancing is a prerequisite for continuously available OS/390 systems, but so is a well-designed environment.

A white paper that provides a list of things to check for, to help you obtain the highest possible application availability, is available on the Web at:

`http://ibm.com/servers/eserver/zseries/library/whitepapers/pdf/availchk_parsys.pdf`

3.1.2 Planned outages

Planned outages constitute by far the largest amount of system down time. According to recent IBM studies, roughly 90% of all outages are planned outages. For an installation achieving 99.5% planned availability, this would equate to roughly 400 hours of planned down time a year. Based on the industry figures quoted in 2.1, “Deciding if Parallel Sysplex is right for you” on page 14, this equates to a cost of between $10,000,000 (US) and $2,600,000,000 (US) per annum per installation. Even allowing that planned outages will be during off-peak hours when the impact will be minimized, it can be seen that the real cost is substantial.

Restarting an image to add new software or maintenance, to make configuration changes, and so on, is costly. Much progress has already been made in this area to avoid IPLs. For example, z/OS has many parameters that can be changed dynamically, I/O configurations can be changed dynamically, and, by eliminating parallel channels, new peripheral devices can be added dynamically. Dynamic I/O reconfiguration even allows a channel path definition to be changed from converter (CVC) to ESCON® (CNC) and FICON® converter (FCV) to FICON.

In Parallel Sysplex, HCD supports cross-system activation of the new I/O configuration. This is discussed in detail in the IBM Redbook *MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*, SG24-4037. Dynamic I/O reconfiguration has been in use for many years at most installations.

Most parts of the hardware allow concurrent repair or maintenance, resulting in fewer outages due to this work.

Capacity Upgrade on Demand (CUoD) and Customer Initiated Upgrade (CIU) allows processor upgrades to add processors for z/OS or ICF upgrades concurrent with normal operation by utilizing a currently spare PU. In LPAR mode, the CP can be added to the shared pool of CPs, or it can be used as a dedicated CP by an LP that uses dedicated CPs–in either case, no POR or image reset is required.

The Capacity Backup Option (CBU) is a disaster recovery option that exploits CUoD to nondisruptively bring additional processor capacity online. A typical use would be in a remote site where a one-way 2084-B16 could be upgraded via CBU to a 12-way 2084-B16 if the primary site is lost in a disaster. IBM System z9 has expanded the availability of CBU to ICFS.

Special Terms and Conditions control the use of CBU. The CBU Users Guide contains more information about the use of CBU.

Concurrent I/O conditioning on zSeries pre-installs support cards (such as FIBBs and channel driver cards) so that channel cards (such as ESCON, FICON, and ICBs) can later be added concurrently. Because FICON and ICBs can only occupy certain I/O slots, installation of this feature may involve moving existing channel cards and CHPID number changes.

On z990 and later processors, only rarely would the addition of an I/O card be disruptive. Some upgrades are still non-concurrent, for example, some storage upgrades or the addition of an I/O cage.

Parallel Sysplex can make planned outages less disruptive. A system can be removed from the sysplex and work will continue to be routed to the remaining systems. From the user perspective, the application continues to be available, (although users may have to log on again to reestablish the session), even if one of the systems is down.

While a system is removed from the sysplex, maintenance or new software levels can be applied. When the system is reintroduced to the sysplex, it can coexist at the new software level with the other systems. New function might not be available until the software is upgraded on all systems, but existing functions are not affected.

In order for this strategy to work, new levels of software must be written to allow compatibility with previous levels. No software maintenance should require a sysplex-wide IPL. See 2.2.3, “Software coexistence considerations” on page 19 for more information about the IBM strategy for this coexistence.

3.1.3 Unplanned outages

For many years, IBM has continued to improve the fault-tolerance and nondisruptive upgradability of the hardware.

The processor module on zSeries or processor book(s) on z9 109 processors contains identical Processing Units (PUs), which can function as z/OS CPs, System Assist Processors (SAPs), ICF CPs, IFL CPs, or zAAP CPs (z990 and z9 109 only), depending on microcode load. Not all processors have spare PUs, because they may all be in operation in normal use–for example, a 2066-004 or a 2086-420 has none.

Sparing involves the utilization of a spare PU in the processor module to replace a failing PU (which could be an OS/390 CP, an SAP, or a specialist engine.

With improvements in sparing on G5 and later CPCs, if a spare PU is available, the failure of an SAP, ICF, or z/OS CP does not cause an outage in most cases.

The zSeries and IBM System z9 RAS strategy is a building-block approach developed to meet the client's stringent requirements of achieving Continuous Reliable Operation (CRO). Those building blocks are: Error Prevention, Error Detection, Recovery, Problem Determination, Service Structure, Change Management, and Measurement and Analysis.

The initial focus is on preventing failures from occurring in the first place. This is usually accomplished by using *Hi-Rel* (highest reliability) components from our technology suppliers, using screening, sorting, burn-in, run-in, and by taking advantage of technology integration. For Licensed Internal Code (LIC) and hardware design, failures are eliminated through rigorous design rules, design walk-throughs, peer reviews, element/subsystem/system simulation, and extensive engineering and manufacturing testing.

The z990 RAS strategy is focused on a recovery design that is necessary to mask errors and make them *transparent* to client operations. There is an extensive hardware recovery design implemented to be able to detect and correct array faults. In cases where total transparency cannot be achieved, the capability exists for the client to restart the server with the maximum possible capacity.

Parallel Sysplex has superior availability characteristics in *unplanned*, as well as in planned, outages, because work continues to be routed to existing systems when one system is removed. In addition, when a system fails or loses connectivity to the rest of the sysplex, the remaining systems can immediately take steps to make sure that the system in question cannot contaminate any existing work; they do this by cutting the failed system off from the running sysplex. The sysplex components, working together, also begin to recover the failing system and any transactions that were in progress when the failure occurred.

The installation-established recovery policies determine what happens when failures occur, including how much is done automatically and how much requires operator intervention.

3.1.4 Scope of an outage

An additional consideration for planning for availability is to limit the scope of the effects outages will have. For example, two smaller, physically separate CPCs have better availability characteristics than one CPC twice as powerful, because a catastrophic hardware failure causes only half the available capacity to be lost. In a CICS environment, splitting an application region (AOR) into two AORs in Parallel Sysplex allows work to flow through one region if another has failed. Putting the two AORs on separate physical hardware devices gives even more availability. Figure 3-1 shows how separating elements can help limit the scope of failure.

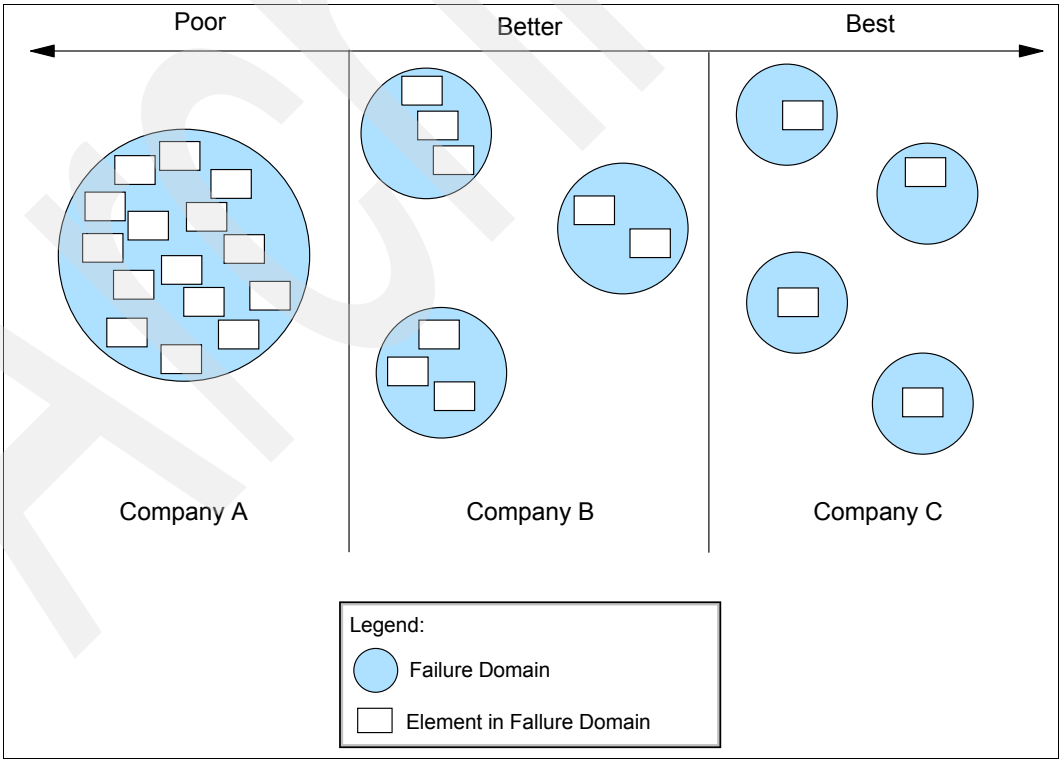


Figure 3-1 Failure Domain example

The more separate pieces, or domains, there are, the less total capacity there is to lose when a domain fails. CICS is an example of how this concept applies. Other examples to create *small* failure domains include:

- Separate power connections to hardware where possible.
- Physically separate channels and CF links that attach the same pieces of hardware. Care should also be taken to spread the channels and CF links over as many cards as possible. The Chpid Mapping Tool provides support to do this to configure for maximum availability on z990 and IBM z9 Processors when assigning chpids to pchids.
- Place the CFs and the z/OS LPs they are coupled to in separate CPCs.
- Define backup NCPs to allow minimal or fast recovery.
- Put backup or spare copies of critical data sets on volumes belonging to different DASD subsystems. An example of this is primary and alternate couple data sets.
- Use disk mirroring or dual copy for all important data sets, such as shared Sysres, all catalogs, other vital system data sets, and, obviously, all critical user data.
- Have a mixture of CF structures and CTCs for XCF signaling.
- Use XCF for inter-VTAM and inter-TCP/IP communication.
- Spread CF structures across multiple CFs according to the current guidelines, and do a rebuild on separate CFs.
- Choose Sysplex Timer redundancy features.
- Distribute channels, CTCs, and control units across multiple switches.

The following sections briefly touch on some of these examples. For a more complete discussion, refer to the following IBM Redbooks:

- *Continuous Availability S/390 Technology Guide*, SG24-2086
- *System/390 MVS Parallel Sysplex Continuous Availability Presentation Guide*, SG24-4502

Various methods exist to assist in determining failure domains or single points of failure. One such method is Component Failure Impact Analysis (CFIA). A CFIA study can be performed by your IBM Services specialist.

3.2 Software considerations for availability

When managing multiple systems, it is much easier to manage multiple copies of a single image than to manage many different images. For that reason, it is sensible to create z/OS systems that are clones of each other. They can share the master catalog, Sysres volumes, and Parmlibs and Proclibs. There is additional information in Appendix A, “Systems Management Products for Parallel Sysplex“, of *OS/390 Parallel Sysplex Configuration, Volume 3: Connectivity*, SG24-5639 about cloned z/OS systems.

However, there are decisions to be made about striking a balance between the benefits of sharing and the risks of creating single points of failure. This is discussed in more detail in the IBM Redbook *Parallel Sysplex - Managing Software for Availability*, SG24-5451.

3.2.1 z/OS considerations

In a non-sysplex environment, certain system data sets are key to high availability. These data sets become even more crucial in Parallel Sysplex because they could now be shared by all the systems in the sysplex and as such become potential single points of failure.

Consideration should be given to the use of disk mirroring techniques or Hiperswap to ensure a live backup at all times. Examples of such resources are:

- Master Catalog
- Couple Data Sets

  Place the alternate couple data set on a separate control unit from the primary.

- Sysres

Refer to *Achieving the Highest Levels of Parallel Sysplex Availability,* SG24-6061 to get more information about considerations for these datasets (such as sharing the Sysres volumes).

Couple data set considerations

When formatting the sysplex couple data set, specifying the GROUP and MEMBER keywords in the format utility determines the number of XCF groups and the number of members per group that can be supported in the sysplex. These values should not be overspecified, as this can lead to elongated recovery or IPL times.

These values cannot be *decreased* nondisruptively using an alternate couple data set and the SETXCF COUPLE,PSWITCH command. A sysplex-wide IPL is required. However, this technique can be used to *increase* the values for GROUP and MEMBER nondisruptively.

For further couple data set considerations, see *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061, and for general details on couple data sets, see *z/OS MVS Setting Up a Sysplex*, SA22-7625.

JES3 and continuous availability

JES3 does not read the Initialization Stream In a Hot Start or a Hot Start with Analysis. JES3 Hot Start with refresh was introduced to enable changes to the initialization stream without a sysplex-wide IPL.

During a Hot Start with Refresh only, changed statements are processed, and statements that cannot be changed are checked for correctness. Table 3 of *z/OS V1R7.0 JES3 Initialization and Tuning Guide,* SA22-7549 lists the parameters that can be changed.

JES3 managed devices are eligible for dynamic I/O reconfiguration. Many years ago, this was not the case.

XES external hang detect

Sometimes, problems can arise in a sysplex where one application in the sysplex issues a request to another member, but never gets a reply. An example might be a subsystem that is using a CF structure and wishes to rebuild the structure to another CF. The subsystem initiating the rebuild will contact all its peers on the other z/OS images in the sysplex, requesting them to take part in the rebuild. If one of those peers does not reply, the rebuild process will hang. Unfortunately, it is not easy to identify which process in the sysplex is the one that is causing the hang.

To address this situation, *XES External Hang Detect* was introduced in OS/390 V2R8. With this support, XES on the system to which an event is delivered for processing by an instance of the structure exploiter will start a timer every time it receives such an event. If the task does not respond within two minutes, XES will issue an IXL040E/IXL041E message; identify the system, jobname, and ASID of the task that is not responding. While this does not alleviate the hang situation, it does identify the causer of the hang, enabling the operators or automation to respond appropriately, thus freeing up the hang.

3.2.2 Subsystem considerations

This section provides pointers to information about subsystems and availability issues.

CICS

The IBM Redbook *Planning for CICS Continuous Availability in a MVS/ESA Environment*, SG24-4593 still addresses most of the fundamental issues associated with CICS availability. See the Unavailability Cause and Solution Checklist in Chapter 1 of that redbook for an analysis of causes of unavailability and strategies for circumventing them. CICS exploitation of the system manages coupling facility duplexing, which can help maximize the availability of your CICSPlex:

- Shared Temporary Storage
- Shared User-defined Data Tables
- Named Counter Server

CICSPlex SM provides the intelligent workload balancing and real time analysis (RTA) to help detect problems in the CICSPlex and maximize availability.

CICS use of VTAM persistent sessions (see below) or TCP/IP use of VIPAs also minimizes the risk of communications failures between users and CICS, and between CICS regions.

DB2

Several years ago, DB2 UDB for OS/390 V6 introduced the ability to duplex the DB2 Group Buffer Pools (GBps). This effectively removed the GBps as a single point of failure, and dramatically reduced recovery time following a CF, CF link, or structure failure. This is the only instance of User Managed Duplexing in IBM products.

GBP duplexing works well if used with an Integrated Coupling Facility (ICF). By placing the primary copy of the GBP in the ICF, you can enjoy the performance aspects of an ICF and still rely upon the availability characteristics of a duplexed structure. For high availability, do not place the DB2 SCA or Lock structures in an ICF that is in the same failure domain as any of the members of the DB2 data sharing group.

DB2 UDB for OS/390 V7 introduced a new startup option, to speed up recovery in a data sharing environment. *Restart Light* allows you to do a cross-system restart of a DB2 member using a smaller DB2 storage footprint and optimized for retained lock freeing (DB2 terminates after releasing as many retained locks as it can). This improves availability in that the resources protected by retained locks can get freed quicker.

See Chapter 13, “Achieving 24x7”, *DB2 UDB for z/OS: Design Guidelines for High Performance and Availability*, SG24-7134 for the latest information.

IMS

IMVS V9, the most recent version of IMS at the time of writing, improved database availability for HALDB databases by providing the long-awaited online reorganization (OLR) enhancement.

The IMS Parallel Sysplex Rapid Network Recovery feature uses the facilities of VTAM MNPS to dramatically reduce the elapsed time required to reconnect all user sessions following a failure of the IMS subsystem.

IMS also provides:

- Shared message queues
- Shared VSO DEDBs
- Shared Fast Path DEDBs with SDEPs
- VTAM generic resource support
- Fast database recovery
- Fast Path DEDB online change

VTAM GR support allows you to access multiple IMS subsystems using a single GR name, offering a single-system image while using the resources of many IMS subsystems. In general, if one IMS subsystem fails, you can log onto another IMS subsystem in that GR group. This is discussed in more detail in the IBM Redbook *Using VTAM Generic Resources with IMS*, SG24-5487.

The Fast Database Recovery (FDBR) feature provides quick access to shared database resources in a sysplex environment by releasing records that might otherwise be locked by a failed IMS until the failed system is restarted. This can significantly reduce the impact of an IMS failure in a data sharing environment, especially if you are using shared message queues.

IMS uses Universal Time Coordinates (UTC) for its internal time stamps. However, this is only a solution for the IMS code, but it cannot address problems with time changes in applications. Clients frequently still take their IMS systems down for this reason.

These features are discussed in 4.1.6, “IMS” on page 200. For further information about all of the enhancements, refer to:

- *IMS Version 9 Implementation Guide: A Technical Overview*, SG24-6398
- *IMS V9 Release Planning Guide,* GC17-7831
- *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908
- *IMS in the Parallel Sysplex Volume II: Planning the IMSplex*, SG24-6928

3.2.3 Subsystem software management

For each of the subsystems (CICS, DB2, MQ, and IMS), you need to have a strategy that will allow you to apply maintenance or install a new release without impacting application availability. You must be able to introduce new levels of code onto one system at a time. You also need to be able to keep the remaining subsystems up and running while you stop one to make a change to it. And you need to have the ability to quickly and easily back out any changes, to allow a subsystem to fall back to a prior software level as quickly as possible. All these requirements, and a suggested way of addressing them, are addressed in the IBM Redbook *Parallel Sysplex - Managing Software for Availability*, SG24-5451.

3.3 VTAM network considerations for sysplex availability

The Communications Server for z/OS provides several facilities that can increase the availability of both SNA and TCP/IP applications in a Parallel Sysplex. An enhancement to the APPN architecture called High Performance Routing (HPR) can increase network availability for SNA applications. TCP/IP has a number of functions that can enhance availability of TCP applications in a Parallel Sysplex. Each of these functions is discussed in this section.

3.3.1 VTAM generic resources function

The ability to view the Parallel Sysplex as one unit of many interchangeable parts is useful from a network perspective. For workload balancing, it is nice to be able to have users simply log on to a generic name, and then route that logon to a system with available capacity, keeping the routing transparent to the user. The user therefore views the system as one unit, while the workload balancing under the covers knows that it really consists of multiple parts. This capability was introduced by VTAM V4.2. The function is called *generic resources*. It allows the installation to specify a single generic name to represent a collection of VTAM applications in the Parallel Sysplex. You may define several generic names to represent different workload collections.

The generic resources function also provides a level of availability in the Parallel Sysplex. If a system fails while a user is logged on, he must log on again, but simply to the same generic name as before. The new session is then established with another subsystem in the Parallel Sysplex. Therefore, the user no longer has to know the name of a backup system and can get back onto the application faster.

For more information about the VTAM generic resources function and how it works, refer to 4.5.1, “VTAM generic resources function” on page 250. For more information about subsystems exploiting the VTAM generic resources function, refer to “Generic resource planning considerations” on page 256

3.3.2 Persistent sessions

Here we discuss persistent sessions.

Single Node persistent session

When a persistence-enabled application program fails, VTAM retains the sessions, saves the allocated resources and control blocks, and shields the network from knowledge of the application program failure. VTAM stores the incoming data so that the network views the session as active but not currently responding. When the failed application program restarts or another application program takes over, VTAM reconnects the sessions.

Multi-Node persistent sessions

VTAM uses the coupling facility in the MVS sysplex to maintain session and HPR connection information for all multi-node persistent session (MNPS) application programs. This allows VTAM to restore application sessions after instances of system or application failure, or as part of takeover of the application.

- For application program failures, an application can be restarted on the same VTAM on which it failed, through the single node persistent session (SNPS) support, or on a different VTAM through MNPS planned takeover or MNPS forced takeover.
- For planned or forced takeover or for operating system, hardware, or VTAM failures, a multi-node persistent session application program can be restarted on any VTAM node that supports multi-node persistent sessions in the sysplex, with the following considerations:
  - If a multi-node persistent application program moves to the VTAM where one of its session partners is located, the sessions with that partner will not be restored. The sessions are terminated and must be restarted. This is also true if the multi-node persistent application program moves to the VTAM that is the partner endpoint of the HPR connections associated with the application session.

– If both session partners are located on the same VTAM when the failure occurs, the sessions between them cannot be restored. The sessions are terminated and must be restarted.
– If the multi-node persistent application moves to a VTAM node that is not connected to the multi-node persistent session coupling facility structure, no sessions are restored.

Until the application program is restarted, incoming data is maintained at the other end of the HPR connection. When the application is restarted and before the sessions are restored, VTAM stores the incoming data.

High Performance Routing

As mentioned previously, HPR is a prerequisite for the Multi-Node Persistent Session support. High Performance Routing meets the following requirements:

Improved APPN data routing

HPR transports data at very high speeds by using low-level intermediate routing and by minimizing the number of flows over the links for error recovery and flow control protocols. The flows are minimized by performing these functions at the endpoints rather than at each hop (link) along the path.

Improved APPN reliability

HPR switches paths within the HPR portion of the network to bypass link and node failures if an acceptable alternate path is available. This occurs transparently to the sessions; in other words, the session is not disrupted.

Functional equivalence

HPR maintains functional equivalence with APPN. To do so, HPR continues to support priority routing, connection networks, and multiple network connectivity. Priority routing allows the capability for higher priority traffic to pass lower-priority traffic in intermediate nodes within the HPR portions of the network. HPR also routes across connection networks or subnetwork boundaries in much the same way as APPN. HPR routes are not given preferential treatment by the APPN routing algorithm. Existing non-HPR APPN routes will also be used if they meet the requirements of the APPN Class of Service.

Seamless migration

HPR is designed for *drop-in* migration. A given APPN node can be upgraded to the HPR level of function without taking down the network, without configuring new parameters at its adjacent nodes, and without any logistical complications or coordination.

3.3.3 VTAM systems management

A number of systems management facilities supported by VTAM can increase overall availability.

Automatic Restart Manager support

Support for the Automatic Restart Manager (ARM) facility of z/OS was introduced in VTAM V4.3. ARM exploitation reduces the impact on users when a VTAM failure occurs. VTAM will register with ARM at startup time. ARM can then automatically restart VTAM after a failure, as long as the ARM function is still active. Other ARM participants are automatically notified after the VTAM recovery. For more information about ARM, refer to 3.7.2, “Automatic restart management (ARM)” on page 128.

Cloning

VTAM V4.3 was the first release to support the cloning facilities in z/OS. VTAM allows you to dynamically add applications in the Parallel Sysplex environment. Model application definitions can be used, so that a single set of VTAMLST members can be used for all VTAM application major nodes.

VTAM can also be an z/OS cloned application itself, using z/OS symbols in all VTAMLST members. It is possible to use a single set of VTAMLST members for all the VTAM systems. It is also possible to use z/OS symbols in VTAM commands.

If the VTAM systems are defined as APPN nodes (either end nodes or network nodes), it is much easier to make full use of the z/OS cloning facilities. VTAM APPN systems require fewer system definitions than VTAM subarea systems. The definitions that are required are simpler and lend themselves readily to the use of the z/OS symbols.

VTAM's use of cloning: VTAM systems that use APPN rather than subarea protocols can make optimum use of the z/OS cloning facilities.

The cloning of applications and the cloning of VTAM itself make it much easier to replicate systems in a Parallel Sysplex environment.

APPN dynamics

APPN dynamics offer many advantages that may enhance availability. APPN features greater distributed network control that avoids critical hierarchical dependencies, thereby isolating the effects of single points of failure; dynamic exchange of network topology information to foster ease of connection, reconfiguration, and adaptive route selection; dynamic definition of network resources; and automated resource registration and directory lookup. APPN extends the LU 6.2 peer orientation for user services to network control and supports multiple LU types, including LU0, LU1, LU2, LU3, and LU6.2.

VTAM V4.4 provides the ability to use XCF services to create *logical* APPN S/390 server-to-S/390 server connections, thus eliminating the need for VTAM CTCs.

3.4 IP network considerations for sysplex availability

Recommended sources of further information for this section:

- *Leveraging z/OS TCP/IP Dynamic VIPAs and Sysplex Distributor for higher availability*, found on the Web at:

  `http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130165.pdf`
- Chapter 8. TCP/IP in a sysplex *z/OS V1R7.0 Communications Server: IP Configuration Guide*, SC31-8775
- *TCP/IP in a Sysplex*, SG24-5235
- "Sysplex Distributor in z/OS 1.7", *z/OS V1R7.0 Hot Topics Newsletter,* GA22-7501 August 2005

Virtual IP Addressing (VIPA) was introduced to remove the dependency of other hosts on particular network attachments on the zSeries or z9 processor. Each IP packet has a source and destination IP address. Intermediate nodes in an IP network, individual links, and IP addresses are important only to other IP routers and switches. If a link fails, then normally the

traffic flows over alternative links. However, the endpoint IP address is critical. Before VIPA, this was the physical port.

VIPA provides an IP address that is owned by a TCP/IP stack (in the z/OS system) and not by any physical adapter, so the failure of a physical adapter does not cause the loss a VIPA. It is always available as long as the owning TCP/IP stack is up. Since the z/OS system should be configured with multiple redundant IP adapters, VIPA provides failure independence from any particular adapter.

If the TCP/IP stack fails or is taken down (or z/OS), the system fails, or is taken down, then the VIPA disappears. A further extension to VIPA - Dynamic VIPA (DVIPA) - was introduced. With this extension, the VIPA can be moved automatically to another TCP/IP stack in the sysplex. TCP/IP stacks in a sysplex exchange information about DVIPAs and the TCP/IP stacks are aware of the status of the other TCP/IP stacks in the sysplex.

To summarize, a (static) VIPA protects an IP address from adapter failures and DVIPAs protects against software failures of a single TCP/IP stack (or z/OS image) image.

See Chapter 7, “Virtual IP Addressing” of *z/OS V1R7.0 Communications Server: IP Configuration Guide*, SC31-8775 for exhaustive details.

3.4.1 Virtual IP Addressing

Virtual IP Addressing (VIPA) can improve availability in a sysplex environment. The VIPA function is used with the routing daemon (RouteD or OMPROUTE) to provide fault-tolerant network connections to a TCP/IP for z/OS system. The RouteD daemon is a socket application program that enables TCP/IP for z/OS to participate in dynamic route updates, using the Routing Information Protocol V1 or V2 (RIP-1). OMPROUTE supports the Open Shortest Path First (OSPF), and RIP routing protocol.outeD is no longer supported starting with z/OS V1R7.

The VIPA functions allow the installation to define a virtual interface that is not associated with any hardware components and thus cannot fail. The IP address that is defined for a virtual interface is therefore always available.

Remote IP clients connect to the VIPA address through one of the physical interfaces of the TCP/IP for z/OS system. Name servers must be configured to return the VIPA address of the TCP/IP for z/OS system and not the IP addresses of the physical interfaces. If a physical interface fails, dynamic route updates will be sent out over the other physical interfaces, and downstream IP routers or @server z/Series will update their IP routing tables to use an alternate path to the VIPA address. The effect is that TCP connections will not be broken, but will recover nondisruptively through the remaining physical interfaces of a TCP/IP for z/OS system.

Figure 3-2 shows an example of TCP/IP VIPA recovery of an inbound connection.

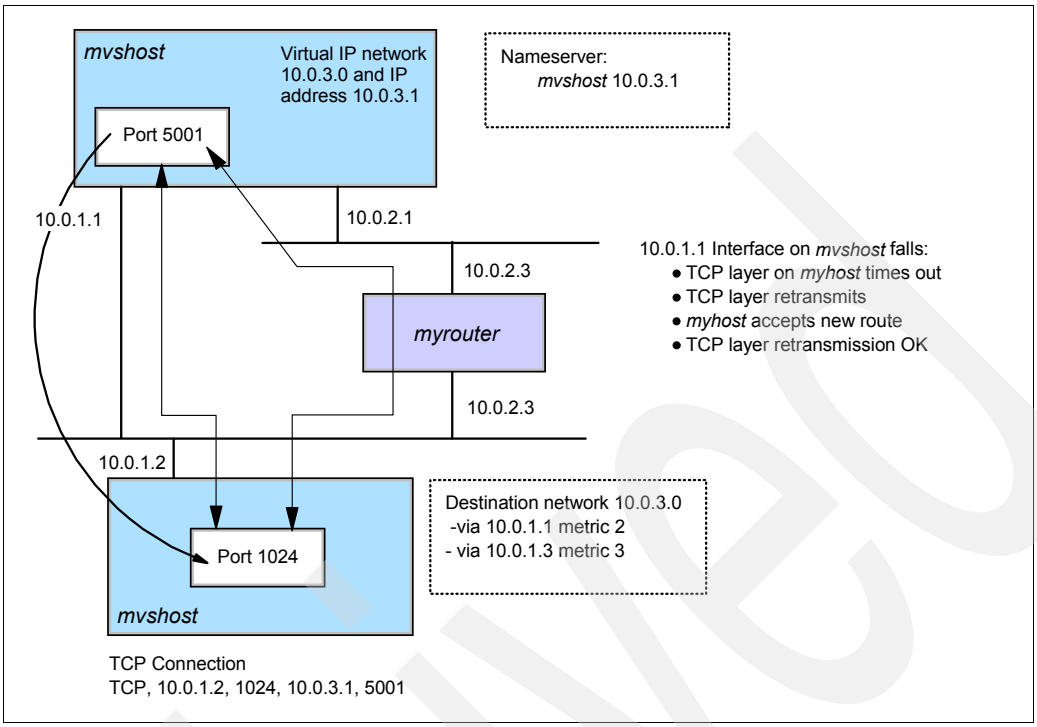


Figure 3-2 TCP/IP VIPA example

If a complete TCP/IP for z/OS system becomes unavailable, connections with clients will be broken. A standby TCP/IP for z/OS system can, in that situation, be dynamically modified to take over the VIPA identity of the failing TCP/IP for z/OS system. After such a takeover action, remote clients can re-connect to the VIPA address of the failing system, and they are connected to a backup system. If the backup system supports the same set of server functions as the failing system supported (such as FTP services, Telnet services, and so on), clients will be able to manually recover by re-establishing their FTP session or Telnet session. Dynamic VIPA Takeover can be used to automatically activate the VIPA on another host.

VIPA may be used with static route definitions, but in case of an interface failure, a manual update of routing tables in both TCP/IP for z/OS and downstream routers is required. This is impractical in any but the smallest network.

For more information, refer to *z/OS V1R7.0 Communications Server: IP Configuration Guide*, SC31-8775.

Because a VIPA is associated with a z/OS TCP/IP stack and is not associated with a specific physical network attachment, it can be moved to a stack on any image in the sysplex or even to any z/OS TCP/IP stack, as long as the address fits into the network configuration. Moving a static VIPA can be done manually by an operator or by client-programmed automation. Movement of the static VIPA allows other hosts that were connected to the primary stack to reestablish connections with a backup TCP/IP stack using the same VIPA. After the primary TCP/IP stack has been restored, the reassigned VIPA address can be moved back. Consider the following when backing up and restoring a z/OS TCP/IP stack:

- All connections on the failing host will be disrupted.
- The client can use any ephemeral port number when reestablishing the connection to the backup server.

- Having a different port number for the backup and primary server is *not* recommended. For example, if the primary FTP used port 21 and the backup FTP used port 1021, when backing up and restoring a z/OS TCP/IP stack, the client would have to know whether to use port 21 or 1021.

3.4.2 Dynamic VIPA takeover and takeback

Here we discuss dynamic VIPA takeover and takeback.

Dynamic XCF

Prior to z/OS 1.7, it was necessary to use DYANAMICXCF for Sysplex Distributor and for DVIPAs. This was because MAC addresses were used for packet forwarding and this required that the destination was exactly one hop away and DYNAMICXCF was a convenient way to ensure this requirement was met.

Use IPCONFIG DYNAMICXCF (Or IPCONFIG6 DYNAMICXCF) statements to create trusted, internal links to other stacks within a sysplex. DYNAMICXCF creates a single IP address by which all other stacks in the sysplex can reach the stack. Dynamic XCF also generates the appropriate DEVICE, LINK, and other definitions and activates the devices to enable the stack to communicate with other stacks in the complex. It can also generate dynamic definitions for TCP/IP stacks on this and other hosts (in the sysplex). From this description, it follows that there can only be one TCPplex per sysplex up to z/OS 1.7. z/OS 1.8 has previewed support for more than one TCPplex per sysplex. There are a number of cases where this could be useful, for example, when merging sysplexes or for outsourcing installations.

z/OS 1.7 Communications Server will allow TCP/IP connectivity through XCF from pure Subarea nodes. This allows users to utilize the full range of TCP/IP sysplex functions without having to redefine the SNA network to use APPN communications. Communications Server provides internal links between TCP/IP stacks on the same z/OS image. This support is called a Same Host (IUTSAMEH) link. If the Communications Server and processor are properly configured, the z/OS 1.7 Communications Server will establish XCF communications between two stacks in LPARs on the same CPC using Hipersockets.

The minimum requirements for any two TCP/IP stacks in a sysplex (whether in one z/OS image or not) to use dynamic XCF are:

- VTAM must have XCF communications enabled.
- DYNAMICXCF must be specified in the profile of each TCP/IP stack.

See Chapter 8, “TCP/IP in a sysplex“, of *z/OS V1R7.0 Communications Server: IP Configuration Guide*, SC31-8775 for examples of definitions generated by Dynamic XCF.

Removal of requirement for DYNAMICCXCF for Sysplex Distributor in z/OS 1.7

When DYNAMIC XCF is not used, TCP/IP will use Generic Routing Encapsulation (GRE) to forward the packet to a unique IP address on the target stack. This address will be configured using a new configuration option in the VIPADYNANMIC block:

```
VIAPAROUTE DEFINE dynxcfIPaddress targetIPaddress
```

This facility could be very useful in multi-site sysplexes when CF link resources might be constrained or if high speed Ethernet facilities are available.

Sysplex problem detection and recovery

z/OS 1.6 and 1.7 have significantly extended the autonomic behavior in error situations of TCP/IP stacks in the IP sysplex.

z/OS TCP/IP monitors its own state to look out for problems and take recovery actions if they occur. The PROFILE.TCPIP file contains a GLOBALCONFIG statement with a TIMERSECS parameter (default of 60 seconds). The sysplex monitor gets control according to this specification and various checks are made, including VTAM address space availability, OMPROUTE availability, and storage availability of various parts of TCP/IP storage (ECSA and so on).

If RECOVERY is specified on the SYSPLEXMONITOR parameter of the GLOBALCONFIG statement (and is recommended), and if the TCP/IP detects one of these conditions, it removes itself from the sysplex. The other stacks are signalled that this is happening and they then can initiate recovery actions, such as moving DVIPAs or removing application instances from distributed application groups.

z/OS 1.7 also has an AUTOREJOIN parameter in which case the TCP/IP stack automatically rejoins the sysplex group when it determines that the condition has cleared.

DVIPA takeover and takeback

DVIPA definitions are contained within a VIPADYNAMIC block. Distributed DVIPAs are defined for and used by the Sysplex Distributor (see below).

During a planned or unplanned outage, the DVIPAs and distributed DVIPAs are taken over by a backup TCP/IP stack. When the primary TCP/IP stack is restarted, the DVIPAs and distributed DVIPAs are taken back.

3.4.3 Network load balancing

Load balancing is the ability of a cluster to spread workload across servers according to some policy. Various techniques for IP load balancing are available for a sysplex:

- Internal load balancing solutions

  These rely on some z/OS component to do the balancing with few dependencies on the network. They have access to load information, which can, in some sense, be used to optimize the load balancing. The two main techniques are DNS/WLM and Sysplex Distributor.

- Sysplex aware external load balancers

  These rely on components inside the sysplex for advice on how to distribute workload. An example of this is the IBM Network Dispatcher. IBM and CISCO have developed a new (open) protocol, Server/Application State Protocol (SASP). The z/OS Load Balancing Advisor (z/OS 1.4 and above) and the CISCO Content Switching Module provides an example of the use of this protocol. Other vendors are able to supply SASP compliant devices. (Of course, SASP supports non-sysplex clusters too, provided they have an analogous advisor.) Using SASP, external load balancers can provide optimized load balancing for a sysplex using sophisticated criteria.

- External load balancing solutions

  These have little awareness of the state of the sysplex and several vendors provide such load balancing solutions.

To be usable, these clustering techniques provide the ability to advertise a single system image or identity to clients. Additionally, clustering techniques should provide for scalability

and systems management. Apart from DNS/WLM, the solutions discussed below use a single IP address to represent the cluster to the client.

For reference, SASP is supported by eWLM on the following platforms:

- IBM AIX® 5L™ Version 5.2
- Microsoft Windows® 2000 Advanced Server, 2000 Server, 2003 Enterprise Edition and 2003 Standard Edition
- Sun™ Microsystems™ Solaris™ 8 (SPARC Platform Edition) and 9 (SPARC Platform Edition)
- zSeries Linux® in the next release of the IBM Virtualization Engine™

Chapter 8, “TCP/IP in a sysplex”, of *z/OS V1R7.0 Communications Server: IP Configuration Guide*, SC31-8775 contains a table giving a detailed comparison between these options. The recommended solutions should use either a SASP-compliant router or Sysplex Distributor.

3.4.4 DNS/WLM

This is only available with Bind 4.9.3 and not with Bind 9. In general, DNS/WLM relies on the host name to IP address resolution as the mechanism by which to distribute load among host servers. Note that the system most suitable to receive an incoming client connection is determined only at connection setup time. DNS/WLM is no longer recommended.

3.4.5 Sysplex Distributor

Sysplex Distributor was announced in OS/390 2.10 and has been enhanced in subsequent z/OS releases. Sysplex Distributor builds on the VIPA and Dynamic VIPA Takeover support to effectively let you have a cloned application, on a number of z/OS systems, with each instance having the same IP address. It provides the benefits of WLM/DNS without requiring WLM/DNS support in the application, nor requiring the client to abide by the Time To Live value specified by the server. Sysplex Distributor also adds the ability to nondisruptively move an application back to its original location after Dynamic VIPA Takeover has been used to move the application after a failure.

Sysplex Distributor (SD) requires no structures in a CF; it uses only XCF (up to z/OS 1‘.6) and can optionally use other network connections in z/OS 1.7.

The single IP address by which the Sysplex Distributor is known is called a distributed DVIPA. The IP entity that advertises this address to the network is itself a single image within the sysplex known as a distributing stack. (If the distributing stack fails, the distributed DVIPA moves to the backup TCP/IP stack, which in its turn becomes the distributing stack.) WLM provides the distributing stack with a WLM recommendation (a server specific WLM weight) for each target server. Sysplex Distributor has the ability to specify certain policies within the Policy Agent so that it can use QoS (Quality of Service) information from target stacks to further modify the WLM recommendation. SD also measures responsiveness of target servers in setting up new TCP connection requests and favors those that are most responsive.

In general, all inbound network traffic passes through the distributing stack, but outbound traffic does not.

From a systems management viewpoint, SD changes do not require much coordination with the network, while the Sysplex aware load balancers are likely to require changes on both host and network.

Sysplex Distributor can be exploited by IMS, CICS, MQ, DB2, and WebSphere Application Manager.

Sysplex Distributor is described in more detail in the *z/OS Communication Server: IP Configuration Guide*, SC31-8775.

Improved workload distribution for Sysplex Distributor in z/OS 1.7

SD uses improved WLM interfaces, Health Indicators for the TCP/IP stacks (such as numbers of connections dropped due to backlog), and values for QoS (Quality of Service) to decide which servers will get new connections, and it can route over any available route, not necessarily over CF links - this is likely to be especially useful in a multi-site sysplex, as we will see in 3.8.1, “Multi-site sysplexes” on page 131, where the CF infrastructure is typically under more stress.

3.5 Hardware considerations for availability

The amount of redundancy to configure is a very important consideration. To a large extent, this will depend on the amount of availability required, and how much cost your business can afford. In most cases, however, the redundant hardware should be used as a normal part of the configuration, rather than having to sit idle until a problem arises.

Generally speaking, and from a pure availability perspective, we recommend at least two of everything. For example, it is strongly suggested that an installation configure at least two CFs, with two CF links to each CPC, which contains an image that is communicating with that CF. The system normally uses all CF links, but if one fails, the others have enough capacity to carry the full load.

The *Parallel Sysplex Availability Checklist*, available on the Parallel Sysplex home page, contains a detailed list of things to consider when configuring hardware for availability.

3.5.1 Number of CPCs in Parallel Sysplex

When planning for continuous availability and a *guaranteed service level*, you might want to configure one more CPC than the number you estimated for performance. In this way, if you suffer a hardware outage, the redistribution of the load to the running CPCs will not cause a performance degradation. Alternately, you can choose to provide extra *capacity* in the CPCs without providing an additional CPC.

3.5.2 Redundant power

zSeries and z9 CPCs have dual power cords that should be connected to separate power sources. In addition, they can be configured with an Internal Battery Feature (IBF). On z9, feature 3210 provides one pair of batteries and the configurator will configure up to three features. The IBF is fully integrated into the server power control/diagnostic system, which provides full battery charge, test, and repair diagnostics.

The IBF can keep a z9 fully operational for between 7 and 13 minutes, depending on configuration.

If power can be restored during this time after a site failure, the IBFs can significantly decrease recovery time for the CF exploiters, because they will not have to allocate and build the structures again. DB2 subsystems can be restarted and do not have to read the logs to rebuild the structures. For a further discussion of UPS and power save mode/state, refer to

the IBM Redbook *System/390 MVS Parallel Sysplex Continuous Availability SE Guide*, SG24-4503.

CF volatility or nonvolatility

The presence of the Internal Battery Feature or UPS determines the *volatility* of the CF; it determines whether the contents of storage are maintained across an external power outage. Certain CF exploiters *react* to the volatility, or change of volatility of the CF. It is important to consider the volatility of the CF when configuring for continuous availability.

The system logger itself (and therefore all exploiters of logger) reacts to a volatile CF as though it were failure dependent (even if it is standalone) and depending on parameter settings, staging datasets will or will not be used depending on whether the CF is volatile or not. (Remember that the logstreams are on the connecting systems, so some connectors may use staging datasets and others may not, to the same structure in the same CF, at least in theory.) This is explained in detail in Chapter 9, “Planning for System Logger Applications”, of *z/OS MVS Setting Up a Sysplex*, SA22-7625.

Other subsystems such as IMS and DB2 simply issue warning messages if they detect a volatile CF.

Special consideration needs to be made for the JES2 Checkpoint; if the checkpoint is lost, a cold start is required. If the JES2 checkpoint is put in a non-volatile CF that becomes volatile, JES2 prompts the operator to start a JES2 reconfiguration dialogue to move it to DASD or another non-volatile CF. Full details are given Chapter 4, “Checkpoint dataset definition and configuration” in *z/OS V1R7.0 JES2 Initialization and Tuning Guide,* SA22-7532.

If you have an external UPS, you need to tell the CF that its status is nonvolatile. You do this with the CF MODE NONVOLATILE command that is entered at the CF console.

3.5.3 Isolate the CF

There are a number of reasons why you may want to isolate some or all of your CFs onto CPCs of their own. The first and most important is for recovery. Depending on the structures residing in the CF, if a CPC containing both a CF LP and an z/OS LP that is connected to the CF LP were to fail, the impact ranges from being minimal to your whole Parallel Sysplex being adversely affected. The exact implications depend on which structures were in the CF that failed.

Coupling Facility Configuration Options, GF22-5042 categorizes the different users of CF into those that do not have to be failure-isolated from those that do. When deciding whether you should have all ICFs, all stand-alone CFs, or some combination in between, you should consult that document to identify the most appropriate configuration for your environment.

An additional reason is discussed in 2.4, “CF architecture” on page 51, and relates to the conversion of synchronous CF requests into asynchronous requests if the z/OS LP is sharing CPs with a CF LP. This could have undesirable performance effects; however, as it does not affect ICF LPs, this should only be a concern if you plan to run a CF in an LP using CP rather than ICF engines.

Generally speaking, if you are doing data sharing with Parallel Sysplex, the data sharing-related structures should reside in a failure-independent CF. If you are only doing resource sharing, it may be acceptable to use CFs that are not failure-independent. Section 2.2.8, “What different 'plexes are there?” on page 28 contains a list of the Parallel Sysplex exploiters that are regarded as resource sharers rather than data sharers. There is also information about the requirements of the different resource sharing exploiters in the IBM Redbook *S/390 Parallel Sysplex: Resource Sharing*, SG24-5666.

System Managed CF duplexing has offered an alternative to CF isolation for some structures. See *System-Managed CF Structure Duplexing*, GM13-0103 and 2.6.1, “Which structures should be duplexed?” on page 66 for more information.

There are a number of options regarding where the CFs could be run. For a production with two CFs in separate CPCs, the viable options are:

- Use at least one stand-alone CF with dedicated CPs. This is the preferred option for a production CF involved in data sharing to contain structures requiring failure isolation.
- Use the ICF facility to run a CF in a CPC that is failure-isolated from the production sysplex z/OS images (this is a variation of the above).
- Use ICFs to run both CFs on CPCs that are *not* failure-isolated from the production sysplex. This is acceptable for resource sharing Parallel Sysplexes, provided the CFs are on separate CPCs.

ICF expansion: You can have an ICF LP with one or more dedicated CPs, plus the ability to expand into the pool of shared operating system CPs. You can have an ICF LP with one or more dedicated CPs, plus the ability to expand into a shared ICF CP or shared MVS CP. This facility is most useful to handle spikes in CF load, such as during structure rebuild and not for normal CF capacity.

See W98028 on Techdocs, Dynamic ICF Expansion for details, found at:

`http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/W98028`

Test CFs

For a test CF, you have more flexibility, mainly due to the reduced performance and recovery requirements for a test sysplex. The options for a test CF are:

- Run in an LP on a stand-alone CF, possibly sharing CPs with a production CF, or with other test CFs. If the CPs are shared between the LPs, you should run the production CF in active wait mode, and run the test CF with Dynamic CF Dispatching enabled. This will ensure acceptable performance for the production Parallel Sysplex, while still having the required functionality on the test Parallel Sysplex.
- Use the ICF facility to place the CF on a CPC. If this really is a test sysplex, it may be acceptable to use a CF that is not failure-isolated, even if you are testing data sharing. IC links could be used to connect to the test z/OS LP if it is running on the same CPC. If the test z/OS is running on another CPC, external CF links must be used.
- Have a CF LP that shares CPs with operating system LPs. If you have a very low MIPS requirement for the test CF, this may be a viable option. To control the amount of resource being consumed by the test CF, Dynamic CF Dispatching should be used. As this is only a test CF, the performance should be acceptable. Again, IC or external CF links could be used to communicate with the z/OS LPs in the test Parallel Sysplex.

The different CF configurations are discussed further in *Coupling Facility Configuration Options*, GF22-5042.

Third CF

Although the recommended configuration of two CFs, each with sufficient resource to accommodate all the allocated structures, can provide industry-leading levels of availability, there may be installations where anything less than 100% is unacceptable. (For example, a requirement for CF redundancy at all times, including during disruptive maintenance of one of the CFs, such as to add storage to a CF, would require at least 3 CFs.)

If your installation falls into this category, you may wish to consider providing a third CF. This can be implemented as either a full production CF, or as a standby, ready to be called into use only when necessary.

Some clients have three CFs in their production sysplex. If you wish to use this configuration, there are no special considerations, other than remembering that all three CFs should be specified in the preference lists, and all three of the CFs should have sufficient MIPS, CF link capacity, and storage to at least support all the critical structures in case of a failure of one of the other two CFs. Rather than splitting your CF load over two CFs, you would split it over three.

If you are using DB2 GBP duplexing, you should especially make sure to specify all three CFs in the preference list for the GBP structures. If one of the DB2 duplexed GBps is lost, DB2 will revert back to simplex mode for that GBP. However, if a third CF is available, and is included in the structure's preference list, DB2 will automatically reduplex into the third CF.

However, if you wish to implement the third CF as a hot standby that only gets used when one of the other CFs is unavailable, the considerations are different.

For a hot standby to be effective, you ideally want it to use little or no resources when it is not being used, but still be able to give production-level response times when it is called into action.

The most likely way that you would implement a hot standby CF is in an LP, using shared CPs, with Dynamic CF Dispatching enabled, and a high weight so that it can quickly get the CPU it needs when called into action.

3.5.4 Additional CF links

For high availability, redundant CF links should be configured between CPCs and CFs. This configuration removes a potential single point of failure in the Parallel Sysplex environment. In most clients, a single CF link should be sufficient to provide an acceptable response time, so two CF links would be sufficient to provide redundancy in case of a CF link failure.

If System Managed CF duplexing is being used, extra care is needed to ensure sufficient CF links.

3.5.5 I/O configuration redundancy

Normal channel configuration techniques should be applied for FICON, ESCON, and any remaining parallel channels in a Parallel Sysplex. Channels should be spread across channel cards/IBB domains and directors to maintain channel path availability. The Systems Assurance Product Review (SAPR) Guide guide for the relevant CPC should be used so that all devices are attached for maximum availability.

Refer to Section 2.7, “DASD”, in *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061 for information about DASD RAS.

Remember that all models of IBM ESCON Directors are now Withdrawn from Marketing (WDFM), meaning that upgrades cannot be ordered.

For further information about FICON and ESCON director features, see Section 2.6, “Switches”, in *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061.

3.5.6 Sysplex Timer redundancy

In any sysplex environment, whether basic or parallel, the Sysplex Timer is a critical device, and can be a single point of failure for the entire sysplex if it is not configured for redundancy. Refer to Section 2.4, “9037 Sysplex Timer Considerations”, in *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061 for more information.

3.5.7 Server Time Protocol (STP)

The 9037 will be WDFM in 2006 and IBM’s recommendation is to migrate to STP over time. It will be possible to migrate nondisruptively from 9037 to an STP network. STP is designed to overcome some limitations in the 9037. From a configuration viewpoint, the major difference is that timer signals will use coupling links and support distances (with repeaters), of up to 100 km.

3.6 Limitations to continuous availability

There are situations in non-sysplex and Parallel Sysplex environments that limit the achievement of continuous availability. It is anticipated that many of these limitations will be fixed over time in further releases of LIC and software. Some of the traditional limitations still exist, but have been addressed to some extent:

- Depending on the types of data you have, and your backup tools, backing up your data may require applications to be stopped so that a coherent, point-in-time backup can be achieved. DB2 and IMS both provide backup facilities that allow the database to be backed up without having to take the data offline. For VSAM, the Backup While Open facility can be used to get a coherent backup without removing access to the data. For other file types, or for full volume dumps, facilities such as the ESS FlashCopy® function can be used to minimize the outage time.
- Database reorganizations are disruptive. This can be offset to some extent through the use of partitioned databases. In this case, the disruption can be reduced to just that part of the database that is being reorganized. DB2 supports partitioned databases and partitions can be dynamically added, and IMS/ESA V7 introduced support for partitioned Full Function databases.

  DB2 has an online reorg capability, whereby the data remains during most (but not all) of the reorg process. Online reorg capability is also available for Fast Path DEDB databases in IMS.
- Some upgrade/maintenance activities on the CF are nonconcurrent. CF duplexing provides a solution for those IBM applications that do not support some form of rebuild.
- Daylight saving time issues for application code: IBM subsystems themselves are able to handle this issue.

Some other infrastructure inhibitors remain for the time being:

- VSAM does not support online reorganization of VSAM files.

With careful planning, it is possible to minimize the disruption caused by these limitations.

3.7 Recovery considerations for availability

When failures occur in a Parallel Sysplex, work can continue on the remaining elements of the system. Also, many features have been added that will make recovery of the failed system easier.

If a failure occurs in a CF, structures may be automatically recovered on an alternate CF if the preference list for the structure contains the alternate CF name. This structure rebuild is initiated by each subsystem that has structures on the CF when the subsystem is informed that a failure has occurred. Not all subsystems support dynamic structure rebuild. See the table in *System-Managed CF Structure Duplexing*, GM13-0103 for more detail.

In the following sections, we discuss Sysplex Failure Management (SFM) and Automatic Restart Manager (ARM). Sometimes confused, in fact they are complementary. SFM is concerned with the big picture and is concerned with keeping the sysplex running, while ARM is concerned with restarting specific workloads if they fail (and the restart could be in the same or another system from where the failure occurred).

3.7.1 Sysplex Failure Management (SFM)

To enhance the availability of a sysplex, XES provides the Sysplex Failure Management (SFM) facility. A recent analysis of multisystem outages indicated that the single change that would have eliminated the largest number of system outages would be the implementation of SFM, using ISOLATETIME(0) to automate the removal of disabled systems. The use of SFM is described in the white paper *Improve Your Availability with Sysplex Failure Management*, which is available on the Web at:

`http://www.s390.ibm.com/products/pso/availability.html`

To use SFM, you must define and activate an SFM policy. The SFM couple data set contains one or more SFM policies that allow an installation to predefine the actions z/OS is to take to handle system failures, signalling connectivity failures, or PR/SM reconfiguration actions.

SFM makes use of some information specified in COUPLExx (and includes all the functionality available through XCFPOLxx).

The SFM Policy can also be used with the REBUILDPERCENT specification in the CFRM policy to determine whether MVS should initiate a structure rebuild when loss of connectivity to a coupling facility occurs.

Failure detection interval and operator notification interval

Each system in the sysplex periodically updates its own status and monitors the status of other systems in the sysplex. The status of the systems is maintained in the sysplex couple data set (CDS) on DASD. *A status update missing* condition occurs when a system in the sysplex does not update its status information in either the primary or alternate couple data set within the failure detection interval, specified on the INTERVAL keyword in COUPLExx, and appears dormant.

The operator notification interval is the amount of time between when a system no longer updates its status and when another system issues IXC402D. This interval is specified in OPNOTIFY in COUPLExx. The operator notification interval must be greater than the failure detection interval.

Overview of SFM policy

An SFM policy has:

- Policy statement
- System statements
- Reconfiguration statements

SFM allows you to define responses for:

- System failures indicated by status update missing condition
- Signalling connectivity failures in the sysplex
- Reconfiguring systems in a PR/SM environment

Only one SFM policy can be active in the sysplex at any one time, but installations may require different recovery actions at different times, for example, during the online day and during the batch window. Different SFM policies can be defined in the SFM couple data set and activated using the SETXCF operator command. This function can be automated using an appropriate automation tool.

Status update missing

SFM allows you to specify how a system is to respond to this condition. The options are PROMPT, ISOLATETIME, RESETIME, and DEACTTIME.

A system that is not updating its status might already be in a disabled wait state. In all cases, we recommend that the Automatic I/O Interface Reset Facility be enabled. This will cause an I/O interface reset and release the systems I/O Reserves. If this option is not enabled, then in some cases, RESERVEs held by the unresponsive system may not be released until the system is eventually manually reset.

If PROMPT (the default) is specified, then IXC402D will be issued. Again, it is important to take the appropriate reset action before responding, otherwise resources or data integrity problems may result. For these reasons, this is not a recommended option.

System isolation allows a system to be removed from the sysplex as a result of the status update missing condition, without operator intervention, thus ensuring that the data integrity in the sysplex is preserved. Specifically, system isolation uses special channel subsystem microcode in the target CPC to cut off the target LP from all I/O and Coupling Facility accesses. This results in the target LP loading a non-restartable wait state, thus ensuring that the system is unable to corrupt shared resources.

The ISOLATETIME option indicates how long SFM will wait after detecting a status update missing condition.

- If a system has not updated its status within the failure detection interval and is producing no XCF signalling traffic, SFM will start to isolate the system at the end of the ISOLATETIME interval.
- If a system has not updated its status within the failure detection interval, but is producing XCF signalling traffic, the operator is prompted to optionally force the removal of the system.

Figure 3-3 on page 125 shows the isolation of a failing system.

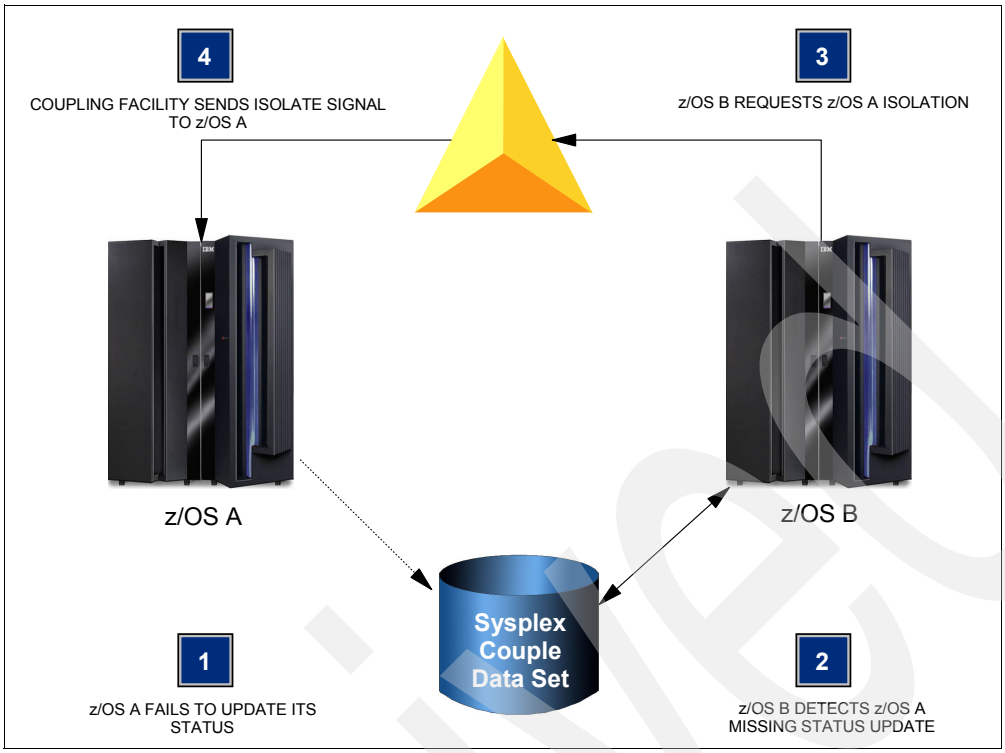


Figure 3-3 Isolating a failing z/OS

In this example, z/OS A fails to update its status and then:

1. This is detected by z/OS B.
2. The z/OS B, in turn, requests, through the CF, the isolation of z/OS A.
3. Having received the request from z/OS B, the CF sends the isolate signal to the CPC that z/OS A is running on.
4. The channel microcode in that CPC then stops all I/O and CF communication for the LP that z/OS A is running in.

Note: This requires that z/OS A is still in communication with the CF.

If an isolation attempt is not successful (because, for example, the failing system is not connected through CF to another system in the sysplex), message IXC102A prompts the operator to reset the system manually so that removal can continue.

Resetting or deactivating a failed system on PR/SM

When a system running fails, another system in the same sysplex running in a partition on the same processor can reset or deactivate the failed system and remove it from the system without operator intervention.

The RESETTIME and DEACTTIME parameters specify how long to wait after the status update missing condition is detected before action is taken.

Handling signalling connectivity failures

All systems in a sysplex must have signaling paths to and from every other system at all times. SFM can eliminate operator intervention when signalling connectivity is lost between two or more systems.

The CONNFAIL parameter indicates whether SFM is to handle signaling connectivity failures for the sysplex. CONNFAIL is specified on the DEFINE POLICY statement of the SFM policy.

SFM allows you to assign a relative value WEIGHT to each system in the Parallel Sysplex. The weight is used for two purposes:

- If there is a partial loss of connectivity between the systems in the sysplex, SFM uses the weight to determine which systems should be isolated in order to let the most important systems continue processing. (Remember full XCF connectivity is needed between every member of a sysplex.)
- The weight is also used to decide whether a structure should be rebuilt after a system loses connectivity to a CF.

In this way, the most important systems in the Parallel Sysplex can continue *without operator intervention* in the event of a failure.

As a simple example, assume there are three systems in a sysplex. z/OS A has WEIGHT(10), z/OS B has WEIGHT(10), and z/OS C has WEIGHT(30). There is an XCF connectivity failure between z/OS B and z/OS C. The alternatives are to continue with z/OS A and z/OS B (total WEIGHT=20), or to continue with z/OS A and z/OS C (total WEIGHT=40). The latter configuration is used; that is, z/OS B is isolated from the sysplex. If all systems in this example had the same weight (that is, if the weight of z/OS C was 10), it would be unpredictable which two systems would be kept in the Parallel Sysplex.

Note: The defined weights are not used when reacting to a *status update missing* condition. In this case, the system that has not updated its status is partitioned, regardless of its weight.

Weights are attributed to systems by the systems programmer based on a number of factors determined by the installation, such as:

- Importance of the work running on a system.
- The role of the system–for example, the system that owns the network might be deemed to be high importance, even if it does not actually run any applications.
- MIPS of the systems in the sysplex.

Weight is any value from 1 to 9999. Specifying no weight is equivalent to specifying WEIGHT(1). If you do not specify a WEIGHT parameter in the SFM policy, every system is given the same importance when it comes to partitioning.

Handling coupling facility connectivity failures

The SFM policy, in conjunction with the CFRM policy, is also used to determine the rebuild activity for those CF structures that support it. The relationship between SFM and CFRM in the area of rebuild is that REBUILDPERCENT (CFRM) is calculated using the WEIGHT values as defined to SFM.

Using the SFM weights, MVS calculates the weighted percentage of lost connectivity and compares this with the user specified REBUILDPERCENT value. If the calculated percentage is greater than or equal to that specified by REBUILDPERCENT, MVS will initiate structure rebuild.

For this reason, we recommend REBUILDPERCENT(1) so that rebuild will take place for any loss of connectivity, since we recommend always to configure with sufficient white space to allow a rebuild to be successful.

Planning PR/SM reconfigurations

After a system running in a PR/SM partition is removed from the sysplex, SFM allows a remaining system in the syplex to reconfigure processor storage for use by the remaining systems.

PR/SM reconfiguration statements are specified on the RECONFIG statement of the SFM policy. PR/SM reconfiguration definitions include the name of the failing system (FAILSYS), the name of the system that is to perform the configuration (ACTSYS), and whether PR/SM is to deactivate a specific system or all logical partitions in the range of the acting system (TARGETSYS). To carry out a PR/SM reconfiguration, ACTSYS and TARGETSYS must be on the same processor, FAILSYS can be on the same system as TARGETSYS, or it can be another system on the same or another processor.

Other SFM considerations

You can configure a CF using ICF and z/OS image in a single CPC, both of which are in the same sysplex. In this configuration, if the CPC fails, both the CF and the z/OS in the same CPC fail. This is called a *double failure*. To avoid the impact of a single point of failure, most components using a CF structure have implemented structure rebuild support, so that the structure can be automatically recovered. But for a double failure situation, some components may take a long time to recover or require manual intervention during the recovery process. Especially in this double failure situation, the most important thing for high availability is to isolate the failing system from the sysplex as quickly as possible so that recovery processing can proceed on the remaining systems in the sysplex. Therefore, it is imperative to:

- Ensure that the INTERVAL value is not too long.
- Ensure that the ISOLATETIME value is very low (ideally zero). This implies that PROMPT is *not* specified in the SFM policy.
- Ensure that operational procedures are in place to quickly detect the issuance of IXC102A and respond to the WTOR. There is specific Parallel Sysplex support added to the product System Automation for z/OS by APAR OW39485 that deals with this, and other, Parallel Sysplex-related messages.
- Ensure that operational procedures are in place to monitor the recovery process.
- Ensure that all exploiters that require rapid recovery are not allowed to have their structures placed in a CF that can allow a double failure to occur.

These considerations are particularly important when you are using CF duplexing, since no duplexed CF requests can complete until the failure of the duplexed CF structures has been detected (and this only happens after the failure of z/OS systems on the failing CPC has been detected).

Recommendation for SFM: A key impetus for moving to Parallel Sysplex is the business need for continuous availability, so minimizing operator intervention and ensuring a consistent and predefined approach to recovery from failure of elements within the sysplex is paramount. SFM achieves this requirement.

SFM is discussed in detail in the IBM Redbook *System/390 MVS Parallel Sysplex Continuous Availability SE Guide*, SG24-4503, Chapter 2, “High-level design concepts for Parallel Sysplex” on page 13, and in *OS/390 V2R10.0 MVS Setting Up a Sysplex,* GC28-1779.

3.7.2 Automatic restart management (ARM)

The purpose of automatic restart management (ARM) is to provide fast, efficient restarts for critical applications. These can be in the form of a batch job or started task (STC). ARM is used to restart them automatically whether the outage is the result of an abend, system failure, or the removal of a system from the sysplex.

When a system, subsystem, or application fails, it may hold database locks that cannot be recovered until the task is restarted. Therefore, a certain portion of the shared data is unavailable to the rest of the systems until recovery has taken place. The faster the failed component can be restarted, the faster the data can be made available again.

ARM is a function in support of integrated sysplex recovery and interacts with:

- Sysplex Failure Management (SFM)
- Workload Manager (WLM)

ARM also integrates with existing functions in both automation (SA for z/OS) and production control (Tivoli® TWS) products. However, care needs to be taken when planning and implementing ARM to ensure that multiple products (TWS and SA for z/OS, for example) are not trying to restart the same elements. SA for z/OS support has proper tracking of ARM-enabled elements to ensure that multiple restarts are avoided.

For more information about WLM, see *System Programmer's Guide to: Workload Manager*, SG24-6472.

ARM characteristics

ARM was introduced in MVS/ESA V5.2. ARM requires a couple data set to contain policy information, as well as status information, for registered elements. Both JES2 and JES3 environments are supported.

The following describe the main functional characteristics:

- ARM provides only job and STC restart. Transaction or database recovery are the responsibility of the restarted applications.
- ARM does not provide initial starting of applications (first or subsequent IPLs). Automation or production control products provide this function. Interface points are provided through exits, event notifications (ENFs), and macros.
- The system or sysplex should have sufficient spare capacity to guarantee a successful restart.
- To be eligible for ARM processing, elements (Jobs/STCs) must be registered with ARM. This is achieved through the IXCARM macro. Some subsystems come with this support built in. For example, CICS registers with ARM at startup time. For products that do *not* register with ARM, there is a program available, called ARMWRAP, that can be inserted into an existing job or STC, and this can be used to do the ARM registration. Your IBM representative can obtain the ARMWRAP package for you from the MKTTOOLS disk in IBM.
- A registered element that terminates unexpectedly is restarted on the same system.
- Registered elements on a system that fails are restarted on another system. Related elements are restarted on the same system (for example, DB2 and its corresponding IRLM address space).

- The exploiters of the ARM function are the jobs and STCs of certain strategic transaction and resource managers, such as the following:
  - CICS
  - CP/SM
  - DB2
  - IMS TM
  - IMS/DBCTL
  - ACF/VTAM
  - TCP/IP
  - Tivoli NetView® for z/OS
  - IRLM

  These products already have the capability to exploit ARM. When they detect that ARM has been enabled, they register an element with ARM to request a restart if a failure occurs.

  There are three exit points in ARM where installation written programs can cancel a restart or do other things to extend control and enhance function.

ARM and subsystems

When a subsystem such as CICS, IMS, or DB2 fails in a Parallel Sysplex, it impacts other instances of the subsystem in the sysplex due to such things as retained locks and so on. It is therefore necessary to restart these subsystems as soon as possible after failure to enable recovery actions to be started, and thus keep disruption across the sysplex to a minimum.

Using ARM to provide the restart mechanism ensures that the subsystem is restarted in a pre-planned manner without waiting for human intervention. Thus, disruption due to retained locks or partly completed transactions is kept to a minimum.

We recommend that ARM be implemented to restart major subsystems in the event of failure of the subsystem or the system on which it was executing within the Parallel Sysplex.

Recommendation: *OS/390 Parallel Sysplex Recovery,* GA22-7286 contains information about:

- Hardware recovery
- CF recovery
- Subsystem recovery

Take a look at this document when planning for the recovery aspects of the Parallel Sysplex.

3.8 Disaster recovery (DR) considerations in Parallel Sysplex

Parallel Sysplex is an excellent framework for disaster recovery: Parallel Sysplex can be useful in a disaster recovery strategy for the following reasons:

- The elements of a Parallel Sysplex may be physically spread over up to 100 kilometers.
- Parallel Sysplex allows you to configure all elements redundantly, thus reducing the risk that a disaster could render your entire Parallel Sysplex inoperable.
- In addition to traditional disaster recovery configurations based on either cold or hot standby CPCs, IBM eServer zSeries and System z9 CPCs offer CBU (Capacity Backup). Cold or hot standby processors will typically be configured as a one-way processor with a number of CBU processors (and on z9 this could include CBU ICFs).
- Parallel Sysplex maximizes the benefits of DASD remote copy. Following a complete failure of one site, the remaining members of the Parallel Sysplex in the alternate site can continue processing using the remote copies of the data from the failed site.

At a SHARE session in Anaheim, California, the Automated Remote Site Recovery Task Force presented a scheme consisting of six tiers of recoverability from disaster scenarios. These are as follows:

- Tier 0 - No DR plan

  No DR plan: All data is lost and recovery is not possible

- Tier 1 - Pickup Truck Access Method

  Pickup Truck Access Method (PTAM): The system, the subsystem, and the application infrastructure along with application data is dumped to tape and transported to a secure facility. All backup data, such as image copies and archived logs, still on site are lost in the event of a disaster (typically up to 24 to 48 hours). DR involves securing a DR site, installing IT equipment, transporting backup tapes from the secure facility to the DR site, restoring the system, the subsystem, and application infrastructure along with data, and restarting the workload (typically more than 48 hours). Cost factors include creating the backup copy of data, backup tape transportation, and backup tape storage.

- Tier 2 - PTAM and hot site

  PTAM and hot site: Same as Tier 1, except the enterprise has secured a DR facility. Data loss is up to 24 to 48 hours and the recovery window will be 24 to 48 hours. Cost factors include owning a second IT facility or a DR facility subscription fee in addition to the Tier 1 cost factors.

- Tier 3 - Electronic vaulting

  Electronic vaulting: Same as Tier 2, except that the enterprise dumps the backup data to a remotely attached tape library subsystem. Data loss will be up to 24 hours or less (depending upon when the last backup was created) and the recovery window will be 24 hours or less. Cost factors include telecommunication lines to transmit the backup data and a dedicated tape library subsystem at the remote site in addition to the Tier 2 cost factors.

- Tier 4 - Active secondary site (electronic remote journaling)

  Active secondary site: Same as Tier 3, except that transaction manager and database management system updates are remotely journaled to the DR site. Data loss will be seconds and the recovery window will be 24 hours or less (the recovery window could be reduced to two hours or less if updates are continuously applied to a shadow secondary database image). Cost factors include a system to receive the updates and disk to store the updates in addition to the Tier 3 cost factors.

- Tier 5 - Two-site two-phase commit

  Two-site two-phase commit: Same as Tier 4, with the applications performing two-phase commit processing between two sites. Data loss will be seconds and the recovery window will be two hours or less. Cost factors include maintaining the application in addition to the Tier 4 cost factors. Performance at the primary site can be affected by performance at the secondary site

- Tier 6 - Zero data loss (remote copy)

  Zero Data Loss (remote copy): The system, the subsystem, and application infrastructure along with application data is continuously mirrored from the production site to a DR site. Theoretically, there is no data loss if using a synchronous remote copy, and only seconds worth of changes is lost if using an asynchronous remote copy. The recovery window is the time required to restart the environment using the secondary disks if they are data consistent (typically less than two hours). The synchronous solution conceptually allows you to reach zero data loss, but performance may be impacted and care must be taken when considering the rolling disaster, which will leave inconsistent data at the secondary site.

Tier 7 has been added since to described DR solutions combining advanced data replication technologies and automation.

- Tier 7 **-** Geographically Dispersed Parallel Sysplex

  GDPS is beyond the SHARE-defined DR tiers, as it provides total IT business recovery through the management of processors, systems, and storage resources across multiple sites. GDPS manages not just the physical resources, but also the application environment and the consistency of the data, providing full data integrity (across volumes, subsystems, operating system platforms, and sites), while providing the ability to perform a normal restart in the event of a site switch, thus keeping to a minimum the duration of the recovery window.

  Additionally, the GDPS/PPRC and GDPS/PPRC HyperSwap™ Manager offerings provide the HyperSwap function, which enables a nondisruptive switch to the secondary copy of the disks. The GDPS offerings are discussed further later in this redbook.

Many installations are currently Tier 2, that is, backups are kept offsite with hardware installed for recovery in the event of a disaster. Numerous installations are now looking at Tier 6 and Tier 7, that is, a backup site is maintained and there is minimal-to-zero data loss.

3.8.1 Multi-site sysplexes

Many installations maintain two or more separate sites, with each site providing disaster recovery capability to the others. Typically, CPCs will be configured with capacity back-up (CBU). A presentation and audio file about this topic, called *Second Data Center Considerations*, is available on the Web at:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1063

In fact, there are many presentations available at this Web site that may be relevant and they were produced in September 2004. The presentation covers both cross site sysplex and recovery to a second site using various remote copy techniques, with the caution that what is technically supported may not be doable in practice, though most questions have an *it depends* answer.

The presentation reports that a client was doing 60,000 lock requests/sec at 5 km with 150 microsecond average response time. The CF subchannel is busy for the whole of the asynchronous request response time. This means that per second, 60,0000*150 microseconds =9 seconds/second of subchannel busy were absorbed. To maintain

subchannel utilization below 30% busy would require 9/(7*.3)=4.2 coupling links (since each link has seven subchannels). If the second site were 20 km away, then response time would approximately double and about twice as many links would be needed. In fact, in this case, there were eight CPCs involved with only one production image per processor, so that in fact two links per CPC would have been adequate. When considerations of diverse routing over two different routes are taken into account, the numbers of links required increases. Clearly, the distance effect needs careful consideration.

For example, what would happen in a two site sysplex 100 km apart is technically feasible. Some things, such as XCF and GRS, are sysplex wide. At 100 km, the propagation delay at 10 microseconds/km becomes 1 millisecond and this could affect performance even in sysplexes, whose primary purpose is to save software costs. We examine this further in the next subsection.

The effects on the CF and on path utilization when duplexing is used are much more severe. We expect the only practical use of a Parallel Sysplex at 100 km will be for GDPS (see 3.9, “GDPS: The e-business availability solution” on page 139.

Chapter 12, “Extended Distance Solutions”, in *IBM System z9 and @server zSeries Connectivity Handbook*, SG24-5444 provides a good introduction to the extension of the various zSeries and IBM System z9 Fibre Channels. Section 12.1 contains a table giving maximum unrepeated fiber link distances. Single mode fiber supports much longer unrepeated distances than multi-mode fiber.

ESCON and ETR are limited to 2 km (multimode 62.5 micrometers) or 3 km (multimode 50 micrometers) without repeaters. Note that all ESCON directors (ESCDs) are now WDFM and the 9036-003, which extends the CPU to timer distance, is also WDFM.

In summary for longer distances, IBM eServer zSeries and System z9 fiber connections (ESCON, FICON, CF, and Gbit Ethernet) can be extended to 103 km (with RPQ 8P2263) with appropriate extender technology, such as DWDM and CWDM (Dense and Coarse Wave Division Multiplexers). DASD using FCP PPRC can be up to 300 km apart and unlimited distance is possible with XRC. The timer to timer connection of 9037 cannot extend beyond 40 km (because loss of synchronization will occur). With STP, the Primary Time server (Stratum 1 server) can use CF links up to 100 km to Stratum 2 processors.

ESCON suffers significant degradation beyond 9 km, but in any case, these days we expect that DASD and tape will use FICON, which can be used at 100 km successfully.

Console connectivity issues in a multi-site sysplex

The question is how to provide MVS consoles cross-site to other processors (possibly over large distances). 3174s are no longer available and in any case would not facilitate cross site consoles. There are three possibilities:

- The 2074-003 is still available and connects via ESCON (either directly or through an ESCD), and it can have one or two ESCON adapters. The consoles themselves are LAN attached PCs (with PCOM). This means that the consoles could be in the other site yet effectively connect locally to a processor in the first site and can be used as NIP consoles.
- The OSA-Integrated Console Controller (ICC) available on z990, z890 and z9. This applies to 1000 base T OSA ports defined as ICC through HCD. As before, the consoles are network attached PCs.
- SNA Consoles are real MCS consoles that can be brought up on any PC that can access VTAM services. They are only accessible after VTAM comes up and so are not suitable for NIP consoles.

Distance effects on CF in a multi-site sysplex

The sync-async conversion algorithm described in 2.4.1, “Synchronous and asynchronous CF requests” on page 53 limits the CPU cost of distance on all CF requests. However, it can do nothing about the elongation of response time due to distance, and this happens to both synchronous and asynchronous requests. At 10 km, the propagation delay is about 100 microseconds. Looking at user response time, if a transaction requires, say, 10 synchronous operations to a local CF, we might expect response to increase by 1000 microseconds (that is, 1 millisecond) if the CF is moved 10 km away. The user will probably not notice this. However, a batch job might do millions of CF operations and a million operations will increase elapsed time by 100 seconds (and more as distance increases at 100 km by 1000 seconds).

If CF duplexing were used cross-site, then, as shown in 2.6.2, “What is System-Managed CF structure duplexing?” on page 68, then any duplexed request requires up to six sequential cross site communications: processor to CF, CF-CF before CF executes request, CF-CF after CF executes request, and CF-processor. So at 10 km, duplexing will add approximately 600 microseconds to each duplexed request.

This discussion was not intended to analyze in depth the consequences of distance on a two site sysplex, only to emphasize that careful planning will be required for multi-site sysplexes actively exploiting sysplex capabilities.

The sample multi-site in Figure 3-4 represents a typical non-GDPS multi-site sysplex that could facilitate a number of recovery scenarios.

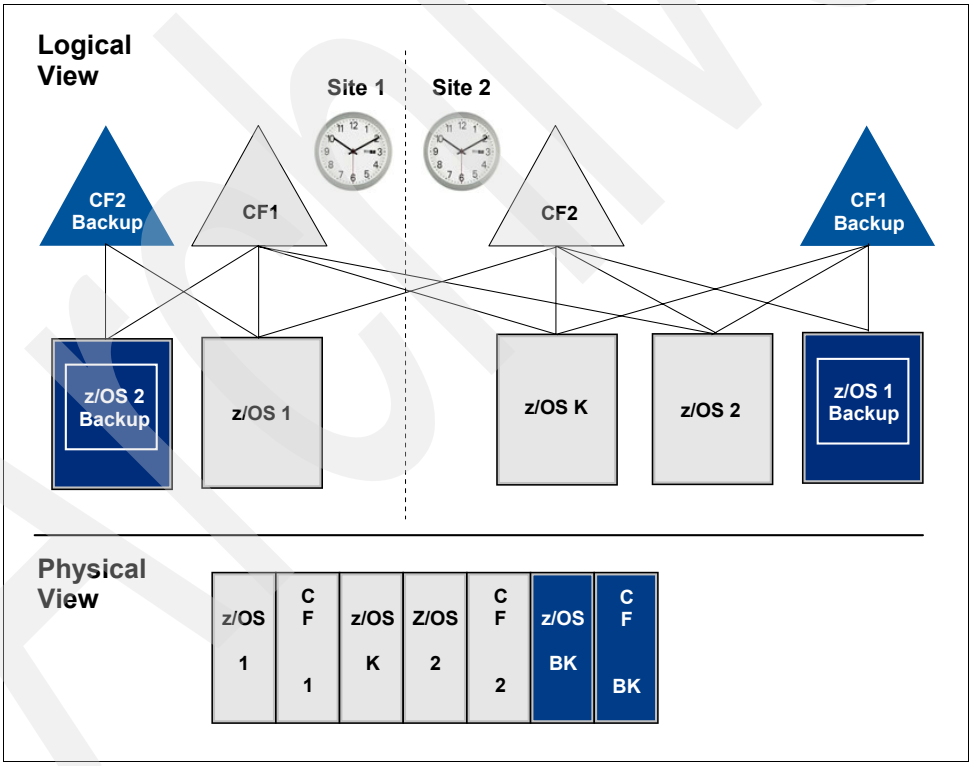


Figure 3-4 Sample multi-site sysplex configuration

- Site 1 and Site 2 can operate as a single Parallel Sysplex. However, the distance between the sites may dictate that spreading production workload across the complete Parallel Sysplex is not viable, due to the response time degradation caused by the distance. Assume then that Site 1 performs high priority work in its own subplex, and Site 2 performs low priority work in its subplex.
- In the event of a loss of a CPC in Site 1, it would be possible for the workload associated with it to be taken over by LP X, for example, until the failed CPC is restored.
- As Site 2 performs low priority work, it has no backup CF. In the event of a failure, therefore, spare CF capacity could be used in Site 1.
- Site 2 can also act as a disaster recovery site for Site 1. By using remote copy techniques, copies of key system data and databases can be maintained at Site 2 for fast recovery of production systems in the event of a complete loss of production at Site 1. As Site 1 is a data sharing environment, the recovery site is required to provide a similar environment.
- Each site can also provide additional capacity for the other for scheduled outages. For example, provided all structures support rebuild, the stand-alone CF in Site 2 can be upgraded without interrupting the applications. If remote copy is used, whole applications can be moved between sites with minimal interruption to applications using techniques, such as GDPS HyperSwap. This might be useful if a complete power down is scheduled in either site.

See 3.8.2, “Disaster recovery data” on page 134 for a discussion of remote copy techniques, and the remaining sections of this chapter for specific subsystem disaster recovery considerations.

3.8.2 Disaster recovery data

A key element of any disaster recovery solution is having critical data available at the recovery site as soon as possible and as up-to-date as possible. The simplest method of achieving this is with offsite backups of such data. However, the currency of this data is dependent on when the outage happened relative to the last backup.

A number of options are available that allow for the electronic transfer of data to a remote site. Two techniques, electronic remote journaling and remote DASD mirroring, are discussed further.

Electronic remote journaling

Electronic remote journaling requires an active CPC at the remote site, with appropriate DASD and tape subsystems. The transaction manager and database manager updates are transmitted to the remote site and journaled. Image copies of databases and copies of the subsystem infrastructure are required at the recovery site. In the event of an outage at the primary site, the subsystem infrastructure is restored, the journaled data is reformatted to the subsystem log format, and recovery of the subsystem (TM/DB) is initiated. The only *data lost* is that in the remote journal buffers, which was not hardened to DASD, and that which was being transmitted at the time of the outage. This is likely to amount to only a few seconds worth of data. The quantity of data lost is proportional to the distance between sites.

Electronic remote journaling can be implemented using the Remote Site Recovery feature of IMS, or the Tracker Site feature of DB2. Another option is the Remote Recovery Data Facility (RRDF) product supplied by E-Net Corporation. More information about RRDF can be found on the Web at:

http://www-304.ibm.com/jct09002c/gsdod/solutiondetails.do?solution=15871&expand=true&lc=en

Remote DASD mirroring

IBM currently provides two options for maintaining remote copies of data. Both address the problem of data made out-of-date by the time interval between the last safe backup and the time of failure. These options are:

- Peer-to-peer remote copy (PPRC)
- Extended remote copy (XRC)

Peer-to-Peer remote copy

PPRC provides a mechanism for *synchronous* copying of data to the remote site, which means that no data is lost in the time between the last backup at the application system and the recovery at the remote site. The impact on performance must be evaluated, since an application write to the primary subsystem is not considered complete until the data has been transferred to the remote subsystem.

Figure 3-5 shows a sample peer-to-peer remote copy configuration.

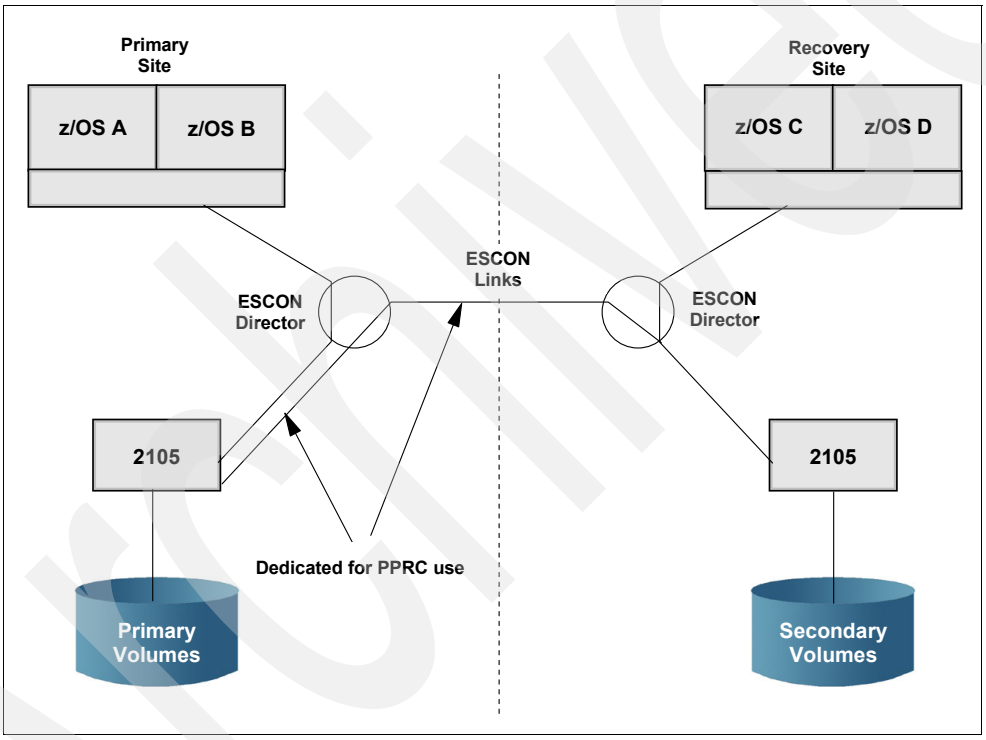


Figure 3-5 Peer-to-Peer remote copy configuration

The peer-to-peer remote copy implementation requires ESCON links between the primary site DASD controller and the remote (recovery) site DASD controller. These links are provided via:

- Direct ESCON connection between controllers
- Multimode ESCON director connection
- XDF ESCON director connection
- 9036 ESCON extender connection

The mode of connection determines the distance limit of the secondary controller. Up to 103 km is available with the Enterprise Storage Server. At the time of writing, Enterprise Storage Server support for PPRC has been available since 2000.

Peer-to-peer dynamic address switching (P/DAS) provides the mechanism to allow switching from the primary to the secondary controller.

Extended remote copy

XRC provides a mechanism for *asynchronous* copying of data to the remote site. Only data that is in transit between the failed application system and the recovery site is lost in the event of a failure at the primary site. Note that the delay in transmitting the data from the primary subsystem to the recovery subsystem is usually measured in seconds.

Figure 3-6 shows a sample extended remote copy configuration.

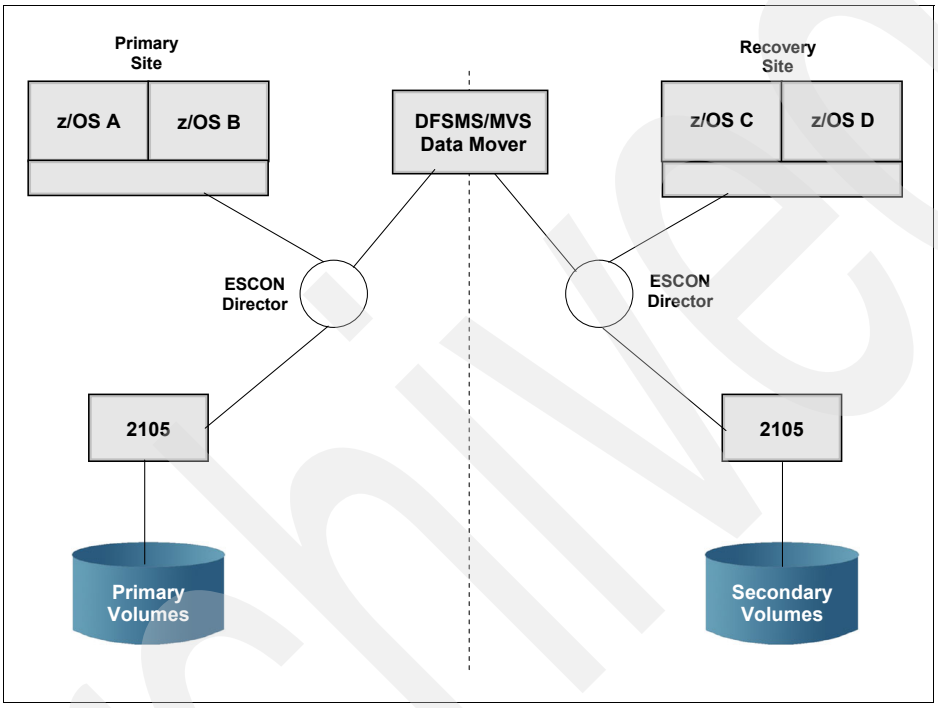


Figure 3-6 Extended remote copy configuration

The extended remote copy implementation involves the transfer of data between the primary subsystem and the recovery subsystem. The recovery subsystem must be attached to another z/OS image running the required level of DFSMS and can exist at the primary site, at the recovery site, or anywhere in between. Figure 3-6 shows an independently sited data mover at ESCON distance limits. However, XRC sites can be separated by distances greater than those supported by ESCON, with the use of channel extenders and high speed data links.

Mixing Remote Copy Solutions: Each remote copy solution uniquely addresses data sequence consistency for the secondary volumes. Combining the techniques may lead to unpredictable results, exposing you to data integrity problems. This situation applies to PPRC, XRC, and specific database solutions, such as IMS RSR.

The DASD Remote Copy solutions are data-independent and semi. That is, beyond the performance considerations, there is no restriction on the data that is mirrored at a remote site using these solutions.

A more complete description of PPRC and XRC functions, configuration, and implementation can be found in the following IBM Redbooks:

- *Planning for IBM Remote Copy*, SG24-2595
- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries*, SG24-5680

3.8.3 DRXRC: Disaster recovery and system logger

z/OS 1.7 introduces a new option to enable the DASD write to the staging datasets to be asynchronous (in fact, a logger buffers the writes and actually writes them out when the buffer is full or after prescribed limit). The purpose of this is to minimize the performance and response time overheads of synchronous mirroring of the staging datasets (and the basic overhead of the synchronous write to the staging dataset). The logstream is defined with a new option DRXRC, which indicates that the staging dataset is for disaster recovery purposes and that asynchronous writes to it should be used. A new XRC keyword, LOGPLUS, is used to add a DRXRC staging dataset volume to an XRC session. This indicates that system logger is responsible for time stamps rather than storage control.

In the event of a disaster, the first system in the secondary site is piled with a new IEASYSxx parameter DRMODE=YES, which indicates that DRXRC datasets are to be included in the log data recovery for coupling facility structure based log streams that were connected before IPL. This is discussed further in a level 9 document dated August 2005 called “Doctor, Doctor Give me the News” in *z/OS V1R7.0 Hot Topics Newsletter,* GA22-7501, and in *z/OS MVS Setting Up a Sysplex*, SA22-7625.

3.8.4 CICS disaster recovery considerations

The way in which recovery data is provided to the secondary site determines what is required in terms of CICS infrastructure copies. For example, if recovery data is provided from offsite backup data, backup copies of VSAM files and a copy of the CICS infrastructure are required. If full remote DASD mirroring is in place, CICS journaling must be duplexed to the CF and LOGR staging data sets; the CICS/VSAM RLS data sharing group must be identical to that of the production site. VSAM files, system logger infrastructure, CICS/VSAM RLS infrastructure, and CICS infrastructure must also be copied to the secondary site.

For a full discussion on the options available for CICS disaster recovery, refer to the IBM Redbook *Planning for CICS Continuous Availability in a MVS/ESA Environment*, SG24-4593.

3.8.5 DB2 disaster recovery considerations

Be aware that if you implement a DB2 data sharing group, your disaster recovery site *must* be able to support the same DB2 data sharing group configuration as your main site. It must have the same group name, the same number of members, and the names of the members must be the same. Additionally, the structure names in the CFRM policy must be the same (although the sizes can be different).

The hardware configuration, however, *can* be different. For example, your main site could be a multisystem data sharing group spread among several CPCs, with CFs and Sysplex Timers. Your disaster recovery site could be a large single z/OS image, which could run *all* of the DB2 subsystems in the data sharing group.

Since some of the benefits of the Parallel Sysplex are lost by bringing the subsystems under one z/OS, all but one of the members could be stopped once the data sharing group has been started on the recovery site.

Figure 3-7 gives some example configurations for DB2 data sharing groups.

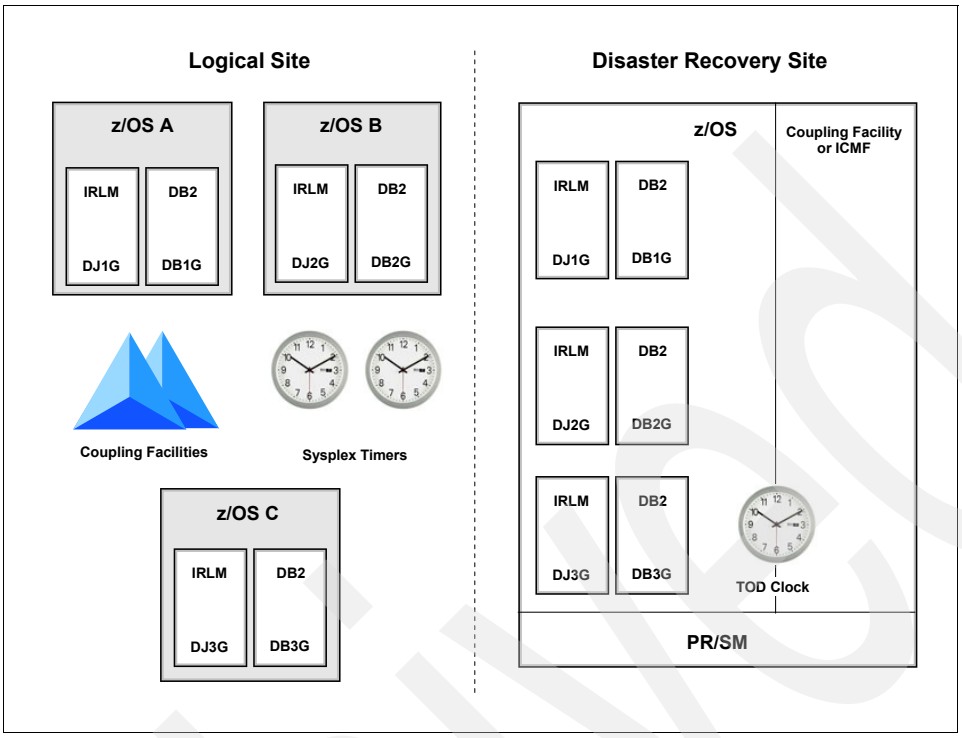


Figure 3-7 Example configurations for DB2 data sharing groups

IBM provides a broad range of functions to be used for disaster recovery of DB2 subsystems, like remote copy services, flashcopy, and tracker site, which are used by the following disaster recovery solutions:

- Split Mirror
- FlashCopy Consistency Group
- Metro Mirror
- Global Mirror for z/OS (XRC)
- Global Mirror
- Geographically Dispersed Parallel Sysplex

More detailed information about disaster recovery solutions can be found in:

- *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370
- *DB2 UDB for z/OS V8 Data Sharing: Planning and Administration,* SC18-7417

3.8.6 IMS disaster recovery considerations

IMS was the first IBM S/390 program product to exploit *hot standby*. Using the extended recovery facility (XRF), one IMS system shadows another and takes over within seconds if the active system fails. With XRF, a number of IMS TM installations worldwide have achieved continuous service availability for more than 1000 days.

XRF is not generally considered a disaster recovery function. When data sharing is used, an IMS XRF active and its alternate must be in the same Parallel Sysplex. XRF is a recovery function for failures of IMS, z/OS, and CPCs.

With the introduction of IMS 5.1, the concept of a standby system was extended from a local perspective to remote site recovery (RSR), which provides system-managed transfer and

application of log data to shadow databases at the remote site. A key feature of RSR is that the remote databases are assured of consistency and integrity. In the event of a site failure, the remote system can be restarted as the production system within minutes.

RSR offers two levels of support, selected on an individual database basis:

- Database Level Tracking

  With this level of support, the database is shadowed at the remote site, thus eliminating database recovery in the event of a primary site outage.

- Recovery Level Tracking

  No database shadowing is provided with this option. The logs are transmitted electronically to the remote site and are used to recover the databases by applying forward recovery to image copies in the event of an outage at the primary site.

RSR supports the recovery of IMS full-function databases, Fast Path DEDBs, IMS message queues, and the telecommunications network. Some of the facilities introduced in IMS/ESA V6 do have an impact on XRF and RSR usage.

Note: The following are important points to note:

- IMS XRF systems cannot participate as members of a generic resource group. However, XRF systems and generic resource members can connect to the same shared message queue group.
- IMS RSR systems cannot participate as members of a generic resource group. However, RSR systems and generic resource members can connect to the same shared message queues group.
- Subsystems that use XRF cannot use Fast DB Recovery.
- A DBCTL system with an alternate standby configuration cannot use Fast DB Recovery.

Additional details can be found in *IMS/ESA V6 Administration Guide: System,* SC26-8730, and *IMS/ESA V5 Administration Guide: System,* SC26-8013.

As with DB2, if you have implemented data sharing at the primary site, then the remote recovery site must also be a Parallel Sysplex environment, and the IMS data sharing group must be identical.

3.9 GDPS: The e-business availability solution

In e-business, two important objectives for survival are systems designed to provide continuous availability (CA) and near transparent disaster recovery (DR). Systems that are designed to deliver continuous availability combine the characteristics of high availability and near continuous operations to deliver high levels of service 24x7.

High availability is an attribute of a system that provides service at agreed upon levels and can mask unplanned outages from users.

Near continuous operations, on the other hand, is the attribute of a system designed to continuously operate and mask planned outages from users. To attain high levels of continuous availability and near-transparent DR, the solution should be based on geographical clusters and data mirroring.

The GDPS solution, based on Peer-to-Peer Remote Copy (PPRC, recently renamed to IBM TotalStorage® Metro Mirror), is referred to as GDPS/PPRC, the GDPS solution based on Extended Remote Copy (XRC, recently renamed to IBM TotalStorage z/OS Global Mirror), is referred to as GDPS/XRC, and the GDPS solution based on IBM TotalStorage z/OS Global Mirror, is referred to as GDPS/GM.

GDPS/PPRC and GDPS/HM are designed with the attributes of a continuous availability and disaster recovery solution. On the other hand, GDPS/XRC and GDPS/GM have the attributes of a Disaster Recovery solution.

In GDPS/PPRC, since IBM Parallel Sysplex clustering technology is designed to enable resource sharing and dynamic workload balancing, enterprises can now dynamically manage workloads across multiple sites, which can enable them to achieve high levels of availability.

With the introduction of GDPS/PPRC HyperSwap Manager, described later, Parallel Sysplex availability can now be extended to disk subsystems, even if multiple sites are not available and the Parallel Sysplex is configured in one site.

GDPS/PPRC complements a multisite Parallel Sysplex implementation by providing a single, automated solution to dynamically manage storage subsystem mirroring (disk and tape) and processor resources designed to help a business to attain *continuous availability* and *near transparent business continuity (disaster recovery)* with no or minimal data loss.

In GDPS/XRC, the production system(s) located in a production site can be a single system, multiple systems sharing disk, or a base or Parallel Sysplex cluster. GDPS/XRC provides a single, automated solution, designed to dynamically manage storage subsystem mirroring (disk and tape) to allow a business to attain *near transparent* disaster recovery with minimal data loss.

GDPS/XRC is designed to provide the ability to perform a controlled site switch for an unplanned site outage, maintaining data integrity across multiple volumes and storage subsystems and the ability to perform a normal Data Base Management System (DBMS) restart – not DBMS recovery – in the recovery site.

Refer to 3.9.2, “Need for data consistency” on page 141 for details on how the SDM provides data update sequence consistency for all volumes participating in the XRC session.

3.9.1 What is GDPS?

GDPS is an integrated, automated application and data availability solution designed to provide the capability to manage the remote copy configuration and storage subsystems, automate Parallel Sysplex operational tasks, and perform failure recovery from a single point of control, thereby helping to improve application availability.

GDPS is a combination of system code and automation that utilizes the capabilities of Parallel Sysplex technology and storage subsystem mirroring to manage processors, and storage subsystems. GDPS is independent of the transaction manager (for example, CICS TS, IMS, and WebSphere) or database manager (for example, DB2, IMS, and VSAM) being used, and is enabled by means of key IBM technologies and architectures:

- Base or Parallel Sysplex
- Tivoli NetView for z/OS
- System Automation for z/OS
- IBM TotalStorage DS6000™ and DS8000™ series and Enterprise Storage Server® (ESS)
- Peer-to-Peer Virtual Tape Server (PtP VTS)
- Optical Dense or Coarse Wavelength Division Multiplexer (DWDM or CWDM)
- Metro Mirror architecture for GDPS/PPRC

- z/OS Global Mirror architecture for GDPS/XRC
- Global Mirror architecture for GDPS/GM
- Virtual Tape Server Remote Copy architecture

These technologies are the backbone of the GDPS solution.

The GDPS solution is a nonproprietary solution, working with IBM as well as Other Equipment Manufacturer (OEM) disk vendors, as long as the vendor meets the specific functions of the Metro Mirror and z/OS Global Mirror architectures required to support GDPS functions as documented in 3.9.14, “Prerequisites” on page 154.

3.9.2 Need for data consistency

Data consistency across all primary and secondary volumes spread across any number of storage subsystems is essential to providing data integrity and the ability to do a normal database restart in the event of a disaster. The main focus of GDPS automation is whatever happens to the primary site (site 1), to allow the secondary copy of the data in the secondary site (site 2) to be data consistent (the primary copy of data will be data consistent for any site 2 failure). Data consistent means that, from an application’s perspective, the secondary disks contain all updates until a specific point in time, and no updates beyond that specific point in time.

Time consistent data in the secondary site allow applications to restart in the secondary location without having to go through a lengthy and time-consuming data recovery process. Data recovery involves restoring image copies and logs to disk and executing forward recovery utilities to apply updates to the image copies. This process can take many hours. Since applications only need to be restarted, an installation can be up and running quickly, even when the primary site (site 1) has been rendered totally unusable.

GDPS/PPRC uses a combination of storage subsystem and Parallel Sysplex technology triggers to capture, at the first indication of a potential disaster, a data consistent secondary site (site 2) copy of the data, using the PPRC freeze function. The freeze function, initiated by automated procedures, is designed to freeze the image of the secondary data at the very first sign of a disaster, even before any database managers are made aware of I/O errors. This can prevent the logical contamination of the secondary copy of data that would occur if any storage subsystem mirroring were to continue after a failure that prevents some, but not all, secondary volumes from being updated.

Data consistency in a GDPS/XRC or GDPS/GM environment is provided by the Consistency Group (CG) processing performed by the System Data Mover (SDM) for XRC, or by the Global Mirror microcode in the disk subsystems. The CG ensures that the data records have their order of updates preserved across multiple Logical Control Units within a storage subsystem and across multiple storage subsystems.

Providing data consistency enables the secondary copy of data to perform normal restarts (instead of performing database manager recovery actions). This is the essential design element of GDPS in helping to minimize the time to recover the critical workload, in the event of a disaster in site 1.

3.9.3 GDPS systems

GDPS consists of production systems and controlling systems. The production systems execute the mission critical workload. There must be sufficient processing resource capacity (typically in site 2), such as processor capacity, main storage, and channel paths available that can quickly be brought online to restart a system's or site's critical workload (typically by

terminating one or more systems executing expendable [non-critical] work and acquiring its processing resource).

The Capacity BackUp (CBU) feature, available on IBM eServer zSeries and z9 servers, could provide additional processing power, which can help you to achieve cost savings. The CBU feature has the ability to increment capacity temporarily when capacity is lost elsewhere in the enterprise. CBU adds Central Processors (CPs) to the available pool of processors and is activated only in an emergency. GDPS-CBU management automates the process of dynamically adding reserved Central Processors (CPs), thereby helping to minimize manual client intervention and the potential for errors. The outage time for critical workloads can potentially be reduced from hours to minutes. Similarly, GDPS-CBU management can also automate the process of dynamically returning the reserved CPs when the temporary period has expired.

The controlling system coordinates GDPS processing. By convention, all GDPS functions are initiated and coordinated by the controlling system.

All GDPS systems run GDPS automation based upon Tivoli NetView for z/OS and Tivoli System Automation for z/OS. Each system can monitor the sysplex cluster, Coupling Facilities, and storage subsystems and maintain GDPS status. GDPS automation can coexist with an enterprise's existing automation product.

3.9.4 GDPS/PPRC

GDPS/PPRC is designed to manage and protect IT services by handling planned and unplanned exception conditions, and maintain data integrity across multiple volumes and storage subsystems. By managing both planned and unplanned exception conditions, GDPS/PPRC can help to maximize application availability and provide business continuity.

GDPS/PPRC is capable of the following attributes:

- Near continuous Availability solution
- Near transparent D/R solution
- Recovery Time Objective (RTO) less than an hour
- Recovery Point Objective (RPO) of zero (optional)
- Protects against localized area disasters (distance between sites limited to 100 km fiber)

Topology

The physical topology of a GDPS/PPRC, (see Figure 3-8 on page 143), consists of a base or Parallel Sysplex cluster spread across two sites, known as site 1 and site 2, separated by up to 100 kilometers (km) of fiber – approximately 62 miles – with one or more z/OS systems at each site.

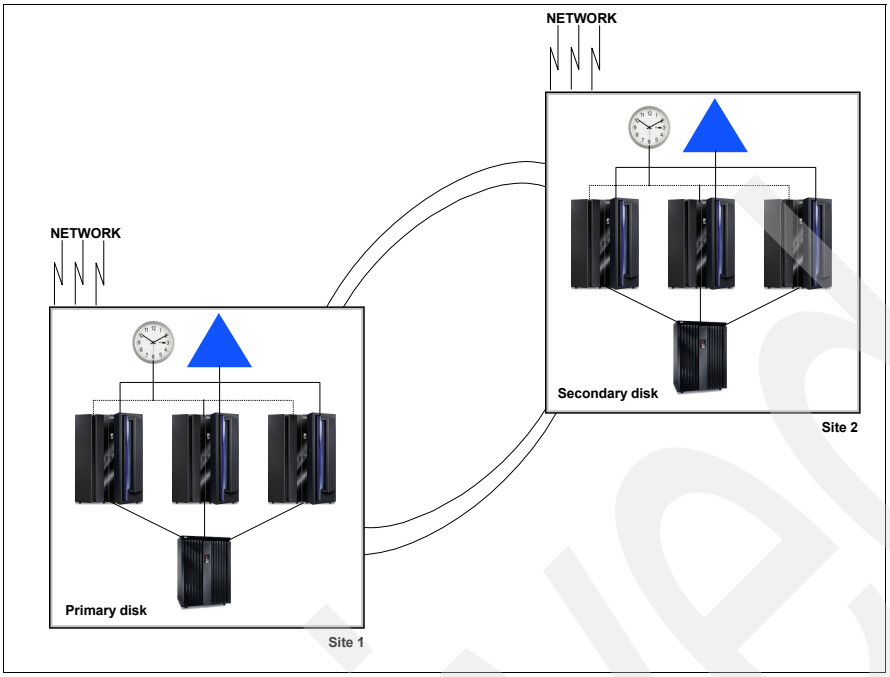


Figure 3-8 Example of GDPS/PPRC configuration

The multi-site sysplex cluster must be configured with redundant hardware (for example, a Coupling Facility and a Sysplex Timer in each site) and the cross-site connections must be redundant. All critical data resides on storage subsystems in site 1 (the primary copy of data) and is mirrored to the storage subsystems in site 2 (the secondary copy of data) via PPRC synchronous remote copy.

Clients have the capability to configure GDPS/PPRC with up to 100 km of fiber between two sites. An immediate advantage of this extended distance is to potentially decrease the risk that the same disaster will affect both sites, thus permitting clients to recover their production applications at another site.

GDPS/PPRC supports Metro Mirror over Fibre Channel Protocol (FCP). Since Metro Mirror over FCP requires only one protocol exchange compared to two or three exchanges when using Metro Mirror over ESCON, it is expected that the distance between sites can be increased while maintaining acceptable application performance. The efficiency of the FCP protocol is also expected to lower the total cost of ownership, since two Metro Mirror FCP links between each pair of ESS disk subsystems are considered sufficient for most workloads, allowing a reduction in cross-site connectivity.

3.9.5 Near continuous availability of data with HyperSwap

Exclusive to GDPS in the PPRC environment is HyperSwap. This function is designed to broaden the near continuous availability attributes of GDPS/PPRC by extending the Parallel Sysplex redundancy to disk subsystems. The HyperSwap function can help significantly reduce the time needed to switch to the secondary set of disks while keeping the z/OS systems active, together with their applications.

With the release of GDPS/PPRC V3.2, the HyperSwap function has been enhanced to exploit the Metro Mirror Failover/Failback (FO/FB) function. For planned reconfigurations, FO/FB may reduce the overall elapsed time to switch the disk subsystems, thereby reducing the time that applications may be unavailable to users. For unplanned reconfigurations, Failover/Failback

allows the secondary disks to be configured in the suspended state after the switch and record any updates made to the data. When the failure condition has been repaired, resynchronizing back to the original primary disks requires only the changed data to be copied, thus eliminating the need to perform a full copy of the data. The window during which critical data is left without Metro Mirror protection following an unplanned reconfiguration is thereby minimized.

3.9.6 Planned reconfiguration support

GDPS/PPRC planned reconfiguration support automates procedures performed by an operations center. These include standard actions to:

1. Quiesce a system's workload and remove the system from the Parallel Sysplex cluster (for example, stop the system prior to a hardware change window).
2. IPL a system (for example, start the system after a hardware change window).
3. Quiesce a system's workload, remove the system from the Parallel Sysplex cluster, and re-IPL the system (for example, recycle a system to pick up software maintenance).

Standard actions can be initiated against a single system or a group of systems. With the introduction of HyperSwap, you now have the ability to perform disk maintenance and planned site maintenance without requiring applications to be quiesced. Additionally, GDPS/PPRC provides customizable scripting capability for user defined actions (for example, planned disk maintenance or planned site switch, in which the workload is switched from processors in site 1 to processors in site 2.

All GDPS functions can be performed from a single point of control, which can help simplify system resource management. Panels are used to manage the entire remote copy configuration, rather than individual remote copy pairs. This includes the initialization and monitoring of the remote copy volume pairs based upon policy and performing routine operations on installed storage subsystems (disk and tape). GDPS can also perform standard operational tasks, and monitor systems in the event of unplanned outages.

The Planned HyperSwap function is designed to provide the ability to transparently switch all primary disk subsystems with the secondary disk subsystems for planned reconfigurations. During a planned reconfiguration, HyperSwap can provide the ability to perform disk configuration maintenance and planned site maintenance without requiring any applications to be quiesced. Large configurations can be supported, as HyperSwap is designed to provide capacity and capability to swap large number of disk devices very quickly. The important ability to re-synchronize incremental disk data changes, in both directions, between primary /secondary disks is provided as part of this function.

Benchmark measurements using HyperSwap for planned reconfiguration

Planned disk reconfiguration conducted at the GDPS solution center with 2900 volumes in the PPRC config demonstrated that the user impact time dropped from 93 seconds without Failover/Failback down to 18 seconds with Failover/Failback.

3.9.7 Unplanned reconfiguration support

GDPS/PPRC unplanned reconfiguration support not only can automate procedures to handle site failures, but can also help minimize the impact and potentially mask a z/OS system, processor, Coupling Facility, disk, or tape failure, based upon GDPS/PPRC policy. If a z/OS system fails, the failed system and workload can be automatically restarted. If a processor fails, the failed systems and their workload can be restarted on other processors.

The Unplanned HyperSwap function is designed to transparently switch to use secondary disk subsystems that contain mirrored data consistent with the primary data in the event of unplanned outages of the primary disk subsystems or a failure of the site containing the primary disk subsystems (site 1).

With Unplanned HyperSwap support:

- Production systems can remain active during a disk subsystem failure. Disk subsystem failures will no longer constitute a single point of failure for an entire sysplex.
- Production systems can remain active during a failure of the site containing the primary disk subsystems (site 1), if applications are cloned and exploiting data sharing across the two sites. Even though the workload in site 2 will need to be restarted, an improvement in the Recovery Time Objective (RTO) is accomplished.

Benchmark measurements using HyperSwap for unplanned reconfiguration

An unplanned disk reconfiguration test using HyperSwap with Failover/Failback, conducted at the GDPS solution center, demonstrated that the user impact time was only 15 seconds to swap a configuration of 2900 volumes of ESS disks while keeping the applications available, compared to typical results of 30-60 minutes without HyperSwap.

What this benchmark does not show is the failover/failback capability to only copying the changed data instead of the entire disk during the resynchronization process. This can save significant time and network resources.

3.9.8 GDPS/PPRC HyperSwap manager

GDPS/PPRC HyperSwap Manager (GDPS/PPRC HM) expands zSeries Business Resiliency to clients by providing a single-site, near continuous availability solution as well as a multi-site entry-level disaster recovery solution.

Within a single site, GDPS/PPRC HyperSwap Manager extends Parallel Sysplex availability to disk subsystems by masking planned and unplanned disk outages caused by disk maintenance and disk failures. It also provides management of the data replication environment and automates switching between the two copies of the data without causing an application outage, therefore providing near-continuous access to data. Figure 3-9 on page 146 shows an example of a GDPS/PPRC HM configuration.

In the multi-site environment, GDPS/PPRC HyperSwap Manager provides an effective entry-level disaster recovery offering for those zSeries clients that have the need for very high levels of data availability. Value is further enhanced by being able to use specially priced Tivoli System Automation and NetView products. In addition, a client can migrate to the full function GDPS/PPRC capability across multiple sites as business requirements demand shorter Recovery Time Objectives provided by a second site. The initial investment in GDPS/PPRC HM is protected when clients choose to move to full-function GDPS/PPRC by leveraging the existing GDPS/PPRC HM implementation and skills.

GDPS/PPRC HM simplifies the control and management of the Metro Mirror (PPRC) environment for both z/OS and Open Systems data. This reduces storage management costs while reducing the time required for remote copy implementation.

GDPS/PPRC HM provides support for FlashCopy. GDPS/PPRC HM can be set up to automatically take a FlashCopy of the secondary disks before resynchronizing the primary and secondary disks following a Metro Mirror suspension event, ensuring a consistent set of disks are preserved should there be a disaster during the re-sync operation.

Figure 3-9 shows an example of a GDPS/HM configuration.

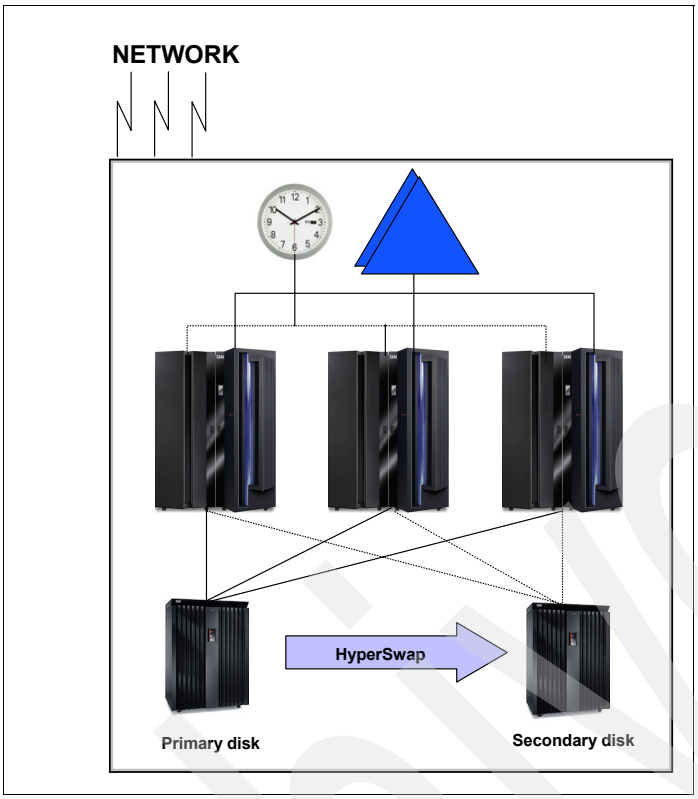


Figure 3-9 Example of GDPS/HM configuration

Near continuous availability of data within a single site

A Parallel Sysplex environment has been designed to reduce outages by replicating hardware, operating systems, and application components. In spite of this redundancy, having only one copy on the data is an exposure. GDPS/PPRC HyperSwap Manager is designed to provide continuous availability of data by masking disk outages caused by disk maintenance or failures. For example, if normal processing is suddenly interrupted when one of the disk subsystems experiences a hard failure, thanks to GDPS, the applications are masked from this error, because GDPS detects the failure and autonomically invokes HyperSwap. The production systems continue using data from the mirrored secondary volumes. Disk maintenance can also be similarly performed without application impact by executing the HyperSwap command.

Near CA of data or disaster recovery solution at metropolitan distances

In addition to the single site capabilities, in a two site configuration, GDPS/PPRC HyperSwapManager provides an entry-level disaster recovery capability at the recovery site. GDPS/PPRC HM uses the Freeze function described in 3.9.2, “Need for data consistency” on page 141. The Freeze function is designed to provide a consistent copy of data at the recovery site from which production applications can be restarted. The ability to simply restart applications helps eliminate the need for lengthy database recovery actions. Automation to stop and restart the operating system images available with the full-function GDPS/PPRC is not included with GDPS/PPRC HyperSwap Manager.

GDPS/PPRC HyperSwap Manager prerequisites

The GDPS/PPRC HyperSwap Manager can use, as prerequisites, IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager with NetView V1.1, which provides the System Automation and NetView requirements, or IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager, V1.1, together with the full-function IBM Tivoli NetView for z/OS product.

These new customized products, together with the GDPS/PPRC HM offering, are designed to bring new levels of affordability to clients who do not require the full-function products.

This makes GDPS/PPRC HM an excellent entry-level offering for those single-site or multisite installations that need higher levels of IT availability.

3.9.9 GDPS/XRC

Extended Remote Copy (XRC, recently renamed to IBM TotalStorage z/OS Global Mirror) is a combined hardware and z/OS software asynchronous remote copy solution. Consistency of the data is maintained via the Consistency Group function within the System Data Mover.

GDPS/XRC includes automation to manage remote copy pairs and automates the process of recovering the production environment with limited manual intervention, including invocation of CBU, thus providing significant value in reducing the duration of the recovery window and requiring less operator interaction.

GDPS/XRC is capable of the following attributes:

- Disaster recovery solution
- RTO between an hour to two hours
- RPO less than two minutes (typically 3-5 seconds)
- Protects against localized as well as regional disasters (distance between sites is unlimited)
- Minimal remote copy performance impact

GDPS/XRC topology

The physical topology of a GDPS/XRC (see Figure 3-10), consists of production systems in site 1. The production systems could be a single system, multiple systems sharing disk, or a base or Parallel Sysplex cluster5. Site 2 (the recovery site) can be located at a virtually unlimited distance from site 1 (the production site).

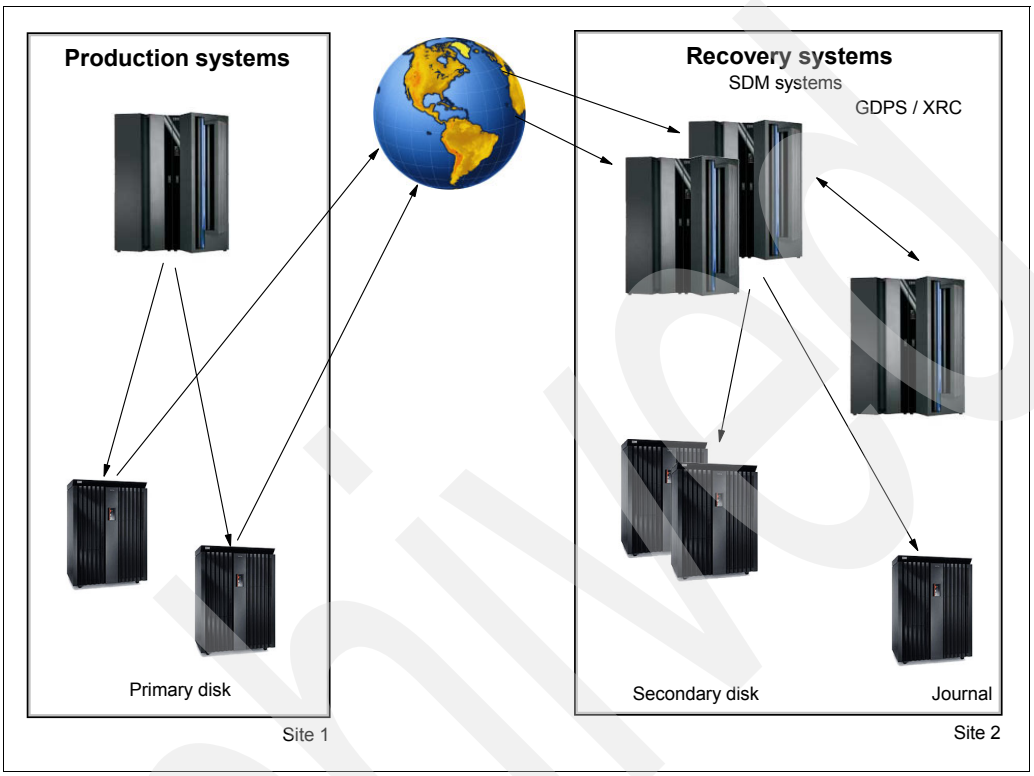


Figure 3-10 Example of GDPS/XRC configuration

During normal operations, the XRC System Data Mover (one or more) execute in site 2 and are in a Base Sysplex with the GDPS controlling system (refer to 3.9.3, “GDPS systems” on page 141 for a definition of the GDPS controlling system). All critical data resides on storage subsystem(s) in site 1 (the primary copy of data) and is mirrored to the storage subsystem(s) in site 2 (the secondary copy of data) via XRC asynchronous remote copy.

Planned reconfiguration support

All the planned reconfiguration actions described in 3.9.4, “GDPS/PPRC” on page 142 are provided by GDPS/XRC for the System Data Mover (SDM) Sysplex in site 2. For example, GDPS/XRC will manage the temporary relocation of the SDM, if it is needed. By managing the SDM Sysplex, GDPS/XRC can also manage the z/OS Metro Mirror remote copy configuration.

GDPS/XRC is designed to automate the process of recovering the production environment with minimal manual intervention, which can provide significant value in minimizing the duration of the recovery window.

3.9.10 Functional highlights (GDPS/PPRC and GDPS/XRC)

The following functions are supported by both GDPS/PPRC and GDPS/XRC.

Peer-to-Peer Virtual Tape Server (PtP VTS) support

GDPS also supports Peer-to-Peer Virtual Tape Server. By extending GDPS support to data resident on tape, the GDPS solution is intended to provide continuous availability and near transparent business continuity benefits for both disk and tape resident data. Enterprises may no longer be forced to develop and utilize processes that create duplex tapes and maintain the tape copies in alternate sites. For example, previous techniques created two copies of each DBMS image copy and archived log as part of the batch process and manual transportation of each set of tapes to different locations.

Operational data, or data that is used directly by applications supporting users, is normally found on disk. However, there is another category of data that *supports* the operational data, which is typically found on tape subsystems. Support data typically covers migrated data, point in time backups, archive data, and so on. For sustained operation in the recovery site, the support data is indispensable. Furthermore, several enterprises have mission critical data that only resides on tape.

The PtP VTS provides a hardware-based duplex tape solution and GDPS can automatically manage the duplexed tapes in the event of a planned site switch or a site failure. Control capability has been added to allow GDPS to *freeze* copy operations, so that tape data consistency can be maintained across GDPS managed sites during a switch between the primary and secondary VTSs.

FlashCopy support

FlashCopy, available on the IBM TotalStorage DS Family and IBM TotalStorage Enterprise Storage Server (ESS), is designed to provide an *instant* point-in-time copy of the data for application usage, such as backup and recovery operations. FlashCopy can enable you to copy or dump data while applications are updating the data. Prior to the release of FlashCopy V2 in 2003, both source and target volumes had to reside on the same logical subsystem. Since this constraint has been removed with FlashCopy V2, GDPS will now allow a FlashCopy from a source in one LSS to a target in a different LSS within the same disk subsystem.

FlashCopy before resynchronization is automatically invoked (based upon policy) whenever a resynchronization request is received. This function provides a consistent data image to fall back to, in the rare event that a disaster should occur while resynchronization is taking place. FlashCopy can also be user-initiated at any time. Clients can then use the tertiary copy of data to conduct D/R testing while maintaining D/R readiness, perform either test/development work, shorten batch windows, and so on.

FlashCopy can operate in either of two modes: the COPY mode, which runs a background copy process, and the NOCOPY mode, which suppresses the background copy. Previously, GDPS/PPRC and GDPS/XRC have provided support for both COPY and NOCOPY.

With the release of GDPS/PPRC V3.2 and GDPS/XRC V3.2, two FlashCopy enhancements are now available. The first enhancement is support for NOCOPY2COPY, which allows changing an existing FlashCopy relationship from NOCOPY to COPY. This gives you the option of always selecting the NOCOPY option of FlashCopy and then converting it to the COPY option when you want to create a full copy of the data in the background at a non-peak time.

Another FlashCopy enhancement available with GDPS V3.2 is support for Incremental FlashCopy. This provides the capability to refresh a volume in a FlashCopy relationship and reduce background copy time when only a subset of the data has changed. With Incremental FlashCopy, the initial relationship between a source and target is maintained after the background copy is complete.

When a subsequent FlashCopy establish is initiated, only the data updated on the source since the last FlashCopy is copied to the target. This reduces the time needed to create a third copy, thus giving you the option to perform a FlashCopy on a more frequent basis.

3.9.11 GDPS Support for heterogeneous environments

Here we discuss GDPS Support for heterogeneous environments.

Management of zSeries operating systems

In addition to managing images within the base or Parallel Sysplex cluster, GDPS can now also manage a client's other zSeries production operating systems – these include z/OS, Linux for zSeries, z/VM, and VSE/ESA™. The operating systems have to run on servers that are connected to the same Hardware Management Console (HMC) Local Area Network (LAN) as the Parallel Sysplex cluster images. For example, if the volumes associated with the Linux images are mirrored using PPRC, GDPS can restart these images as part of a planned or unplanned site reconfiguration. The Linux for zSeries images can either run as a logical partition (LPAR) or as a guest under z/VM.

GDPS/PPRC management for Open Systems LUNs (Logical Unit Number)

GDPS/PPRC technology has been extended to manage a heterogeneous environment of z/OS and Open Systems data. If installations share their disk subsystems between the z/OS and Open Systems platforms, GDPS/PPRC can manage the Metro Mirror and FlashCopy for open systems storage. GDPS/PPRC is also designed to provide data consistency across both z/OS and Open Systems data. This allows GDPS to be a single point of control to manage business resiliency across multiple tiers in the infrastructure, improving cross-platform system management, and business processes.

GDPS/PPRC Multi-Platform resiliency for zSeries

GDPS/PPRC has been enhanced to provide a new function called *GDPS/PPRC Multi-Platform Resiliency for zSeries*. This function is especially valuable for clients who share data and storage subsystems between z/OS and z/VM Linux guests on zSeries, for example, an application server running on Linux on zSeries and a database server running on z/OS.

With a multi-tiered architecture, there is a need to provide a coordinated near Continuous Availability/Disaster Recovery solution for both z/OS and zLinux. GDPS/PPRC can now provide that. z/VM 5.1 provides a HyperSwap function, so that the virtual device associated with one real disk can be swapped transparently to another disk. HyperSwap can be used to switch to secondary disk storage subsystems mirrored by Peer-to-Peer Remote Copy (PPRC, or Metro Mirror). If there is a hard failure of a storage device, GDPS coordinates the HyperSwap with z/OS for continuous availability spanning the multi-tiered application. For site failures, GDPS invokes the Freeze function for data consistency and rapid application restart, without the need for data recovery. HyperSwap can also be helpful in data migration scenarios to allow applications to migrate to new disk volumes without requiring them to be quiesced.

GDPS/PPRC will provide the reconfiguration capabilities for the Linux on zSeries servers and data in the same manner as for z/OS systems and data. To support planned and unplanned outages, GDPS provides the recovery actions, such as the following examples:

- Re-IPL in place of failing operating system images
- Site takeover/failover of a complete production site
- Coordinated planned and unplanned HyperSwap of disk subsystems, transparent to the operating system images and applications using the disks.
- Linux node or cluster failures
- Transparent disk maintenance or failure recovery with HyperSwap across z/OS and Linux applications
- Data consistency with freeze functions across z/OS and Linux

3.9.12 GDPS/GM

GDPS/Global Mirror is a GDPS offering to help manage disk mirroring on behalf of zSeries and open system servers for the Global Mirror copy technology as well as providing automation facilities for reconfiguration of zSeries servers and failover of zSeries systems to the recovery site in the event of a disaster in the application site. Monitoring and alerting facilities are included so that the operator is notified in a timely manner of exceptions and deviations from what is expected.

Global Mirror is IBM's asynchronous PPRC replication technology. When an application issues a write request to a primary device that is part of the mirroring configuration, the I/O completes as soon as the request is successfully received by the primary control unit. Soon thereafter, the updated data is sent to the secondary disk subsystem using PPRC-XD and at regular intervals FlashCopy is used to save consistent images of all disks in the Global Mirror session. Because of the asynchronous nature of Global Mirror, it is possible to have the secondary disk subsystem at greater distances than would be acceptable for synchronous PPRC.

Channel extender technology can be used to place the secondary disk subsystem up to thousands of kilometers away.

Figure 3-11 shows the physical topology of a GDPS/GM configuration. GDPS/GM requires two continually running systems: the Controlling system at the application site with the zSeries and Open servers running production, and the Recovery system at the remote recovery site.

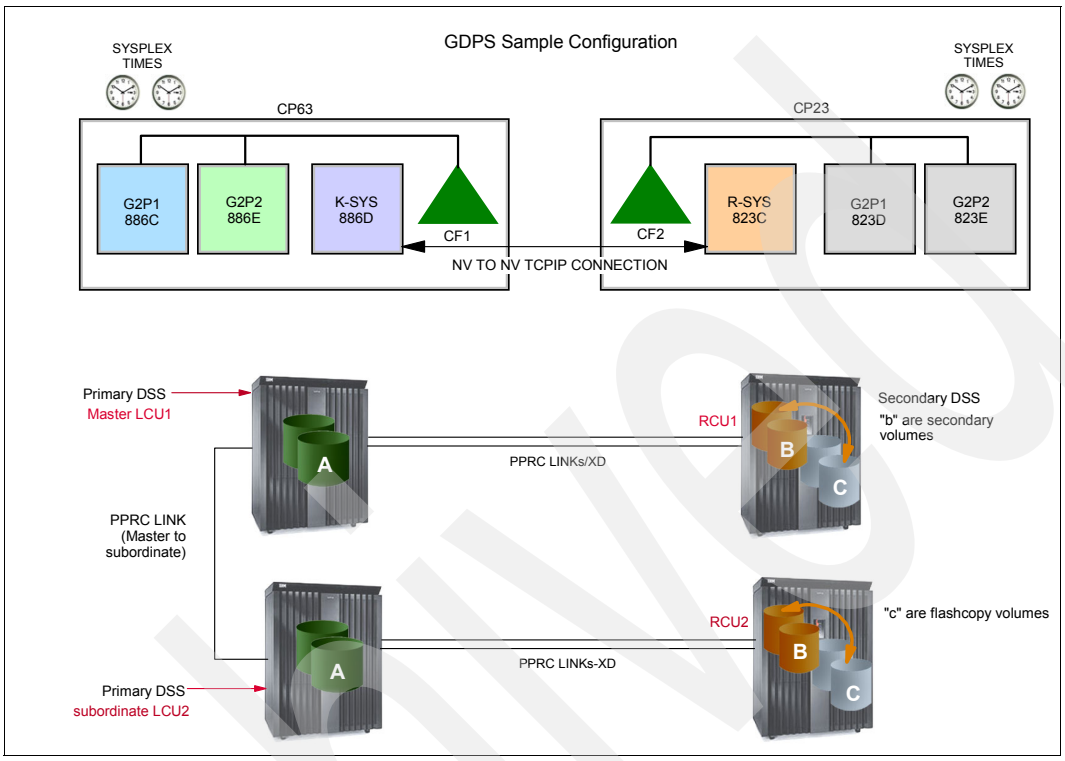


Figure 3-11 Example of GDPS/GM configuration.

The *Application site* is the site where production applications whose data is to be mirrored normally run and it is the site where the Global Mirror primary disks are located. You may also see this site referred to as the *local site* or the *A-site*.

The *Recovery site* is the site where a mirrored copy of the production disks are located and it is the site into which production systems are failed over to in the event of a disaster. You may also see this site referred to as the *remote site* or the *R-site*.

The *Controlling systems*, also known as the *K-sys* and *R-sys*, are the systems that run GDPS code to manage the mirroring environment and carry out the tasks for site failover.

The K-sys is a system in the application site, and its primary role is to manage the Global Mirror environment.

A *Production system* can run GDPS code and function as the K-sys, although it is desirable to isolate the K-sys in a separate z/OS image.

The R-sys is a system in the recovery site whose primary role is to perform the failover actions required to restart production systems using the mirrored copy of the data in the event of a disaster. Specifically, R-sys will recover the Global Mirror secondary data up to the last completed consistency point; it will reconfigure the recovery site resources and restart the production systems in the recovery site using the recovered disks.

The Controlling and Recovery systems heartbeat and communicate alerts, and status information via NetView to NetView communication between these two systems.

Configuration updates are specified on the Controlling system and are propagated to the Recovery system, again, via NetView to NetView communication.

GDPS/GM Controlling system

The Controlling system can be one of the following:

- A stand-alone monoplex system that is not a member of any sysplex in the local site
- A member of an existing sysplex in a local site environment.

  The Controlling system is a member of an existing sysplex; it must be the only system in this sysplex that is running GDPS automation code.

GDPS/GM Recovery system

The Recovery system can be one of the following:

- A stand-alone monoplex system that is not a member of any sysplex in the recovery site.
- A member of an existing sysplex in the recovery site.

  If the Recovery system is a member of an existing sysplex, it must be the only system in this sysplex that is running GDPS automation code.

3.9.13 IBM Global Services offerings

The following GDPS services and offerings are provided by IBM Global Services.

Technical Consulting Workshop (TCW)

TCW is a two day workshop where IBM Global Services specialists work with client representatives to understand the business objectives, service requirements, technological directions, business applications, recovery processes, cross-site, and I/O requirements. High-level education on GDPS is provided, along with the service and implementation process. Various remote and local data protection options are evaluated.

IBM Global Services specialists present a number of planned and unplanned GDPS reconfiguration scenarios, with recommendations on how GDPS can assist the client in achieving their objectives. At the conclusion of the workshop, the following items are developed: acceptance criteria for both the test and production phases, a high level task list, a services list, and project summary.

Remote Copy Management Facility (RCMF)

With this service, the RCMF/PPRC or RCMF/XRC automation to manage the remote copy infrastructure will be installed, the automation policy customized, and the automation verified along with providing operational education for the enterprise.

GDPS/PPRC HyperSwap Manager

IBM Implementation Services for GDPS/PPRC HyperSwap Manager helps simplify implementation by working with the client to get GDPS/PPRC HyperSwap Manager and its prerequisites up and running with limited disruption to client's business. On-site planning, configuration, implementation, testing, and education activities are part of the IBM Implementation Services for GDPS/PPRC HyperSwap Manager solution.

GDPS/PPRC and GDPS/XRC

GDPS IBM Implementation Services for GDPS/PPRC or GDPS/XRC assists client with planning, configuration, automation code customization, testing, onsite implementation assistance, and training in the IBM GDPS solution. Either option supports the Peer-to-Peer Virtual Tape Server (PtP VTS) form of tape data mirroring.

3.9.14 Prerequisites

For IBM to perform these services, you must have the following elements. The prerequisites listed in Table 3-1 may not contain all of the requirements for this service. For a complete list of prerequisites, consult your sales representative.

Table 3-1 Prerequisites required

Prerequisites	RCMF	GDPS
Supported version of z/OS or z/OS.e z/VM V5.1 or higher (note 1)	OK	OK
IBM Tivoli System Automation for Multiplatforms V1.2 or higher (note 1)		OK
IBM Tivoli System Automation for z/OS V2.2 or higher (note 2)		OK
IBM Tivoli NetView V5.1 or higher (note 2)	OK	OK
Storage subsystem with PPRC Freeze function (CGROUP Freeze/RUN) (note 3 + note 4)	OK	OK
XRC support with Unplanned Outage support (note 4)	OK	OK
Multisite Base or Parallel Sysplex (GDPS/PPRC)		OK
Common Timer Reference (Sysplex Timer) for XRC	OK	OK

- Note 1: z/VM is a prerequisite if GDPS/PPRC Multiplatform Resiliency is required.
- Note 2: For GDP/PPRC HyperSwap Manager, the following software products are required:
  - IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager with NetView V1.1 or higher, or
  - IBM Tivoli NetView for z/OS V5.1 or higher together with one of the following:
    - IBM Tivoli System Automation for GDPS/PPRC HyperSwap Manager V1.1 or higher, or
    - IBM Tivoli System Automation for z/OS V2.2 or higher
- Note 3: GDPS/PPRC HyperSwap requires PPRC support for Extended CQuery. GDPS/PPRC Management of Open Systems LUNs requires support for Open PPRC, management of Open PPRC via CKD device addresses, and Open PPRC SNMP alerts.
- Note 4: GDPS FlashCopy support requires FlashCopy V2 capable disk subsystems. In addition, visit http://www.ibm.com/servers/storage/support/solutions/bc for a listing of all recommended maintenance that should be applied, as well as review Informational APAR II12161.

3.9.15 More information about GDPS

- *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374
- The GDPS home page on the Web at:

  http://www.ibm.com/systems/z/gdps/
- Achieving Near Continuous Availability at:

  http://www.ibm.com/servers/eserver/zseries/pso/

3.10 Recommended sources of disaster recovery information

The following is a list of books that provide valuable information for many aspects of disaster recovery:

- *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370

4

Workloads in Parallel Sysplex

In this chapter, we review the important workload considerations for the Parallel Sysplex environment. We examine the most common workload types and look at the characteristics of their exploitation of the Parallel Sysplex. Also, where possible, *any changes to your configuration* caused or recommended by the implementation of a Parallel Sysplex are highlighted.

Sizing CF structures: Just in case you have not looked at the earlier chapters or have forgotten, there is a Web-based tool to help you determine the size of each of your CF structures at:

`http://www.ibm.com/servers/eserver/zseries/cfsizer/`

The tool has been deliberately kept simple to make it as easy to use as possible, and to minimize the amount of data you need to be able to use it. In addition to providing recommended structure sizes, the tool also provides IXCMIAPU statements that can be tailored to your configuration and used to help set up the structures.

Recommended sources of further information: The following sources provide support for the information in this chapter:

- *OS/390 V2R9.0 Parallel Sysplex Application Migration,* GC28-1863
- *OS/390 R4 Implementation*, SG24-2089
- *OS/390 Release 5 Implementation*, SG24-5151
- *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*, SG24-5277
- *System/390 MVS Parallel Sysplex Migration Paths*, SG24-2502
- *SNA in a Parallel Sysplex Environment*, SG24-2113
- *TCP/IP in a Sysplex*, SG24-5235
- *Revealed! Architecting e-business Access to CICS*, SG24-5466
- *Securing Web Access to CICS*, SG24-5756
- *System Programmer's Guide to: Workload Manager*, SG24-6472
- *IBM Web-to-Host Integration Solutions*, SG24-5237
- *DB2 UDB for z/OS V8 Data Sharing: Planning and Administration,* SC18-7417
- *DB2 UDB for z/OS V8 Administration Guide,* SC18-7413
- *DB2 UDB for z/OS V8 Release Planning Guide,* SC18-7425

Recommended sources of further information: The following sources provide support for the information in this chapter:

- *A Performance Study of Web Access to CICS*, SG24-5748
- *Batch Processing in a Parallel Sysplex*, SG24-5329
- *CICS and VSAM Record Level Sharing: Implementation Guide*, SG24-4766
- *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765
- *CICS and VSAM Record Level Sharing: Recovery Considerations*, SG24-4768
- *CICS Transaction Server for OS/390: Version 1 Release 3 Implementation Guide*, SG24-5274
- *CICS Transaction Server for OS/390 V1R3 CICS Intercommunication Guide*, SC33-1695
- *CICS Transaction Server for OS/390 V1.3 Migration Guide*, GC34-5353
- *CICS Transaction Server for OS/390 V1R2 Release Guide*, GC33-1570
- *CICS for OS/390 and Parallel Sysplex*, GC33-1180
- *CICS Transaction Server for OS/390: Web Interface and 3270 Bridge*, SG24-5243
- *CICSPlex SM Business Application Services: A New Solution to CICS Resource Management*, SG24-5267
- *CICS Transaction Server for OS/390 V1R3 Planning for Installation*, GC33-1789
- *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*, GG24-4286
- *CICS Transaction Server for OS/390 V1.3 CICSPlex SM Concepts and Planning*, GC33-0786
- *DB2 on MVS Platform: Data Sharing Recovery*, SG24-2218
- *DB2 UDB for OS/390 and Continuous Availability*, SG24-5486
- *DB2 UDB for z/OS V8 What's New*, GC18-7428
- *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908
- *IMS in the Parallel Sysplex Volume II: Planning the IMSplex*, SG24-6928
- *IMS in the Parallel Sysplex Volume III IMSplex Implementation and Operations*, SG24-6929
- *IMS Version 9 Implementation Guide: A Technical Overview*, SG24-6398
- *IMS Connectivity in an On Demand Environment: A Practical Guide to IMS Connectivity*, SG24-6794
- *IMS e-business Connect Using the IMS Connectors*, SG24-5427
- *JES3 in a Parallel Sysplex*, SG24-4776
- *OS/390 MVS Multisystem Consoles Implementing MVS Sysplex Operations*, SG24-4626
- z/OS MVS Parallel Sysplex Test Report:

  `http://www.ibm.com/servers/eserver/zseries/pso`
- *z/OS V1R2 Communications Server: APPC Application Suite Administration*, SC31-8835

4.1 e-business and Parallel Sysplex

When the first version of this book was written, few people had heard of the World Wide Web, and even fewer had access to it. Now the World Wide Web is all-pervasive. You hardly see an advertisement that does not have the company's URL at the bottom of the ad, and you take it as a given that you can sit at a Web browser, enter `www.widgets.com`, and be taken to the Widgets company Web site.

The role that Parallel Sysplex plays in e-business is also rapidly evolving. When the Internet started merging with the business world, the majority of Web interactions involved downloading static HTML pages. In that environment, and given the level of HFS sharing available at the time, Parallel Sysplex was not seen as a big player in the Internet world.

Since then, things have changed dramatically. No longer is the Web used simply as a marketing medium; it is now the way companies actually transact business. Interactions might still include HTML pages, but they also include a lot more, like talking to back-end CICS, DB2, and IMS systems. Companies are leveraging their existing applications and data to enable real e-business, in a much more efficient manner, and with a worldwide client base.

Together with this move to e-business comes a requirement for much higher application availability. If your company cannot respond to a client request, that client can now be in your competitor's *shop* about 10 seconds later – meaning that lack of availability really does mean quantifiable lost business.

For these reasons, clients doing serious e-business are looking more and more at Parallel Sysplex as a way of meeting the divergent requirements of:

- Maintaining near-continuous application availability
- Still having to process the traditional workloads, including batch jobs and data backups
- Being able to keep the system software up to date, in order to keep pace with competitors

The software in this area is evolving so rapidly that it is just about impossible to document it in a static publication like this. Hardly a month goes by without a product announcement that adds new or more flexible features. Therefore, in this section, we will only briefly discuss the options currently available, and refer you to documents that discuss these options in far greater detail.

4.1.1 Sysplex components for e-business applications

In this section, we briefly introduce some of the major components that need to be well understood and used appropriately for e-business applications:

- Hierarchical File System (HFS)
- System Logger
- Resource Recovery Services (RRS)
- Workload Manager (WLM)

Sharing of the files in Hierarchical File System (HFS)

In the past, a possible inhibitor to exploiting a Parallel Sysplex as a Web Server was the lack of shared read/write support for the Hierarchical File System (HFS). Prior to OS/390 V2R9, if you wished to share a HFS between more than one z/OS image, *all* of the images had to have the HFS mounted in read-only mode. This made it difficult to schedule updates to the HTML pages and Java™ application files stored in the HFS while still maintaining the service.

This inhibitor has been removed in OS/390 V2R9. OS/390 V2R9 provides support for multiple z/OS systems to share an HFS and still have the ability for any of the systems to make

updates to the HFS. As a result, it is now possible to maintain the availability of the Internet service running across a number of z/OS images (for improved availability) and still be able to make any required changes to the HFS content.

Notes: To help you understand Shared HFS function, we recommend that you refer to the animations that are available at:

`http://www.ibm.com/servers/eserver/zseries/zos/bkserv/animations/ussanims.html`

In order to share a file across systems, all the systems that are sharing the HFS in question must be running OS/390 V2R9 or a later release and must be in the same sysplex. As shown in Figure 4-1 on page 161, the systems consist of three kinds of HFS datasets, as described below:

- The sysplex root HFS data set

  This HFS dataset contains directories and symbolic links that allow redirection of directories. Only one sysplex root HFS is allowed for all systems participating in shared HFS.

  The sysplex root is used by the system to redirect addressing to other directories. It is very small and is mounted read-write. There are sample jcl to build the sysplex ROOT HFS dataset in SYS1.SAMPLIB(BPXISYSR).

- The system-specific HFS data sets

  This HFS dataset contains data specific to each system, including the /dev, /tmp, /var, and /etc directories for one system. There is one system-specific HFS data set for each system participating in the shared HFS capability.

  The system-specific HFS data set is used by the system to mount system-specific data. It contains the necessary mount points for system-specific data and the symbolic links to access sysplex-wide data, and should be mounted read-write. There are sample jcl to build the system-specific HFS dataset in SYS1.SAMPLIB(BPXISYSS).

- The version HFS

  This HFS dataset contains system code and binaries, including the /bin, /usr, /lib, /opt, and /samples directories. IBM delivers only one version root; you might define more as you add new system levels and new maintenance levels.

  The version HFS has the same purpose as the root HFS in the non-sysplex world. It should be mounted read-only. IBM supplies this HFS in the ServerPac.

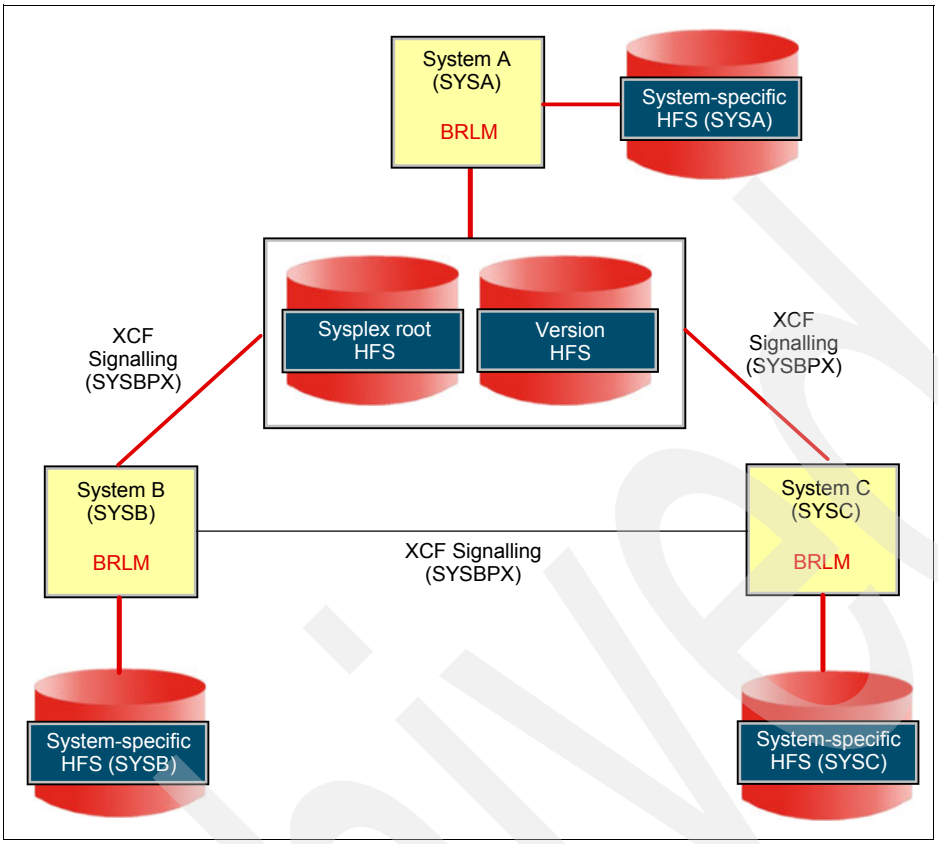


Figure 4-1 Main HFS datasets in Shared HFS environment

Furthermore, OMVS couple dataset is also needed with this function. There are sample jcl to build OMVS couple dataset in SYS1.SAMPLIB(BPXISCDS). The HFS files are accessed through XCF signalling from each system in a shared HFS environment. Then you must be conscious of the mount mode (read-only or read-write) and dataset owner for each HFS dataset, which influences the performance of the access to a file in HFS. It's necessary to define the action of each dataset in a owner system failure. For APARs about dataset management in Shared HFS environment, the following Web site may be of assistance:

`http://www.ibm.com/servers/eserver/zseries/zos/unix/pdf/ow54824.pdf`

information about setting up a shared HFS is available in *z/OS V1R7.0 UNIX System Services Planning,* GA22-7800 and the redbook *Hierarchical File System Usage Guide*, SG24-5482.

Important: In system environment with this Shared HFS function, apply APAR OW52293. This distributes Byte Range Lock Manager (BRLM) for the files in HFS to each system in the sysplex environment. This APAR can help you avoid the problem where an application locks a file in HFS (like the inetd and cron daemons). Before making this effective, add ITEM NAME(DISTBRLM) to the OMVS couple dataset.

For more information about BRLM, refer to the following Web site:

`http://www.ibm.com/servers/eserver/zseries/zos/unix/apps/brlm.html`

Writing of log data by System Logger

System Logger is a z/OS component that allows an application to log data from a sysplex. You can log data from one system in a sysplex or from multiple systems across the sysplex. A System Logger application writes log data into a log stream (in the Coupling Facility or DASD-only). A Coupling Facility log stream can contain data from multiple systems, allowing a system logger application to merge data from systems across the sysplex. In a DASD-only log stream, interim storage for log data is contained in local storage buffers on the system.

All compatible Resource Managers in the sysplex have access to the log stream of DASD-only or of the structure in the Coupling Facility. The System Logger implementation provides for fault tolerant management of log data. There is no single point of failure if you have two CFs installed. While the log data is passed quickly and efficiently to the Coupling Facility, it is also logged on the generating system in a staging file or data space. Periodically, when the log threshold is met, the logs are *off-loaded* to VSAM linear data sets. This process ensures integrity of the log data while providing a highly efficient recovery mechanism for sysplex-wide resources. For detail information about system logger, refer to *Systems Programmer's Guide to: z/OS System Logger*, SG24-6898, and *z/OS MVS Setting Up a Sysplex*, SA22-7625

WebSphere Application Server for z/OS is predefined as a z/OS system logger application, so you can use a log stream as the product's error log. And WebSphere Application Server for z/OS is an RRS-compliant resource manager and will participate in transactional commits with DB2. Thus, WebSphere Application Server for z/OS will require RRS to start writing data to its system logger log streams. If you plan to configure WebSphere to run in a sysplex, we strongly recommend that you configure the system logger to use the Coupling Facility log stream for the best throughput.

Transaction management by Resource Recovery Services (RRS)

Resource Recovery Services (RRS) consists of the protocols and program interfaces that allow an application program to make consistent changes to multiple protected resources. z/OS, when requested, can coordinate changes to one or more protected resources, which can be accessed through different resource managers and reside on different systems. z/OS ensures that all changes are made or no changes are made. For detailed information, refer to *z/OS V1R7.0 MVS Programming: Resource Recovery,* SA22-7616.

For e-business applications in the WebSphere Application Server product, The Object Transaction Service (OTS) manages transactions and provides the CORBA OTS interface to z/OS Resource Recovery Services (RRS), which coordinates resource recovery across several Resource Managers. On the Servant side, this is a very thin layer that delegates requests to the Control Region. The Object Transaction Service (OTS) requires the existence of RRS. RRS in its turn requires the definitions of logstreams in a Couple dataset or DASD-only.

The IMS, CICS, and DB2 Resource Managers have also implemented an interface to RRS so that, in the case of a WebSphere client application, resource coordination between the various Resource Managers can be driven by RRS. When the WebSphere client application completes its work and issues a commit, then the Control Region notifies RRS of this event and the two phase commit process begins. The client is notified about the outcome and may then continue with the next transactional Unit of Work (UOW). And the activation of RRS implies the use of the RRS Attachment Facility (RRSAF) when using the DB2 JDBC™ (type-2) driver. For more information about RRS and the products related RRS, refer to *Systems Programmer's Guide to Resource Recovery Services (RRS)*, SG24-6980.

Queueing and balancing for applications by Workload Manager (WLM)

With workload management, you define performance goals and assign a business importance to each goal. You define the goals for work in business terms, and the system decides how much of the resources, such as CPU, storage, and channels, should be given to meet its the goal. WLM constantly monitors the system and adapts processing to meet the goals you set. For detailed information about Workload Manager, refer to *z/OS V1R7.0 MVS Planning: Workload Management,* SA22-7602, *z/OS MVS Programming: Workload Management Services,* SA22-7619, and *System Programmer's Guide to: Workload Manager*, SG24-6472.

The components like TCP/IP, Web Servers, and WebSphere Application Server in z/OS system can receive the following benefits from WLM services:

- Routing

  WLM can establish and manage connections between a client and a server address space (Routing Manager Services). This services performs two main functions:

  - Automatically starting and maintaining server address spaces as needed by the workload across the sysplex.
  - Balancing the workload among the servers in the sysplex by deciding on the best server and providing the server routing information when a server is requested by the routing manager.

- Queuing and address space management

  WLM has a service that queues work requests to workload management for execution by one or more server address spaces (Queueing Manager Services). This service allows the system to:

  - Dynamically start and stop server address spaces based on workload.
  - Control the number of server instances per server address space.
  - Manage the work queues associated with the server address spaces to meet the performance goals set by the client.

  With the dynamic management of server address spaces, an installation does not need to calculate the exact number of address spaces to process work, nor do they have to monitor workload fluctuations that change the number of address spaces needed. Clients can still segregate work requests into different server address spaces if this is important for security or integrity.

  Enclaves must be used with the queuing manager services. This means that clients can define velocity and discretionary goals for work as well as response time goals. Multiple period control is also available for work running in enclaves.

  Queueing Manager Services provide an incentive to subsystems who run with multiple tasks in one address space to switch to multiple address spaces. Queueing Manager Services make it easier for installations to isolate individual work requests from each other, by running only one work request in each execution address space, with WLM managing the number of execution address spaces.

- Prioritizing work to meet performance goals

  This is called Work Manager Services and is the most basic service provided by WLM. This services allow systems to recognize:

  - A subsystem work manager and the transactions it processes.
  - The service class goals associated with the transactions.
  - The address spaces that are processing the transactions.

Based on this information, workload management can determine whether goals are being met, and which work requests need resources to meet their goals.

The Work Manager Services allows:

- Your clients to define performance goals to your subsystem work manager's transactions.
- MVS to recognize the goals, and match resources to the work to meet the goals.
- Your clients to get reports from performance monitors like RMF on how well work is executing and whether the goals are being met.

Using the work manager services in your product allows your clients to specify goals for your work the same way they specify them for MVS-managed work.

The work manager services allow workload management to associate incoming work with a service class. When the work is associated with a service class, MVS knows the performance goal and importance level associated with the work, as well as understanding which address spaces are involved in processing the work request.

To exploit the benefits of WLM, the Web server must be enabled for WLM support, that is, it has to be defined in Scalable Server mode. This means that the Web server is configured for WLM support using the ApplEnv directive and is started using the -SN (subsystem name) parameter. The ApplEnv directive is used by the Web server to divide incoming requests into Application Environments in WLM goal modes and to route those requests. For detailed information about Web server, refer to *z/OS HTTP Server Planning, Installing and Using, V1R7,* SC34-4826.

WebSphere Application Server also requires that WLM is running in goal mode and Application Environments (AE) are defined. WebSphere uses the services of WLM to balance work on the system. For example, the WLM queuing service is used to dispatch work requests from the WebSphere Application Server for z/OS Control Region to one or more WebSphere Application Server for z/OS Servants. This tells WLM that this server would like to use WLM-managed queues to direct work to other servers, which allows WLM to manage server spaces to achieve the specified performance goals established for the work. And WebSphere delegates the responsibility for starting and stopping Servants to the WLM Address Space management service. This allows WLM to manage application server instances in order to achieve the performance goals specified by the business.

For more information about WebSphere products, refer to the following Web sites:

http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp
http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

Tip: For z/OS V1R2 version or later, Dynamic Application Environment function for WLM is supported by APAR OW54662.

4.1.2 Web Server

For most Web transactions, the primary means of communication will be via the Web Server, either running on z/OS or on another platform. Starting with the replacement for the Lotus® Domino® Go Web server (DGW), the Web Server was actually broken into two components. The Java servlet support was moved into a product called WebSphere Application Server. The remaining functions from DGW were provided by a new product called IBM HTTP Server of z/OS.

Figure 4-2 on page 165 shows the history of Web servers and WebSphere Application Server products for z/OS, which also contains a summary of the related OS/390 and z/OS releases.

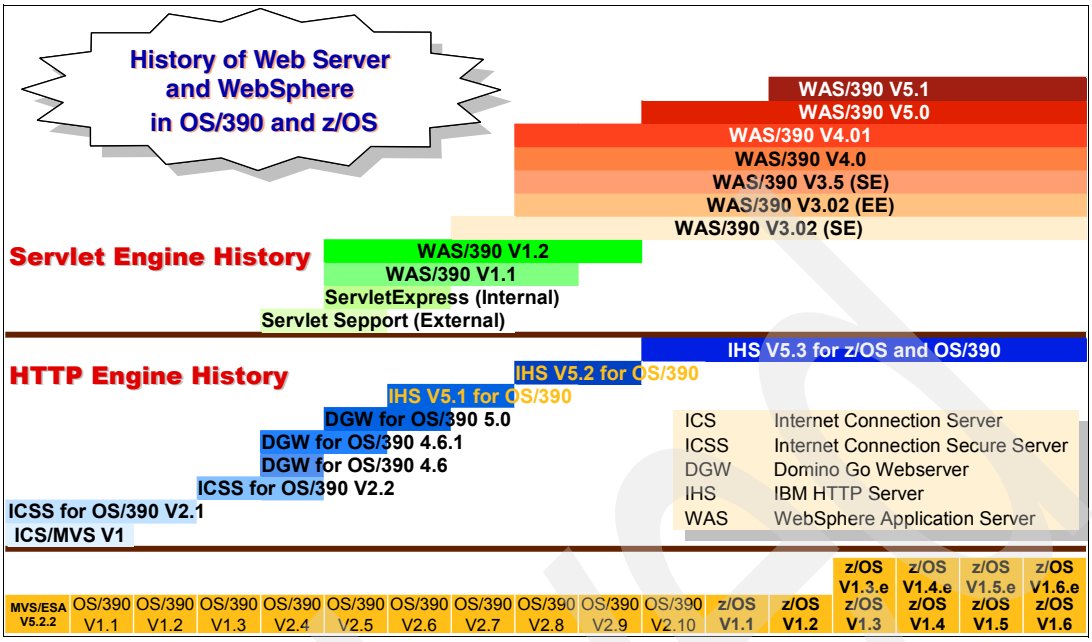


Figure 4-2 History of Web servers and WebSphere in z/OS and z/OS

Notes: For the most current HTTP server documentation and information updates, go to the following Web site at:

`http://www.ibm.com/software/webservers/httpservers/doc53.html`

We used the IBM HTTP Server for z/OS Version 5.3, which was the latest version at the timing of writing. Refer to *z/OS HTTP Server Planning, Installing and Using, V1R7,* SC34-4826 for more information.

There are three execution modes to run the IBM HTTP server in z/OS:

- Stand-alone server

  This mode is typically for simple HTTP server-only implementations. It is often used for simple Web sites that provide static information about a business and can also be used for a corporate intranet with limited function. Its main role is to provide a limited exposure to the Internet. Since it has a finite capacity based on resource definitions, it may not be able to adequately respond to changes in demand.

- Scalable server

  This mode is typically for interactive Web sites where the traffic volume can decline and increase dynamically and the ability to react is needed. It is also meant to be used in a more sophisticated environment where servlets and JSPs are invoked and performance demands have to be met. The WebSphere Application Server Environment is designed with this particular goal in mind. The HTTP Server can be started as a queue manager. Queue servers are spawned as a result of WLM adjusting workloads to accomplish throughput objectives.

- Multiple servers

  The combination of a stand-alone server and a scalable server, or even multiple scalable servers, can improve scalability and security throughout the system. An HTTP stand-alone server functioning as the gateway to scalable servers creates a barrier at which user authentication can be handled without exposing the entire system to unwanted eyes. The stand-alone server can then provide URL rerouting (links) to other servers listening on different ports in the same TCP/IP stack, or on different stacks.

For detailed information about each mode of running IBM HTTP Server, refer to the following Redbooks: *e-business Cookbook for z/OS Volume I: Technology Introduction*, SG24-5664. *e-business Cookbook for z/OS Volume II: Infrastructure*, SG24-5981, and *e-business Cookbook for z/OS Volume III: Java Development*, SG24-5980.

Among many other features, the Web Server provides support for a feature known as Scalable Web Serving. This support was added as part of ICSS 2.2. Using this support, one Web Server will take responsibility for accepting all incoming requests. Based on the number of requests awaiting execution and whether the Web Server is meeting its performance objectives, WLM may start additional Web Server or address spaces. Figure 4-3 contains a sample configuration showing the use of Network Dispatcher with the Scalable Web Server.

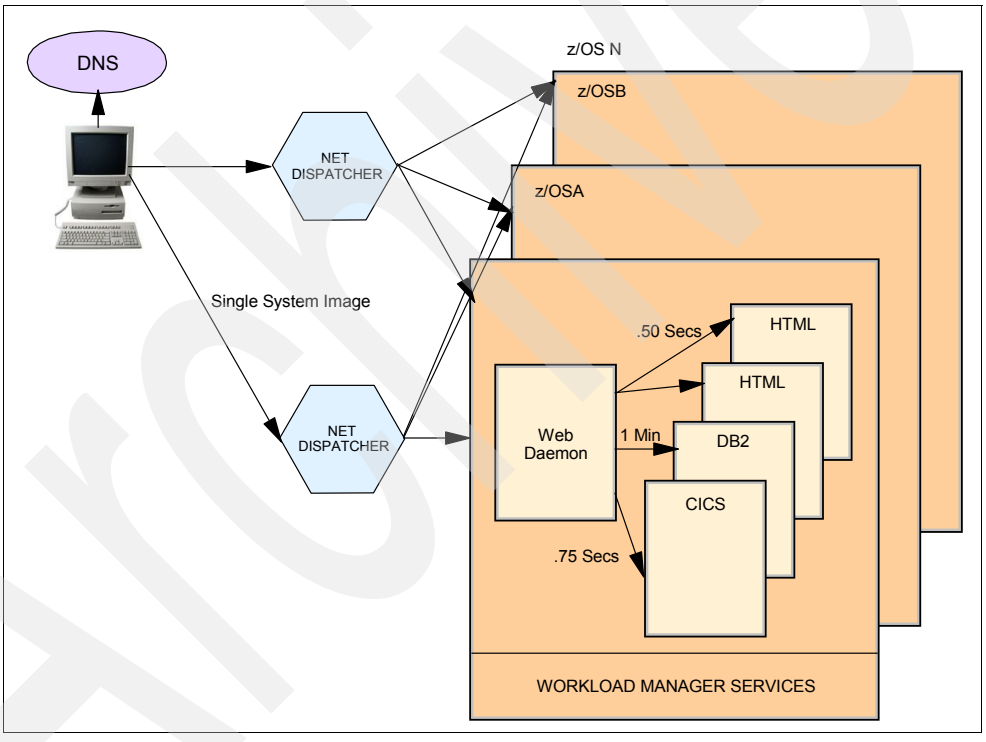


Figure 4-3 Sample scalable Web Server configuration

The combination of Network Dispatcher and the Scalable Web Server not only improves performance for the Web requests, it also provides improved availability by letting you spread the processing of Web requests across multiple server address spaces in multiple images.

More information about how to set up the Scalable Web Server can be found in the IBM Redbook *OS/390 e-business Infrastructure: IBM HTTP Server V5.1 for OS/390*, SG24-5603.

The Web Server can be used to:

- Serve HTML pages
- Communicate with CICS
- Communicate with DB2
- Communicate with IMS

One of the first things to consider when selecting a Web-enabling solution is what business process or processes you wish to make available to the Web. This will help to determine the required data sources and the preferred Web solution architecture.

If the requirement is simply to make existing 3270 applications available to Web users, then a Java 3270 emulator may be the quickest and best solution. If the requirement is to enhance the user interface, then there are several options that we will list that should be assessed to determine if they meet the user's needs.

If an entirely new application is required in order to re-engineer existing business processes, the WebSphere development environment would be the best choice, with the various Connectors for access to applications and data.

The next thing to consider, from a Parallel Sysplex point of view, is the availability characteristics of each option:

- If the solution you choose includes a Web Server or some other intermediate address space (which we will generically call the front-end), that address space should support workload balancing.
- You also have to consider if it is necessary for the front end to reside on the same system as your back-end application manager (CICS, DB2, or IMS).
- You need to decide on the location of the Web Server. Are you going to have a two-tier or a three-tier architecture?
- You have to consider the profile of the transactions. Is data carried from one transaction to another? If so, you either have to make sure that every part of the transaction is executed on the same servers, or you need to use a solution that will make this transient data available across multiple servers.
- Finally, you should check whether the front end can communicate with any instance of the application manager, or if it relies on a hardcoded applid or subsystem name.

In the following sections, we provide this information for each of the listed solutions.

To improve the performance for serving HTML pages, a feature known as Fast Response Cache Accelerator was introduced in OS/390 V2R7. The Fast Response Cache Accelerator uses Adaptive Fast Path Architecture (AFPA) to speed up the processing of static Web pages by the WebSphere Application Server. Web pages are cached within the TCP/IP stack, and requests are handled without traversing the entire kernel or entering the user space. From the perspective of the TCP/IP stack, the Cache Accelerator consists of a set of exits that are executed during stack processing in order to support transient, in-kernel *quick* connections. As a result, the Cache Accelerator executes as an extension of the TCP/IP stack rather than as a functional layer above the TCP/IP stack.

The technology developed for the High Speed Web Access facility, introduced in OS/390 V2R5 to improve performance for Web serving, was integrated into the level of TCP/IP delivered with OS/390 V2R6.

4.1.3 WebSphere Application Server in a Parallel Sysplex environment

The IBM WebSphere family of products represents the core IBM offering for clients who want to transform their business to e-business.

WebSphere Application Server

IBM WebSphere Application Server represents one of the IBM foundations of a On Demand Business infrastructure. It is the Java 2 Enterprise Edition (J2EE™) and Web services technology-based application platform, offering one of the first production-ready application servers for the deployment of enterprise Web services solutions for dynamic e-business.

WebSphere Application Server Version 6 provides an integration of deployment model, administration point, programming model, and integrated application development environment. It is fully J2EE 1.4 compatible. It delivers multiple configuration options for a wide range of scenarios, from simple administration of a single server to a clustered, highly available, high-volume environment. It offers advanced server technology, a production-ready environment, and visual workflow support for key Web services standards.

WebSphere Application Server for z/OS overview

WebSphere Application Server for z/OS provides a foundation for delivering the latest enterprise Java technologies to support e-business application development and deployment. Using WebSphere Application Server, you can develop and deploy new high-volume transaction e-business applications and Web-enable existing applications and databases on z/OS.

The WebSphere Application Server for z/OS product lets you create unlimited numbers of stand-alone application servers. However, the real strength of the product is the ability to create and manage a Network Deployment environment, in which you have installed managed application server nodes across a z/OS sysplex. The main reason to use managed nodes in a cell versus using the same number of stand-alone application servers is the centralized administration that the deployment manager provides for the cell. Another advantage is the ability to create clusters in a managed node that the deployment manager can then manage with simple workload balancing.

Figure 4-4 on page 169 describes the history of Java support in the WebSphere Application Server product. The Java technology used in this product has progressed quickly and is continuing to progress quickly, so check for the latest information in the manuals and at the following Web site:

http://www.ibm.com/software/webservers/appserv/zos_os390/features/index.html

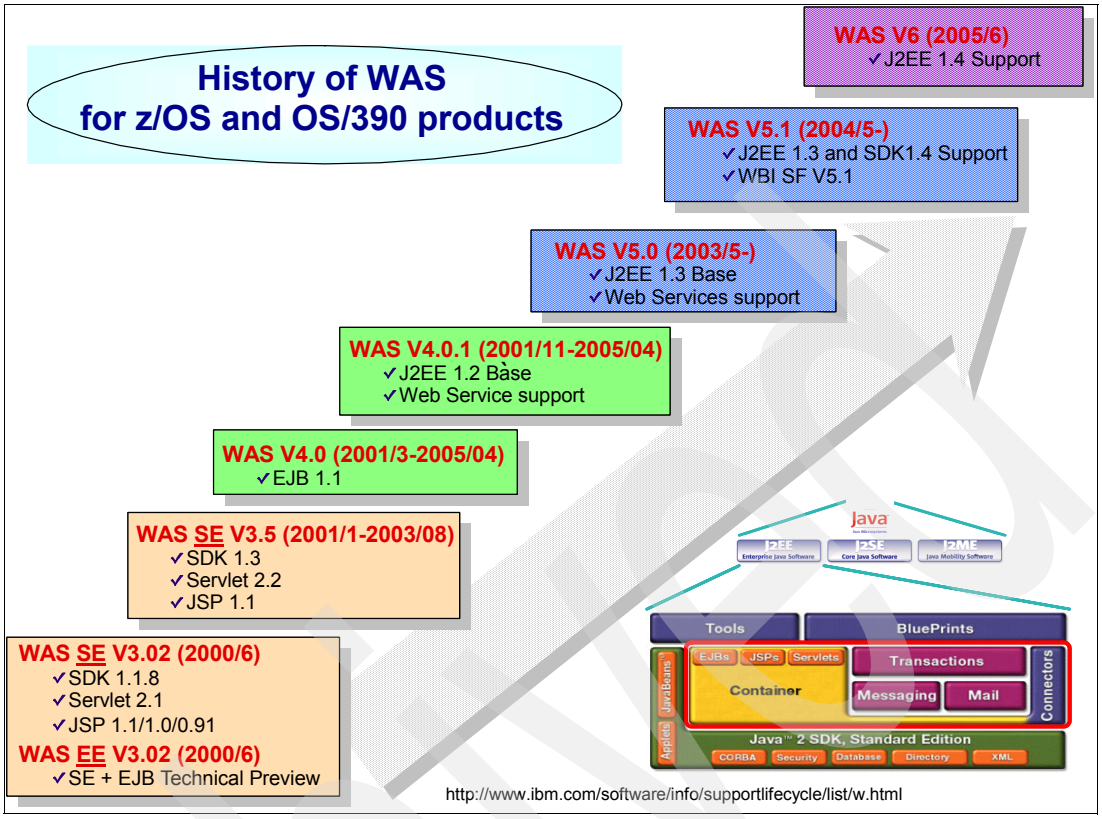


Figure 4-4 WebSphere levels

Notes: This redbook is written based on the information of WebSphere Application Server for z/OS V6.0, which is the latest version for that product available at the time of writing.

See the Infocenter site for WebSphere Application Server for z/OS V6.0 at:

`http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp`

Figure 4-5 shows the major components of WebSphere for z/OS in a sysplex environment on zSeries.

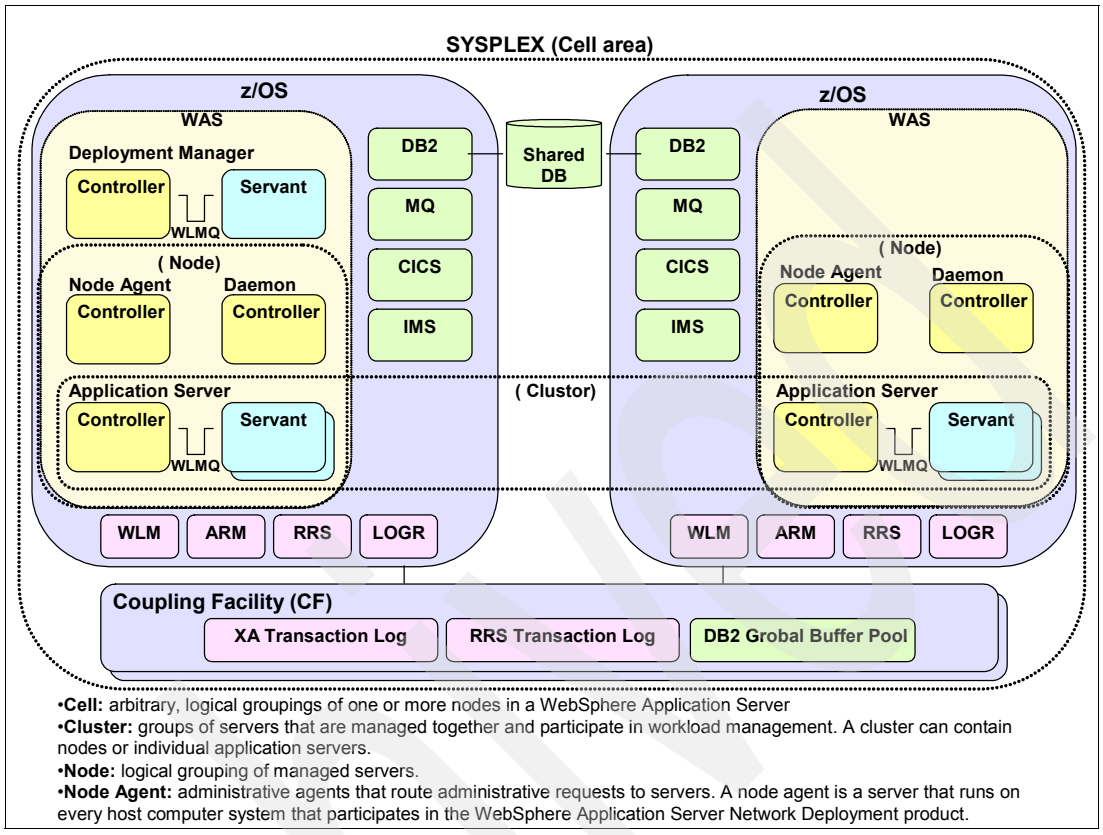


Figure 4-5 Components of WebSphere

The benefits of working in a sysplex environment include:

- Performance Management

  You can balance the workload across multiple systems, thus providing better performance management for your applications.

  Clustering exists in WebSphere products on all platforms for workload balancing and high availability. And the *Cluster* means the logical unit of a server cloned and the structure against which the identity of application is secured in a *Cluster* unit.

  The transaction work in z/OS environment are managed by WLM not only within a single system, but also between sysplex members. So, in the case where the *Cluster* consists of WebSphere Application Server for z/OS between members (z/OS systems) in a sysplex environment, this means that the Web applications can be managed for performance and resources based on WLM policy definition.

- Scalability

  As your workload grows, you can add new systems to meet demand, thus providing a scalable solution to your processing needs.

  When WebSphere Application Server servers are built for Network Deployment in a *Cell* area, servers can be added to respond to the amount of workload in a *Node* unit, including *Node Agent*. If a *Cell* unit is in a Parallel Sysplex environment, it will be easy to change the numbers of sysplex members (z/OS systems) and *Node*s.

- Availability

  By replicating the runtime and associated business application servers, you provide the necessary system redundancy to assure availability for your users. Thus, in the event of a failure on one system, you have other systems available for work.

  The transactions management on WebSphere Application Servers for z/OS is carried out by Resource Recovery Services (RRS) and the system logger function, which synchronizes point processing for application and transaction recovery in a failure and so on. This specification is completely different from WebSphere products on another platform; it exploits 2-phase commits to handle transactions in a heterogeneous environment. If a server fails, other servers in the cluster take over the work and the server is recovered, without re-starting a failed server.

- Manageability

  Administration of servers and other objects is centralized.You can manage all of the application servers as a group or cell by using the Administrative Console of the deployment manager. A deployment manager manages the configuration for all the managed nodes in its cell and deploys applications to any managed node in the cell.

  You can upgrade the Application Server from one release or service level to another without interrupting service to your users. And with the Sysplex Distributor function in a Parallel Sysplex environment, servers can be removed or added dynamically from the object of load management by a command.

For more information about this topic, refer to the redbook *Architecting High Availability Using WebSphere V6 on z/OS*, SG24-6850.

Understanding WebSphere Application Server for z/OS terminology

In WebSphere Application Server for z/OS, the functional component on which applications run is called a server. Servers comprise address spaces that actually run code.

Within each server are two kinds of address spaces: controllers and servants.

- A controller runs system authorized programs and manages tasks, such as communication, for the server.
- A servant is the address space in which the JVM™ resides. It runs unauthorized programs, such as business applications.

Note: The location service daemon and node agent are specialized servers and have no servants. The control region adjunct (new with WebSphere v6) is a specialized servant that interfaces with the new service integration buses to provide messaging services.

As shown in Figure 4-6 and Figure 4-7 on page 173, WebSphere Application Server for z/OS uses the Workload Manager (WLM) function of z/OS to start and manage servers in response to workload activity. When work builds up, WLM dynamically starts additional servants to meet the demand. Each J2EE application server in a WebSphere Application Server for z/OS cell uses WLM services to start servants as WLM application environments. Thus, each application server must associate with a WLM application environment name. WebSphere Application Server for z/OS makes use of dynamic WLM application environments when available.

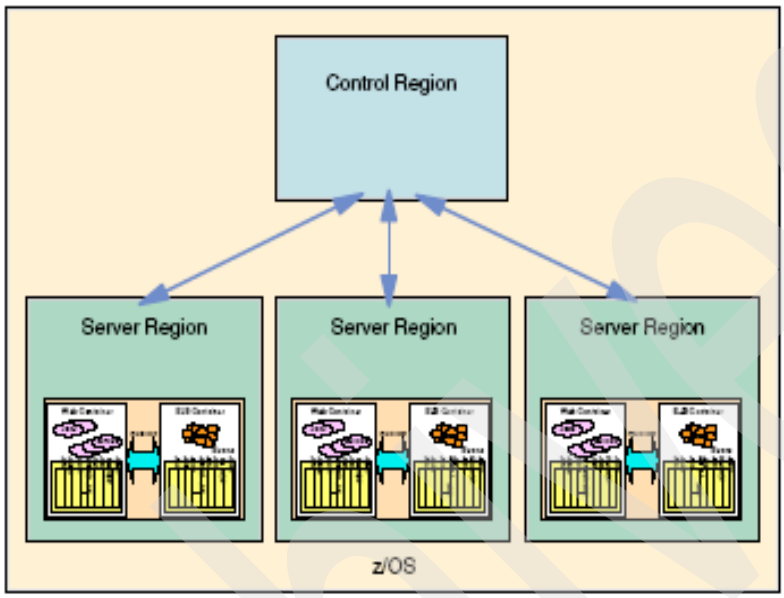


Figure 4-6 Workload management with WebSphere

Note: The WLM service that added dynamic application environments is a prerequisite for WebSphere Application Server for z/OS Version 6.0.1. See SPE (APAR OW54622, included in z/OS Version 1.5 and above), which is described at:

http://www-1.ibm.com/support/docview.wss?uid=isg1OW54622

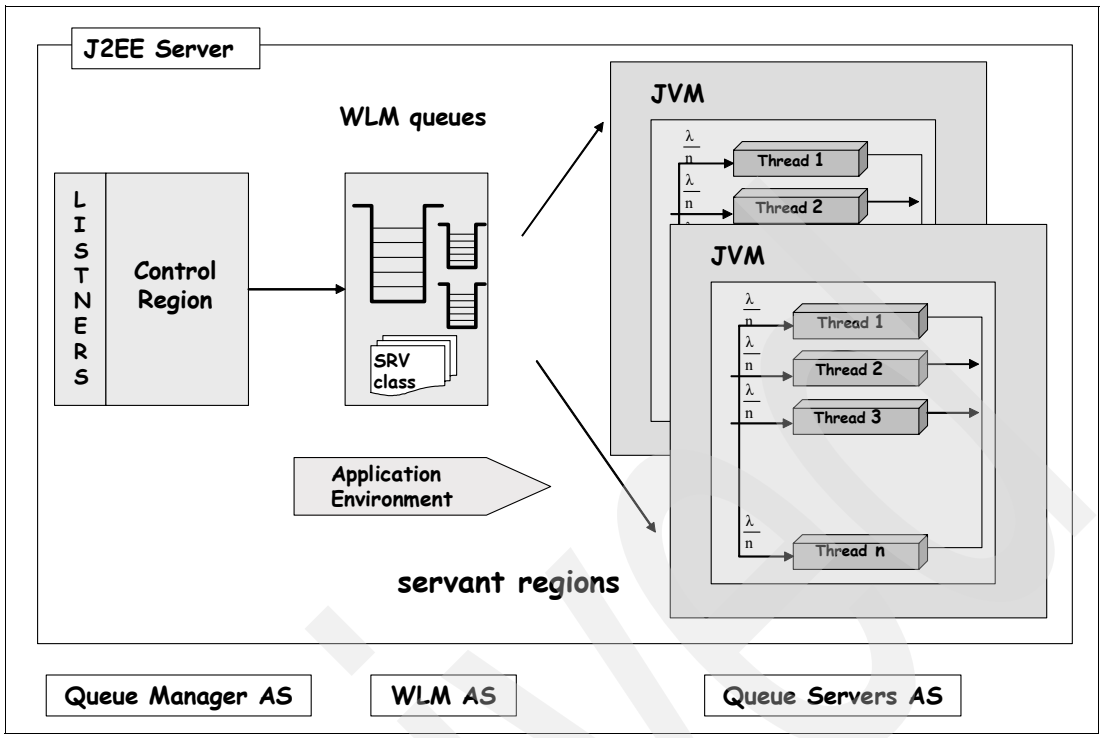


Figure 4-7 Workload management queues

Figure 4-8 on page 174 describes the breakdown of the different server types on your system:

- Unmanaged (stand-alone) application server: The application server that was set up during stand-alone configuration that hosts your J2EE applications.
- Managed (Network Deployment) application server: The application server set up during Network Deployment configuration that hosts your J2EE applications.
- Location service daemon**:** A server that is the initial point of contact for client requests in either configuration.
- JMS server hosts the JMS function in the WebSphere Application Server for z/OS, which controls the MQ broker and queue manager in either configuration. The JMS server no longer exists, as in previous versions of WebSphere Application Server for z/OS. Its function has been replaced with new service integration buses.
- Deployment manager: A specialized application server that hosts the administrative console application (it hosts only administrative applications) and provides cell-level administrative function in a Network Deployment configuration. The administrative console application administers servers (grouped into nodes) on many different systems. The deployment manager is the sole occupant of its own node structure, which does not need a node agent, because there are no application servers in the node, and a cell may have only one deployment manager.

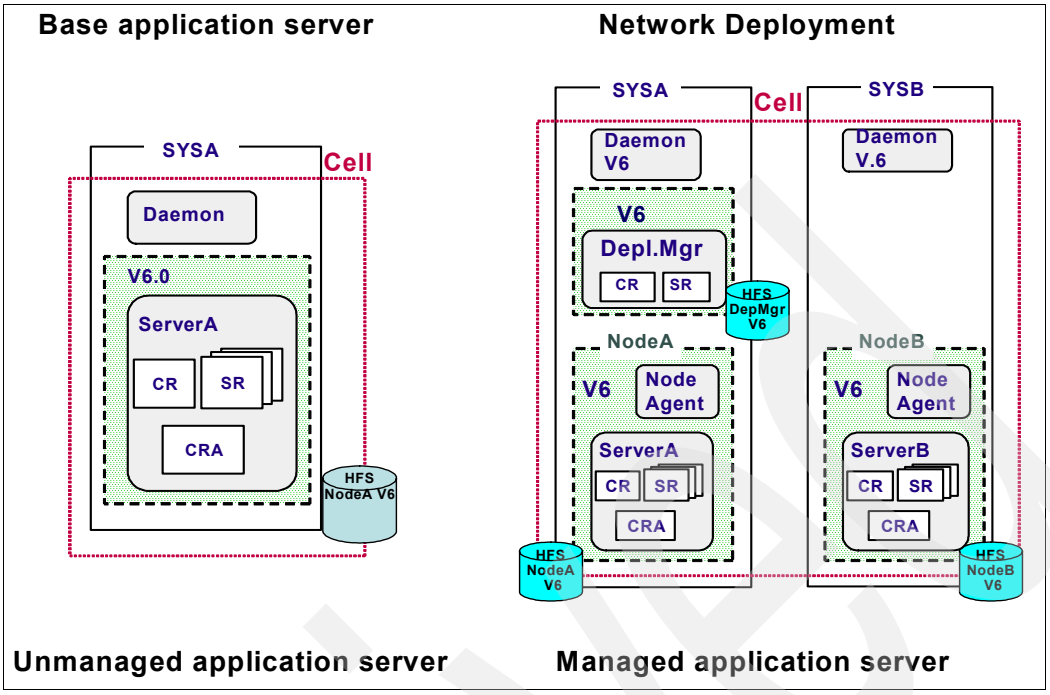


Figure 4-8 Server types breakdown

Note: The version of the administrative console application that runs in the deployment manager is designed to manage multinode environments, while the version in the stand-alone application server is for single node environments only.

- Node agent: Provides node-level administrative function in a Network Deployment configuration. A node contains servers that may be part of a cluster. The cluster may span nodes as long as all involved nodes are in the same cell. Here is a quick breakdown of clusters, nodes, and cells:
  - Cluster: A logical collection of like-configured application servers that provides performance, reliability, and administration advantages:
    - A cluster can span nodes and systems within the same cell.
    - Clusters are not a layer in the cell/node/server hierarchy. Instead, they are a way of grouping servers that host the same applications within a cell. A cluster can span nodes and systems within the same cell.
  - Node: A logical collection of servers on a particular z/OS system in the cell:
    - The cell to which a node belongs can span several systems, but the node must remain within a single z/OS system.
    - A z/OS system can contain multiple WebSphere Application Server for z/OS nodes, belonging to the same or different cells. A stand-alone WebSphere Application Server for z/OS cell consists of a single node. Due to administrative constraints, this node should have only a single application server in it.
    - A network deployment cell consists of a deployment manager node, which is responsible for cell-wide administrative tasks, and any number of managed nodes. Each managed node contains a node agent, which handles communication with the cell's deployment manager, and any number of application servers.

– Cell, as shown in Figure 4-9: A logical collection of WebSphere Application Server for z/OS nodes that are administered together. The cell is the largest unit of organization:
  - All nodes that comprise a cell must reside on systems in the same sysplex or on the same z/OS monoplex.
  - A z/OS sysplex or monoplex can contain multiple WebSphere Application Server for z/OS cells. Different cells may have nodes on the same systems, though a given node can be a member of only one cell.
  - There are two kinds of WebSphere Application Server for z/OS cells: stand-alone application server cells, and Network Deployment cells. To help you understand the interaction between servers, clusters, nodes and cells, Figure 4-9 various configurations you can set up in your Network Deployment sysplex.

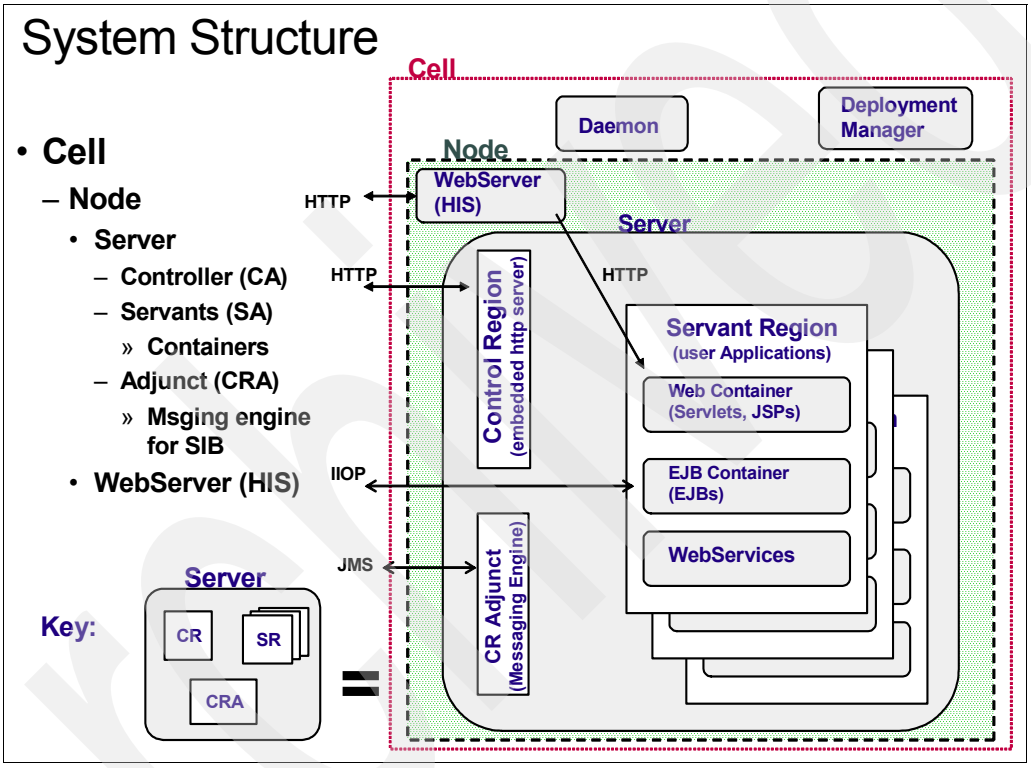


Figure 4-9 System structure

Planning a configuration HFS or zFS

Note: You can also use a zFS file system for the configuration directory; the term HFS can be used interchangeably with the term zFS.

This topic describes the planning decisions you need to make when setting up a WebSphere Application Server for z/OS configuration HFS.

Relationship between the configuration HFS and the product HFS

The configuration HFS contains a large number of symbolic links to files in the product HFS (/usr/lpp/zWebSphere/V6R0, by default). This allows the server processes, administrator and clients to access a consistent WebSphere Application Server for z/OS code base.

Cell, node, and server settings, as well as deployed applications, are stored in the WebSphere Application Server for z/OS configuration directory or WebSphere Application Server for z/OS configuration HFS, as shown in Figure 4-10.

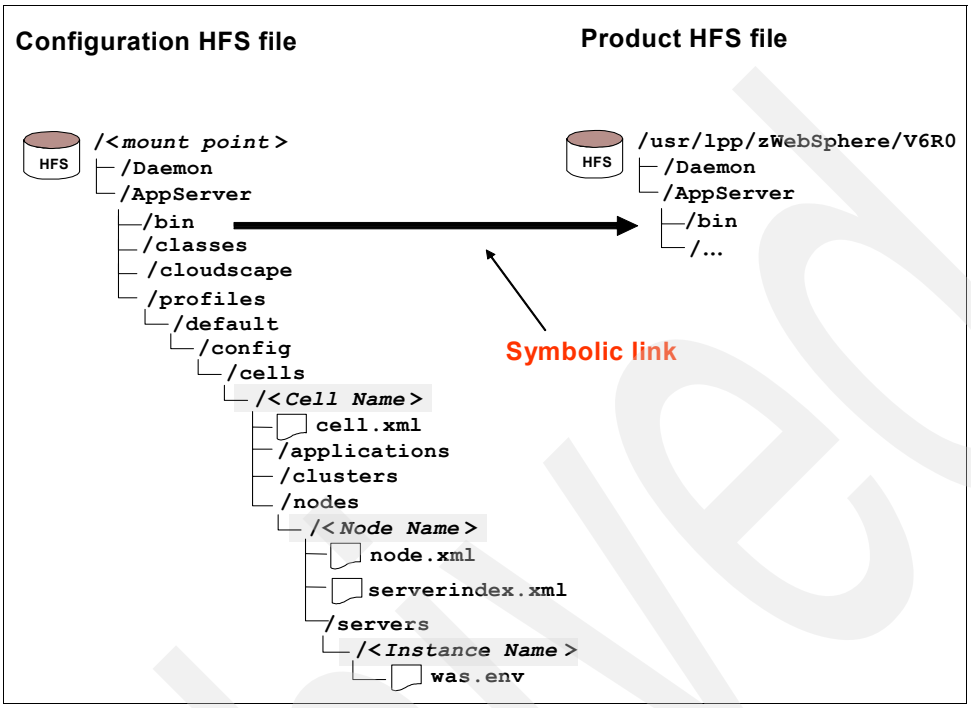


Figure 4-10 HFS configuration file

Note: These symbolic links are set up when the WebSphere Application Server home directory is created, and are very difficult to change. Therefore, systems that require high availability should keep a separate copy of the WebSphere Application Server for z/OS product HFS and product data sets for each maintenance or service level in use (test, assurance, production, and so on.) to allow system maintenance, and use intermediate symbolic links to connect each configuration HFS with its product HFS.

For more information about setting up a WebSphere Application Server for z/OS configuration that allows you to apply maintenance nondisruptively to a cell, see the Washington Systems Center white paper *Planning for Test, Production and Maintenance*. You can find this document on the Web at `http://www.ibm.com/techdocs` by searching for document number WP1000396 under the category *White Papers*.

The following sections present various elements with inherent decisions to make when setting up your WebSphere Application Server for z/OS configuration HFS, as well as give you the information to best make those decisions based on the needs of your planned configuration. Each node needs a home directory, and every WebSphere Application Server for z/OS node, whether it is a stand-alone application server, deployment manager, managed application server node, or location service daemon, requires a read/write home directory.

Sharing the configuration HFS between cells

Two or more WebSphere Application Server for z/OS cells (stand-alone application server, Network Deployment, or both) can share a WebSphere Application Server for z/OS configuration HFS, given the following:

- All cells using the configuration HFS must be set up using the same security domain. In particular, each must have the same administrator user ID and configuration group.
- The cells must have distinct cell short names.
- Each node must have its own WAS_HOME directory that is not shared with any other node or cell. As noted above, you may share the daemon home directory (/Daemon) between cells, as it has subdirectories farther down for each cell in the configuration HFS.

Note: Beware that sharing a configuration HFS between cells increases the likelihood that problems with one cell may cause problems with other cells in the same configuration's HFS.

Sharing the configuration HFS between systems

Two or more z/OS systems can share a configuration HFS, provided the z/OS systems have a shared sysplex and the configuration HFS is mounted R/W. All updates are made by the z/OS system that "owns" the mount point. For a Network Deployment cell, this is generally the z/OS system on which the cell deployment manager is configured.

Which HFS mount point for your WebSphere Application Server for z/OS

The choice of WebSphere Application Server for z/OS configuration HFS mount points depends on your z/OS system layout, the nature of the application serving environments involved, and the relative importance of several factors: ease of setup, ease of maintenance, performance, recoverability, and the need for continuous availability.

- In a single z/OS system:

  If you run WebSphere Application Server for z/OS on a single z/OS system, you have a wide range of choices for a z/OS configuration HFS mount point. You may wish to put several stand-alone application servers in a single configuration HFS with a separate configuration HFS for a production server or for a Network Deployment cell. Using separate configuration HFS data sets improves performance and reliability, while using a shared configuration HFS reduces the number of application server cataloged procedures you need.

- In a multi-system z/OS sysplex with no shared HFS:

  In a multi-system sysplex with no shared HFS, each z/OS system must have its own configuration HFS data sets. For stand-alone application servers and for Network Deployment cells that do not span systems, the options are the same as for a single z/OS system.

- In a multi-system z/OS sysplex with a shared HFS:

  If your sysplex has a shared HFS, you can simply mount a large configuration HFS for the entire cell. When using the Customization Dialog, specify the common configuration HFS mount point on each system. As noted above, you should update the configuration HFS from the z/OS system hosting the deployment manager. Performance will depend on the frequency of configuration changes, and ensure you devote extra effort to tuning if this option is chosen. Alternatively, you can mount a separate configuration HFS on each system, perhaps using the system-specific HFS mounted at /&SYSNAME on each system.

Recommendations:

- On a single z/OS system:
  - Create a configuration HFS at /WebSphere/V6R0 and use it to create a stand-alone server. Place home directories for additional non-production stand-alone application servers in the same configuration HFS.
  - Create a separate configuration HFS at /WebSphere/V6R0_<cell_short_name> for each product stand-alone server or Network Deployment cell.
- On a multisystem sysplex with no shared HFS:
  - Follow the recommendations above for a single z/OS system. This will allow you to use common cataloged procedures for each cell.
  - Establish separate mount points on each system for any cell that you may need to recover on an alternate system in the sysplex.
- On a multisystem sysplex with a shared HFS:
  - Use a shared configuration HFS when performance is not an issue or when a shared HFS is required to support specific WebSphere Application Server for z/OS functions.
  - Use non-shared configuration HFS data sets when performance is an issue, or when you must avoid a single point of failure.

Transaction management by WebSphere Application Server and RRS

Many computer resources are so critical to a company's work that the integrity of these resources must be guaranteed. If changes to the data in the resources are corrupted by a hardware or software failure, human error, or a catastrophe, the computer must be able to restore the data. These critical resources are called protected resources or, sometimes, recoverable resources.

Resource recovery is the protection of the resources. Resource recovery consists of the protocols and program interfaces that allow an application program to make consistent changes to multiple protected resources.

z/OS, when requested, can coordinate changes to one or more protected resources, which can be accessed through different resource managers and reside on different systems. z/OS ensures that all changes are made or no changes are made. Resources that z/OS can protect include:

- A hierarchical database
- A relational database
- A product-specific resource

Preparing Resource Recovery Services

WebSphere Application Server for z/OS uses Resource Recovery Services (RRS) to support two-phase transaction commit. RRS must be up and running before WebSphere Application Server for z/OS servers are started. See *z/OS V1R7.0 MVS Programming: Resource Recovery,* SA22-7616 for more information. Normally, all systems in a sysplex shared a common set of RRS logs for sync point processing. If you wish to associate specific systems in a sysplex for sync point processing, you can specify a log group, which is a group of systems within a sysplex that share an RRS workload name when you start RRS. The default log group name is the sysplex name. If you specify a different log group name when you start RRS, it will coordinate sync point processing with all systems in the sysplex that use the same RRS log group name.

Implementing local transactions on z/OS involves the participation of four separate entities

The four separate entities are:

- The Transaction Coordinator (RRS)

  RRS is responsible for keeping track of when there is an active global or a local transaction associated with a given work context.

  RRS is also responsible for:

  - Ensuring that global transaction functions, such as commit and rollback, are not permitted when a local transaction is active for a given work context
  - Preventing the start of a global transaction when the current UR state is in-flight or beyond
  - Rejecting global commit functions against URs which are in local transaction mode, known as local URs, which are in any state except in-reset
  - Notifying resource managers, by driving their appropriate sync point exit routines, when a work manager or an application has ended a global transaction

- A work manager, such as WebSphere for z/OS:

  A work manager is responsible for ensuring that the correct transactional environment is established or restored before it dispatches the application. A work manager is also responsible for:

  - Ensuring that the default transaction mode is correctly set either for the address space or for each individual context
  - Enforcing transaction policy on the application or method exit and invoking End_Transaction to take the appropriate action

- A resource manager, such as DB2:

  A resource manager must manage its resources correctly regardless of whether the transaction mode is local or global. A resource manager is also responsible for:

  - Notifying RRS if it supports local transaction mode
  - Ensuring that its local transactional functions are not executed when the transaction environment for the work context is global
  - Registering its uncommitted local interest with RRS
  - Deleting its interest when the local resource is committed or rolled back via the resource manager's local transaction functions correctly using the local transaction flag in its COMMIT, BACKOUT, and COMPLETION exit routines
  - Using its own logs to recover its local transactions during restart, because local URs are not written to RRS logs

- An application

  An application defines the demarcation between local transaction mode and global transaction mode, based on the transaction mode set by a work manager when it dispatches the application.

Local and global interactions

RRS assumes that the logic of current z/OS applications depends on a global transaction, so RRS processing is based on implicit global transaction mode. In this mode, a global transaction is always active for a given application, and, if it accesses resources via an RRS-compliant resource manager, those resources are committed globally, along with any other resources the application might have accessed. In this mode, RRS performs a global commit for a transaction that ends normally, and a global rollback for an application that ends abnormally.

Consider the following scenario shown in Figure 4-11: The work manager dispatches the application, and the work manager must call Set_Environment to set up the transaction mode correctly, dispatching the application initially with local transaction mode as the default. Thus, when the application creates two connections to different databases and accesses them, it is already in local transaction mode. When the application begins a global transaction, the resource managers switches the transaction mode to global.

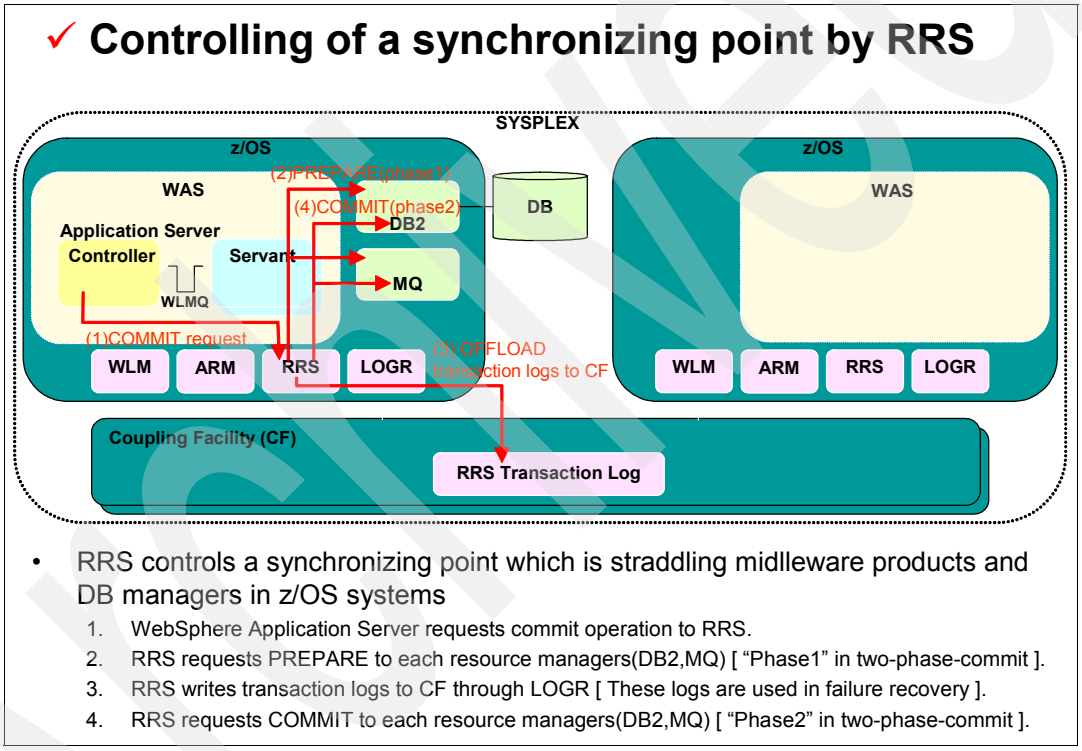


Figure 4-11 Controlling of a synchronizing point by RRS

RRS logging requirements

As shown in Figure 4-12 on page 181, Resource Recovery Services log streams use five log streams that are shared by the systems in the log group. Every MVS image that runs RRS needs access to the coupling facility, and the DASD on which the system logger log streams for its log group are defined.

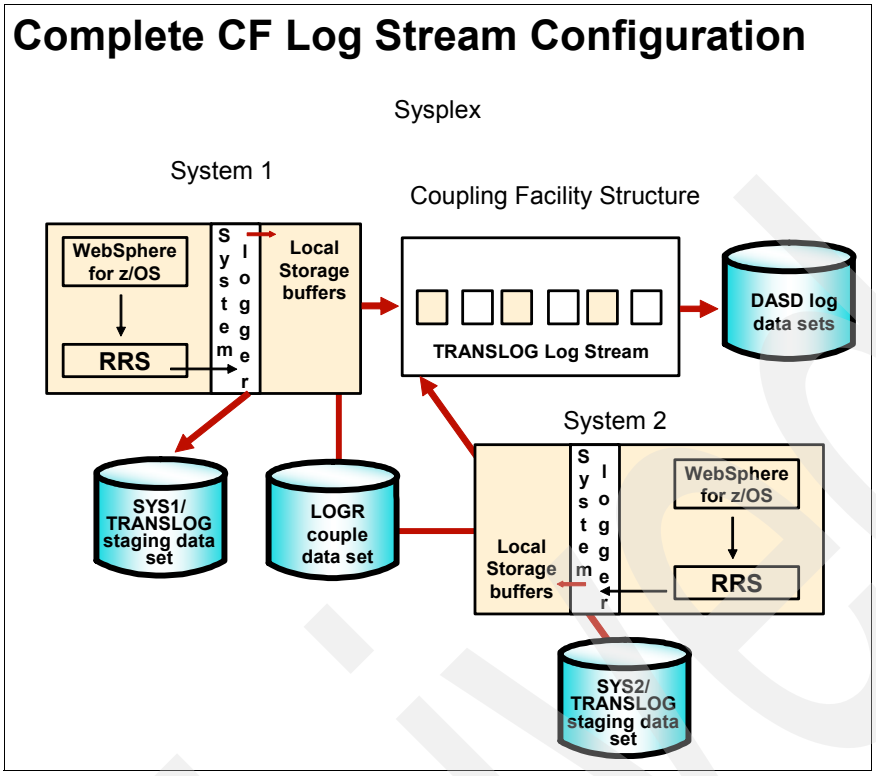


Figure 4-12 Log stream configuration

Note: You can define RRS log streams as coupling facility log streams or as DASD-only log streams. If using coupling facility log streams, the RRS images on different systems in a sysplex run independently, but share log streams to keep track of the work. If a system fails, an instance of RRS on a different system in the sysplex can use the shared logs to take over the failed system's work. Use DASD-only log streams only in either single system sysplexes with one RRS image or a sysplex in which information should not be shared among RRS images.

RRS uses five log streams that are shared by the systems in a sysplex. Every MVS image with RRS running needs access to the coupling facility and the DASD on which the system logger log streams are defined. *z/OS V1R7.0 MVS Setting Up a Sysplex*, SA22-7625 contains information about the tasks you need to perform related to logging.

Note: You may define RRS log streams as coupling facility log streams or as DASD-only log streams.

If using coupling facility log streams, the RRS images on different systems in a sysplex run independently, but share log streams to keep track of the work. If a system fails, RRS on a different system in the sysplex can use the shared logs to take over the failed system's work. Implementation of coupling facility log stream is a multisystem and multiple applications solution. It allows a system logger application to merge data from systems across the sysplex.

DASD-only logstreams, shown in Figure 4-13, should only be used either in single system sysplexes with one RRS image, or in a sysplex in which information should not be shared among RRS images.

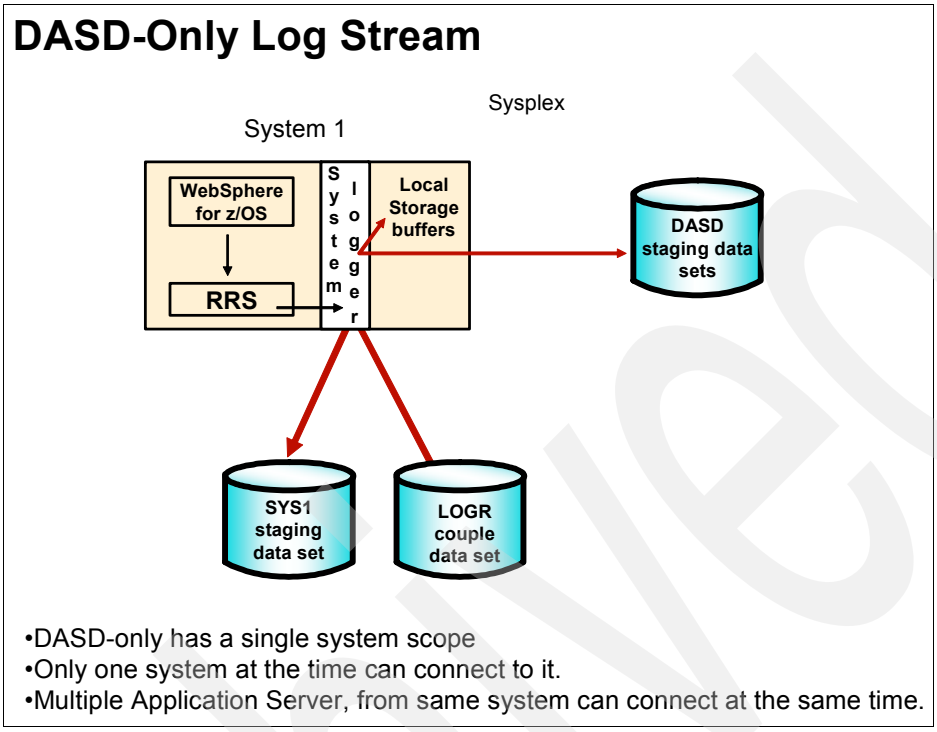


Figure 4-13 DASD-only log stream

Planning for recovery

This article helps you plan for any recovery measures you may need to take.

- Decide whether or not to implement automatic restart.
- Review the recommendations for starting a deployment manager on a different MVS image.

Automatic restart management

If you have an application that is critical for your business, you need facilities to manage failures. z/OS provides rich automation interfaces, such as automatic restart management, which you can use to detect and recover from failures. Automatic restart management, as shown in Figure 4-14 on page 183, handles the restarting of servers when failures occur. WebSphere Application Server for z/OS uses the z/OS Automatic Restart Management (ARM) to recover application servers. Each application server running on a z/OS system (including servers you create for your business applications) are automatically registered with an ARM group. Each registration uses a special element type called SYSCB, which ARM treats as restart level 3, assuring that RRS restarts before any application server.

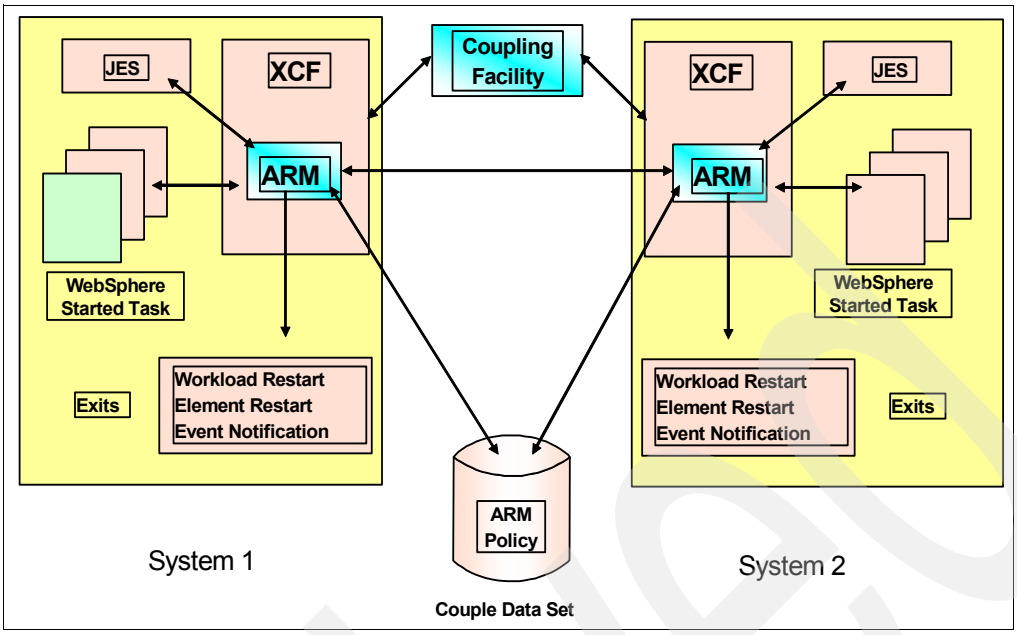


Figure 4-14 Automatic Restart Manager

Some things of note:

- If you have automatic restart management (ARM) enabled on your system, you may wish to disable ARM for the WebSphere Application Server for z/OS address spaces before you install and customize WebSphere Application Server for z/OS. During customization, job errors may cause unnecessary restarts of the WebSphere Application Server for z/OS address spaces. After installation and customization, consider enabling ARM.
- If you are ARM-enabled and you cancel or stop a server, it will restart in place using the **`armrestart`** command. It is a good idea to set up an ARM policy for your deployment manager and node agents.
- If you start the location service daemon on a system that already has one, it will terminate.
- Every other server will come up on a dynamic port unless the configuration has a fixed port. Therefore, the fixed ports must be unique in a sysplex.
- If you issue STOP, CANCEL, or MODIFY commands against server instances, be aware of how automatic restart management behaves regarding WebSphere Application Server for z/OS server instances.

Figure 4-15 shows how Automatic Restart Manager and RRS interact to fulfill the WebSphere recovery functions.

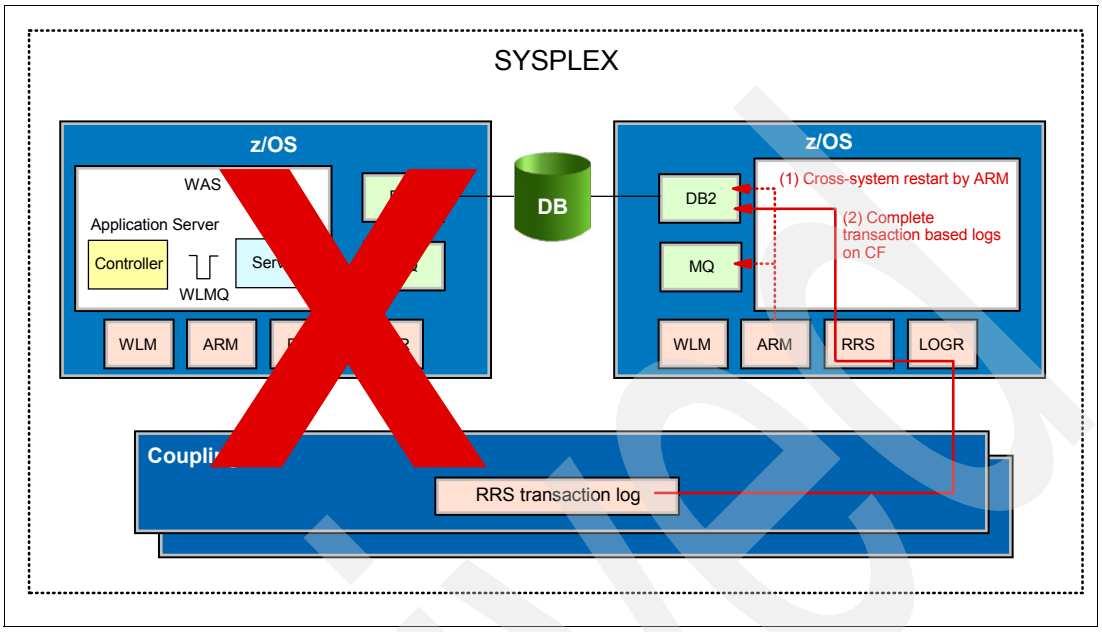


Figure 4-15 Transaction recovery by RRS

4.1.4 e-business and Parallel Sysplex CICS

Over the past 35 years, developers have created two major types of CICS applications, or assets: CICS COMMAREA programs and CICS terminal-oriented programs.

- CICS COMMAREA programs receive requests and send responses through an area of memory called the COMMunications AREA (COMMAREA). CICS programs can be written in COBOL, PL/I, C, C++, Assembler, or Java. In general, CICS COMMAREA programs are similar to subroutines in that they are unaware of how they were invoked. They are often stateless, with CICS, on behalf of the program, automatically managing the transactional scope and security context, which are typically inherited from the caller and a transaction definition.
- CICS terminal-oriented programs are sometimes known as 3270 programs because they are designed to be invoked directly from an IBM 3270 Display Station or similar buffered terminal device. Invocation usually corresponds to a single interaction in an user dialog, starting with receipt of a message from the terminal and ending with transmission of a reply message to the same device. Input data from the terminal device is carried in a data stream, which the application acquires through a RECEIVE command. After processing, an output data stream is transmitted back to the terminal device through a SEND command. Terminal-oriented programs must be capable of analyzing device-specific input data streams and building output data streams to be transmitted to the terminal. CICS also provides a service known as Basic Mapping Support (BMS), which simplifies application programming for terminals such as the IBM 3270 Display Station.

e-business access to COMMAREA programs

A best practice in CICS application design for a number of years has been to separate the key elements of the application, in particular:

- Presentation logic
- Business logic
- Data access logic

Figure 4-16 shows a transaction made up of three separate programs: a terminal-oriented program (P), a business logic program (B), and a data access logic program (D). Communication between programs (P) and (B), and between (B) and (D), uses the CICS COMMAREA.

This separation provides a framework that enables reuse of business logic and data access logic programs as subroutines within a larger application, as well as reuse with alternative implementations of presentation logic.

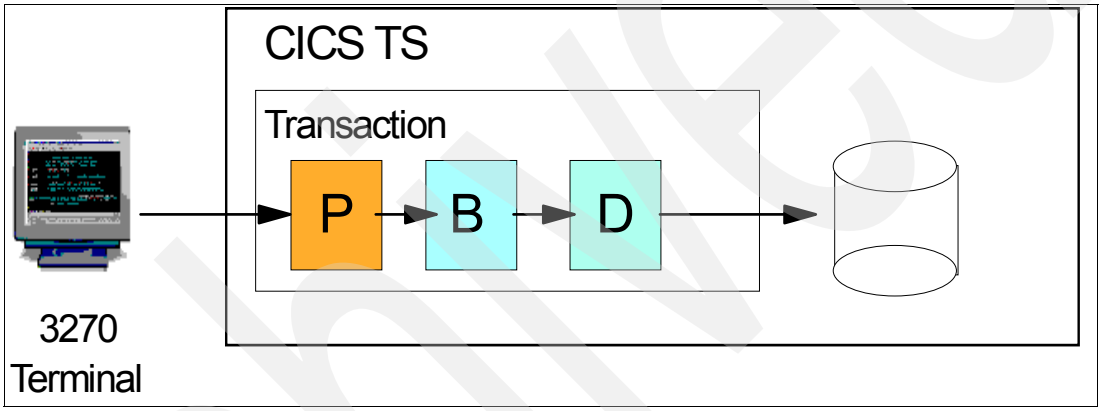


Figure 4-16 Separation of application and presentation logic in CICS

This example shows a transaction made up of three separate programs, a terminal-oriented program (P), a business logic program (B) and a data access logic program (D). Communication between programs (P) and (B), and between (B) and (D) uses either a CICS COMMAREA or CICS Channels and containers (the strategic alternative to COMMAREA provided in CICS TS Version 3.1). This separation provides a framework that enables reuse of business logic and data access logic programs as subroutines within a larger application, as well as reuse with alternative implementations of presentation logic. With this kind of separation, you can reuse the business and data logic in both Business-to-Client (B2C) and Business-to-Business (B2B) solutions.

CICS COMMAREA programs can be relatively easily enabled for access from a variety of different client applications running on a wide range of platforms. Typical e-business clients include:

- Web service requester
- Java servlet or Enterprise JavaBean (EJB™) running in a Java 2 Platform
- Enterprise Edition (J2EE) application server
- An application running in a Microsoft .NET environment
- Web browser

Adapters and connectors

In most cases, connections from an e-business client will use a combination of:

- Internal adapters
- External connectors
- Standard Internet Protocol (IP) based protocols

An adapter is simply a program that accepts a request and converts the data from an external format to the internal format used by the CICS business logic program.

Figure 4-17 shows how a terminal-oriented program (P) and internal adapters (A) can access the same CICS business logic program (B). For example, an adapter may convert a SOAP1 message to a COMMAREA format. The transport mechanism used, to invoke the adapter may by synchronous or asynchronous. An adapter can be implemented in any language supported by CICS, is often independent of the specific transport protocol used and may be hand-coded or tool generated. The COMMAREA that is used to exchange information between the existing terminal-oriented program (P) and the business logic program (B) is the same as that used to exchange information between the adaptors (A) and business logic program. In fact, it is likely that the business logic program is unaware that it is being invoked by an adapter as opposed to the terminal-oriented program.

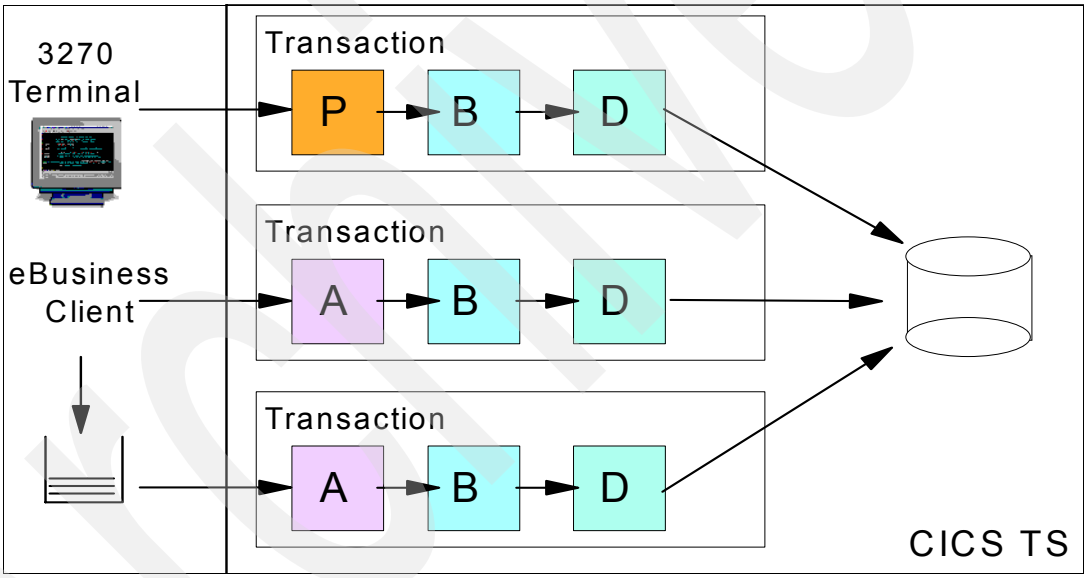


Figure 4-17 Using an internal adapter

An external connector provides a remote call interface and implements a private protocol to invoke an application running under CICS. Figure 4-18 on page 187 shows how an external connector (C) can be used by an e-business client to invoke a CICS business logic program (B). In this case, the COMMAREA is composed by the e-business client. The most well-known example of an external connector for CICS is the CICS Transaction Gateway, which implements the Common Client Interface (CCI) specified by the J2EE Connector Architecture (JCA).

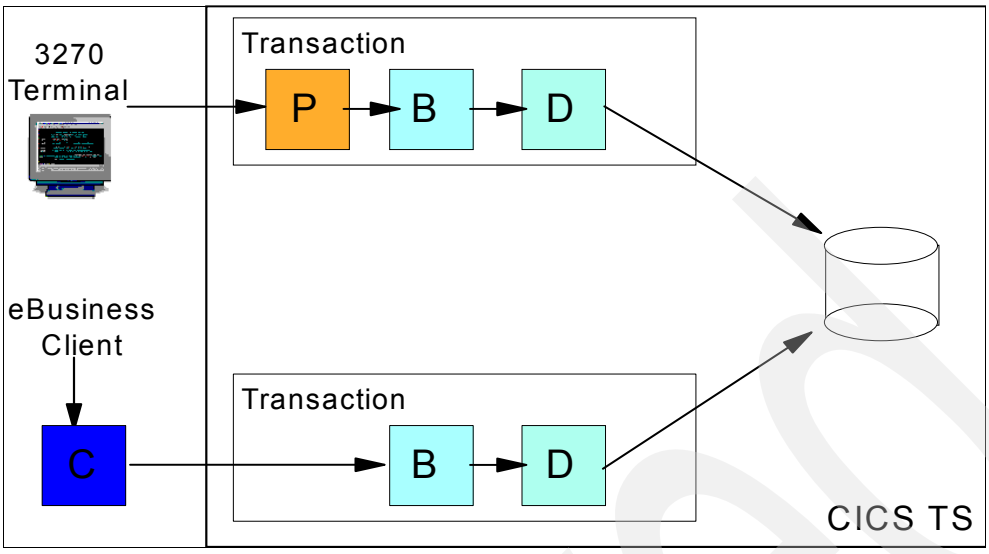


Figure 4-18 External connector

Figure 4-18 shows the simple use of an external connector to access an existing CICS COMMAREA program. In most real life situations, it is necessary to use a combination of internal adapters and external connectors.

e-business access to terminal-oriented programs

There are a number of existing CICS programs that do not have a clear separation of different application logic components. Instead, the presentation logic (P) and business logic (B) are tightly integrated within a single program for which there is only a 3270 interface (see Figure 4-19).

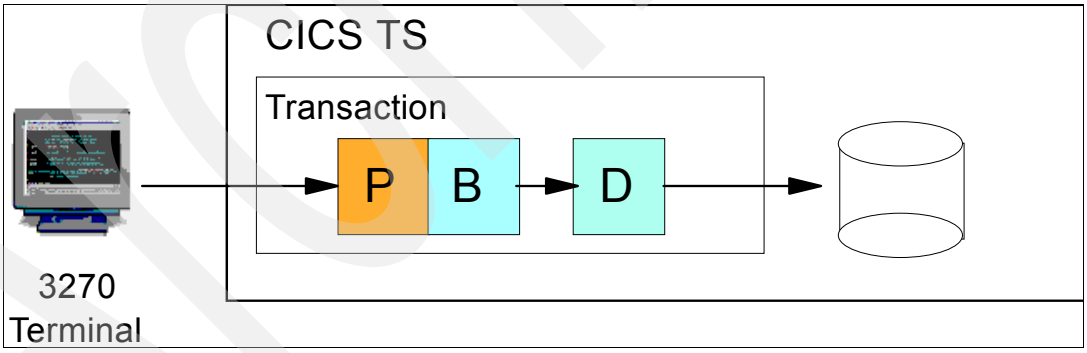


Figure 4-19 Tightly coupled presentation and business logic

To address this problem, an adaptor must convert a request from an e-business client into a data stream understood by the terminal oriented program. To help with this task, the Link3270 bridge feature of CICS Transaction Server provides a 3270 wrappering function with a linkable COMMAREA interface. No changes are required for the existing application code, and knowledge of 3270 data streams is not generally needed. Thus, the Link3270 bridge provides a programmatic interface for an important class of terminal-oriented programs, enabling them to be reused without resorting to less efficient and more fragile screen scraping. Figure 4-20 on page 188 shows how a bridge adapter can be used to convert a request from an e-business client into the bridge vectors that are used by the Link3270 bridge to invoke a terminal oriented program. The terminal oriented program is not aware that it is being invoked by an e-business client rather than a 3270 terminal.

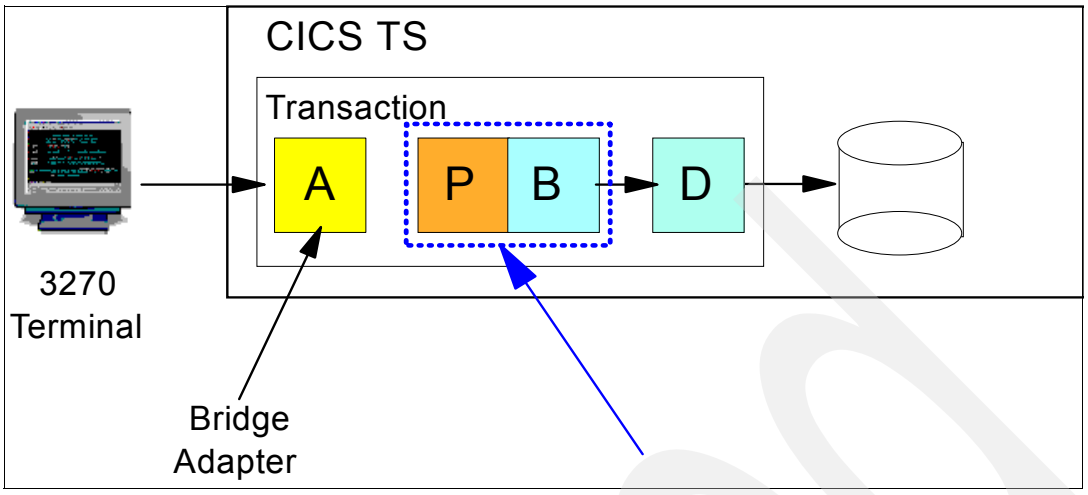


Figure 4-20 Bridge adapter using the Link3270 bridge

IBM has implemented the bridge adapter concept in the IBM WebSphere MQ Integrator Agent for CICS (MQIAC), which enables middle-tier applications to access existing back-end applications.

CICS programs are typically grouped into application suites, or components, for performing a common set of business actions. Identifying the CICS programs that provide flexible public interfaces and understanding these interfaces is the first key step in reuse. The next is to decide the best access options to support your e-business solution.

CICS e-business technologies

This section introduces the major technologies that can be used to build CICS e-business solutions. Solutions based on these architectures benefit from the comprehensive set of development tooling, which is provided to help in the generation of applications.

If you are addressing the large scale architecture of an e-business solution, we recommend that you consult IBM Patterns for e-business. The Patterns approach is based on a set of layered assets that can be exploited by any existing development methodology. These layered assets are structured in a way that each level of detail builds on the last. These assets include:

- Business patterns that identify the interaction between users, businesses, and data.
- Integration patterns that tie multiple Business patterns together when a solution cannot be provided based on a single Business pattern.
- Composite patterns that represent commonly occurring combinations of Business patterns and Integration patterns.
- Application patterns that provide a conceptual layout describing how the application components and data within a Business pattern or Integration pattern interact.
- Runtime patterns that define the logical middleware structure supporting an Application pattern. Runtime patterns depict the major middleware nodes, their roles, and the interfaces between these nodes.
- Product mappings that identify proven and tested software implementations for each Runtime pattern.
- Best-practice guidelines for design, development, deployment, and management of e-business applications.

The main purpose of the Patterns for e-business is to capture e-business solutions that have been tested and proven. The information captured is thought to fit the majority of situations. By referring to the patterns, you can save valuable time and effort in the design of your e-business solutions. Many of the existing patterns and supporting documentation are applicable to building a CICS e-business solution, in particular:

- *Patterns: Connecting Self-Service Applications to the Enterprise*, SG24-6572
- *Patterns on z/OS: Connecting Self-Service Applications to the Enterprise*, SG24-6827
- *Patterns: Direct Connections for Intra- and Inter-enterprise*, SG24-6933
- *Patterns: Self-Service Application Solutions Using WebSphere for z/OS V5*, SG24-7092
- *Patterns: Service-Oriented Architecture and Web Services*, SG24-6303

For more general information about the Patterns for e-business, visit the Patterns Web site:

`http://www.ibm.com/developerWorks/patterns`

CICS Transaction Gateway

Historically, the CICS Transaction Gateway (CICS TG) is the standard way to access CICS applications from a Java environment; the CICS TG provides a comprehensive set of Java classes to do this. If you are writing new CICS TG applications, we highly recommend that you only use the J2EE Connector Architecture resource adapters (see below) and the CCI programming interface for developing new applications rather than the other ECI and EPI Java classes provided with the CICS TG.

The J2EE connector Architecture (JCA) defines a standard for connecting from the J2EE platform to heterogeneous Enterprise Information Systems. CICS is an example of an Enterprise Information System (EIS). The JCA enables an EIS vendor to provide a standard *resource adapter,* which is a middle-tier connector (that is, a *logical* middle-tier, as the resource adapter runs on the same machine as the J2EE application server) between a Java application and an EIS, which permits the Java application to connect to the EIS. Remember, the IBM J2EE application server is WebSphere Application Server.

The CICS Transaction Gateway is the preferred implementation for JCA connectors to access all CICS servers from WebSphere Application Server. CICS TG provides two resource adapters, one for ECI (External Call Interface) and one for EPI (External Presentation Interface).

ECI resource adapter
: This adapter is required for making calls to CICS COMMAREA-based programs.

EPI resource adapter
: This adapter is required for making calls to CICS terminal-oriented programs. It provides a record-based interface to terminal-oriented CICS applications.

The ECI resource adapter is the simplest to use and the most commonly used CICS TG resource adapter. Figure 4-21 shows how the CICS TG enables e-business access to a CICS business logic program.

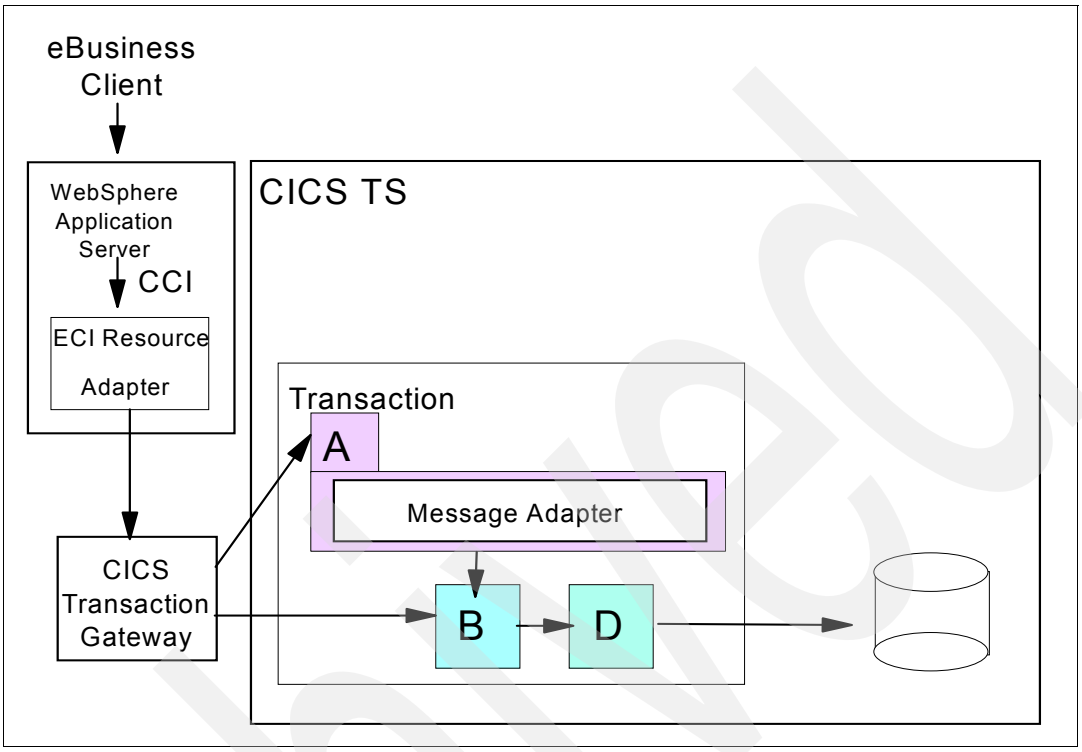


Figure 4-21 CICS Transaction Gateway

A J2EE application uses the CCI programming interface to invoke the CICS ECI resource adapter. The CICS TG ECI classes are packaged with the ECI resource adapter and are used to pass the application request to the CICS TG. The J2EE application can invoke the CICS business logic program (B) directly if no message transformation is required. In this case, Rational® Application Developer can be used to create a Java bean to represent a COMMAREA formatted as COBOL types, with Java methods for getting and setting fields.

A message adapter in CICS is required only if the message is to be transformed, for example, the request is in XML and the CICS business logic program requires a COBOL record format. The length of the message is subject to the normal CICS COMMAREA message length limitation of 32,500 bytes.

The CICS TG is the preferred implementation for JCA connectors to access all CICS servers from WebSphere Application Server, for e-business applications that require a high performing, secure, and scalable access option with tight integration to existing CICS applications. The CICS TG benefits from ease of installation, flexible configuration options, and requires minimal changes to CICS and in most cases no changes to existing CICS applications.

Because the CICS TG is a multi-platform product, it can be deployed in many different topologies. Figure 4-22 on page 191 shows the most common topologies used. We are going to take a closer look at topology 3, where both CICS TG and WebSphere Application Server are both executing on z/OS.

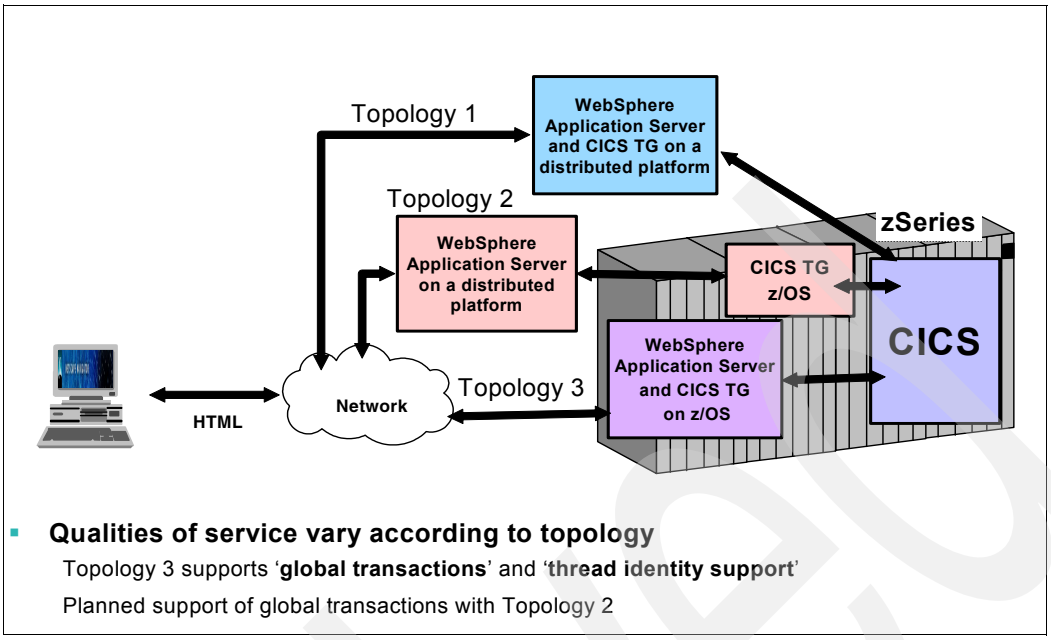


Figure 4-22 Common topologies for using CICS TG with WebSphere Application Server

In a zSeries topology, WebSphere Application Server can be deployed on either a z/OS system or on a Linux operating system. The qualities of service of the Linux topology are largely the same as for those on distributed platforms. Deploying both WebSphere Application Server and CICS TG on z/OS offers some significant benefits in terms of improved qualities of service.

The communication between the CICS TG and the target CICS TS region is the External CICS Interface (EXCI). The EXCI is a z/OS specific application programming interface that enables a non-CICS program (a client program) running in z/OS to call a program (a server program) running in a CICS region and to pass and receive data by means of a communications area. The CICS application program is invoked as though linked-to by another CICS application program. If the client and server are in the same z/OS LPAR, the multiregion operation (MRO) facility of CICS interregion communication (IRC) facility is used by the EXCI. If they are in different LPARs, XCF/MRO provides the underlying communication.

The most common z/OS configuration makes use of a local CICS TG. On z/OS, this results in a direct cross-memory EXCI connection between the application server and CICS.
Figure 4-23 shows a J2EE application deployed to WebSphere Application Server Version 5 for z/OS using a local CICS TG. The highest qualities of service are achieved when this configuration is used. This is the only topology that provides thread identity support and two phase-commit transactions.

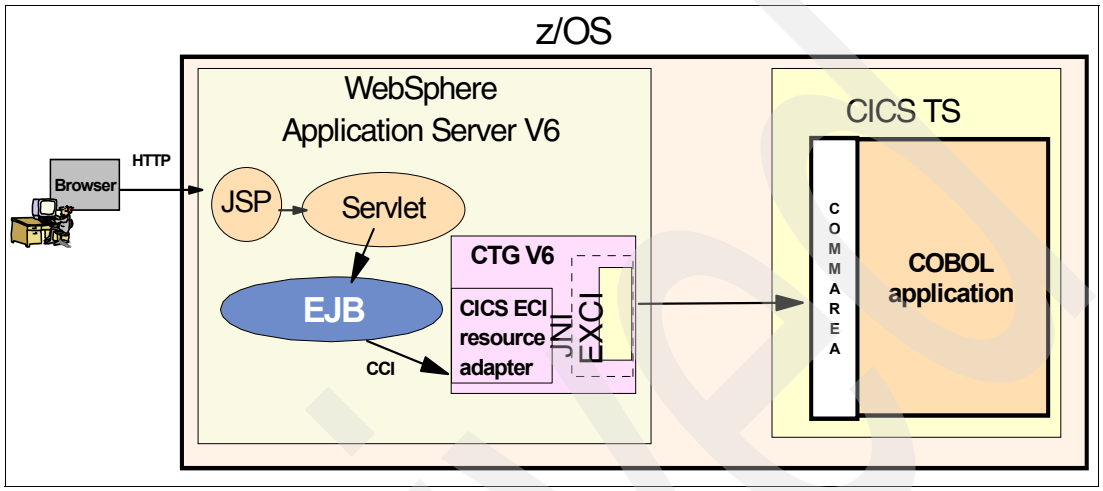


Figure 4-23 WebSphere Application Server and CICS TG deployed on z/OS

- The unique thread identity support in WebSphere Application Server for z/OS allows the application server to automatically pass the user ID of the thread (for example, the caller's identity) to CICS when using the ECI resource adapter. This satisfies a common end-to-end security requirement of automatically propagating the authenticated caller's user ID from WebSphere Application Server to CICS.

  The two-phase commit capability for this topology is provided through MVS resource recovery services (RRS), an integral part of z/OS. RRS acts as the external transaction coordinator for z/OS in managing the transaction scope between WebSphere Application Server, CICS, and other z/OS subsystems, including IMS/TM, IMS/DB, IBM DB2, and WebSphere MQ systems.
- A second important topology for z/OS is shown in Figure 4-24.

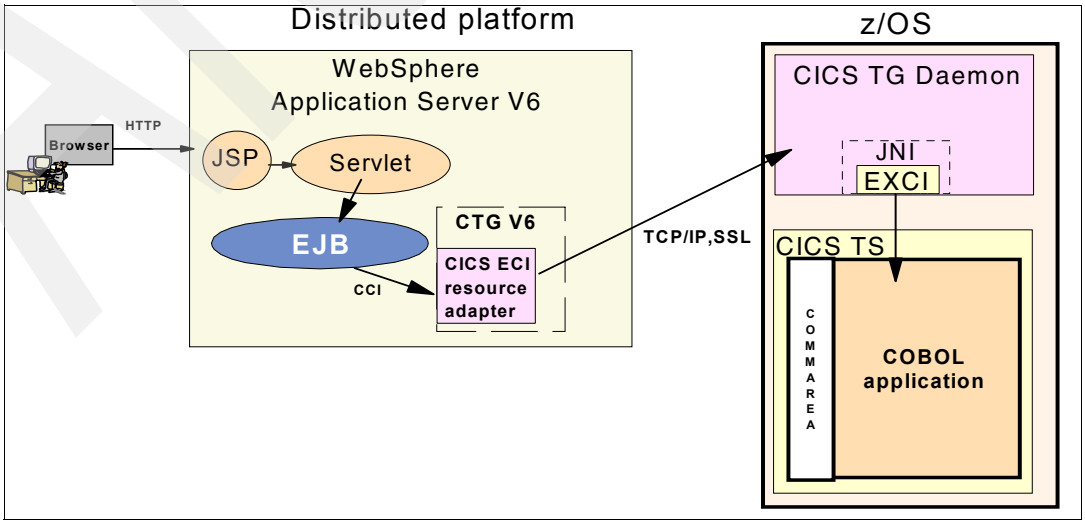


Figure 4-24 CICS TG daemon on z/OS

Here we can see that the connection to CICS TS is made from a distributed platform (which could be a WebSphere Application Server running in Linux for zSeries) via a CICS TG daemon. The job of the CICS TG daemon is to receive the client ECI request (the z/OS daemon only supports ECI), unpackage the client request, invoke the target CICS TS program using the EXCI, and return the results. Connectivity to the CICS TG Daemon can be made over:

- TCP/IP
- APPC
- TCP62 (APPC flows over a TCP/IP network)

Again, the EXCI is used to provide communication between the CICS TG and the target CICS region.

CICS Web services

If you are considering implementing Web services in CICS TS, it is *highly recommended* that you work with CICS TS Version 3.1, which contains a lot of new Web services-related functions that are not available in earlier releases. Prior to CICS Version 3.1, some SOAP for CICS functionality was introduced as supportpac CA1M, and the supportpac was replaced by by an optional feature for CICS Transaction Server V2.2 or V2.3.

CICS Web Services support allows CICS to operate as a service requester (that is, client) or service provider (that is, server) within a Web services architecture. The Web services standards allow service requesters and service providers to exchange XML based messages, securely and reliably, independent of platform, or application language. This allows application developers to rapidly build open standards based applications independently of the CICS business logic program they will interact with.

Figure 4-25 shows the structure of a CICS service provider implementation. You may choose to invoke the business and data logic directly from your service provider pipeline, in which case you do not have to write any new code, or you may choose to write an adapter program, which will provide additional processes (for example, extra mapping of data or error handling not done by the tooling), which will then call the business function with the expected commarea.

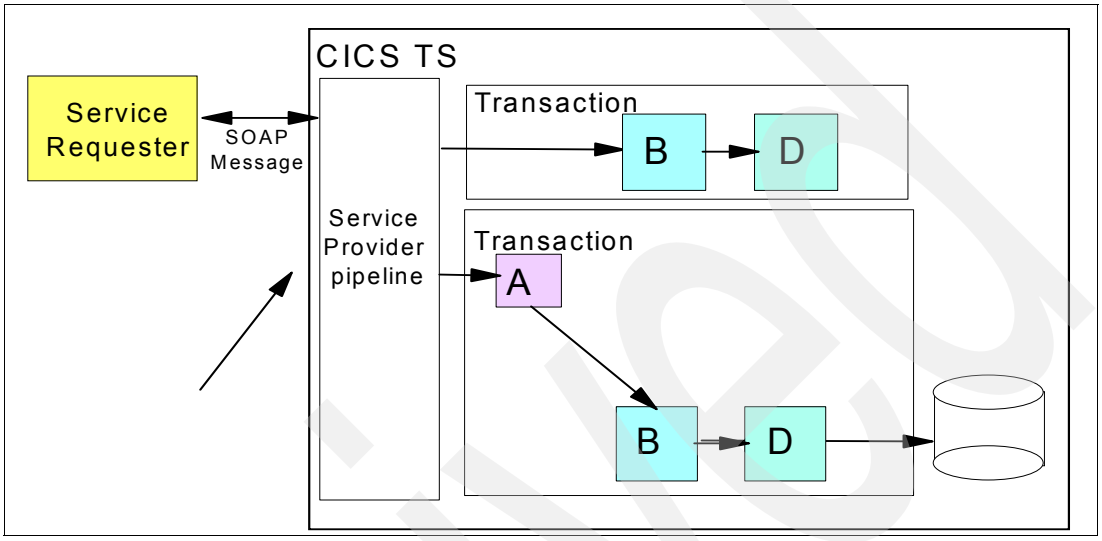


Figure 4-25 CICS TS service provider implementation

CICS Web services support allows CICS business logic programs to be deployed (unmodified in most cases) as service providers, or you can develop new CICS applications that can act as service requesters; these interactions are based on industry-standard Web services implementation. The service provider and service requester functions within CICS are run as pipelines, using the new Channel/Container APIs in CICS TS Version 3.1 (the SOAP for CICS optional feature uses CICS Business Transaction Services (BTS) containers).

The Web services support in CICS TS relies heavily on the XML parsing technology built into Enterprise COBOL Version 3 and Enterprise PL/I Version 3. Both provide built-in XML parsing functionality for the PL/I and COBOL languages, which can simplify the development of parsing code.

A lot of emphasis has been placed on simplifying application development for CICS Web Services applications. There is a Web services assistant supplied with CICS TS Version 3.1 that will generate the runtime environment for a CICS TS service provider application, using only the commarea of the target business application as input. Alternatively, WebSphere Developer for zSeries (based on IBM Rational Application Developer) contains wizards that can do the same thing.

The SOAP messages on which the Web services standards are based are not bound to any specific transports. CICS supports two transports:

- Messaging (WebSphere MQ)
- HTTP

Different pipelines are provided for the two different transports.

There is an alternative way to deploy existing CICS business commarea applications as Web services, and that is to use the Web Services support provided by WebSphere Application Server; if you choose this method, then you will be using the CTG resource adapter to

connect to the CICS business function encapsulated in commarea-style applications (see “CICS Transaction Gateway” on page 189).

CICS Web support (CWS)

CICS Web support (CWS) is a set of resources supplied with CICS TS for z/OS that provide CICS with a subset of the HTTP serving functions found in a general-purpose Web server. This allows CICS applications to be invoked by and reply to HTTP requests (usually but not always sent by a Web browser). A summary of how a CICS application can be Web-enabled using the CWS is illustrated in Figure 4-26.

The Web browser (HTTP client) attaches directly to CICS using the sockets interface. Using the workload balancing features in TCP/IP, the client connect request could in fact be routed to any CICS region in the CICSplex that has the Sockets listener enabled, thereby shielding the client from the unavailability of a CICS region.

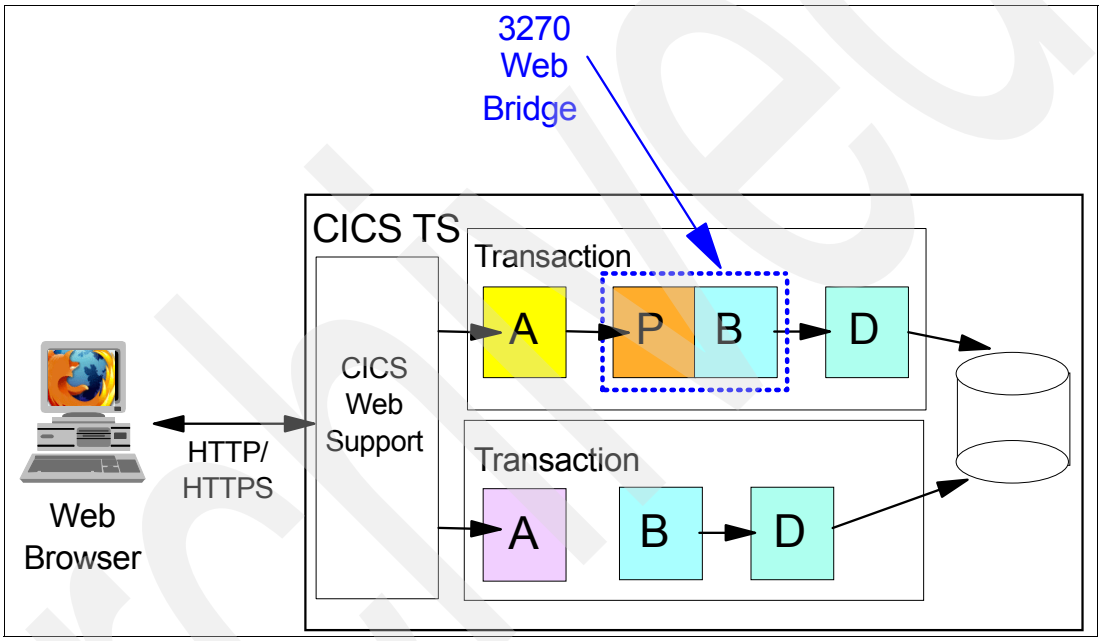


Figure 4-26 CICS Web support

To invoke a commarea style application, you need to write an adapter that will contain your *Web-aware* logic, taking the HTTP request received from the Web browser, and mapping it to the commarea expected by the business function; once the business function has executed, your adapter will need to build the HTTP response to be returned. CICS provides some very powerful APIs to simplify the parsing and building of HTTP requests and responses. To invoke a 3270 transaction, the facilities of the 3270 bridge are used. The transaction remains unchanged and the 3270 output is converted to HTML.

While CICS TS is not a *traditional* HTTP server, direct access to CICS TS over HTTP or HTTPS can be very useful both for business and for CICS systems management purposes. The Web User Interface that is used to manage your CICSPlex SM environment is actually a sophisticated CICS Web Support application.

Further information about CICS Web support may be found in *CICS Transaction Server for OS/390 V1.3 CICS Internet Guide,* SC34-5445, and the IBM Redbook *CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge*, SG24-5480.

CICS Enterprise JavaBean support

Enterprise JavaBeans™ (EJBs) are reusable Java server components written to Sun Microsystem's Enterprise JavaBeans (EJB) specification. They can be used in an application server called an Enterprise Java Server (EJS). The EJS provides interfaces and services defined by the EJB specification.

Enterprise beans execute within a container provided by the EJS. They are located by looking up their names in a name server using the Java Naming and Directory Interface™ (JNDI). The EJB container provides services such as transaction support, persistence, security, and concurrency.

CICS TS Version 2 and later provides partial support for Version 1.1 of the EJB specification, in that it supports one kind of EJB: *session beans*. The EJB container within CICS provides the services required by enterprise beans running within the CICS EJB server.

A session bean is instantiated by a client and represents a single conversation with the client. In most cases, this conversation only lasts as long as the client is active. From this point of view, the session bean is very similar to a traditional CICS pseudo-conversational transaction.

A session bean performs business actions such as transferring data, or performing calculations on behalf of the client. These business actions can be transactional or non-transactional. If the actions are transactional, the session bean can manage its transaction using the Object Transaction Service (OTS), or it can use the container-managed transaction services provided by the EJB container.

The CICS EJB server, like any other EJB server that complies with EJB specification, provides support for stateful as well as stateless session beans. The option as to whether an enterprise bean is to be deployed as a stateful or stateless session bean is specified in its deployment descriptor.

Other options

There are a number of other methods of accessing CICS from the Web, including WebSphere Host On-Demand, WebSphere Host Access Transformation Services, WebSphere MQSeries, Sockets, Templates, and others. However, these methods are heavily dependent on products other than CICS, so are not discussed in this section.

A redbook can help you determine the best way of leveraging your existing CICS applications and knowledge in an e-business world is available on the Internet at:

`http://www.redbooks.ibm.com/abstracts/sg245466.html?Open`

4.1.5 DB2

Of the three subsystems we discuss in this section, DB2 is the simplest from an e-business access perspective. If you have DB2 and wish to make DB2 data available via the Web, you have four basic options:

- Use CICS.
- Use IMS.
- Use some form of Java programming.
- Use Net.Data®.
- Use a C/C++ CGI/GWAPI program with WebSphere.

If you are going to use CICS or IMS, the e-business considerations for those subsystems are the same regardless of whether the data they are accessing is in DB2, IMS DB, or VSAM. And, because we cover those subsystems in CICS in 4.2.2, "CICS transactions in a Parallel Sysplex" on page 209 and IMS in 4.3.2, "IMS DB data sharing" on page 236, we will not

discuss them again in this section. Instead, we will concentrate on the Java and Net.Data options.

Both the Net.Data and Java options provide flexibility as to where the *front-end* runs – it can either be in the same OS/390 as the DB2 subsystem resides on, or it can be on another system (S/390 or otherwise). If the front end runs on the same OS/390 as the DB2 subsystem you wish to access, then local DB2 access is used to communicate with DB2. If the front end runs on another system, then DRDA must be used to communicate with DB2. Figure 4-27 shows these two options diagrammatically.

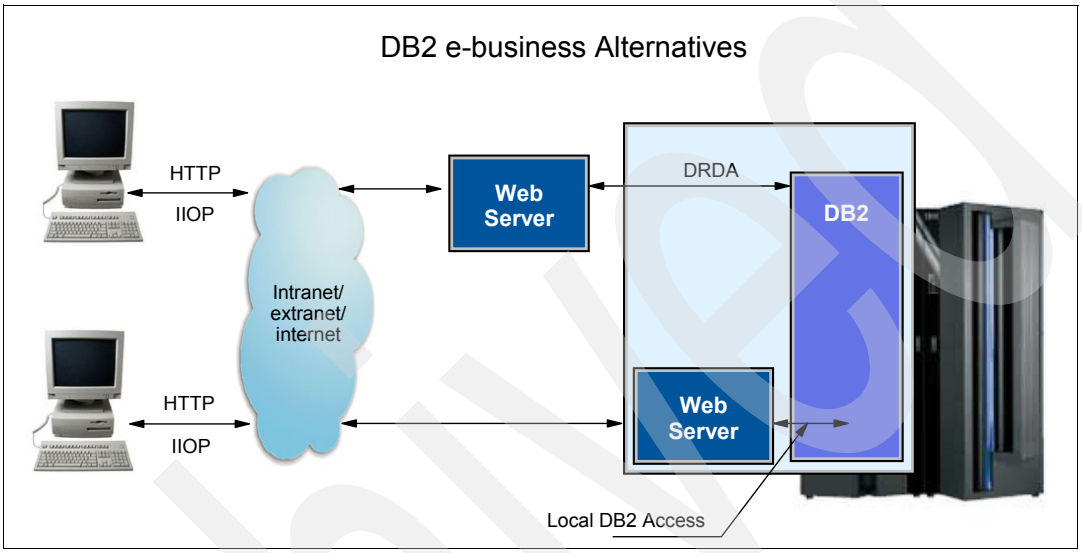


Figure 4-27 DB2 access options

Other options are:

- If your front end is not on the same system as the target DB2, you must use DRDA to communicate with DB2. DRDA can use either LU6.2 or TCP/IP to communicate with DB2. If you are using TCP/IP, you can use one of the TCP/IP workload balancing mechanisms to distribute the requests across the members of the DB2 data sharing group.
- If you are using LU6.2, you *can* use the VTAM GR support to achieve the same thing; however, this is not the most efficient option. Web Servers typically have many sessions, and VTAM GR would funnel all of those into the one DB2 subsystem. A better choice is to use the WLM-based workload balancing for SNA sessions, which is built into both DB2 for OS/390 (with DDF) and DB2 Connect™.
- If you are using local access to communicate with DB2, there must be a DB2 subsystem on the same system as the front end. This DB2 does not necessarily have to be the one that you will get the data from; it is possible to have the local DB2 use a DRDA connection to get your data from a remote DB2 subsystem.
- If you are interested in the highest levels of availability, one option to consider is to place two DB2 subsystems that are in the same data sharing group on the OS/390 systems that are running your front ends. In the normal course of events, whichever subsystem is placed first in IEFSSN will be the one that receives all DB2 calls that specify the Group Attach Name. However, if that DB2 is stopped, perhaps to apply maintenance, all calls using the Group Attach Name will get routed to the remaining DB2 subsystem. When defining the initialization parameters for Net.Data or the Web Server, you have the option of specifying the DB2 subsystem name or the Group Attach Name. Specifying the Group Attach Name allows you to use this two-DB2 setup. This does *not* provide any workload

balancing; however, it does provide improved availability if you ensure that only one of the two DB2 subsystems is ever stopped at a time.

DB2 Web components

There are a number of terms and components that will be used in most of the DB2 Web connection options. Before we get into discussing the options, we will briefly describe each of these components:

JDBC Java Database Connectivity (JDBC) allows the use of dynamic SQL statements in a Java program, giving you the ability to access a relational database (in this case, DB2) from a Java program. JDBC is actually a driver that must be available when the Java program runs.

SQLJ Structured Query Language for Java (SQLJ) is basically an enhancement to JDBC that permits the use of static SQL from within a Java program. As with JDBC, SQLJ is a driver that must be available when the Java program executes.

Net.Data Net.Data is a product that provides a set of macros that can be used to access DB2 data. Net.Data programs consist of a set of Net.Data macros, and run under the control of a Web Server. Net.Data is included as part of DB2 for OS/390, and is a replacement for the earlier DB2 www Connection.

DB2 Connect DB2 Connect is the IBM product that provides an interface to DRDA (there are similar products available from other vendors). DB2 Connect can be used as the middleware to connect any distributed server to DB2. There is a DB2 Connect Personal Edition, which is designed to be installed on the client workstation and permit access to DB2 directly from that workstation. There is also a DB2 Connect Enterprise Edition, which is designed to run on a distributed server platform. In this case, clients communicate with the server using JDBC or SQLJ, and the server communicates with DB2 using DB2 Connect to translate the SQLJ and JDBC calls into DRDA calls.

DB2 Connect V6 introduced specific sysplex support. If WLM is running in Goal mode, and the SYSPLEX installation option is specified in DB2 Connect, WLM will return information to DB2 Connect about the members of the DB2 data sharing group, and the available capacity on the systems each of the members are running on. DB2 Connect then uses this information when deciding which DB2 member to send a given request to. More information about this aspect of DB2 Connect can be found in *DB2 Connect Quick Beginnings for UNIX*, available on the Internet at:
`ftp://ftp.software.ibm.com/ps/products/db2/info/vr6/pdf/letter/db2ixe60.pdf`

DRDA Distributed Relational Data Architecture (DRDA) is an architecture that permits external access to a relational database. In the context of what we are discussing here, DRDA would normally be used as a way for distributed platforms to communicate directly with DB2. DRDA also supports communication between two DB2 for OS/390 subsystems that are not in the same data sharing group.

DB2 Java options

JDBC and SQLJ are used to include SQL statements in Java programs. The decision regarding where to place the Java program (as an applet on the client platform, as a servlet on a distributed platform, or as a servlet running under the Web Server on OS/390) is determined more by performance considerations than functionality.

It is possible to include the JDBC or SQLJ code in an applet; however, the size of the resulting file (which would have to be downloaded to the client for every invocation) would probably not provide very good performance. Also, the client would have to have DB2 Connect or a similar product installed; this might be an acceptable situation for an intranet application, but not one that will be used for clients coming in from the Internet.

The solution that is recommended to most installations is to run the Java program as a servlet on OS/390, using VisualAge® for Java to compile the Java programs for improved performance.

Net.Data

The other option is to use Net.Data. Net.Data can be run in either a distributed Web Server or on a Web Server on OS/390. To invoke the Net.Data program, the URL specified by the client will include the name of the Net.Data program. When the client call reaches the Web Server, the Net.Data program will be invoked. If the Web Server is running on a distributed platform, DB2 Connect would then be used to communicate with DB2 on OS/390.

There is no difference in functionality, and once again, your decision is likely to be driven by performance and system management considerations.

Other options

There are numerous other options for accessing DB2 data from the Web. Some of the ones that are provided by IBM are:

- CGI
- GWAPI
- QMF™ for Windows
- Host Publisher
- DB2 Forms

There are also a number of products available from other vendors.

Like Net.Data and Java, CGI and GWAPI programs can run in either a Web Server that is local to DB2, or in a distributed Web Server. However, the use of CGI and GWAPI is no longer considered strategic; new applications should be developed using Java or Net.Data instead.

QMF for Windows, Host Publisher, and DB2 Forms are all aimed at specific application requirements, and would not be as generic as Java or Net.Data.

Further information

For more information about the use of DB2 with the Web, refer to the IBM Redbooks *Accessing DB2 for OS/390 Data from the World Wide Web*, SG24-5273 and *WOW! DRDA Supports TCP/IP: DB2 Server for OS/390 and DB2 Universal Database*, SG24-2212.

4.1.6 IMS

Just as for CICS and DB2, there are numerous options available that allow existing IMS applications and data to be included as a viable part of the e-business world. These options are in a state of evolution and include connectivity solutions for both SNA and TCP/IP networks.

Once the request reaches the IMS environment to be processed, IMS additionally provides a variety of Parallel Sysplex capabilities that answer the requirements for performance, capacity, and availability.

To better understand this whole environment for IMS, this section provides a brief overview of:

- Communication architectures and solutions for IMS
- IMS workload balancing methods
- Parallel Sysplex solutions

IMS communication architectures and solutions

IMS can be accessed across SNA and TCP/IP networks using a variety of communication protocols and architectures. It is worth noting that regardless of which option is chosen, there is no corresponding requirement to modify the IMS transaction. The same transaction can be invoked concurrently from different environments. The IMS solutions were created to allow the transactions to be independent from, and unaware of, the network interfaces. IMS Communication Options, as shown in Figure 4-28, show some of the different options.

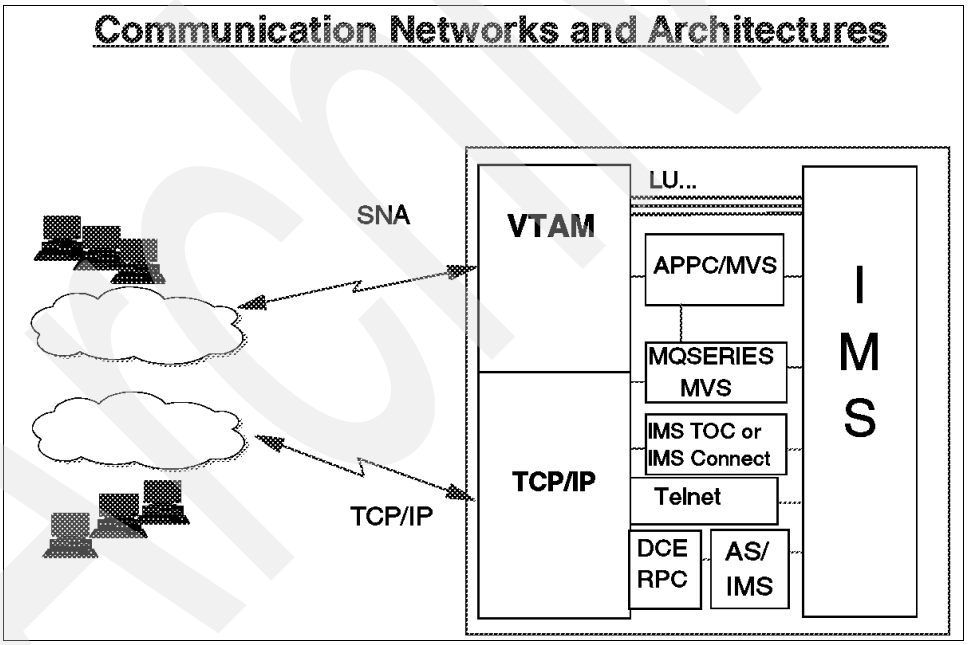


Figure 4-28 IMS Communication Options

The traditional SNA communication protocols (LU1, LU2, and so on) can be used to access IMS applications, since IMS is a VTAM application. The most commonly used SNA protocol in the e-business world is LU 6.2, with its associated implementation called Advanced Program to Program Communications (APPC). APPC provides a standard, connection-oriented, Application Programming Interface (API) for one program to directly communicate with another, even when separated by a network.

Access to IMS across a TCP/IP network falls into three primary categories:

- Remote login using Telnet and 3270 emulation
- The Remote Procedure Call (RPC) interface using Distributed Computing Environment (DCE) services
- Using the standard TCP/IP connection-oriented API called *sockets*

> **Note:** The sockets API provides a program-to-program communication interface for the TCP/IP environment, much as APPC provides for SNA networks. Although IMS is not a TCP/IP application, accessibility across all three categories is fully supported.

- For remote login, the Telnet server actually creates an LU2 session into IMS through VTAM.
- For DCE/RPC, a product called Application Server/IMS communicates with remote DCE clients and interfaces with IMS.
- For sockets support, IMS provides a connector (called IMS TOC in IMS/ESA® Version 6, and IMS Connect in IMS/ESA Version 7 – we will refer to them generically as IMS TOC/IMS Connect in this document) that provides the conversion between sockets-based messages and IMS messages.

MQSeries on MVS provides a bridge that retrieves the inbound message from the MQ queue, sends it to the IMS message queue, and then receives replies from IMS and sends them to the appropriate MQ outbound queue. MQSeries hides the network specifics from the application programs and supports deployment across both SNA and TCP/IP networks.

Any of these protocols and associated products can be used when developing e-business applications that access IMS. To assist in the development and implementation effort, several options and solutions are currently available. These include:

- IMS WWW templates (IWT)
- Component Broker support
- IMS TOC/IMS Connect
- The IMS Client for Java
- The IMS Connector for Java
- Open Database Access (ODBA)
- Classic Connect
- Host on Demand
- Net.Commerce™

A brief overview of each of these options follows.

Accessing IMS transactions using APPC

The *IMS WWW templates* are C programs, written and supplied by IBM, that provide a flexible, interactive, and portable interface to IMS transactions using APPC. The templates function as a Web Server program that access IMS transactions and maps the results into an attractive page for the browser users. They are provided with a choice of Web application interfaces:

- Common Gateway Interface (CGI)
- Internet Connection API (ICAPI), which is a higher-performing interface than CGI and is unique to IBM Web Servers
- The more recent Java servlet wrapper

These programs can be used as *canned* solutions or as models when creating user-written Web Server programs.

Component Broker is the IBM implementation of Common Object Request Broker Architecture (CORBA) and provides the capability for applications to access transactions and data using standard object technology. An application adaptor for IMS is provided to facilitate access to IMS transactions. Remote access to IMS uses APPC. When Component Broker and IMS reside in the same OS/390 image, access is provided using a direct interface to the Open Transaction Manager Access (OTMA) component of IMS. If Component Broker is not on the same OS/390 image as IMS, APPC will be used as the interface.

Open Transaction Manager Access (OTMA)

It is worth noting at this point a function called Open Transaction Manager Access (OTMA), which was introduced in IMS V5. OTMA supports a high-performance interface between IMS and any OS/390 application for the exchange of transaction messages and commands. The OS/390 applications are expected to use MVS XCF services and to prepare the messages in a format IMS understands. Messages that are received by the OTMA component in IMS are processed in the same manner as any other transaction, that is, the message queue services, logging, and other components of IMS are used as required by the transaction. Examples of applications that use OTMA include:

- The MQSeries Bridge for IMS
- AS/IMS, which supports DCE/RPC access
- IMS TOC/IMS Connect, which supports TCP/IP sockets access

Due to the use of XCF as the communications vehicle, OTMA can be used even if IMS is not active in the same z/OS image as the client.

Accessing IMS transactions using TCP/IP sockets

IMS TOC/IMS Connect provides the architectural framework that supports connectivity to IMS from any TCP/IP sockets application. The IMS TCP/IP OTMA Connection (IMS TOC, also known as ITOC) product code can be obtained from the IMS Web site, rather than on standard product tapes. At the time of writing, IMS TOC is scheduled to run out of service in September 2001. IMS Connect, which is a separately priced feature of IMS V7 Transaction Manager, replaces IMS TOC and provides enhanced functionality.

IMS TOC/IMS Connect provides services to translate (including ASCII-to-EBCDIC, if needed) data sent from the TCP/IP client into a format understood by the IMS OTMA component. Optionally, the data is also validated (user ID and password/passticket checks). Reply messages are also translated into a format understood by the TCP/IP sockets program.

It is important to note that IMS TOC/IMS Connect can interface with multiple IMS systems, even if an IMS resides on a different MVS in the sysplex. Figure 4-29 on page 203 shows the role of IMS TOC/IMS Connect.

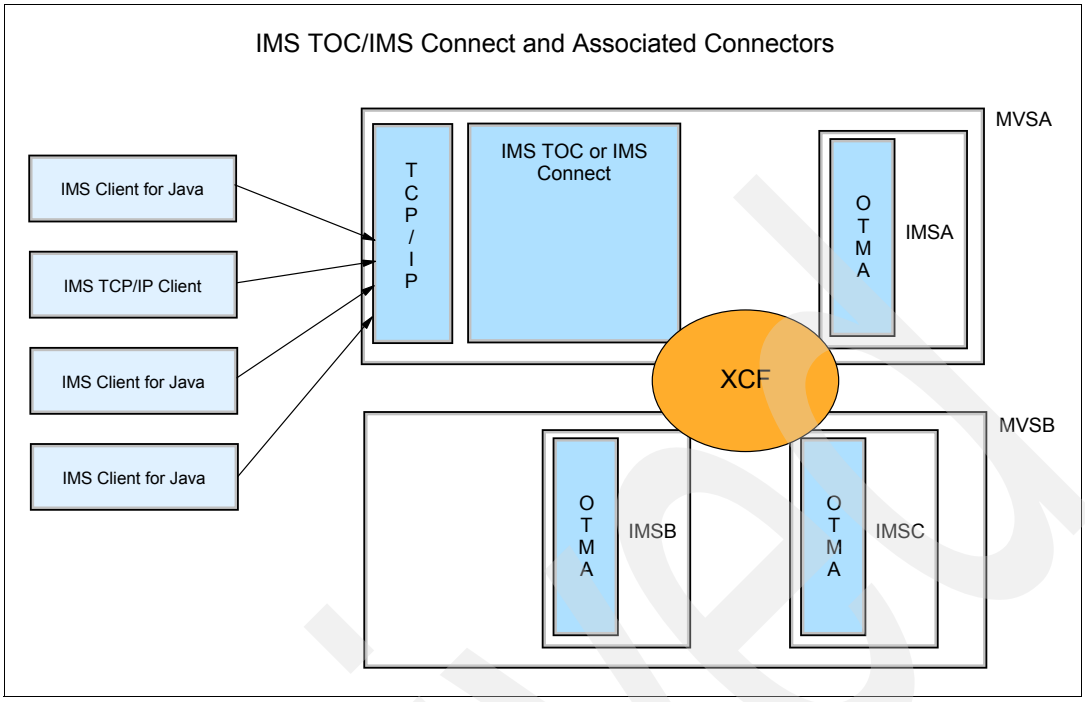


Figure 4-29 Role of IMS TOC/IMS Connect

- The *IMS Client for Java* consists of both a sample Java program that issues TCP/IP socket calls to access IMS transactions and an associated user exit routine for IMS TOC/IMS Connect. The exit translates the message into the format required by the IMS OTMA interface. Both the sample Java program and exit routine are delivered with IMS TOC/IMS Connect. They are provided as a prototype of TCP/IP connectivity into IMS and can be used as is or, since the source is provided, modified as needed.
- The *IMS Connector for Java* provides a way to generate and create Java applications using a standard toolkit. It is delivered as a component of the VisualAge for Java (VAJava) development environment, and provides the code to invoke IMS transactions using TCP/IP sockets.

With additional support from IBM WebSphere Studio and the WebSphere Application Server, the capability is also provided to build and run Java servlets.

Accessing IMS databases

Three main options are available to access the IMS database:

- *Open Database Access (ODBA)* is an IMS function, introduced in IMS V6, that provides a callable interface for OS/390 applications to directly access IMS databases. The use of ODBA supports the allocation of one or multiple PSBs and the ability to issue any DL/I call against the associated databases. The z/OS application can be a stored procedure invoked by a remote SQL client, an z/OS Web server program, or any MVS application.
- *Classic Connect* is a separately orderable component of DataJoiner® that provides read-only access to data stored in IMS databases and Virtual Storage Access Method (VSAM) data sets. Classic Connect, in conjunction with a DataJoiner instance, allows users from a variety of client platforms to submit a standard SQL query that accesses IMS DB and VSAM data. For example, a user on a PC (with DB2 client software) can issue an SQL join across IMS, VSAM, DB2 for MVS, Informix®, Oracle, Sybase, and DB2 common server data sources.

- *IMS DataPropogator* maximizes the value of IMS DB assets by enhancing IMS DB and DB2 coexistence. IMS DataPropagator™ enables the propagation of changed data between IMS DB and DB2 databases. A companion product, DataPropagator Relational, can further distribute this data to distributed DB2 systems across the enterprise.

Other IBM-provided solutions for Web-enabling IMS

Other less well known options exist:

- *Host on Demand (HOD)* provides a quick and easy Web interface to back-end enterprise systems like IMS. Access is provided across a TCP/IP network and uses 3270 emulation and Telnet. HOD also provides the necessary tools to create an attractive presentation interface for the Web browser.
- *Net.Commerce* provides a solution that quickly, easily, and securely enables electronic commerce on the World Wide Web. It helps companies integrate existing business processes and existing systems with the Web, as well as grow new Web-based businesses. Net.Commerce is scalable, open to customization and integration, and capable of handling any level of transaction volume growth. It comes complete with catalog templates, setup wizards, and advanced catalog tools to help build effective and attractive electronic commerce sites.

Workload balancing methods

Once the e-business application is developed and deployed, the next area to consider is the requirement to support growth and availability. Several workload balancing solutions are provided that address access across both SNA and TCP/IP networks. It is assumed that the target IMS subsystems are part of an IMS data sharing group, and thus are capable of accepting and subsequently processing or routing any transaction request.

SNA

VTAM Generic Resources (VTAM GR) is a function provided by VTAM to minimize the knowledge that an user needs to log on to one of several like-instances of an application (for example, one of the IMSs) in a Parallel Sysplex. To the user, there is a common name, called the Generic Resource Name, which is used by VTAM to refer to *any* of the members of a Generic Resource Group. VTAM decides which member to use when actually establishing a session request from an user node that logs on using the generic name.

IMS Workload Balancing with VTAM GR, as described in Figure 4-30 on page 205, shows how remote applications and devices that use SNA communication protocols, for example, LU 2 (3270 devices) and LU 6.2 (APPC) programs, can connect to an IMS system in a Parallel Sysplex using a generic name. Any of the options that use SNA to access IMS can therefore get the availability benefits provided by VTAM GR.

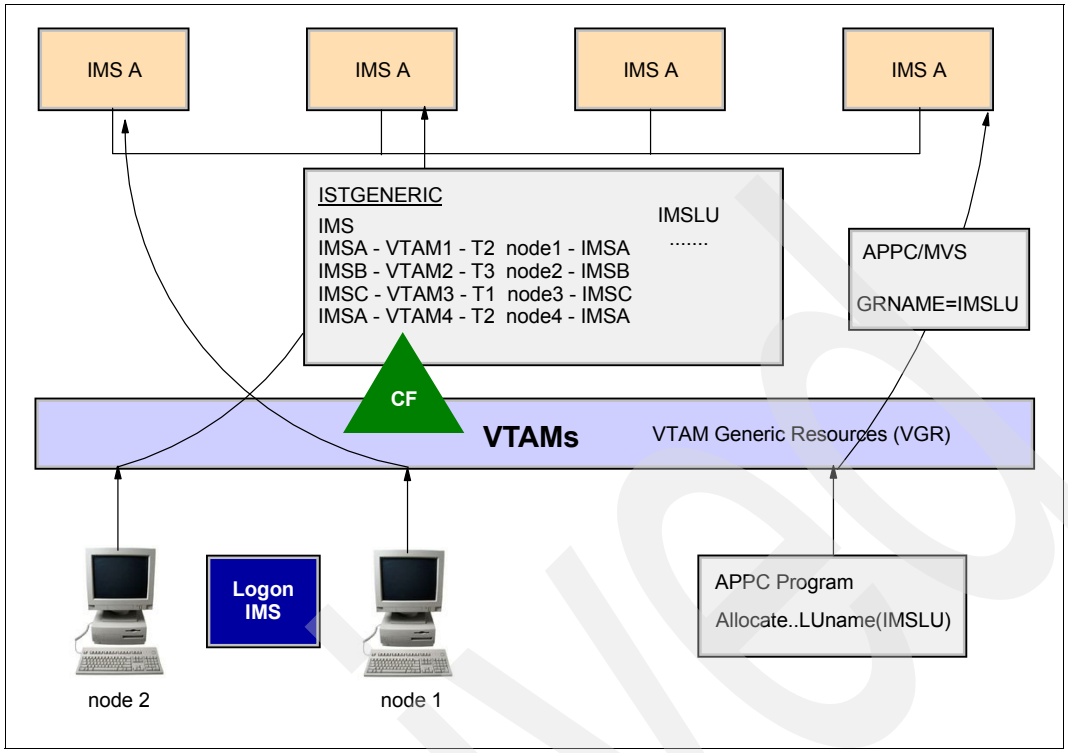


Figure 4-30 IMS Workload Balancing with VTAM GR

TCP/IP

TCP/IP provides a number of mechanisms for balancing workloads across servers:

- For long-running sessions and connections that use Telnet, the Telnet server on OS/390 supports DNS/WLM. Telnet registers with WLM, allowing incoming requests to be routed to the most appropriate Telnet server in the sysplex. Once a specific Telnet server is picked to process the request, the Telnet-to-IMS session (which uses VTAM) can take advantage of VTAM GR to access one of the IMS subsystems in the data sharing group. This is shown in Figure 4-31.

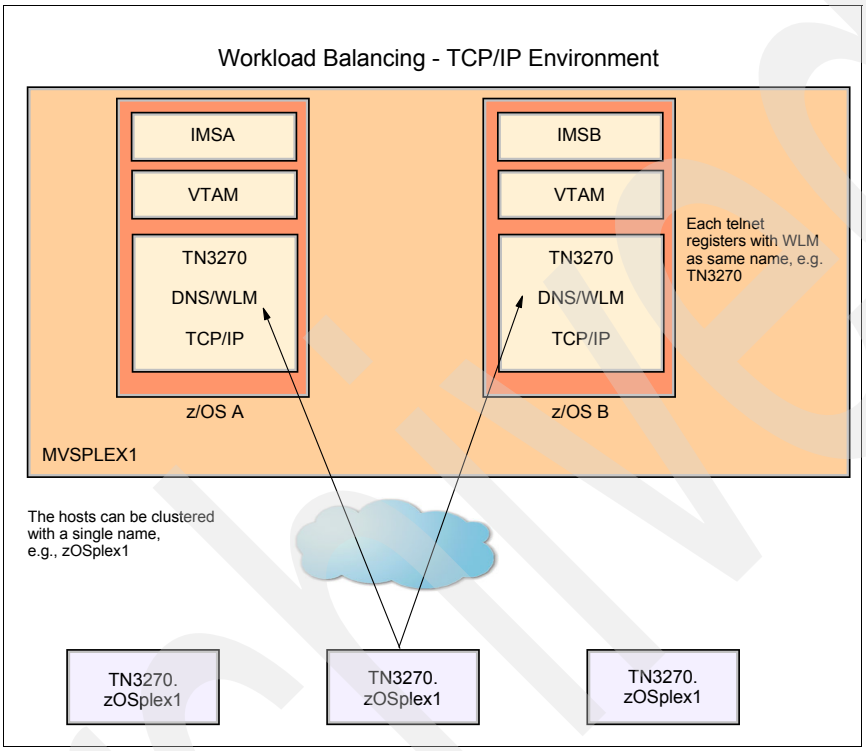


Figure 4-31 Load Balancing across Telnet servers

> **Note:** At the time of writing, neither IMS itself or IMS TOC provide support for DNS/WLM–the use of DNS/WLM in relation to IMS is limited to the Telnet Server's support of DNS/WLM.

- For short-lived requests, such as Web requests to IMS TOC/IMS Connect, the use of DNS/WLM is not suitable because of the overhead of the DNS/WLM interaction for each transaction request. Prior to the availability of OS/390 V2R10, the supported methods for such short-lived requests consist of the use of functions provided by the Interactive Network Dispatcher (IND) and external devices, such as the IBM 2216 router and CISCO MNLB.

  These types of capabilities allow for a generic TCP/IP host name to resolve to a specific IP address, for example, that of the router. The software on the router chooses and establishes a connection to one of the IP stacks in the sysplex using a mechanism, such as a round-robin approach or even by querying the WLM. Each of the IP stacks in the sysplex specifies an IP address loopback alias equal to the IP address of the front-end router. All the inbound messages are sent via the router to the target IP address, but reply messages are sent directly by the back-end IP stack (this can be done because of the loopback alias definition). This is shown in Figure 4-32 on page 207.

Once a back-end IP stack is chosen, an application, such as IMS TOC/IMS Connect, can be configured on the same port number on each IP stack. This is important because a socket connection is actually a combination of IP address and port number.

The IMS TOC/IMS Connect address space can then be configured to communicate with all the IMS systems in the sysplex, using OTMA.

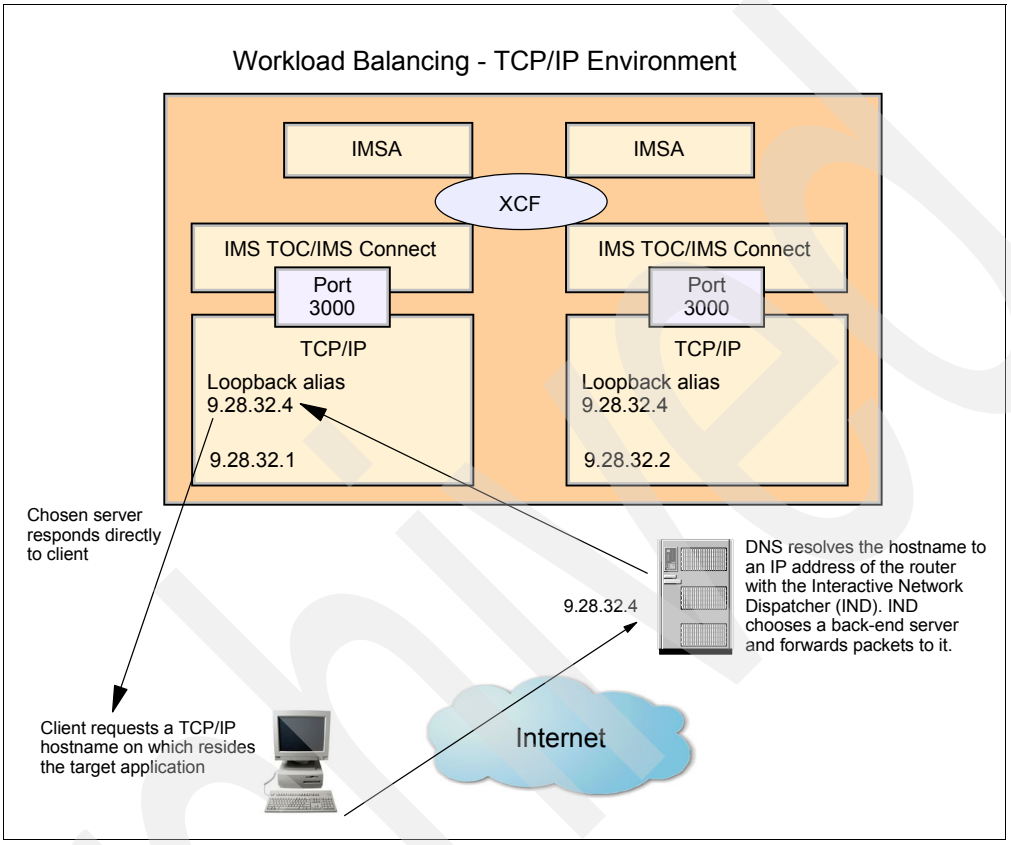


Figure 4-32 IMS Load Balancing with IND

IMS TOC provides an exit that can be used to decide which IMS subsystem to send a given transaction to. The exit can contain the names of a number of IMS subsystems. However, the exit has no knowledge of whether a given subsystem is available or not. If the subsystem that the exit selects is unavailable, the transaction will fail.

One of the new capabilities provided with IMS Connect (available as a feature in IMS V7) is a datastore table that keeps track of the current status (active or inactive) of all the IMS systems that IMS Connect is configured to reach. An associated exit interface allows user-written code to take action based on this information. IMS Connect, as a member of the XCF group, is informed whenever the status of any other members of the group changes, for example, if an IMS subsystem starts or stops. The message exits can use this information to reroute the request to an alternative IMS system in case the target IMS is not available. Figure 4-33 is an example of what can be done. Note that this capability is not available with IMS TOC.

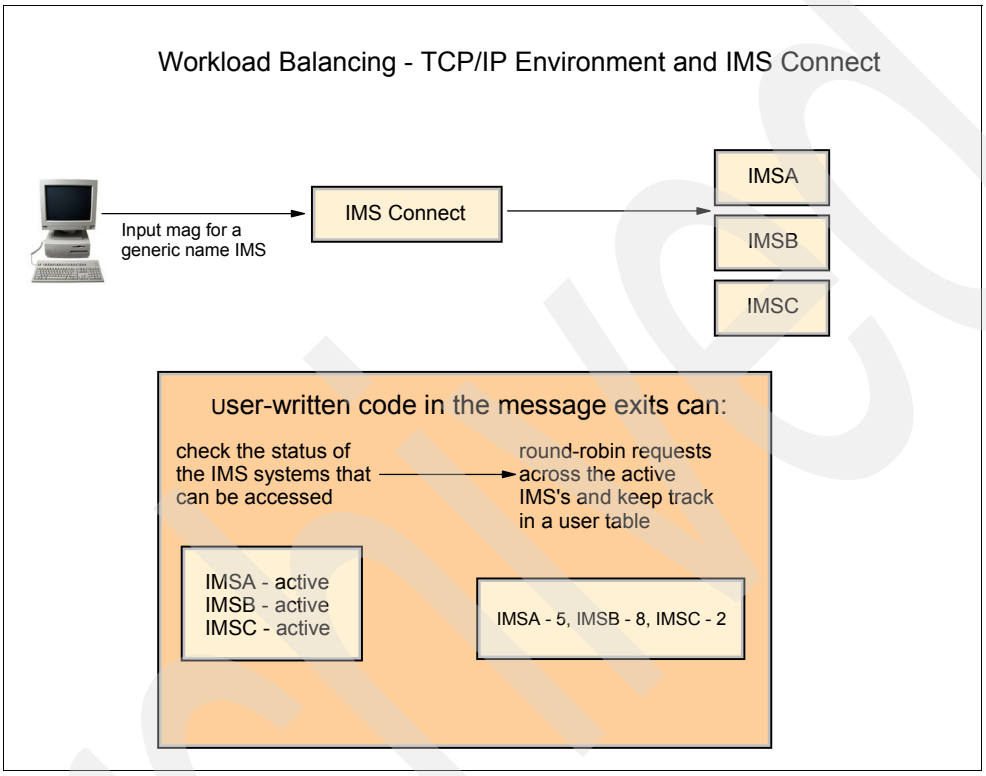


Figure 4-33 IMS connect Workload Balancing capability

Parallel Sysplex support

Once the transaction reaches the IMS message queue, the normal IMS Parallel Sysplex features can be used to ensure the transaction gets processed by one of the members of the data sharing group. The Parallel Sysplex capabilities of IMS are described in detail in 4.1.6, “IMS” on page 200 and in 4.3.2, “IMS DB data sharing” on page 236.

Additional information

Further information about each of these options is available on the IMS Web site at:

`http://www.ibm.com/software/data/ims/imswwwc.html#9`

In addition, various white papers, manuals, and redbooks are pointed to at the Web site:

`http://www.ibm.com/software/data/ims/library.html`

IBM Redbooks that may prove useful are:

- *IMS e-business Connect Using the IMS Connectors*, SG24-5427
- Chapter 10 of *IBM Web-to-Host Integration Solutions*, SG24-5237

- *IIMS/ESA V6 Parallel Sysplex Migration Planning Guide for IMS TM and DBCTL*, SG24-5461
- *Connecting IMS to the World Wide Web: A Practical Guide to IMS Connectivity*, SG24-2220

4.2 Transaction management in Parallel Sysplex

The following sections describe the functions available to manage transactions in a Parallel Sysplex environment.

4.2.1 Dynamic transaction routing

Dynamic transaction routing is the ability to route the execution of an instance of a transaction to any system, based on operational requirements at the time of execution. Operational requirements are such things as required performance or transaction *affinity.* Affinities are further discussed in “Affinities and CICS” on page 213, and in Affinities and in “IMS data sharing groups” on page 237. Dynamic transaction routing in a Parallel Sysplex delivers dynamic workload balancing.

To establish a dynamic transaction routing environment, the following steps are required:

1. Understand any transaction affinities.
2. Remove or tolerate these affinities.
3. Establish multiple application regions.
4. Clone the application.

For a CICS application to be suitable for cloning, it must be able to run in more than one CICS region. The first application that is chosen to use the Parallel Sysplex data sharing environment is likely to be one that can be cloned without change. Cloning can occur on a single image. Cloning across images provides additional availability. The transaction manager definitions must be modified to route the transactions accordingly.

Now we will take a look at two IBM transaction managers, CICS, and IMS TM.

4.2.2 CICS transactions in a Parallel Sysplex

The simplest possible CICS configuration you can have is a single CICS region that handles both communications with the users and access to the data. Today, a CICS system with only a single CICS address space is rare. CICS systems (CICSPlexes) are now the norm.

In a multiple-system environment, each participating system can have its own local terminals and databases, and can run its local application programs independently of other systems in the network. It can also establish links to other systems, and thereby gain access to remote resources. This allows resources to be distributed among and shared by the participating systems. CICS intercommunication facilities make it possible for multiple CICS regions to share selected system resources, and to present a *single-system* view to terminal operators. At the same time, each region can run independently of the others, and can be protected against errors in other regions.

The support within CICS that enables region-to-region communication is called *interregion communication (IRC)*. CICS to CICS communication outside the scope of a single region or Parallel Sysplex is achieved by using *intersystem communication (ISC)*.

Multiregion operation

For CICS-to-CICS communication within an LPAR or Parallel Sysplex, CICS provides an interregion communication facility that is independent of SNA access methods. This form of communication is called multiregion operation (MRO). MRO can be used between CICS systems that reside:

- In the same z/OS LPAR
- In the same Parallel Sysplex

CICS Transaction Server for z/OS(R) can use MRO to communicate with:

- Other CICS Transaction Server for z/OS systems
- CICS Transaction Server for OS/390 systems
- CICS Transaction Gateway for z/OS (using EXCI)
- z/OS batch programs using the external CICS interface (EXCI)

Note: The external CICS interface (EXCI) uses a specialized form of MRO link.

The support within CICS that enables region-to-region communication is called interregion communication (IRC). IRC can be implemented in three ways:

- Through support in CICS terminal control management modules and by use of a CICS-supplied interregion program (DFHIRP) loaded in the link pack area (LPA) of MVS. DFHIRP is invoked by a type 3 supervisor call (SVC).
- By MVS cross-memory services, which you can select as an alternative to the CICS type 3 SVC mechanism.
- By the cross-system coupling facility (XCF) of z/OS. XCF is required for MRO links between CICS regions in different MVS images of an MVS sysplex. It is selected dynamically by CICS for such links, if available.

Intersystem communication

For communication between CICS and non-CICS systems, or between CICS systems that are not in the same z/OS LPAR or Parallel Sysplex, you normally require an SNA access method, such as ACF/VTAM(R), to provide the necessary communication protocols. Communication between systems through SNA is called intersystem communication (ISC).

CICS intercommunication facilities

For communicating with other CICS, IMS, DB2, or APPC systems, CICS provides the following facilities:

- Function shipping
- Asynchronous processing
- Transaction routing
- Distributed program link (DPL)
- Distributed transaction processing (DTP)

Function shipping

CICS function shipping lets an application program access a resource owned by, or accessible to, another CICS system. Both read and write access are permitted, and facilities for exclusive control and recovery and restart are provided. The remote resource can be:

- A file
- A DL/I database
- A transient-data queue
- A temporary-storage queue

Application programs that access remote resources can be designed and coded as though the resources were owned by the system in which the transaction is to run. During execution, CICS ships the request to the appropriate system.

The advent of function-shipping for file requests allowed the creation of the File-Owning Region (FOR), which could own the access to specific files, and which could be invoked from multiple TORs within the CICSPlex.

Similarly, the function-shipping of Temporary Storage and Transient Data queues allowed the creation of the queue-owning region (QOR), which could own the access to specific TS and TD queues.

Asynchronous processing

Asynchronous processing allows a CICS transaction to initiate a transaction in a remote system and to pass data to it. The remote transaction can then initiate a transaction in the local system to receive the reply. The reply is not necessarily returned to the task that initiated the remote transaction, and no direct tie-in between requests and replies is possible (other than that provided by user-defined fields in the data). The processing is therefore called asynchronous.

CICS transaction routing

CICS transaction routing permits a transaction and an associated terminal to be owned by different CICS systems. Transaction routing can take the following forms:

- A terminal that is owned by one CICS system can run a transaction owned by another CICS system.
- A transaction that is started by automatic transaction initiation (ATI) can acquire a terminal owned by another CICS system.
- A transaction that is running in one CICS system can allocate a session to an APPC device owned by another CICS system.

Distributed program link (DPL)

CICS distributed program link enables a CICS program (the client program) to call another CICS program (the server program) in a remote CICS region. Here are some of the reasons you might want to design your application to use DPL:

- To separate the user interface (for example, BMS screen handling) from the application business logic, such as accessing and processing data, to enable parts of the applications to be ported from host to workstation more readily.
- To obtain performance benefits from running programs closer to the resources they access, and thus reduce the need for repeated function shipping requests.

In many cases, DPL offers a simple alternative to writing distributed transaction processing (DTP) applications.

Distributed transaction processing (DTP)

When CICS arranges function shipping, distributed program link, asynchronous transaction processing, or transaction routing for you, it establishes a logical data link with a remote system. A data exchange between the two systems then follows. This data exchange is controlled by CICS-supplied programs, using APPC, LUTYPE6.1, or MRO protocols. The CICS-supplied programs issue commands to allocate conversations, and send and receive data between the systems. Equivalent commands are available to application programs, to allow applications to converse with CICS or non-CICS applications. The technique of distributing the functions of a transaction over several transaction programs within a network is called distributed transaction processing (DTP).

DTP allows a CICS transaction to communicate with a transaction running in another system. The transactions are designed and coded specifically to communicate with each other, and thereby to use the intersystem link with maximum efficiency.

The communication in DTP is, from the CICS point of view, synchronous, which means that it occurs during a single invocation of the CICS transaction and that requests and replies between two transactions can be directly associated. This contrasts with the asynchronous processing described previously.

Evolution of multi-region operation in CICS

Figure 4-34 shows the evolution of CICS from a single system to multiple regions running on multiple images. This evolution was driven on the one hand by expanding CICS workloads and the need to improve performance and ability, and on the other hand, by improvements to the hardware and software available - in particular, the emergence of Parallel Sysplex.

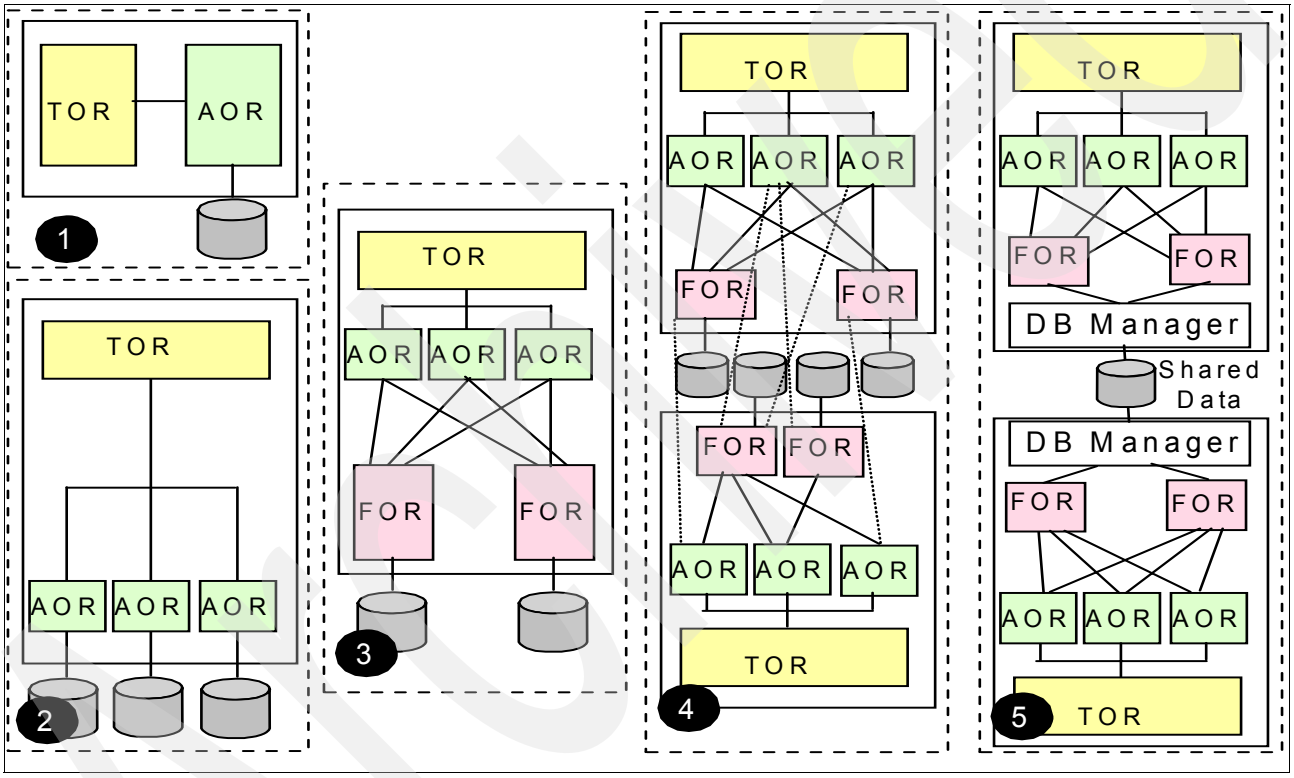


Figure 4-34 Evolution of multi-region operation in CICS

The steps shown in Figure 4-34 are:

1. In the first box, we can see the simplest possible CICSPlex, a TOR that handles communication with the user and then routes the work to an AOR, with the AOR handling access to the database (which could be IMS, DB2, or VSAM).
2. As workloads increased and databases become more complex, the CICSPlex evolved into that shown in the second box, where we see multiple AORs with each AOR owning the access to its subset of the data.
3. In the third box, we see a situation where there is a requirement to share data between multiple AORs. Prior to VSAM RLS or DB2 data sharing, this could only be achieved by creating a File-Owning Region (FOR), which owned the data to be shared between multiple AORs.

4. In boxes one, two, and three, we see the CICPlex in only a single z/OS. In the fourth box, we see how Parallel Sysplex technology evolved to allow us to have two z/OS images, each with their own TORs, AORs, and FORs, but with a requirement to share data between the two z/OS images. We can see that MRO/XCF or ISC has been used to allow AORs in each z/OS to communication with the FORs owning data in the other z/OS.
5. Finally, with the advent of VSAM RLS and DB2 data sharing, we are in a position to genuinely share data across the Parallel Sysplex. This breakthrough removes the need for FORs for shared data, thereby removing a potential "single point of failure" in the CICSPlex. It is often the case that a file-owning region within the CICSplex is still required to provide access to data files that do not support data sharing through the CF, such as BDAM files, or VSAM files that are not accessed in RLS mode.

Dynamic transaction routing

We have already seen that CICS resources, such as transactions and programs, required in one region may be owned by another. For example, you may have a terminal-owning region (TOR) that requires access to transactions owned by an application-owning region (AOR). You can specify the location of a resource when you are designing your system. Then, requests of a specific resource are always routed to the same region. Typically, the location of the resource is specified in the installed resource definition. This is known as *static routing*.

With *dynamic routing*, the location of the resource is decided at run time. This has the advantage that an intelligent dynamic routing program can make the decision about which CICS region to route the transaction to based on such criteria as:

- What potential target CICS regions are available
- Which of those potential target CICS regions is likely to process the transaction the fastest
- What affinities (if any) are associated with this transaction

CICSPlex SM (see 4.2.3, "CICSPlex SM" on page 216) provides such a dynamic routing program, which does all of these, and can even check with z/OS WLM to identify which of the potential targets is best placed (given the current workload) to process the request.

You can use the dynamic routing program to route:

- Transactions initiated from user terminals
- Transactions initiated by a subset of terminal-related EXEC CICS START commands
- Function-shipped Program-link requests
- Function-shipped Program-link requests to program DFHL3270 to execute a CICS transaction using the bridge

If you are using dynamic transaction routing, your dynamic routing program needs to be aware of any affinities (see "Affinities and CICS" on page 213) when making its decision about which CICS region is to be the target region. In an ideal world, your application will have no affinities - if there are affinities, then your dynamic routing program must be able to manage those affinities.

Affinities and CICS

CICS transactions use many different techniques to pass data from one transaction to another. Some of these techniques require that the transactions exchanging data must execute in the same CICS region, and therefore impose restrictions on the *dynamic* routing of transactions. If transactions exchange data in ways that impose such restrictions, there is said to be an affinity between them.

There are two categories of affinity: intertransaction and transaction-system affinity.

The restrictions on dynamic transaction routing caused by transaction affinities depend on the duration and scope of the affinities. Clearly, the ideal situation for a dynamic transaction routing program is to have no transaction affinities at all. However, even when transaction affinities do exist, there are limits to the scope of these affinities. The scope of an affinity is determined by:

- Affinity relation: This determines how the dynamic transaction routing program is to select a target AOR for a transaction instance associated with the affinity.
- Affinity lifetime**:** This indicates how long the affinity exists.

Intertransaction affinity is an affinity between two or more CICS transactions. It is caused when transactions pass information between one another, or synchronize activity between one another, by using techniques that force them to execute in the same CICS AOR. Intertransaction affinity, which imposes restrictions on the dynamic routing of transactions, can occur in the following circumstances:

- One transaction terminates, leaving state data in a place that a second transaction can access only by running in the same CICS AOR.
- One transaction creates data that a second transaction accesses while the first transaction is still running. For this to work safely, the first transaction usually waits on some event, which the second transaction posts when it has read the data created by the first transaction. This synchronization technique requires that both transactions are routed to the same CICS region.

Transaction-system affinity is an affinity between a transaction and a particular CICS AOR (that is, it is not an affinity between transactions themselves). It is caused by the transaction interrogating or changing the properties of that CICS region.

Transactions with affinity to a particular system, rather than another transaction, are not eligible for dynamic transaction routing. Usually, they are transactions that use INQUIRE and SET commands or have some dependency on global user exit programs.

An affinity lifetime is classified as one of the following types:

System The affinity lasts while the target AOR exists, and ends whenever the AOR terminates (at a normal, immediate, or abnormal termination). The resource shared by transactions that take part in the affinity is not recoverable across CICS restarts.

Permanent The affinity extends across all CICS restarts. The resource shared by transactions that take part in the affinity is recoverable across CICS restarts. This is the most restrictive of all the intertransaction affinities.

Pseudo-conversation The LUname or user ID affinity lasts for the entire pseudo-conversation, and ends when the pseudo-conversation ends at the terminal.

Logon The (LUname) affinity lasts for as long as the terminal remains logged on to CICS, and ends when the terminal logs off.

Signon The (user ID) affinity lasts for as long as the user is signed on, and ends when the user signs off.

Note: For user ID affinities, the pseudo-conversation and signon lifetime are only possible in situations where only one user per user ID is permitted. Such lifetimes are meaningless if multiple users are permitted to be signed on at the same time with the same user ID (at different terminals).

An affinity relation is classified as one of the following:

Global A group of transactions in which all instances of transactions initiated from any terminal must execute in the same AOR for the lifetime of the affinity. The affinity lifetime for global relations is system or permanent.

LUname A group of transactions whose affinity relation is defined as LUname is one in where all instances of transactions initiated from the same terminal must execute in the same AOR for the lifetime of the affinity. The affinity lifetime for LUname relations is pseudo-conversation, logon, system, or permanent.

User ID A group of transactions whose affinity relation is defined as user ID is one in which all instances of transactions initiated from a terminal and executed on behalf of the same user ID must execute in the same AOR for the lifetime of the affinity. The affinity lifetime for user ID relations is pseudo-conversation, signon, system, or permanent.

In a dynamic transaction routing environment, your dynamic transaction routing program must consider transaction affinities to route transactions effectively. Ideally, you should avoid creating application programs that cause affinities, in which case the problem does not exist. However, where existing applications are concerned, it is important that you determine whether they are affected by transaction affinities before using them in a dynamic transaction routing environment.

The CICS Interdependencies Analyzer (CICS IA) can help you with this task. The affinity-related functions of CICS IA are designed to help users of CICS dynamic routing, who need to determine whether any of the transactions in their CICS applications use programming techniques with inter-transaction or transaction-system affinities. CICS IA can also be used by application programmers to detect whether the programs they are developing are likely to cause transaction affinities. Using CICS IA, you can:

- Collect data about potential affinities
- Load the affinity data into DB2 databases
- Use the Query interface to analyze the affinities data by means of SQL queries
- Use the Scanner to check a load module library for programs that issue commands that may cause transaction affinities
- Use the Affinities Reporter to produce detailed affinity reports
- Use the Builder to create a file of affinity-transaction-group definitions suitable for input to CICSPlex SM

Recommendation to use CICS Interdependencies Analyzer: We strongly recommend that you use the CICS IA to identify possible affinities in your CICS systems. IBM Global Services can provide assistance in analyzing the data provided by this utility, which can result in significant time savings.

In this section, we look at the topology for CICS in a Parallel Sysplex, and also at how affinities may affect the Parallel Sysplex configuration.

4.2.3 CICSPlex SM

It is evident that CICS has come a long way and become a lot more complex since the days of the single CICS region. CICSPlex SM, which is delivered as part of CICS TS Version 3.1 (and of all earlier releases since CICS TS Version 1.3), is designed to simplify the management of today considerably more complex CICS environment, and has functionality built into it that allows it to use z/OS and Parallel Sysplex technology when managing your CICS environment.

Some examples of the facilities provided by CICSPlex SM are:

- *Dynamic Transaction routing*: The CICSPlex SM supplied dynamic routing program EYU9XLOP supports both workload balancing and workload separation. You define to CICSPlex SM which requesting, routing, and target regions in the CICSplex can participate in dynamic routing, and any affinities that govern the target regions to which particular work requests must be routed. The output from the CICS Interdependency Analyzer can be used directly by CICSPlex SM.
- *Single system image*: CICSPlex SM keeps track of the location and status of every CICS resource, allowing CICS regions to be managed as a single system. Actions affecting multiple resources can be achieved with a single command, even though the resources are spread among multiple CICS systems on different CPCs.
- *Single point of control*: CICSPlex SM allows all the CICS systems in an enterprise to be managed from a single point of control. This supports installations who have business reasons for organizing their systems into more than one CICSplex (for example, one for testing and another for production).
- *Run Time Analysis (RTA)*: RTA provides management by exception, drawing attention to potential deviations from service level agreements. CICS users will see improved reliability and higher availability of their systems, because potential problems are detected and corrected before they become critical.

The CICSPlex SM address space

A CMAS is the hub of any CICSPlex SM configuration, managing and reporting on CICS regions and their resources. A CICSplex is managed by one or more CMASes. In our case, we have installed one CMAS in each image. The CMAS does the monitoring, real time analysis, workload management, and operational functions of CICSPlex SM, and maintains configuration information about the CICS regions for which it is responsible. See *CICS for OS/390 and Parallel Sysplex,* GC33-1180 for more information about a CMAS.

CICS and Parallel Sysplex: CICS TS and the Parallel Sysplex have a very complementary structure, especially when using MRO. We recommend that you use XCF MRO services when connecting CICS regions within a Parallel Sysplex. CICS systems that are part of a Parallel Sysplex can be at different release levels. All releases of CICS that are still in support can coexist within a single CICSplex, sharing a single CSD if you wish. The use of CICSPlex SM is highly recommended when running CICS in the Parallel Sysplex environment.

What will my data sharing configuration look like?

This section describes an example of a target data sharing configuration that is recommended for a Parallel Sysplex. The target configuration is designed with considerations for:

- *Availability*: It provides maximum availability of online applications by cloning as many of the sysplex components as possible. These include:
  - z/OS images
  - VTAM nodes
  - Transaction processing subsystems (CICS and IMS)
  - The DBCTL, DB2, and VSAM record level sharing (RLS) subsystems, providing IMS DB, DB2, and VSAM multisystem data sharing
- *Capacity*: Provides a growth capability for adding additional capacity without disrupting production work. The sysplex provides this kind of nondisruptive growth capability, enabling you to easily upgrade or add a new CPC.
- *Systems management*: It provides better systems management of multiple images, with z/OS clones offering easier installation and administration of additional images.

You should aim for as much symmetry as possible in a Parallel Sysplex. For example, there is no reason why you should not install a CICS terminal-owning region on each image.

4.2.4 The target CICS configuration in a Parallel Sysplex

The target configuration, as shown in Figure 4-35 on page 218, includes the following elements:

- One terminal-owning region (TOR1 through TOR4) in each of the z/OS images.
- Twelve application-owning regions allocated across the four z/OS images.

> **Note:** Although the transaction processing workload that runs in this Parallel Sysplex configuration is assumed to be a mixture of DBCTL, DB2-based, and VSAM applications, all the application-owning regions are capable of running all transactions. That is, they are clones of each other, and any workload separation is controlled by workload management policies and the CICS dynamic routing mechanism. Therefore, all the application-owning regions require a connection to DBCTL, DB2, and VSAM RLS subsystems in their respective images.

- Four DBCTL environments allocated across the four z/OS images support the CICS-DL/1 workload processed by the application-owning regions. Each DBCTL consists of a database control (DBCTL) address space, a database recovery control (DBRC) address space, and a DL/1 separate address space (DLISAS).
- Four DB2 subsystems allocated across the images support the CICS-DB2 workload processed by the application-owning regions.
- Four SMSVSAM systems allocated across the images support the CICS-VSAM workload processed by the application-owning regions.
- Four DFHCFMN server regions allocated across the images support access to data tables in the Coupling Facility.
- Eight Integrated Resource Lock Managers (IRLMs), two for each image (a separate IRLM is required for DBCTL and DB2 on each image).

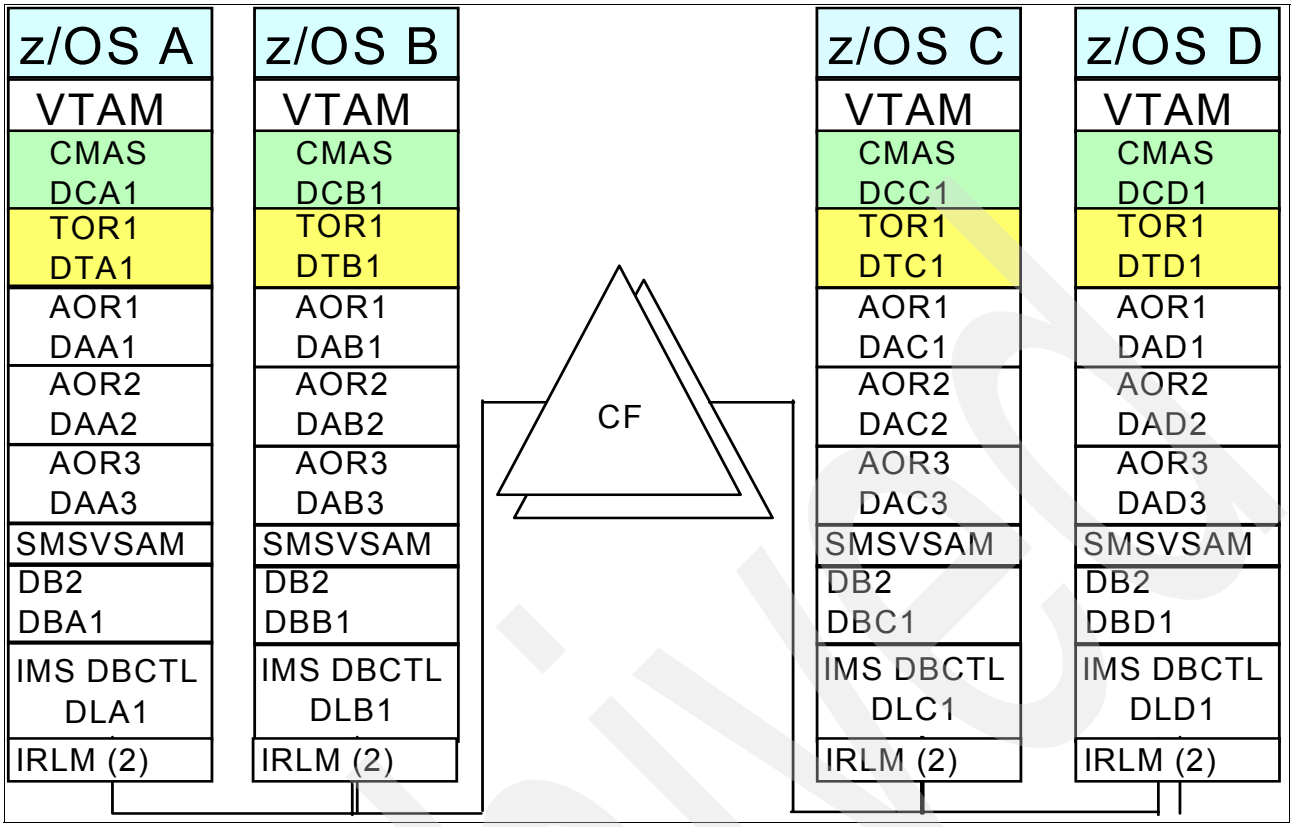


Figure 4-35 Sample CICS Data Sharing Subsystem configuration

The CICS terminal-owning regions

With VTAM generic resources function, all the terminal-owning regions in the CICSplex can be represented by one generic application (APPL) name, and appear as one TOR to terminal users. This means that regardless of which application the users want, they log on to only a single CICS application ID. VTAM generic resources resolve the generic name to the specific APPLID of one of the terminal-owning regions. Thus, the CICSplex appears as a single system to the user.

Fast TOR restart can be implemented by using VTAM persistent session support.

The terminal-owning regions are identical in every respect except for their external identifiers. This means that:

- They have different specific APPLIDs to identify them to VTAM and their partner MRO regions.
- They each have a unique local name specified on the SYSIDNT system initialization parameter.
- They each have a unique CICS monitoring subsystem identifier for RMF performance reporting, specified on the MNSUBSYS system initialization parameter.

Generally, apart from the identifiers just listed, you should try to make your terminal-owning regions identical clones, defined with identical resources (such as having the same system initialization parameters).

Exact cloning may not be possible if you have some resources that are defined to only one region. For example, if your network needs to support predefined auto-connected CICS

terminals, you have to decide to which region such resources should be allocated and specify them accordingly. In this situation, you cannot use the same group list GRPLIST system initialization parameter to initialize all your terminal-owning regions. However, the GRPLIST system initialization parameter allows you to specify up to four group list names, which makes it easier to handle variations in CICS startup group lists.

The reasons for having multiple terminal-owning regions are as follows:

- For continuous availability

  You need to ensure that you have enough terminal-owning regions to provide continuous availability of the CICSplex.

  Fewer users are impacted by the failure of one terminal-owning region. If a terminal-owning region fails, the users connected to other terminal-owning regions are unaffected, while the users of the failed region can log on again immediately, using the VTAM generic resource name, without waiting for the failed terminal-owning region to restart.

  Furthermore, Multi-Node Persistent Session support allows CICS sessions to remain connected across VTAM, hardware, or software failures. When CICS is restarted in the same system or another system, the session can be re-established with no disruption to the user, other than the need to log on again.

- For performance

  To service several application-owning regions requires many MRO send and receive sessions. It is better to allocate the required sessions across several terminal-owning regions than to try to load them all into just one or two systems.

  In the sample configuration, we balanced the number of subsystems and CICS regions to fully exploit images running on multiprocessor CPCs.

- For faster restart

  If a terminal-owning region fails, restarting is faster because of the smaller number of sessions to be recovered.

The CICS application-owning regions

The application-owning regions are defined as sets, with each set containing identical regions (AOR clones). Each set of clones should be capable of handling one or more different applications. The terminal-owning regions achieve workload balancing and availability by dynamically routing the incoming transactions to the best candidate application-owning region within a cloned set. CICSPlex SM provides the dynamic routing function, by interfacing with WLM to establish the best target CICS region.

If you have split your CICS regions into separate regions based on applications, the data sharing, workload balancing environment of the Parallel Sysplex allows you to collapse regions together again. If your reason for splitting applications into separate regions is to provide some form of storage protection between applications, the introduction of transaction isolation in CICS/ESA 4.1 may make splitting no longer necessary.

Note: The AORs can be at different levels of CICS, as long as each AOR provides support for the facilities used by the applications in those AORs.

The CICS file-owning regions

Prior to VSAM RLS, a file-owning region was required if you wished to share a VSAM file between multiple CICS regions. However, files that are accessed in RLS mode are now accessed directly from each CICS region, rather than having to be routed through a central owning region. We have not shown any file-owning region in our configuration, but you might need to have one or more where VSAM RLS is not possible.

The CICS queue-owning regions

The inclusion of a queue-owning region in the CICSplex is important where it is not possible to use shared Temporary Storage. Shared temporary storage queues are stored in the coupling facility, and are managed by a dedicated Temporary Storage server address space, allowing the shared TS queues to be accessed concurrently by multiple CICS TS regions. Before shared temporary storage, TS and TD queues had to be owned by a single CICS region, and requests to read or write to those queues would be function shipped to that region. This prevented any intertransaction affinities occurring with temporary storage or transient data queues. Defining queues to the application-owning regions as remote queues, accessed through a queue-owning region, ensures that they are accessible by any application-owning region through function shipping requests. An alternative to a queue-owning region is to make the file-owning region a combined FOR/QOR.

With CICS TS for OS/390 and later, there is no need for the QOR. CICS TS for OS/390 introduces shared temporary storage queues between different CICS systems using the CF. Shared temporary storage provides multisystem data sharing for non-recoverable temporary storage queues; the queues are stored in CF list structures.

The CICS Listener region

A recent addition to the set of CICS regions is the Listener region, which is used exclusively for handling transactions initiated over TCP/IP - this could be from:

- CICS Transaction Gateway clients
- HTTP clients
- Service requesters
- EJB clients

Figure 4-36 on page 221 shows the function of a Listener region.

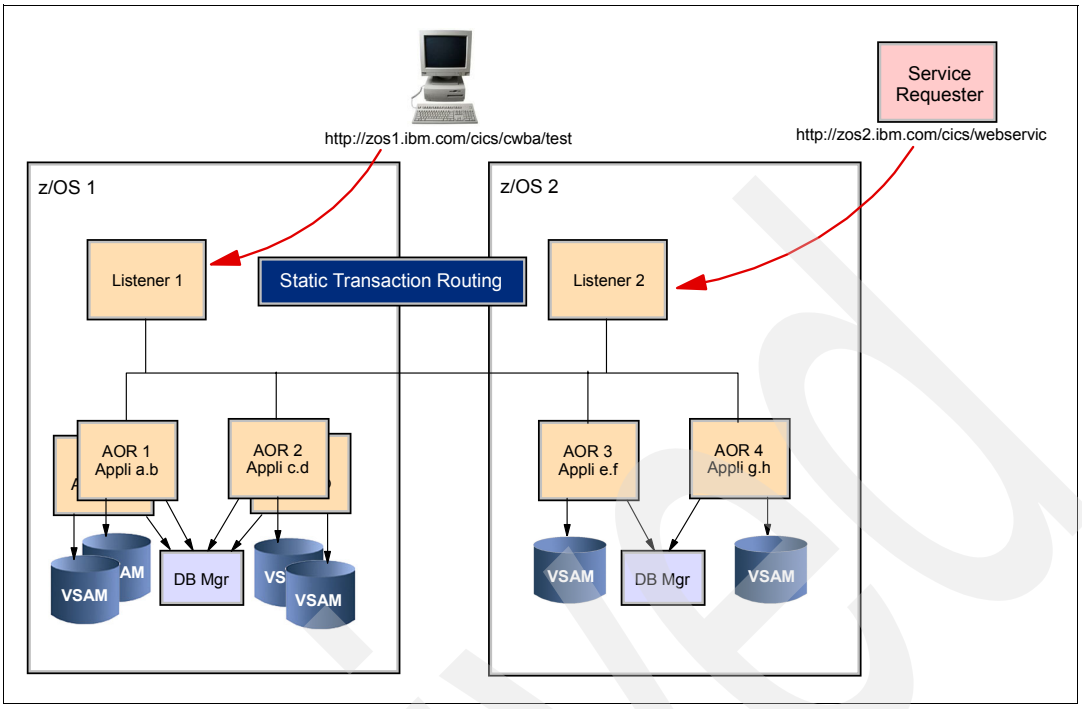


Figure 4-36 e-business CICSplex: Listener region

There are a number of reasons why you may wish to use a separate region for this function:

- You may wish to protect your Web users from problems that may arise with traditional applications (or vice-versa).
- The Listener region may contain special functions (such as the TCP/IP listener) that you have not activated on your other CICS regions.
- You may wish to be able to control the performance of the Listener regions separately from the traditional regions. This could either be to guarantee them good performance, or to restrict them to a certain amount of resource usage.
- For security reasons, you may wish to place the Listener region in a different LPAR than your AORs.
- Channeling TCP/IP-based transactions through a Listener region gives you the ability to route the incoming transactions across multiple AORs, in the same way that 3270 workloads are distributed via a TOR.

The IMS DBCTL environment

To exploit IMS data sharing, our target configuration includes one IMS DBCTL environment in each z/OS that has CICS application-owning regions, with each such region connected to the DBCTL in its image.

The DB2 subsystems

To exploit DB2 data sharing, our target configuration includes one DB2 subsystem in each z/OS that has CICS application-owning regions, with each such region connected to the DB2 in its image.

The CICSPlex SM address space

Our configuration shows a CICSPlex SM address space (CMAS) in each z/OS image that runs CICS regions. The CMASes combine to provide a single system image of the CICSplex (see 4.2.3, “CICSPlex SM” on page 216).

4.2.5 CICS TS for OS/390 and Parallel Sysplex

Now that we have a good understanding of the nature of a CICSplex, we can look at how the advantages of the Parallel Sysplex can best be brought to bear on our CICSplex.

CICS/VSAM Record Level Sharing (RLS)

RLS is a VSAM function exploited by CICS TS. RLS enables VSAM data to be shared, with full update capability, between many applications running in many CICS regions. RLS provides many benefits to CICS applications, improving the way CICS regions can share VSAM data. Using RLS, you may:

- *Improve availability*: Availability is improved in a number of ways:
  - The FOR is eliminated as a single point of failure. With an SMSVSAM server in each image in the Parallel Sysplex, work can be dynamically routed to another image in the event of a system failure.
  - Data sets are not taken offline in the event of a backout failure. If a backout failure occurs, only the records affected within the unit of work remain locked; the data set remains online.
- *Improve integrity*: Integrity is improved in RLS mode for both reading and writing of data. RLS uses the shared lock capability to implement new read integrity options. CICS supports these options through extensions to the application programming interface (API). For CICS/VSAM RLS usage and sizing related to the CF, refer to the CF Sizer tool, available at:

  `http://www.s390.ibm.com/cfsizer/`
- *Reduce lock contention*: For files opened in RLS mode, VSAM locking is at the record level, not at the control interval level, which can improve throughput and reduce response times.
- *Improve sharing between CICS and batch*: Batch jobs can read and update, concurrently with CICS, *non-recoverable* data sets that are opened by CICS in RLS mode. Conversely, batch jobs can read (but not update), concurrently with CICS, *recoverable* data sets that are opened by CICS in RLS mode.
- *Improve performance*: Multiple CICS application-owning regions can directly access the same VSAM data sets, avoiding the need to function ship to file-owning regions. The constraint imposed by the capacity of an FOR to handle all the accesses for a particular data set, on behalf of many CICS regions, does not apply.

Temporary storage data sharing

The CICS TS for OS/390 temporary storage data sharing facility supports the Parallel Sysplex environment by providing shared access to CICS non-recoverable temporary storage queues.

Temporary storage data sharing enables you to share non-recoverable temporary storage queues between many applications running in different CICS regions across the Parallel Sysplex. CICS uses temporary storage data sharing to store shared queues in a structure in a Coupling Facility, access to which is provided by a CICS temporary storage server address space.

CICS stores a set of temporary storage queues that you want to share in a temporary storage pool. Each temporary storage pool corresponds to a Coupling Facility list structure defined in a CFRM policy. You can create single or multiple temporary storage pools within the Parallel Sysplex, to suit your requirements, as the following examples show:

- You could create separate temporary storage pools for specific purposes, such as for production or for test and development.
- You could create more than one production pool, particularly if you have more than one CF and you want to allocate temporary storage pool list structures to each CF.

The benefits of temporary storage data sharing include:

- *Improved performance compared with the use of remote queue-owning regions*: Access to queues stored in the CF is quicker than function shipping to a QOR.
- *Improved availability compared with a QOR*: The availability of a temporary storage server is better than that with a QOR because you can have more than one temporary storage server for each pool (typically, one server in each image in the Parallel Sysplex). If one temporary storage server or image fails, transactions can be dynamically routed to another AOR on a different image, and the transaction can then use the temporary storage server in that image.
- *Elimination of intertransaction affinities*: Temporary storage data sharing avoids intertransaction affinity.

  For CICS temporary storage usage and sizing related to the CF, refer to the CF Sizer tool available at:

  `http://www.s390.ibm.com/cfsizer/`

The CICS log manager

The CICS log manager replaces the journal control management function of earlier releases. Using services provided by the system logger, the CICS log manager supports:

- The CICS system log, which is also used for dynamic transaction backout. (The CICS internal dynamic log of earlier releases does not exist in CICS TS for OS/390.)
- Forward recovery logs, auto journals, and user logs (general logs).

The system logger is a component of z/OS that provides a programming interface to access records on a log stream.

The CICS log manager uses the services of the system logger to enhance CICS logging in line with the demands of the Parallel Sysplex environment. In particular, it provides online merging of general log streams from different CICS regions that may be on different images in the Parallel Sysplex. The CICS log manager, with the system logger, improves management of system log and dynamic log data (all of which are written to the system log stream) by:

- Avoiding log wraparound
- Automatically deleting obsolete log data of completed units-of-work

All CICS logs (except for user journals defined as type SMF or DUMMY) are written to system logger log streams. User journals of type SMF are written to the SMF log data set.

There are a number of tasks that you must complete in order to set up the CICS TS for OS/390 log manager environment. Refer to *CICS Transaction Server for OS/390 V1.3 Migration Guide,* GC34-5353. Specifically, we recommend that you:

- Carefully plan your CF configuration. The system logger requires at least one CF. However, the ability to rebuild CF structures becomes vital if you want to avoid disruptions to your processing environment in case of a CF failure. Therefore, we recommend that you have two CFs.
- When planning structure sizes, ensure that you allow enough storage to prevent the log stream associated with the CICS system log from spilling to DASD.
- Use the CICS logger CF sizing utility (DFHLSCU) to calculate the amount of CF space you need and the average buffer size of your log streams.
- Make sure that your specifications for the log streams are such that the system logger copies to staging data sets if the CF is (or becomes) volatile.
- Specify each staging data set to be at least the same size as the log stream share of the CF, but round up the average block size to a multiple of 4096.
- Make sure you define a naming convention that is sysplex-wide consistent for CF structures, DASD log data sets, and DASD staging data sets. This will help you in identifying these resources.

The CICS log manager provides several benefits for all users. Refer to the *CICS Transaction Server for OS/390 V1R2 Release Guide,* GC33-1570 for further explanation.

Enhancements to VTAM generic resources

Support for VTAM generic resources is enhanced in CICS TS for OS/390 R1, with the aim of improving the usability of generic resources with LU6.2 (APPC) connections. The main benefit of these changes is to facilitate inter-sysplex communication, in which both sysplexes use generic resources. The changes mean the following:

- There is more flexibility in communicating by APPC links between generic resources in partner sysplexes, because routing of connections between sysplexes is controlled by CICS.
- You can inquire about an APPC connection between generic resource sysplexes to see which member of the remote generic resource is in session with a member of the local generic resource.
- Generic resource members can now use APPC as well as MRO, connections within the CICSplex.

TCP/IP port sharing

You can configure your TCP/IP so that multiple CICS TS for z/OS Listener regions in the same z/OS LPAR are able to listen on the same TCP/IP port or ports, thereby providing improved availability and performance within the z/OS image.

Similarly, you can run multiple images of the CICS Transaction Gateway Listener daemon, and configure TCP/IP so that they can listen on the same TCP/IP port or ports.

Sysplex Distributor

You can use the function provided by Sysplex Distributor to balance incoming TCP/IP requests destined for CICS TS or CICS Transaction Gateway ports, this time across z/OS images, and using WLM to decide the most appropriate Listener region or CICS Transaction Gateway daemon to process the request. If you are going to use Sysplex Distributor, you have to ensure that the CICS or CICS TG configurations sharing a port are true clones of each other.

CF data tables support

CICS CF data tables support allow user applications running in different CICS regions that reside in one or more z/OS images within a Parallel Sysplex to share working data with update integrity.

Data in a CF data table is accessed through the CICS file control API, enabling existing applications to use it, either without any modification, or with minimum changes, depending on the level of function required. CF data tables provide efficient sharing of data with integrity, and behave much like a sysplex-wide equivalent of user-maintained data tables. Key lengths greater than 16 bytes are not supported.

A CF data table (CFDT) pool is a CF list structure, and access to it is provided by a CFDT server. A CFDT server is similar to a shared data tables FOR in terms of the function it performs, and it is operationally similar to a temporary storage data sharing server.

For any given CFDT pool, there must be a CFDT server in each z/OS that wants to access it. CFDT pools are defined in the CFRM policy, and the pool name is then specified in the startup JCL for the server.

The CFDT server runs as a separate address space within a z/OS image, as either a started task or a batch job. If all of the CFDTs are in a single pool, only one CFDT server is needed in each z/OS image in the sysplex. If the CFDTs are divided into two separate pools, two CFDT servers are needed in each z/OS, and so on.

The CFDT server runs in a non-CICS address space, and requires little management or maintenance. The execution environment for CFDT server regions is provided by a runtime package called the Authorized Cross-Memory (AXM) server environment. The AXM server supports CFDT server regions and cross-memory connection services. This environment is also used by the TS data sharing server.

Sysplex-wide enqueue (ENQ) and dequeue (DEQ)

The sysplex-wide (global) enqueue and dequeue function enables CICS transactions running in the same region, or in different regions within a sysplex, to serialize on a named resource using the existing CICS API. By extending the scope of the CICS enqueue mechanism, a major source of inter-transaction affinity is removed, enabling better exploitation of Parallel Sysplex environments, improving price/performance, capacity, and availability.

For example, serialization makes it possible for concurrent updates to shared Temporary Storage queues by multiple CICS transaction instances, while locking a shared Temporary Storage queue against concurrent updates. This eliminates the race problem created by relying on serial reuse of a principal facility.

The main points of the changes to the CICS enqueue/dequeue mechanism are as follows:

- Sysplex enqueue and dequeue expands the scope of an EXEC CICS ENQ/DEQ command from region to sysplex, by introducing a new CICS resource definition type, ENQMODEL, to define resource names that are to be sysplex-wide.
- ENQSCOPE, an attribute of the ENQMODEL resource definition, defines the set of regions that share the same enqueue scope.
- When an EXEC CICS ENQ (or DEQ) command is issued for a resource whose name matches that of an installed ENQMODEL resource definition, CICS checks the value of the ENQSCOPE attribute to determine whether the scope is local or sysplex-wide, as follows:
  - If the ENQSCOPE attribute is left blank (the default value), CICS treats the ENQ/DEQ as local to the issuing CICS region.

- If the ENQSCOPE attribute is non-blank, CICS treats the ENQ/DEQ as sysplex-wide, and passes a queue name and the resource name to GRS to manage the enqueue. The resource name is as specified on the EXEC CICS ENQ/DEQ command, and the queue name is made up by prefixing the 4-character ENQSCOPE with the letters DFHE.

- The CICS regions that need to use the sysplex-wide enqueue/dequeue function must all have the required ENQMODELs defined and installed. The recommended way to ensure this is for the CICS regions to share a CSD, and for the initialization GRPLISTs to include the same ENQMODEL groups.

Named counter server

The named counter sequence number server provides a facility for generating sequence numbers for use by application programs, both CICS and batch, in a Parallel Sysplex. The named counter server is modeled on the other CF used by CICS and has many features in common with the CFDT server. The unique number generated could typically be used for an order or invoice number, implemented as a new key in a keyed file. Each named counter is held in a pool of named counters, which resides in a CF list structure. Retrieval of the next number in sequence from a named counter is done through a callable programming interface available to CICS and batch programs.

Resource Definition Online (RDO) for CICS temporary storage

RDO for temporary storage eliminates the need to prepare a temporary storage table (TST) for batch assembly and link-edit. There is now no need to shut down and restart CICS in order to make changes to TS queue definitions. RDO support for TS queues is part of the CICS high availability and continuous operations strategy.

4.2.6 CICSPlex SM workload management in a Parallel Sysplex

CICSPlex SM workload management is done in either:

- Goal mode
- Queue mode

Goal mode

The aim of the goal algorithm is to select the target region that is best able to meet the defined, average response-time goals for all work in a workload.

The goal is defined by associating transactions, via the Workload Manager component of z/OS, to a service class. Service classes are assigned on a transaction, LU name, and user ID basis. Service classes can define several types of response-time goals, but CICSPlex SM recognizes only average response-time goals. If transactions are given velocity, percentile, or discretionary goals, they are assumed to be meeting their goals. CICSPlex SM manages at the service-class level (it has no internal knowledge of the transaction characteristics). By consistently allocating service classes to sets of target regions, it minimizes the amount of resource reallocation by the MVS Workload Manager.

It is important for the Service Level Administrator to define goals that are realistic for the underlying capacity of the target systems. Transactions of like attributes (for example, transactions that have similar resource consumption, or pseudo-conversational transactions) should be assigned to distinct service classes. (The response-time goals can be the same for several service classes.) CICS statistics should be used to help you define these transaction sets. (See the Performance Guide for your release of CICS for information about CICS statistics.)

In order for the goal algorithm to be used, all requesting regions, routing regions, and target regions must be on z/OS images running in goal mode.

The goal algorithm is best suited to a symmetrical target region/MVS configuration (in terms of the number of target regions per MVS image), with a number of service classes that is comparable to the number of target regions in a given MVS image.

When CICSPlex SM is operated in goal mode with average response time goals, the following events occur:

1. A transaction arrives at a CICS terminal-owning region or Listener region (the routing region).
2. The routing region passes the transaction's external properties, such as LU name, user ID, and so on, to z/OS WLM.
3. z/OS WLM uses this information to assign a service class. The service class name is passed back to the routing region.
4. The TOR calls DFHCRP for transaction routing. Among other information, the service class name is passed in a comm_area.
5. DFHCRP in turn calls EYU9XLOP (CICSPlex SM).
6. If CICSPlex SM does not already have information about the goal for that service class, it will request that information from WLM.
7. Having the goal of the transaction, CICSPlex SM selects the *best* AOR. The name of this AOR is passed back to the routing region, which then routes the transaction to the selected AOR. The selection process is the following:

   a. Route all transactions belonging to a service class that are failing to meet their goals to a specific AOR.

   b. Those transactions that are meeting their goals are routed to another AOR.

   c. Those transactions that are exceeding their goals are routed to another AOR.

   These AORs could be in the same z/OS, or in another z/OS in the Parallel Sysplex, or in a remote z/OS. However, the algorithm will favor local AOR regions, in an effort to minimize the routing overhead. AORs that are prone to abend will not be favored, although they may appear to have a very short transaction response time. Refer to the IBM Redbook *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager,* GG24-4286 for more information.

Note the following:

- Other than requesting a copy of the service policy, there is no other interaction between CICSPlex SM and the WLM component of z/OS.
- All CICS regions: AORs, TORs, FORs, Listener regions, and so on, continue to report performance information directly to z/OS WLM. This behaves in the same manner regardless of the presence or absence of CICSPlex SM.
- CICSPlex SM has agents that run in all of the CICS regions and pass performance information back to the CICSPlex SM CMAS so that it is aware of the performance of all the regions that are part of that CICSplex.

Queue mode

If z/OS is being operated in goal mode, you still have the choice of running the CICS transactions with the *join shortest queue* algorithm. This is sometimes referred to as *queue mode*. If you are not currently running in goal mode and all your transactions are achieving their goal, there is no immediate need to switch CICSPlex SM to goal mode. However, if the the workload starts to increase, and some transactions start missing their goals, it is probably a good time to make the switch to goal mode.

If you have at least one CICS region operating in compatibility mode, the routing decision is made as follows:

1. A transaction arrives in a CICS TOR or Listener region.
2. CICS passes control to CICSPlex SM for dynamic workload balancing.
3. CICSPlex SM selects the least used AOR, which is the one that has the *smallest queue* (also called queue mode) of waiting transactions. This AOR could be in the same z/OS, or in another z/OS in the Parallel Sysplex, or in a remote z/OS. However, the algorithm tends to favor local AOR regions.

Note: If you have CICS and you do not install CICSPlex SM, it is possible to implement dynamic workload distribution by writing exit routines in the TOR or Listener region.

4.3 Database management in Parallel Sysplex

In this section, we look at database management software. The main focus is on how to set up the software for optimal exploitation of the Parallel Sysplex environment. The following database management software is covered:

- DB2
- IMS DB
- CICS/VSAM RLS

4.3.1 DB2 data sharing considerations

DB2 supports two types of data sharing:

- Shared read-only, which does not exploit the sysplex and allows multiple DB2 systems share data with read access only.
- Full read/write sharing, which requires a Parallel Sysplex and allows multiple DB2 subsystems to have read *and* write access to shared databases.

DB2 read-only data sharing

In a DB2 read-only data sharing environment, one DB2 owns data in a given shared database and has exclusive control over updating the data in that database. A database is a logical construct that contains the physical data in the form of index spaces and tablespaces. We use the term owner or owning DB2 to refer to the DB2 subsystem that can update the data in a given database. Other DB2s can read, but not update, data in the owner's database. These other DB2s are read-only DB2s, or readers.

You do not create any physical data objects on the reader, but you do perform data definition on the reader. This is so that the reader's catalog can contain the data definitions that mimic data definitions on the owning DB2. This allows applications running on the reader to access the physical data belonging to the owner. With shared read-only data, the owning DB2 *cannot update* at the same time the other DB2s are reading. Any read-only access must be stopped

before any updates can be done. Support for shared read-only data has been dropped beginning with DB2 UDB for OS/390 V6.

DB2 read-write data sharing

In DB2 V4, IBM introduced a function that provides applications with full read and write concurrent access to shared databases. DB2 data sharing allows users on multiple DB2 subsystems to share a single copy of the DB2 catalog, directory, and user data sets. The DB2 subsystems sharing the data belong to a DB2 data sharing group. The DB2 subsystems must reside in a Parallel Sysplex and use of a CF is required. The advantages of DB2 data sharing are as follows:

- Availability
- Flexible configurations
- Improved price performance
- Incremental growth and capacity
- Integration of existing applications

Data sharing group

A data sharing group is a collection of one or more DB2 subsystems accessing shared DB2 data. Each DB2 subsystem belonging to a particular data sharing group is a member of that group. All members of the group use the same shared catalog and directory. The maximum number of members in a group is 32. A data sharing environment means that a group has been defined with at least one member. A non-data sharing environment means that no group has been defined. A DB2 subsystem can only be a member of one DB2 data sharing group.

It is possible to have more than one DB2 data sharing group in a Parallel Sysplex. You might, for example, want one group for testing and another group for production data. Each group's shared data is unique to that group. DB2 assumes that all data is capable of being shared across the data sharing group.

Actual sharing is controlled by CPC connectivity and by authorizations. However, DB2 does not incur unnecessary overhead if data is not actively shared. Controls to maintain data integrity go into effect only when DB2 detects inter-subsystem read/write interest on a page set.

The following DASD resources must be shared by a data sharing group:

- Single shared OS/390 catalog: User catalogs for DB2 must be shared to avoid ambiguities.
- Single shared DB2 catalog: Any resource dedicated to one system (for example, any DB2 tablespace), must be unique in the DB2 sharing group.
- Single shared DB2 directory.
- Shared databases.
- The LOG data sets are unique to each DB2 member; however, they are read by all members of the data sharing group.
- The boot strap data sets (BSDSs) are unique to each DB2 member; however, they are
- We recommend that DB2 work files also be shared between data sharing members. This enables you to restart DB2 on other systems when needed. Shared work files are required in order to enable the use of Sysplex Query Parallelism.

As we see in Figure 4-37, each DB2 system in the data sharing group has its own set of log data sets and its own bootstrap data set (BSDS). However, these data sets *must* reside on DASD that is shared between all members of the data sharing group. This allows all systems access to all of the available log data in the event of a DB2 subsystem failure.

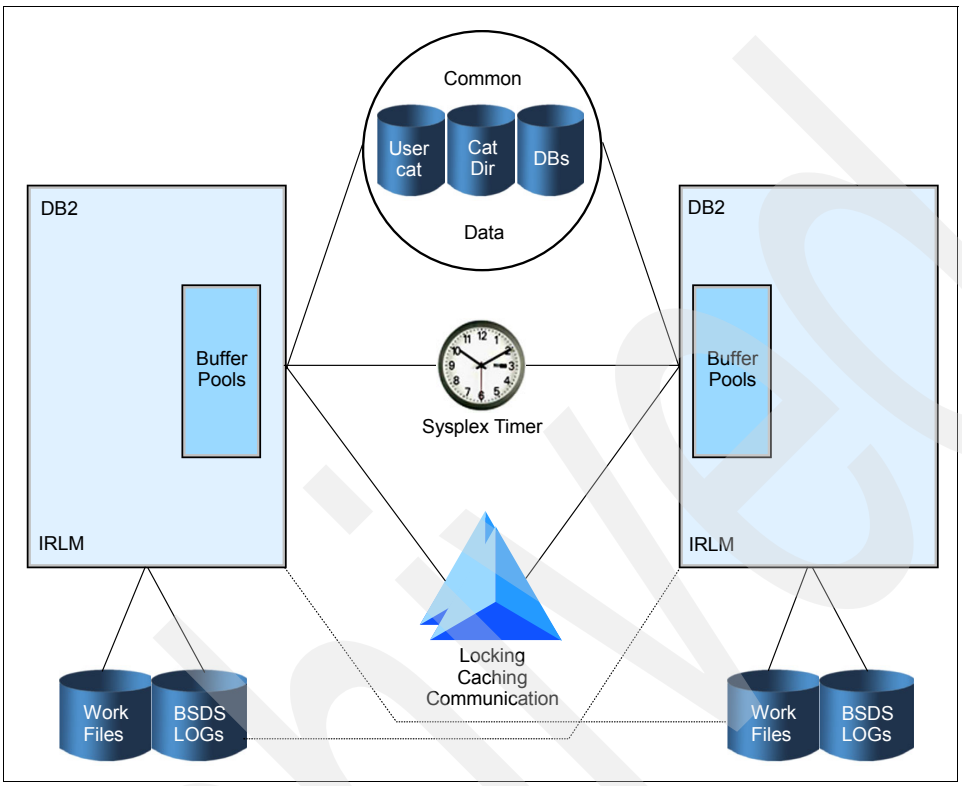


Figure 4-37 DB2 Data Sharing Group in a Parallel Sysplex

Single system image

A data sharing group presents a single system image using one DB2 catalog and conceptually sharing all user databases. Up to 32 DB2 subsystems can read and write to the same databases. The CF allows tables to be treated like local tables from any DB2 in the data sharing group on different OS/390s.

A transaction executed in one system is dependent on its log files. Therefore, for a non-distributed transaction, only the DB2 that began the transaction keeps all of the information needed to successfully complete it. The same is true for the transaction manager.

All data across a DB2 data sharing group is capable of being shared. Any table or tablespace is assumed to be shared across the DB2 data sharing group, including the DB2 catalog and directory. The physical connections required for data sharing are assumed to be available. DB2 dynamically optimizes data access when only one DB2 is accessing it.

Although authorization changes are effective across the DB2 data sharing group, actual sharing is controlled by physical DASD/CPC connectivity. GRANT and REVOKE need to be issued only once and are valid across the data sharing group.

Data access concurrency is supported at every level, and data sharing is transparent to the DB2 user. For example, row level locking appears to be the same whether done in a data sharing environment or not. Locking is done only by the IRLM - IRLM then uses XES services to communicate with the CF; however, this is transparent to the user or application developer.

How DB2 data sharing works

This section provides background information about how shared data is updated and how DB2 protects the consistency of that data. For data sharing, you must have a Parallel Sysplex.

Data is accessed by any DB2 in the group. Potentially, there can be many subsystems reading and writing the same data. DB2 uses special data sharing locking and caching mechanisms to ensure data consistency.

When one or more members of a data sharing group have opened the same tablespace, index space, or partition, and at least one of them has been opened for writing, then the data is said to be of *inter-DB2 R/W interest* to the members (we shorten this to *inter-DB2 interest*). To control access to data that is of inter-DB2 interest, DB2 uses the locking capability provided by the CF. DB2 also caches the data in a storage area in the CF called a GBP structure, whenever the data is changed.

When there is inter-DB2 interest in a particular tablespace, index, or partition, it is dependent on the GBP, or GBP-dependent. You define GBP structures using CFRM policies. For more information about these policies, see *OS/390 V2R10.0 MVS Setting Up a Sysplex,* GC28-1779.

There is mapping between a GBP and the local buffer pools of the group members. For example, each DB2 has a buffer pool named BP0. For data sharing, you must define a GBP (GBP0) in the CF that maps to buffer pool BP0. GBP0 is used for caching the DB2 catalog, directory tablespaces and index along with any other tablespaces, indexes, or partitions that use buffer pool 0.

To make a particular database eligible for sharing, you would define a Group Buffer Pool corresponding to the local buffer pool being used by the database. If you have a database that you do not wish to share, you would simply not define a GBP. You can put GBps in different CFs. GBps are used for caching data of interest to more than one DB2, to cache pages read in from DASD, and as a cross-invalidation mechanism for buffer coherency across DB2 members.

When a particular page of data is changed by one DB2, DB2 caches that page in the GBP. The CF invalidates any image of the page in the local buffer pools of any other members that currently have a copy of that page. Then, when a request for that same data is subsequently made by another DB2, it looks for the data in the GBP.

Figure 4-38 shows the process of DB2 data sharing in more detail.

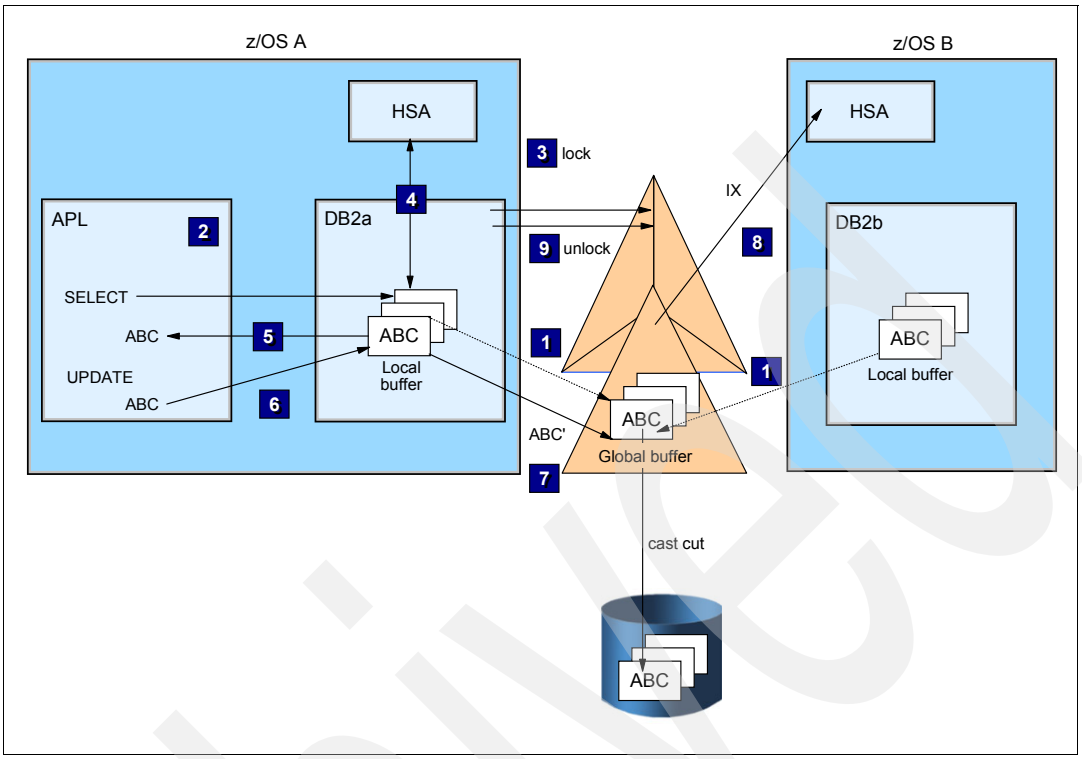


Figure 4-38 DB2 data sharing in a Parallel Sysplex

In 4.3.1, “DB2 data sharing considerations” on page 228, both OS/390 A and OS/390 B have registered with the CF:

1. Their interest in page ABC, a current copy of which exists in each systems local buffers.
2. An application in OS/390 A needs to update the page to ABC'.
3. DB2 in OS/390 A calls the CF to obtain an exclusive lock for the update.
4. OS/390 A checks its HSA vector table.
5. To ensure that the page in its buffer is a valid copy, OS/390 A changes the page to ABC', which is subsequently changed in its local buffer and written to the GBP in the CF.
6. The CF now invalidates OS/390 B's local buffer.
7. By changing the bit setting in the HSA vector associated with the page ABC, the lock held by OS/390 A on the page is released.

For a further description of this process, refer to *DB2 UDB for OS/390 V6 Data Sharing: Planning and Administration,* SC26-9007. For a discussion on buffer pool sizing for DB2, use the CF Sizing tool, available at:

`http://www.s390.ibm.com/cfsizer/`

Writing changed data to DASD

DB2 uses a *castout* process to write changed data to DASD from a GBP. When data is cast out from a GBP to DASD, that data must first pass through a DB2's address space because there is no direct connection from a CF to DASD. This data passes through a private buffer, not DB2's virtual buffer pools. You can control the frequency of castouts with thresholds and checkpoints for each GBP. Any of the DB2 subsystems in the data sharing group may do the castout. The DB2 that is assigned ownership of castout is the DB2 subsystem that had the

first update intent (except during restart) on the page set or partition. After the castout ownership is assigned, subsequent updating DB2 subsystems become backup owners. One of the backup owners becomes the castout owner when the original castout owner no longer has read/write interest in the page set.

DB2 usage of CF structures

During startup, DB2 members join one XCF group, and the associated integrated resource lock managers (IRLMs) join another XCF group. To join, they use the names you specify during DB2 installation.

The Sysplex Timer keeps the CPC time stamps synchronized for all DB2s in the data sharing group. DB2 uses a value derived from the time stamp to replace the RBA when sharing data.

At least one CF must be installed and defined to OS/390 before you can run DB2 with data sharing capability. Before starting DB2 for data sharing, you must have defined one lock structure and one list structure. One-way data sharing does not require the definition of a cache structure; it is only when you go to more than one-way data sharing that you must define at least one cache structure (GBP 0). DB2 uses the three types of CF structures as follows:

- *Cache structures*: Cache structures are used as GBps for the DB2 data sharing group. DB2 uses a GBP to cache data that is of interest to more than one DB2 in the data sharing group. GBps are also used to maintain the consistency of data across the buffer pools of members of the group by using a cross-invalidating mechanism. Cross-invalidation is used when a particular member's buffer pool does not contain the latest version of the data.

  If data is not to be shared (that is, it will be used only by one member), then choose a buffer pool for those non-shared page sets that do not have a corresponding GBP. Assume you choose BP6. Every other member must define its virtual buffer pool 6 with a size of 0 and there should not be a GBP6 defined.

  Depending on the GBPCACHE option specified, GBps may cache shared data pages, which are registered in the GBP directory. This registration allows XES cache structure support to cross invalidate the data pages when necessary. Changes to a registered resource invalidate all other registered copies of that resource in the local buffer pools.

  Reuse of an invalidated resource results in a reread of that resource. The XES cross invalidation advises DB2 to reread (from either CF or DASD) the page when needed. The CF invalidates a given data buffer in all of the local buffer pools when the data is changed by another subsystem.

  Cache structure services, accessed by DB2, provide the following functions:

  - Automatic notification to affected DB2s when shared data has been changed (cross-system invalidation). The CF keeps track of which DB2s are using a particular piece of data, and the CF updates a bit in the HSA on each CPC that contains a copy of that record.
  - Maintenance of cache structure free space. The CF maintains lists of which entries in a cache structure have been changed and which have not. These lists are kept in the order of most recently used. When the CF needs to reclaim space in a structure, it does so from the list of unchanged entries, using the oldest (or least recently used). Entries on the changed list are not eligible for reclaim.
  - Data may be read and written between a GBP and a local buffer pool owned by a single DB2.
  - Maintenance of a secondary copy of the GBP. GBP duplexing is available with DB2 V5 (via APAR PQ17797) and later releases.

The GBP in DB2 V4 and later releases is implemented as a CF cache structure. Each GBP contains:

- Directory entry

  The directory entry contains references for each page represented in the GBP. It has slots for all users of that particular page, and is used, for example, to notify all users of cross-invalidation for that page. The entry indicates the position of the data. Shared data can be located in the related local buffer pool of more than one member of a data sharing group at the same time.

- Data entry (GBPCACHE ALL, CHANGED, or SYSTEM)

  The GBP pages are implemented as data entries in the cache structure of the CF. The GBP is maintained by the participating DB2s. It contains the GBP-dependent pages. Data entries are either 4 KB, 8 KB, 16 KB, or 32 KB.

- *List structure*: There is one list structure per data sharing group used as the shared communications area (SCA) for the members of the group. The SCA keeps DB2 data sharing group member information; it contains recovery information for each data sharing group member. The first connector to the structure is responsible for building the structure if it does not exist.

  The SCA is used in DB2 V4 and later releases to track database exception status.

- *Lock structure*: One lock structure per data sharing group is used by IRLM to control locking. The lock structure contains global locking information for resources on behalf of the requesting IRLM and DB2. It protects shared resources and allows concurrency. The system lock manager (SLM), a component of XES, presents the global lock information to the lock structure.

  A lock structure is used to serialize on resources such as records/rows, pages, or tablespaces. It consists of two parts:

  - A CF lock list table, which consists of a series of lock entries that associate the systems with a resource name that has been modified, including a lock status (LS) of that resource.

    A resource is any logical entity, such as a record/row, a page, partition, or an entire tablespace. DB2 defines the resources for which serialization is required. A CF lock list table has information about the resource used by a member DB2 and is used for recovery if a DB2 fails. One common set of list elements is used for all members of the data sharing group.

  - CF lock hash table. Each hash entry is made up of one SHARE bit for each member of the data sharing group and one EXCLUSIVE byte.

Data sharing enhancements in DB2 V5 and V6

Since the initial introduction of data sharing support in DB2 V4, there have been numerous enhancements to DB2 to further improve performance and availability in a data sharing environment. In this section, we briefly review those enhancements. For more information, refer to *DB2 for OS/390 V5 Release Guide,* SC26-8965, *DB2 UDB for OS/390 V6 Release Planning Guide,* SC26-9013, and *DB2 Universal Database™ Server for OS/390 V7 What's New?,* GC26-9017.

The sysplex-related enhancements in DB2 V5 and V6 are as follows:

- Improvements in query processing:

  The full power of a Parallel Sysplex can be used not only to split a read-only query into a number of smaller tasks, but also to run these tasks in parallel across multiple DB2 subsystems on multiple CPCs in a data sharing group. *Sysplex Query Parallelism* is supported by combining the data from all parallel tasks, regardless of the data sharing

member on which they were executed. You still get an application view of the thread, just as for *CP Query Parallelism* in DB2 V4.1. TCB times of parallel tasks running on CPCs with different processor speeds are adjusted so that they can be analyzed in a meaningful way.

- Enhanced data sharing support in DB2 V5
  - Simplifies the monitoring of applications in a data sharing environment, *group-scope* reports can be generated for accounting. This is especially helpful if you use OS/390 Workload Management to dynamically schedule your applications on different data sharing members. Accounting group-scope reports help you get a complete picture of the resources an application has used, regardless of which member it ran on.
  - GBP rebuild makes CF maintenance easier and improves access to the GBP during connectivity losses. Automatic GBP recovery accelerates GBP recovery time, eliminates operator intervention, and makes data available faster when GBps are lost because of CF failures.
  - Improved restart performance for members of a data sharing group reduces the impact of retained locks by making data available faster when a group member fails.
  - Continuous availability with GBP duplexing in a Parallel Sysplex makes recovery simpler and faster in the event of a CF failure (requires APAR PQ17797).
- DB2 Usage of Sysplex Routing for TCP/IP:

  DB2 provides a service called sysplex routing for TCP/IP DRDA requesters. This allows systems connecting to the DB2 data sharing group using DRDA over TCP/IP connections to have their requests routed to the DB2 server that WLM determines to be the least loaded.

  At the time of writing, the DRDA requesters enabled for this function are another OS/390 for DB2 requester, DB2 Connect Enterprise Edition, and the DRDA client on a PC. This function is similar to that introduced for DRDA requesters using APPC in DB2 V4.1.
- Query performance enhancements in DB2 V6 include:
  - Query parallelism extensions for complex queries, such as outer joins and queries that use non-partitioned tables.
  - Faster restart and recovery with the ability to postpone backout work during restart, and a faster log apply process.
  - Increased flexibility with 8 KB and 16 KB page sizes for balancing different workload requirements more efficiently, and for controlling traffic to the CF for some workloads.
  - An increased log output buffer size (from 1000 4 KB to 100000 4 KB buffers) that improves log read and write performance.
- Data sharing enhancements in DB2 V6:
  - Continuous availability with GBP duplexing in a Parallel Sysplex makes recovery simpler and faster in the event of a CF failure.
  - More caching options for using the CF improve performance in a data sharing environment for some applications by writing changed pages directly to DASD.
- Data sharing enhancements in DB2 V7:
  - Restart Light

    A new feature of the START DB2 command allows you to choose Restart Light for a DB2 member. Restart Light allows a DB2 data sharing member to restart with a minimal storage footprint, and then to terminate normally after DB2 frees retained locks. The reduced storage requirement can make a restart for recovery possible on a system that might not have enough resources to start and stop DB2 in normal mode. If

you experience a system failure in a Parallel Sysplex, the automated restart in light mode removes retained locks with minimum disruption.

Consider using DB2 Restart Light with restart automation software, such a Automatic Restart Manager.

– Persistent structure size changes

In earlier releases of DB2, any changes you make to structure sizes using the SETXCF START,ALTER command might be lost when you rebuild a structure and recycle DB2. Now you can allow changes in structure size to persist when you rebuild or reallocate a structure.

– Faster shutdown of DB2 data sharing members

You can more easily apply service or change system parameters. A new CASTOUT(NO) option on the -STO DB2 command enables a faster shutdown of DB2 data sharing members.

– New global scope for IFI calls and for options of several commands

Several recent enhancements to the Instrumentation Facility Interface (IFI) and some commands help you manage your data sharing environment more easily. Now, information from all the data sharing members can be available with a single call from any one of the members.

4.3.2 IMS DB data sharing

IMS DB has long supported two levels of *data sharing:*

- Database level data sharing
- Block level data sharing

For the purposes of this redbook, we shall only consider *block level data sharing,* because that is the level of sharing that exploits the Parallel Sysplex.

IRLM has always been used to provide data integrity; however, there were limitations. On any image, IRLM could support multiple IMSs, but cross-image connectivity was limited to two IRLMs. This restriction was primarily due to the performance of the connectivity, which was using VTAM CTCs.

IMS DB data sharing in a Parallel Sysplex

When multiple IMS systems share data across *multiple* images and use a CF, it is known as *sysplex data sharing*.

Two elements necessary for data sharing in a Parallel Sysplex are:

- The ability to lock data for update across up to 32 images
- The notification of those sharing systems when data has been modified

Since the introduction of IMS/ESA V5 DB and IRLM V2.1, the number of IRLM connections has increased to 32, within a Parallel Sysplex by the use of a lock structure.

For information about the sizing of this structure, refer to the CF Sizer tool, available at:

`http://www.s390.ibm.com/cfsizer/`

The notification of change to data is an IMS DB function that was introduced in IMS/ESA V5. The implementation was through the use of two cache structures, one for OSAM and one for VSAM. These structures were *directory only* and therefore did not contain actual database data, only a list of the subsystems with an interest in the validity of that data.

In IMS/ESA V6, the OSAM structure was changed so that it now holds data as well as the directory; OSAM structures use the store-through cache algorithm. This structure may therefore be substantially larger than the corresponding IMS/ESA V5 structure.

For information about the sizing of these structures, refer to the CF Sizer tool, available at:

http://www.s390.ibm.com/cfsizer/

Fast path databases

IMS/ESA V6 introduced support for n-way data sharing of Data Entry Data Bases (DEDBs) with Virtual Storage Option (VSO) areas and Sequential Dependents (SDEPs).

The support for VSO is implemented by the use of one or two store-in cache structures for each area. The data is initially written to the cache and periodically cast out to DASD. This provides the equivalent function to IMS/ESA V5, where the data was held in a data space and cast out from there.

The support for Sequential Dependents (SDEPs) does not use CF structures; it is implemented through changes to the segments and CIs combined with an algorithm for selecting which CIs the SDEPs are stored in.

IMS data sharing groups

As with DB2, there is the concept of a data sharing group. Figure 4-39 shows a sample IMS data sharing group, with the basic components required for n-way data sharing.

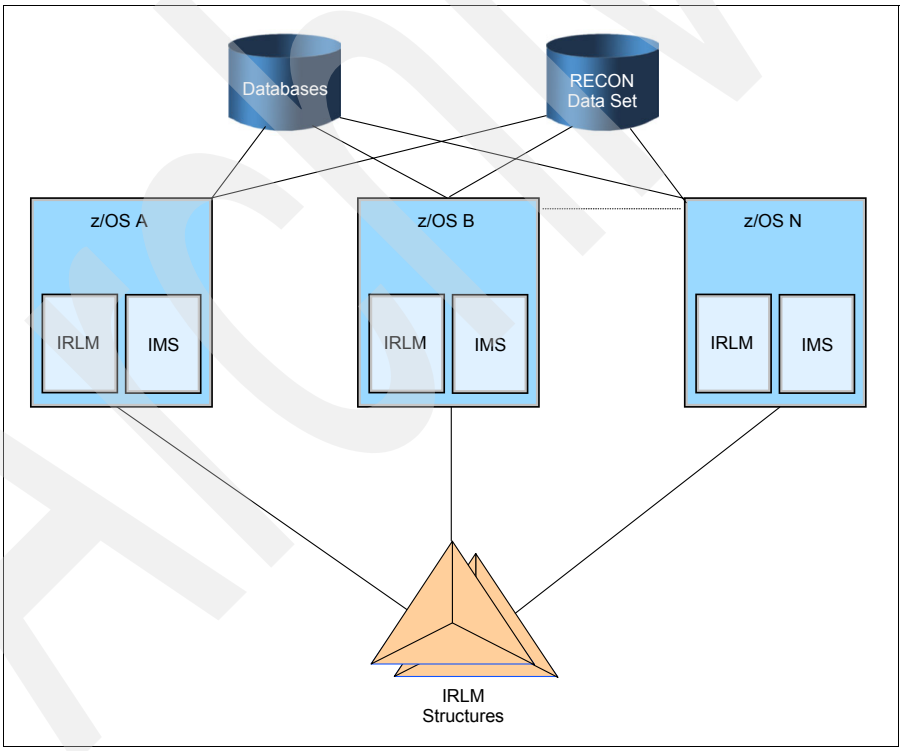


Figure 4-39 Sample data sharing group in an IMS environment

The members (IMS subsystems) of the data sharing group share:

- Databases
- A single set of RECON data sets
- One or more CFs

- A single IRLM lock structure in the CF (hereafter called an IRLM structure)
- OSAM, VSAM, and VSO DEDB cache structures in the CF (hereafter called OSAM, VSAM, and VSO DEDB structures)

A data sharing group is connected to at least one CF. Without this, IRLM V2.1 is unable to allow inter-processor data sharing. If a CF is not available, IRLM V2.1 only allows intra-processor data sharing. A lock structure is required for sharing lock information between IRLMs.

Record updating and buffer invalidation

When a block of data is read into a local buffer by any of the sharing IMSs, it is registered in an entry in the appropriate cache structure. Each entry consists of a field for the buffer ID (known to OS/390 as the resource name) and 255 slots. The slots are for IMS subsystems to register their interest in an entry's buffer. If it is VSAM data, there is no further action; if it is OSAM, then the data may be cached, depending on the definition of the IOBF subpool. There are three options:

- *No caching*: In which case, processing is the same as for VSAM.
- *Cache all data*: In which case, the data is stored in the structure.
- *Cache only changed data*: In which case, the data is only stored in the cache after it has been changed and written back to DASD.

When an IMS wants to modify a block, it must first request a lock from IRLM. The IRLM structure is checked to see if the block is already locked. If the block is not locked, the structure is updated and the lock is granted. When the block has been updated and written back to DASD, the appropriate cache structure is updated to mark the buffer invalid. This causes the CF to notify all IMSs that have registered an interest that their local copy of the block is invalid. Finally, the lock may be released.

Whenever IMS tries to access a buffer, it checks whether the buffer is still valid. If the buffer is invalid, IMS rereads the data from the structure (if it is there) or from DASD. With IMS/ESA V6, OSAM and VSO buffers may be refreshed from the CF rather than from DASD. Thus, data integrity is maintained. Buffer invalidation works in all IMS sysplex database environments: DB/DC, DBCTL, and DB batch. In the sysplex environment, IMS supports buffer pools for VSAM, VSAM Hiperspace™, OSAM, OSAM sequential buffering, and VSO DEDB.

Connectivity to structures

For a data sharing group, the first IMS to connect to an IRLM determines the data sharing environment for any other IMS that later connects to the same IRLM. When identifying to IRLM, IMS passes the names of the CF structures specified on the CFNAMES control statement (in the IMS PROCLIB data set member DFSVSMxx), plus the DBRC RECON initialization time stamp (RIT) from the RECON header. The identify operation fails for any IMS not specifying the identical structure names and RIT as the first IMS. The term *connection* can refer to the following subsystems: either an IMS TM/DB subsystem, IMS batch, or a DBCTL subsystem for use by, for example, CICS.

Recommendation to convert IMS batch to BMP batch: It is worthwhile to determine if any of the IMS batch jobs can be converted to batch message processing programs (BMPs). The primary reason for converting an IMS batch job to a BMP is to take advantage of the availability benefits of BMPs. When BMP batch jobs abend, IMS automatically backs them out and does not create any lock reject conditions. This is rarely true for batch jobs. Also, when an IRLM abends or the lock structure fails, batch jobs abend. BMPs do not abend in this situation. Secondly, BMPs provide a better granularity of locking than IMS batch jobs. Of relatively minor importance (since the number of connections to a cache structure was raised from 32 to 255) is the fact that a single IMS batch job constitutes one connection to the cache structure. Up to 255 connections are allowed to a cache structure. The BMPs will run using the single connection of the IMS control region.

There may be some additional overhead associated in converting to BMPs, caused by the sharing of the database buffer pools with other dependent regions.

Recommendation to release locks frequently: Ensure that applications issue CHKP calls periodically to release locks held when executed as a BMP. The same consideration applies to batch.

IMS APARs PQ26416 and PQ26491 and OS/390 APAR OW38840 deliver new function where the overhead of connecting and disconnecting to the CF cache structures for batch DL/I jobs is significantly reduced. This is especially beneficial for jobs with very short elapsed times, where the communication with XES can make up a considerable portion of the total elapsed time of the job.

Another enhancement of interest is Long Lock Detection Reporting. This was introduced by IMS APARs PN84685 (IMS V5), PQ07229 (IMS V6), PN79682 (IRLM 2.1), and OW20579 (RMF). These APARs provide support for a new RMF report that shows which tasks have been holding a lock for a long time, and which tasks are impacted as a result.

IMS database types eligible for data sharing

The following database organizations are supported by IMS/ESA V6.1 for data sharing:

- HIDAM
- HDAM
- HISAM
- SHISAM
- Secondary indexes
- DEDBs

Programs using these types of databases usually do not require any changes to function in a data sharing environment.

IMS DB V5 database types not eligible for data sharing

The following database organizations are *not* supported by IMS/ESA V5.1 for data sharing:

- DEDBs which use either of the following options:
  - Sequential dependent segments (SDEPs)
  - Virtual storage option (VSO)
- MSDBs

IMS/ESA V5 and V6 coexistence for OSAM data sharing

Data sharing is valid with a mixture of IMS/ESA V5 and IMS/ESA V6 systems, as long as you do not use OSAM data caching.

Data sharing of OSAM data is allowed as long as IMS/ESA V6 does not put data in the structure. At the time of writing, IMS/ESA V6 code is being added to allow IMS/ESA V6.1 to allow the user to specify the ratio of directory entries to data elements. The CFOSAM parameter is used to specify the directory-to-element ratio with few data elements. The ratio may be as high as 999:1. The IOBF statement in the DFSVSMxx member can be used to control the amount of data caching, and should be set to specify *no* data sharing.

Fast database recovery

IMS/ESA V6 introduced the Fast Database Recovery (FDBR) region which, using XCF monitoring, can track an active IMS while either being on the same or another image.

In the event of the active system failing, the FDBR code will dynamically back out, in-flight, full function database updates, invoke DEDB redo processing, and purge retained locks from IRLM.

XRF and Sysplex data sharing

XRF can be used in a sysplex data sharing environment if the CF is available and connected when the alternate system is brought up.

4.3.3 CICS/VSAM record level sharing considerations

CICS/VSAM record level sharing (RLS) is a data set access mode that allows multiple address spaces, CICS application owning regions (AORs) on multiple systems, and batch jobs to access VSAM data at the same time. With CICS/VSAM RLS, multiple CICS systems can directly access a shared VSAM data set, eliminating the need for function shipping between AORs and file owning regions (FORs). CICS provides the logging commit and rollback functions for VSAM recoverable files. VSAM provides record-level serialization and cross-system caching. CICS, not VSAM, provides the recoverable files function. Figure 4-40 on page 241 is an illustration of CICS/VSAM RLS implementation in Parallel Sysplex.

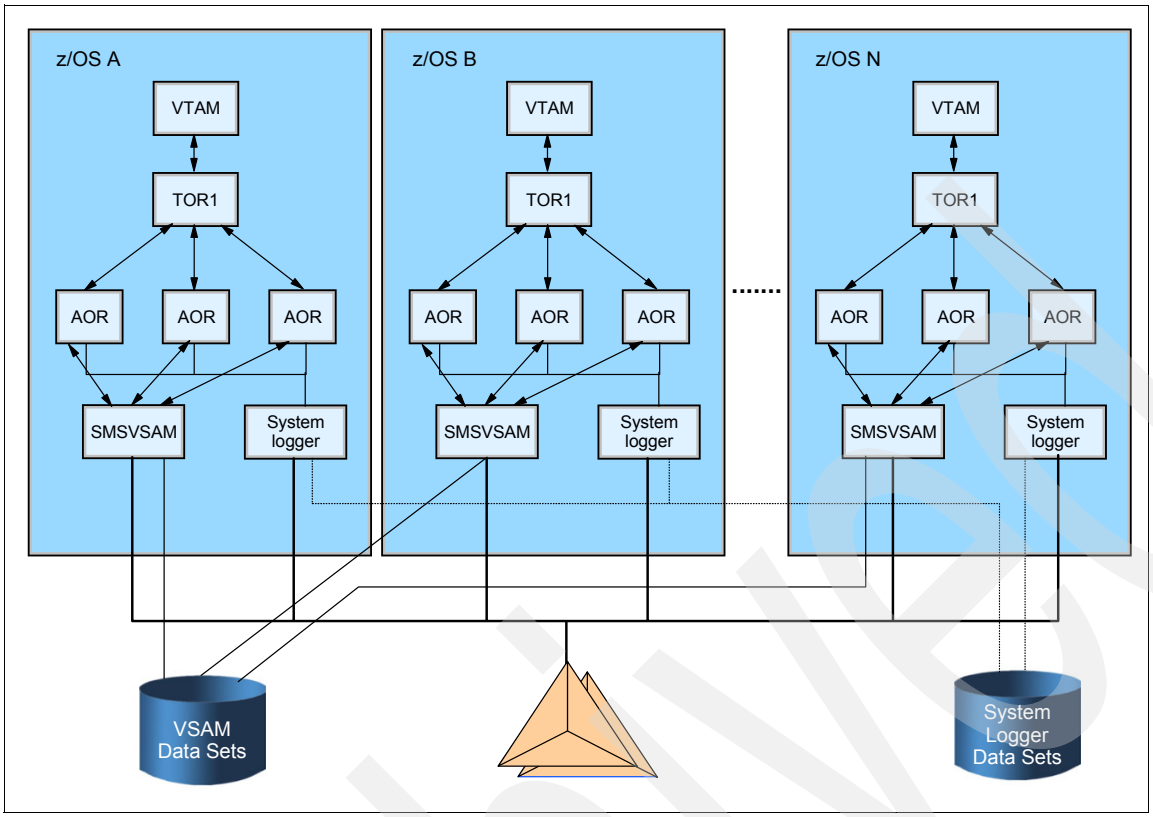


Figure 4-40 VSAM record level sharing in a Parallel Sysplex

The CICS/VSAM Record Level Sharing (RLS) data access mode allows multisystem access to a VSAM data set while ensuring cross-system locking and buffer invalidation. CICS/VSAM RLS uses XES services to perform data set level locking, record locking, and data caching. CICS/VSAM RLS maintains data coherency at the control interval level. It uses CF caches as store-through caches. When a control interval of data is written, it is written to both the CF cache and to DASD. This ensures that a failure in the CF cache does not result in the loss of VSAM data.

VSAM RLS improves the availability of VSAM data sets during both planned and unplanned outages, because if one of the OS/390 systems or CICS regions is not available, applications can access their data from another system or CICS region. It also provides data integrity and availability through the use of common locking and buffering.

VSAM RLS improves availability: VSAM RLS provides improvements for data integrity and availability.

The SMSVSAM server is the system address space used for CICS/VSAM RLS. The data space associated with the server contains most of the VSAM control blocks and the system-wide buffer pool used for data sets opened for record-level sharing. SMSVSAM assumes responsibility for synchronizing this control block structure across the sysplex.

The LOG attribute on the VSAM cluster (contained in the ICF catalog entry) defines a data set as recoverable or non-recoverable. Because CICS maintains the log of changed records for a data set (thus allowing transactional changes to be undone), only VSAM data sets under CICS are recoverable. Whether a data set is recoverable or not determines the level of sharing allowed between applications:

- Both CICS and non-CICS jobs can have concurrent read/write access to non-recoverable data sets.
- Non-CICS jobs can have read-only access to recoverable data sets, concurrent with read/write access by CICS transactions. Full read integrity is ensured.

More information about how VSAM RLS actually works, and comparisons between VSAM RLS processing and traditional VSAM processing, see the IBM Redbooks *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765, and *Batch Processing in a Parallel Sysplex*, SG24-5329.

Implementing CICS/VSAM RLS

To enable CICS/VSAM RLS, you must define one or more CF cache structures and add these to your SMS base configuration. Cache set names are used to group CF cache structures in the SMS base configuration. In setting up for CICS/VSAM RLS processing, you also need to define the CF lock structure.

A data set is assigned to a CF cache structure based on your SMS policies. If the storage class for a VSAM data set contains a non-blank cache set name, the data set is eligible for record-level sharing. When the data set is opened for RLS-processing, the cache set name is used to derive an eligible CF cache structure to use for data set access.

Restrictions are discussed in the CICS TS, OS/390, DFSMS, and sysplex reference libraries. Refer to the *CICS TS Program Directory* for specific product requirements. For planning and implementation information relation to VSAM RLS, refer to the IBM Redbook *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765.

Refer to CICS APAR II09698, Installation planning information, for the latest installation recommendations.

For more information about VSAM RLS, refer to *DFSMS/MVS V1R5 Planning for Installation,* SC26-4919.

Software dependencies

CICS/VSAM RLS support has the following prerequisites:

- MVS/ESA V5.2 or higher
- DFSMS 1.3 or higher
- CICS TS 1.1 or higher

In addition, some mechanism (GRS or an equivalent function) is required for multisystem serialization.

There are additional dependencies on the following program products, or equivalent products, for full-function support:

- CICSVR V2.3 or higher
- RACF V2.1 or higher
- EPDM V1.1.1 or higher
- Appropriate levels of COBOL, PL/I, FORTRAN, and Language Environment® runtime libraries, for batch applications that will use VSAM RLS data access

All products listed in this section need to be at a specific service level or have function-enabling PTFs applied.

Hardware dependencies

To exploit this support, you must have at least one CF connected to all systems that will be taking part in VSAM record level sharing. If you have multiple CFs, those CFs must be accessible from any system that may be using VSAM RLS.

Maximum availability recommendation: For maximum availability, we recommend that you set up at least two CFs with connectivity to all CPCs in the Parallel Sysplex. If a CF is not operational, this allows the storage management locking services to repopulate its in-storage copy of the locks to the secondary CF. A CF must be large enough to contain either a lock structure or a cache structure (or both), and have enough *white space* to allow the structures to be increased in size. You define a single CF lock structure, IGWLOCK00, to be used for cross-system record-level locking. The CF lock structure must have connectivity to all systems that will be taking part in VSAM RLS. A nonvolatile CF for the lock structure is not required, but it is highly recommended for high availability environments. If you have a volatile CF lock structure, and a power outage causes a Parallel Sysplex failure resulting in loss of information in the CF lock structure, all outstanding recovery (CICS restart and backout) must be completed before new sharing work is allowed.

In order for VSAM RLS to use the structures you have defined in the CFRM policy, you must also define them to SMS. CF cache structures associated with a given storage class must have, at a minimum, connectivity to the same systems as the storage groups mapped to that storage class.

In summary, we recommend that you set up multiple CFs for maximum availability and workload balancing, and ensure that these CFs have connectivity to all systems that will use CICS/VSAM RLS.

Coexistence issues

Within a Parallel Sysplex, an SMS configuration can be shared between systems running DFSMS/MVS V1.3 and other DFSMS/MVS systems; however, toleration PTFs must be applied to all pre-DFSMS/MVS V1.3 systems so that they do not conflict with RLS access. An SMS configuration containing VSAM RLS information must be activated by a system that is at least at the DFSMS V1.3 level.

For fallback from DFSMS/MVS V1.3, a DFSMS/MVS V1.3-defined SMS control data set is compatible with down-level DFSMS/MVS systems, if all toleration PTFs are applied to those systems.

GRS is required to ensure cross-system serialization of VSAM resources and other DFSMS/MVS control structures altered by CICS/VSAM RLS. Open requests from a system that is not at the DFSMS V1.3 (or higher) level are not allowed when RLS access to a data set is active, or if RLS transaction recovery for the data set is pending.

In a JES3 environment, you must be careful to define cache set names only in the SMS storage classes that are used by data sets opened for RLS processing. A non-blank cache set name causes a job to be scheduled on an RLS-capable system. If all storage classes have non-blank cache set names, then all jobs accessing SMS-managed data sets are scheduled to DFSMS/MVS V1.3 systems, causing an imbalance in workload between the DFSMS/MVS V1.3 systems and down-level systems.

SMS changes

To implement CICS/VSAM RLS, you need to modify your SMS configuration in the following ways:

- Add the CF cache structures to your SMS base configuration.

  You can have up to 255 CF cache set definitions in your base configuration. Each cache set can have one to eight cache structures (or buffer pools) defined to it, allowing data sets to be assigned to different cache structures in an effort to balance the workload.

  CF cache structures associated with a given storage class must have, at a minimum, the same connectivity as the storage groups mapped to that storage class.

  Having multiple CF cache structures defined in a cache set could provide improved availability. If a CF cache structure fails and a rebuild of the structure is not successful, SMS dynamically switches all data sets using the failed CF structure to other CF structures within the same cache set.

- Update storage class definitions to associate storage classes with CF cache set names. You also need to define the direct and sequential CF weight values in the storage classes.
- Change the ACS routines to recognize data sets that are eligible for VSAM RLS so that they can be assigned to storage classes that map to CF cache structures.

Storage requirements for SMS control data sets have also changed to accommodate CICS/VSAM RLS information.

Batch considerations

Transactional VSAM Services (TVS) adds the ability for batch jobs to update recoverable VSAM files concurrent with updates from CICS regions. While CICS provides its own logging capability, TVS will provide this capability for batch jobs. VSAM RLS will be used for locking and buffer coherency. It is envisioned that TVS will be used for files that require near-24x7 availability, but also have to be updated from batch. More on TVS can be found in *ABCs of z/OS System Programming Volume 3,* SG24-6983.

There are a number of restrictions relating to the use of VSAM files by batch jobs, where those files are also accessed in RLS mode by CICS. For a full list of the considerations for batch, refer to Chapter 14, “Using VSAM Record-Level Sharing” in *DFSMS/MVS V1R5 Using Data Sets,* SC26-4922, and the IBM Redbook *CICS and VSAM Record Level Sharing: Implementation Guide*, SG24-4766.

CICS considerations

There are also considerations for CICS in relation to its use of VSAM data sets in RLS mode:

- *Changes in file sharing between CICS regions*: Prior to VSAM RLS, shared data set access across CICS Application-Owning Regions (AORs) was provided by CICS function-shipping file access requests to a CICS File-Owning Region (FOR). CICS/VSAM RLS can now provide direct shared access to a VSAM data set from multiple CICS regions. The highest performance is achieved by a configuration where the CICS AORs access CICS/VSAM RLS directly.

  However, a configuration that places a CICS FOR between CICS AORs and CICS/VSAM RLS is supported as well. FORs can continue to exist to provide distributed access to the files from outside the sysplex.

- *LOG and LOGSTREAMID parameters for DEFINE CLUSTER*: These parameters replace the corresponding definitions in the CICS File Control Table (FCT). LOG specifies whether the VSAM sphere is recoverable (where CICS ensures backout) or non-recoverable. LOGSTREAMID specifies the name of a forward recovery log stream to use for the VSAM sphere.

- *BWO parameter for DEFINE CLUSTER*: If CICS/VSAM RLS is used, this is the mechanism for specifying backup-while-open in a CICS environment.
- *Sharing control*: Sharing control is a key element of this support. Careful consideration must be given to managing the sharing control data sets that contain information related to transaction recovery.
- *Recovery*: New error conditions affect subsystem recovery. These include loss of the lock structures, loss of a CF cache structure, SMSVSAM server failure, or errors during backout processing.
- *Batch job updates of a recoverable VSAM sphere*: CICS/VSAM RLS does not permit a batch job to update a recoverable sphere in RLS access mode (however, refer to the *Stop Press* note earlier in this section referring to Transactional VSAM Services). The sphere must be RLS-quiesced first, using a CICS command, and then the batch job can open the VSAM sphere for output using non-RLS protocols. If it is necessary to run critical batch window work while transaction recovery is outstanding; there are protocols that allow non-RLS update access to the VSAM sphere. Backouts done later must be given special handling. If you intend to use this capability, you need to plan for the use of the IDCAMS SHCDS PERMITNONRLSUPDATE and DENYNONRLSUPDATE commands and for the special handling of these transaction backouts.

Refer to *CICS Transaction Server for OS/390 V1R2 Release Guide,* GC33-1570 and *CICS Transaction Server for OS/390 V1.3 Migration Guide,* GC34-5353 for more information.

4.4 Batch workload considerations

In this section, we review how the job entry subsystem exploits the Parallel Sysplex environment. Further, we look at techniques available for workload balancing in a batch environment today.

Since MVS/ESA SP V5.1, there is a component called JESXCF. This component provides the cross-systems communication vehicle (through XCF services) for the use of the job entry subsystem.

At subsystem initialization time, each JES member automatically joins a JES XCF group. All the members attached to the group then use XCF services to communicate with each other.

4.4.1 JES2 considerations in Parallel Sysplex

Note: At the current level of z/OS, for systems operating in a MAS configuration, a sysplex (not necessarily Parallel Sysplex) is mandatory. This allows the JESXCF component to provide communication to all members in the MAS through XCF services.

JES2 can exploit the CF. In addition to its use of XCF, which may be using the CF for communication, JES2 can also place its checkpoint information into a structure in the CF. The coupling facility is faster for read operations than cached DASD, but slightly slower for writes when comparing the JES2 checkpoint I/O operations. The real advantage of the coupling facility lies in its FIFO queuing of lock requests. This ensures round-robin (equitable) sharing of the checkpoint, delivering it to the members in the order requested. As the number of members of the MAS increases, this is important because of the increased contention for the primary checkpoint data set (CKPT1). For further details, refer to:

http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/W9748B

Allowable JES-BCP combinations:

With z/OS, the JES levels supported by a given release are the same as the JES levels that may coexist in the same multi-access spool (MAS) or multisystem complex with the JES delivered in that z/OS release. Thus, the JES releases you can use with the z/OS V1R7 BCP are:

- For JES2: z/OS V1R7 JES2, z/OS V1R5 JES2 and z/OS V1R6 JES2 (both are functionally equivalent), and z/OS V1R4 JES2
- For JES3: z/OS V1R7 JES3, z/OS V1R5 JES3 and z/OS V1R6 JES3 (both are functionally equivalent), and z/OS V1R4 JES3

As of z/OS V1R2, compliance with the allowable JES-BCP combinations is enforced. Therefore, failure to use one of the above JES releases with the z/OS V1R7 BCP results in an abend and subsequent termination of the JES address space

4.4.2 JES3 considerations in Parallel Sysplex

Starting with OS/390 V2R8, JES3 provides support for WLM-managed batch initiators. With this enhancement, the system programmer can specify, by job class group, whether WLM or JES3 will manage batch initiators. When the initiators are managed by WLM, they can be dynamically stopped and started as needed, and placed on systems with sufficient capacity to process the related batch jobs.

4.4.3 Can I have JES2 and JES3 in the same sysplex?

There is no problem having them in the same SYSPLEX. Be aware, however, that the JES service organization does not consider this a supported configuration if they are running in the same MVS image.

4.4.4 Batch workload balancing and Parallel Sysplex

Beginning with OS/390 V2R4, you have the option of using a set of initiators where WLM, rather than JES, controls how many are active, and the rate at which queued jobs are initiated.

WLM provides a mechanism for classification of work and controls the allocation of system resources (for example, CPU and memory) to units of work during the execution phase. However, prior to the introduction of WLM-Managed Initiators, WLM had no control over when or where batch jobs would be started, and had to work in reaction mode as JES placed jobs into execution independent of current conditions.

JES has been extended to classify batch jobs following converter processing and to provide job backlog information to WLM, by service class, on each image in the JES MAS. WLM acquires the job backlog information and uses it in determining whether or not installation goals are being met. WLM starts, stops, or changes job selection criteria for WLM-Managed initiators based upon availability of work, goal attainment, and processor resource availability within the sysplex. For example, if WLM determines that a more important workload, like CICS, is missing its goals on a system because it is being delayed by a resource that batch is using a lot of, WLM will reduce the number of available WLM-Managed initiators on that system, thereby reducing the number of batch jobs being run.

In keeping with its goal to maximize system throughput, WLM will attempt to keep the CPU as busy as possible, and may start new initiators (as appropriate, based on goal achievement)

until the available capacity in that system is less than 5%. At that point, WLM will check to see if other systems in the JES MAS have more spare capacity. If they do, it will stop initiators on this system. The result is that the goal achievement of the service class used for batch jobs will decrease. This in turn will prompt WLM on one of the other systems in the JES MAS to start more initiators. Note, however, that WLM on one system will not *direct* WLM on a different system to start more initiators. Each WLM operates semi-independently, reacting to a drop in the goal achievement of the service class at the sysplex level. Also, it is important to understand that in relation to WLM-Managed initiators, WLM only controls the stopping and starting of its initiators. If a job in a WLM-Managed initiator ends and there is an appropriate job waiting to run, that job will be selected by the initiator - WLM is not involved in the decision to start that job. This behavior is currently the same for both WLM- and JES-Managed initiators.

WLM-Managed Initiators are a far more effective way of maximizing batch throughput and system capacity than traditional JES initiators. JES has no understanding of the capacity of the system; all it knows is how many initiators the installation has defined for each class, and whether they are busy or not. If the CPU is completely over-committed, but there are available initiators, JES will happily start more jobs. Equally, if all the initiators are being used, but the CPU is only 50% busy, JES will be unable to start more work to utilize the available capacity. So, in general, WLM-Managed initiators are more effective at exploiting available capacity, and the experiences of clients that exploit them has generally been positive. Note that there are some special cases where JES-managed initiators continue to be more suitable, for example for jobs that must be started immediately after they are submitted. These special cases are documented in the IBM Redbook *System Programmer's Guide to: Workload Manager*, SG24-6472.

There are various others techniques available in JES2 to control where a job will run within a JES2 MAS environment. These can provide a form of balancing based on JOBCLASS, SYSAFF, and initiator setup. This can be achieved by operator command, initialization options, JCL/JECL[1] changes, or by coding JES2 exits.

Another technique is to use WLM scheduling environments: WLM scheduling environments provide an easy way to control which systems in a sysplex are eligible to run a batch job. In the batch job, you specify a scheduling environment name on the job card (with the SCHENV parameter). The job can then only run on systems where that scheduling environment has a status of ON. The status of the scheduling environment is set to RESET when the system is IPLed, and the operator (or automation) then sets the status to ON or OFF using the MODIFY WLM command.

Note: Be aware that if system affinity is used, JCL conversion can take place on any member of the MAS for which the job has affinity.

Note for JES3: JES3 balances the workload among CPCs by considering the resource requirements of the workload. JES3 can route work to specific systems by enabling initiator groups/classes and JECL (*MAIN statement). JES3 is aware of tape and DASD connectivity and thus will schedule work to the CPC that has the correct devices attached. Starting with OS/390 V2R8, JES3 supports WLM-managed initiators.

[1] JECL: JES2 control language. See *z/OS V1R7.0 JCL Reference*, SA22-7594 for more information.

4.4.5 IBM BatchPipes for OS/390

BatchPipes for OS/390 (5655-D45) is a major enhancement and replacement product for BatchPipes/MVS.

BatchPipes uses *parallelism* and *I/O optimization* to reduce the batch processing time and balance the workload across the system or Parallel Sysplex. By giving you the ability to run the jobs and job steps in parallel, BatchPipes can reduce the elapsed time of batch jobs and job streams. In addition, elapsed time can be further reduced when BatchPipes is used to minimize I/O resource usage.

BatchPipes provides the following capabilities:

- It allows two or more jobs that formerly ran serially to run in parallel.
- It allows individual job steps in a multi-step job to run in parallel.
- It reduces the number of physical I/O operations where possible by transferring data through CPC storage rather than DASD or tape.

Data piping between jobs

The traditional batch job stream uses an intermediate data set as a vehicle to pass data from one process to the next process. Job1 writes data to a data set on either tape or DASD. When it completes, the data set is closed and Job2 can start. In essence, Job1 is a *writer* to the data set while Job2 is a *reader* of the data set.

In a BatchPipes environment, the data set is replaced by a *storage buffer*. After Job1 writes the first block of data to the buffer, Job2 can read this block from the buffer. The storage medium that BatchPipes uses to transfer data from writer to reader is known as a *pipe*.

Data in a pipe always flows in one direction, from writer to reader. A writer_pipe_reader set is known as a *pipeline*. The pipe can exist in either a data space or in a CF structure.

Data piping between systems

The processing of related jobs across systems is achieved by using a common CF structure known as the *pipe*. This set of BatchPipes subsystems is known as a *Pipeplex*.

All the subsystems in the Pipeplex must have the same name and reside on separate systems in the Parallel Sysplex. Cross-system piping can only occur between systems within the same Pipeplex. The Pipeplex can include every system in the Parallel Sysplex, or a subset of the systems. See Figure 4-41 on page 249 for the recommended Pipeplex configuration.

It is also possible to run multiple Pipeplexes within a system. In Figure 4-41 on page 249, the Pipeplex BP01 includes all systems in the sysplex. However, Pipeplex BP02 includes only two of the four systems.

Note: BP01 and BP02 cannot be in the same Pipeplex.

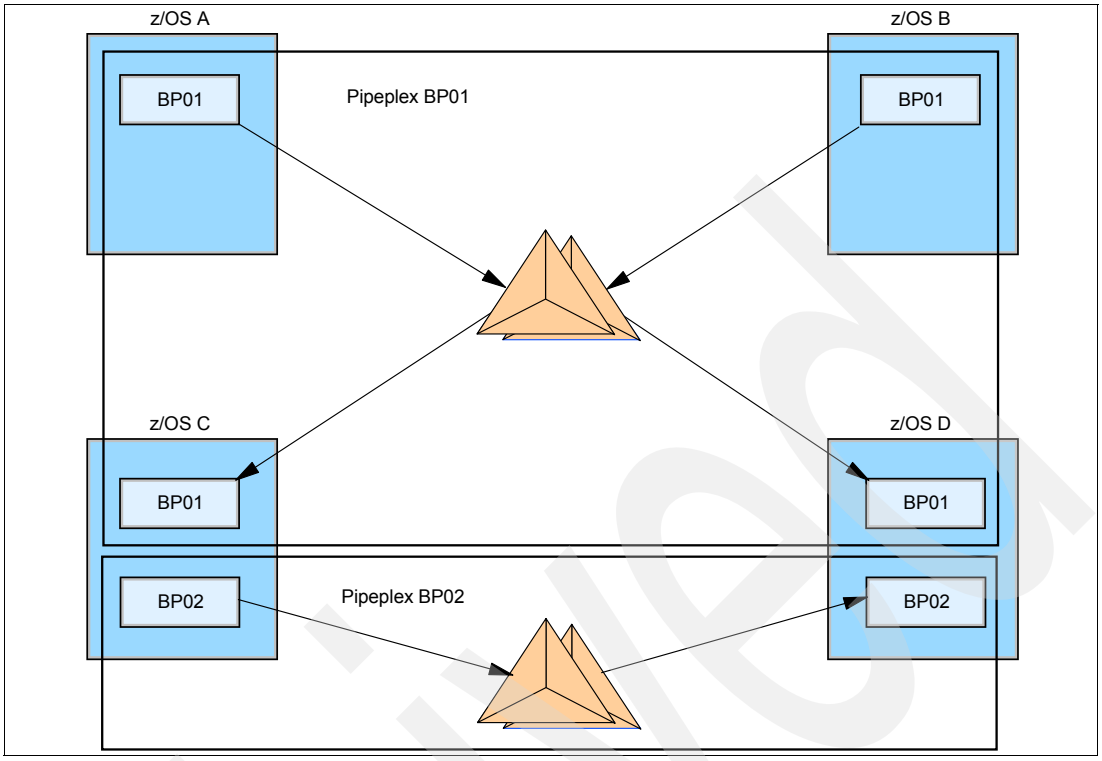


Figure 4-41 Pipeplex in a Parallel Sysplex

Considerations for a Parallel Sysplex

With BatchPipes, jobs that were previously run serially can now run concurrently. Therefore, all resources for those work units must be available concurrently:

- *Initiators*: JES initiators must be available for all jobs in the pipeline. Unless *all* the jobs that will be connected to a pipe have initiators, the jobs will stop when the pipe with no corresponding job fills up. We recommend that all jobs in a pipeline be assigned one job class and a set of initiators be dedicated to that job class.
- *Tape and DASD*: There should still be sufficient devices to simultaneously hold all job and step data that is outside the control of BatchPipes.
- *Virtual Storage*: Each job has its own need for virtual storage. When multiple jobs run concurrently, the combined storage must be available. Additional storage may be required for a CPC storage pipe.
- *CF storage*: A Pipeplex requires enough CF storage to hold the maximum number of cross-system pipes.
- *CP*: Each job and job step has its own need for CPs. Since multiple jobs and job steps run concurrently, CPU utilization may increase while the jobs are running.

BatchPipes affect on chargeback

BatchPipes may have a major effect on variables that are used in charge-back formulas. For example, TCB time is likely to increase.

Without BatchPipes (that is, when the I/O subsystem moved the data), a significant portion of the system services (that is, TCB time) was not charged to the address space that invoked them.

With BatchPipes, the system resources that the job uses are included in the TCB time. For more details, see IBM BatchPipes for *BatchPipes OS/390 V2R1 Users Guide and Reference,* SA22-7458. Figure 4-42 contains information about the impact that BatchPipes may have on your chargeback mechanisms, and contains information about the impact that BatchPipes may have on your chargeback mechanisms.

Variable	How BatchPipes Affects the Variable
TCB Time	TCB time increases. Without BatchPipes (that is, when the I/O subsystem moved the data), significant portion of the system services (that is, TCB time) were not charged to the address space that invoked them. With BatchPipes, the system resources that the job uses are included in the TCB time.
EXCP	EXCP counts decrease. BatchPipes eliminates EXCPs to the pipe data sets.
Tape Mounts	Tape mounts are reduced. Tape and tape mounts for the intermediate data sets that are piped are eliminated, along with tape mount processing and tape storage in a library.
DASD Storage	DASD storage is reduced. DASD storage for the intermediate data sets that are piped is eliminated.
SRB Time	SRB time is reduced. With BatchPipes, fewer SRBs are required to perform data movement.
Service Units (SUs)	• The IOC field increases, where IOC refers to the I/O service of an address space. • The CPU field increases, where CPU refers to the accumulated CPU (that is, TCB) SUs. • The MSO field increases, where MSO refers to accumulated storage SUs.
Uncaptured CPU Time	Uncaptured CPU time is reduced. Compared with accountability of non-BatchPipes jobs, accounting of CPU time is more accurate for BatchPipes jobs.
Working Set Size	BatchPipes does not affect the working set size (WSS). Since processing of the job occurs at a faster rate, the system is likely to keep more pages of the job in the CPC. For example, with fewer physical I/O operations, pages are not paged out while the job waits for those physical I/O operations to occur.
CPU Charges	Because jobs are split and may run on different CPCs, charges for jobs may vary based on the charge rates for the CPC.

Figure 4-42 Effects of BatchPipes on chargeback variables

BatchPipes services offering

IBM offers an Integrated Services Offering (ISO) to help you get started using BatchPipes.

4.5 Network workload balancing capabilities

There are various ways to connect the network of users to applications in the sysplex. The two main application types considered in this book are *SNA applications* and *TCP applications*. VTAM provides network access to the sysplex for SNA applications and TCP/IP provides network access to the sysplex for TCP applications. Both VTAM and TCP/IP provide many specific functions to exploit the Parallel Sysplex. Only VTAM, however, currently exploits the Coupling Facility, through its use of Multi Node Persistent Sessions and Generic Resources support.

In this section, we will describe the use of VTAM GR in some detail.

For TCP/IP, the role of workload balancing is more closely connected to how your network is configured and connected than is the case for SNA.

4.5.1 VTAM generic resources function

To allow an user to easily connect to an application that may be available on multiple systems, VTAM provides a function known as *generic resources*. The use of generic resources provides the user with a single system image of the application no matter where it runs in the Parallel Sysplex.

Using generic resources

The generic resources function is an extension to the existing USERVAR support. It was introduced in VTAM V4.2 to support applications in a Parallel Sysplex.

The user accesses the desired application by using the *generic resource name* of the application. VTAM determines the *actual application instance*[2] for this session based on workload and other performance criteria. The generic resources function allows you to add CPCs and applications, and to move applications to different images on the same or different CPCs without impacting the user.

APPN requirement

The generic resources function requires the use of Advanced Peer-to-Peer Networking® (APPN) in the Parallel Sysplex. This means that the VTAM systems that are running generic resources must be configured either as APPN end nodes or as APPN network nodes. The VTAM end nodes must have a direct connection to at least one VTAM network node inside the Parallel Sysplex. A VTAM end node will select one of the VTAM network nodes to be its network node server, and a direct connection is required for this.

Sessions with generic resource applications can be established from either subarea nodes or APPN nodes. The generic resource function can support sessions from all LU types, including LU0, LU1, LU2, LU3, LU6, LU6.1, and LU6.2.

Do you have to migrate the network to APPN?: Generic resources require the use of APPN protocols inside the Parallel Sysplex. This does not mean that the entire network must be running APPN to access these applications.

If the network contains subarea nodes that require access to generic resource applications, one (or more) of the VTAM network nodes in the Parallel Sysplex will provide a boundary between the subarea network and APPN. A VTAM image that provides this function is called an *Interchange Node (IN)*.

Generic resource definition

There are no new VTAM definitions to define generic resources. Subsystems that want to use the VTAM generic resources function will inform VTAM using the SETLOGON macro. The SETLOGON macro is issued after an application opens its ACB with VTAM. The SETLOGON ADD option (with the generic resource name specified in the NIB of the GNAME operand) will indicate to VTAM that this application is to be added into a generic resource. VTAM takes this information, and then updates the CF structure with the new application instance.

A subsystem can remove itself from a generic resource at any time, without having to close its ACB. It does this with the SETLOGON macro and the DELETE option. If this happens, and the ACB is still open, users can still log on using the application name, but not using the generic name.

Figure 4-43 on page 252 shows a Parallel Sysplex where an user has logged onto a generic resource name called APPL. There are a number of actual application instances of this generic resource, called APPL1, APPL2, and APPL3. These actual application instances reside on systems VTAMC, VTAMD, and VTAME. One of the network nodes, VTAMA or VTAMB, is responsible for doing the workload balancing, which results in the actual application instance APPL2 being selected for this session request.

[2] Meaning which application on which image.

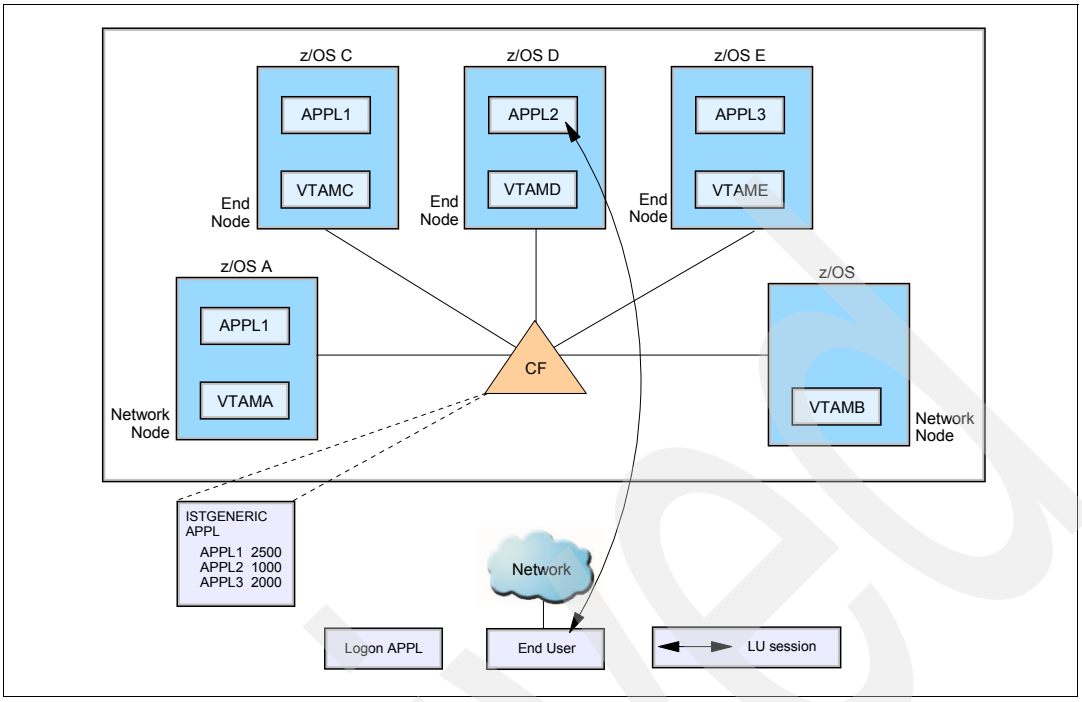


Figure 4-43 Using VTAM generic resources in a Parallel Sysplex

VTAM workload balancing for generic resources

VTAM does workload balancing for generic resources using information stored in the CF. For VTAM to use the CF, there must be an active CFRM policy defined for the Parallel Sysplex. All the VTAMs in the Parallel Sysplex that are part of the same generic resource configuration must be connected to the same CF.

VTAM use of the CF

When VTAM on an OS/390 in a Parallel Sysplex is started, it automatically connects to the CF structure, after first checking that the CFRM policy is active. The first VTAM to become active in the Parallel Sysplex will allocate the storage for the CF structure.

VTAM uses a list structure in the CF to keep the information about the generic resources in the Parallel Sysplex. The default name for this list structure is ISTGENERIC. Starting with VTAM V4.3, you can override the default name by specifying a different structure name on the VTAM STRGR start option. This allows you to have multiple CF structures for generic resources within your network, which might be useful if you wish to separate test and production structures. For more information about CF usage, sizing, placement, and rebuild configurations, refer to the CF Sizer tool, available at:

`http://www.s390.ibm.com/cfsizer/`

Inside the VTAM structure, the CF keeps information about the session counts for each application instance in a generic resource. These session counts are updated automatically each time a session is established or terminated.

When a session setup request for a generic resource name is sent into the network from an user, the request will eventually reach one of the VTAM network nodes in the Parallel Sysplex. The VTAM network node that receives the request will then query the CF structure. The CF will return to the VTAM network node a list of actual application instances, and a session count for each one. The VTAM network node will then choose the appropriate actual application instance. There are a number of ways this is done:

- *VTAM session balancing*: VTAM will use the session counts provided by the CF to balance the number of active sessions on each actual application instance. As a request for a new session is being resolved, the network node VTAM will select the application instance that is running the lowest number of sessions.

  This support was introduced in VTAM V4.2.

- *VTAM call to the OS/390 Workload Manager*: After making the selection based on the session counts, the VTAM network node calls WLM, which can override the instance chosen using VTAM's own logic. WLM will look at the available capacity of the systems running the application instances of the generic resource. The system that has the highest available capacity is selected and the name of the application instance on this system is returned to VTAM.

  This support was introduced in VTAM V4.3.

- *VTAM Generic Resource exit*: Finally, VTAM will call the user-written VTAM generic resource exit (if it exists), which will override the first two choices. IBM provides a default exit called ISTEXCGR, which can be modified to select the application instance based on user criteria. For more information, refer to *VTAM for MVS/ESA V4.4 Customization*, LY43-0068 (available to IBM-licensed clients only).

  This support was introduced in VTAM V4.2.

  A sample VTAM exit is described in the IBM Redbook *Parallel Sysplex - Managing Software for Availability*, SG24-5451.

The reason for providing options to override VTAM's session balancing logic is to take into account the different relative processing capabilities of the systems. In 4.5.1, “VTAM generic resources function” on page 250, system C may have three times the capacity of system D and system E. In this case, using VTAM's session balancing logic alone may result in the underutilization of system C.

Session affinity

Some sessions have special requirements, such as a dependence on a specific application instance of a generic resource. Examples of sessions with affinities are:

- Parallel LU6.2 sessions
- Parallel LU6.1 sessions

In addition to the application information described earlier, the CF also keeps information about individual LUs in the generic resources structure. If affinities exist, then the VTAM network node uses this information from the CF when it selects the application instance.

In 4.5.1, “VTAM generic resources function” on page 250, if the user wishes to establish a parallel session after the first session is established with generic resource APPL, the network node doing the instance selection will ensure that APPL2 is selected. In this way, both sessions are between the user and APPL2 and hence are parallel.

APPN directory services

After the actual application instance has been decided, the VTAM network node must then locate the application. It uses APPN's directory services to do this.

When applications on VTAM end nodes first open their ACBs, information is dynamically sent to the VTAM network node server for that end node, using the APPN control point session. There is no longer any need to predefine VTAM applications as cross-domain resources (CDRSCs) on other VTAMs. This process is known as *APPN registration*. If an application closes its ACB, then the VTAM end node will dynamically de-register itself from the network node server.

In the Parallel Sysplex environment, the VTAM network nodes will use the APPN *central directory server* (CDS) database. The CDS is used on every network node in a Parallel Sysplex. Starting with VTAM V4.4, there is no longer a requirement for every network node in the sysplex to be a CDS. In VTAM V4.4, none, some, or all the network nodes may be a CDS.

The CDS will return the name of the VTAM end node where the application instance resides, and also the name of the VTAM network node server for that end node. These two names are used by APPN to verify the application and also to calculate the route to be used for the user session.

Once the actual application instance has been located and verified, the session request is sent to the VTAM where the instance is located. The session is then established using the selected route.

Local access to generic resources

Special logic is used for VTAM workload balancing. If there are local LUs attached to a VTAM system that runs an actual application instance of a generic resource, the resource selection logic is different. Local LUs can be applications or user LUs. If a local LU requests a session with a generic resource name, VTAM will attempt to select the local instance on the same node.

This can result in unbalanced numbers of sessions among the instances of the generic resource. For example, assume there is a large number of local channel-attached devices connected to VTAM. When the users on these devices request sessions with the generic resource name of APPL, they will *always* be connected to actual application instance APPL3.

Generic resource subsystem support

The following IBM subsystems and their follow-on releases support the VTAM generic resources function:

- CICS/ESA V4.1
- DB2 V4.1
- IMS/ESA V6
- APPC/MVS for OS/390 R3
- TSO/E for OS/390 R3
- NetView Access Services V2

In addition, Computer Associates markets a session manager product called *Teleview* that also supports generic resources.

CICS generic resources support

CICS/ESA V4.1 introduces the exploitation of the VTAM generic resources function to *distribute CICS logons* across different TORs in images in the Parallel Sysplex.

It is quite usual to have more than one CICS TOR on the same VTAM system, and the generic resources function can support this.

DB2 generic resources support

DB2 V4.1 and follow-on releases can use the VTAM generic resources function to *balance sessions* from remote DRDA requesters.

DB2 uses the DRDA architecture to communicate with remote requesters, using LU6.2 sessions. The component of DB2 at the DRDA server called the Distributed Data Facility (DDF) is the part that connects to the network. It also opens the VTAM ACB, and can be part of a generic resource. DB2 has another way of doing workload balancing, referred to as *member routing*. With member routing, when a DRDA requester (single user or gateway) connects to a DB2 server in a data sharing group, the server returns a list of available members of the group, together with weighting information supplied by WLM to enable the requester to select the best member of the group to which to direct their request. The DRDA requester must be enabled to support this feature. For SNA DRDA requesters, the requester must support the APPC sysplex transaction program and, at the time of writing, the only DRDA requester that can use this function is DB2 for MVS/ESA V4 and upwards.

DB2 workload balancing: For remote gateways with many parallel LU6.2 sessions from clients, use DB2's member routing. For remote clients that are single DRDA users, use VTAM's generic resources.

IMS generic resources support

IMS/ESA V6 uses the VTAM generic resources function to balance sessions across IMS Transaction Managers.

APPC/MVS generic resources support

Starting with OS/390 R3, APPC/MVS can associate a VTAM generic resources name with APPC/MVS LUs, to improve availability of APPC/MVS resources and to balance sessions among APPC/MVS LUs. Using VTAM generic resources can also reduce the effort, complexity, and cost of managing a distributed processing environment that includes OS/390 systems.

TSO/E generic resources support

Starting with OS/390 R3, TSO/E introduces the use of VTAM generic resources function to balance sessions across all systems in the sysplex.

NetView access services generic resources support

Just as this redbook was going to press, support for Generic Resources was added to NetView Access Services (NVAS). The support is added via APAR PQ35801. With this support, NVAS will register with VTAM as a generic resource, providing the ability to log on to one of a group of NVAS address spaces using a single VTAM name. It also provides support for logging on to applications, using generic resource names, from NVAS.

This support adds a new CF list structure that NVAS uses to record information about user sessions with any of the NVAS address spaces in the generic resource group.

More information about this support can be obtained in INFO APAR II12403, which is described at:

http://www-1.ibm.com/support/docview.wss?rs=0&q1=II12403&uid=isg1II12403&loc=en_US&cs=utf-8&cc=us&lang=en

Generic resource planning considerations

The following points summarize the planning considerations for generic resources.

- VTAM systems, which support generic resources, must be configured to use APPN (as APPN network nodes or APPN end nodes).
- There must be at least one VTAM network node in the *same* Parallel Sysplex as the VTAM end nodes.
- Each VTAM end node must have a control point session with one of the VTAM network nodes.
- All instances of a generic resource must be in the *same* Parallel Sysplex.
- All VTAM systems that will run generic resources must be in the same network (have the same netid).
- An application can only be known by one generic resource name.
- The generic resource name must be unique in the network. It cannot be the same as another LU or application name.
- A generic resource name cannot be the same as a USERVAR name. This means that any applications that are using generic resources cannot also use XRF.
- It is possible to have more than one actual application instance of a generic resource on a VTAM system.

4.5.2 TCP workload balancing

With the increasing emphasis on e-business applications on OS/390, TCP/IP has assumed a critical role in many OS/390 installations. Functions have been added to the TCP/IP implementation on OS/390 to support dynamic routing and workload management within a sysplex.

As we stated previously, which option you use to balance your TCP/IP workload is closely related to how your clients are connected to your systems.

Workload balancing mechanisms

In TCP/IP, there are two basic mechanisms for balancing connection requests across multiple servers. You can distribute connections based on the IP host names of the servers (for example, WTSC63OE), or on an IP address associated with the servers.

If you are going to do the balancing based on host names, you would use Domain Name Servers to do the distribution. TCP/IP V3.2 introduced a function called *connection optimization*, whereby DNS and WLM cooperate to route connection requests to the most appropriate server. The advantage of this mechanism is that each request gets directed to the best server. The disadvantage is that *every* request must go through the overhead of DNS resolution. Also, this mechanism relies on clients not caching and reusing the provided IP address – something that is completely out of your control in a Web environment.

The other mechanism is to distribute the requests based on the IP address of the server. This mechanism would typically be used by an outboard router. The advantage of this mechanism is less overhead, as there is no DSN resolution required. The disadvantage is that server information is typically sampled instead of communicating with WLM for every connection request, so requests can potentially be routed to a less-than-optimal server.

4.6 APPC/MVS and Parallel Sysplex

Starting with VTAM V4.4, an installation can exploit APPC/MVS support of VTAM generic resources to improve availability of APPC/MVS resources. If one LU in the generic resource group or one system is brought down or fails, APPC/MVS work can continue, because other group members are still available to handle requests that specify the generic resource name. Work from remote systems is less affected by the removal of any single APPC/MVS LU or OS/390 system. Additionally, changes in system configuration, capacity, and maintenance have less effect on how APPC/MVS work.

APPC/MVS support of VTAM generic resources provides a single-system image for a multisystem APPC/MVS configuration. With generic resource names, transaction programs (TPs) from remote systems can establish conversations with APPC/MVS partner TPs on any system; programmers do not need to know specific partner LU names or update TPs whenever the APPC/MVS configuration changes.

APPC/MVS and generic resources use: APPC/MVS TPs can use generic resource names only for partner LUs, not for local LUs.

With generic resource support, it is easier to expand the APPC/MVS configuration. Additional APPC/MVS LUs associated with the same generic resource name can provide immediate improvement in performance and availability, with few or no required changes for APPC/MVS TPs or side information.

Also, the distribution of work among two or more active APPC/MVS LUs on a single system or in a sysplex is possible, so that each LU is used as efficiently as possible. VTAM and WLM distribute session workload among members of generic resource group, thus reducing contention for specific LUs, and improving the performance of systems and TPs.

4.7 TSO/E and Parallel Sysplex

Starting with VTAM V4.4 and OS/390 R3, TSO/E exploits VTAM generic resource support. Functions in the Workload Manager component of OS/390 R3 and the Communications Server allow users to log on to TSO as before, but now using a combination of generic resources and dynamic workload balancing, their sessions are directed to the systems that have the lightest load.

TSO/E and generic resources: TSO/E generic resource support provides the capability to balance session distribution across TSO systems in the sysplex. The ability to reconnect to the original task, if the TSO/E user loses connection, is also included in this support.

If dynamic workload balancing in a TSO environment is of importance to you prior to OS/390 R3, there is a function in VTAM called USERVAR that you may use. Through the USERVAR function, you can provide a pseudonym for multiple TSO systems. The USERVAR exit routine can then be implemented to distribute TSO users to these systems. This allows you to balance your workload, but it is not possible to have functions, such as reconnect, work in a predictable way. However, prior to OS/390 R3, the USERVAR function might satisfy your needs.

4.7.1 MAS considerations

The same TSO user ID cannot be logged onto more than one member in the MAS at any one time. It is possible to make a modification in JES2 code to allow the same TSO user ID to be logged onto multiple members in the MAS. This, however, is recommended only for short term migration purposes.

HASPCNVT modification:

```
BZ  XTDUPEND      *** Instruction Deleted ***  @420P190 05990900    B   XTDUPEND    Skip
duplicate logon check MODIFICATION 05990901
```

Note: The source for module HASPCNVT is found in SYS1.HASPSRC.

With this modification, beware of the following exposures and problems:

- You must be careful not to edit or update the same data set from duplicate TSO user IDs on different systems because there is no SYSTEMS level ENQ to guarantee serialization.
- TSO user notifications are *random*. (They go to the first TSO user logged on in the sysplex.)
- Watch out for TSO logon ENQ - if it is SYSTEMS scope in the GRS RNL, then TSO will reject the second logon with the message `already logged on`.
- Ensure the ISPF data set names are unique by qualifying them with the system name.

4.7.2 Query management facility workload considerations

One of the typical workloads for TSO users in a DB2 environment is the use of the query management facility (QMF) for online, often read-only, data analysis. Sometimes QMF uses the *uncommitted read* bind option (ISOLATION(UR)).

Since the access is often read-only, it may be acceptable to use data that is current. If this is true, you may be able to offload the QMF workload to another image and not worry about that DB2 being part of the data sharing group. The QMF queries could then run against the database without having to worry about data sharing overheads. If, however, there is any update of the data required, then the DB2 subsystem will have to be part of the data sharing group. Note, however, that TSO attach cannot span across several OS/390s.

4.8 Test considerations in Parallel Sysplex

In this section, we look at implications for testing in a Parallel Sysplex. General testing considerations will not be covered. Depending on the environment implemented for test purposes in Parallel Sysplex, your configuration may vary considerably.

4.8.1 Testing implications in Parallel Sysplex

Testing in the Parallel Sysplex environment, which could have up to 32 systems, provides an opportunity for *creativity* because applications should be tested in an environment that mimics the production environment. The challenge is to provide for a thorough, effective test without using excessive resources, and ensuring errors are not introduced when moving or cloning systems. Specific areas for consideration are:

- CF functions
- Dynamic transaction routing
- Failure and recovery scenarios
- Variations or unique functions in systems
- Stress testing
- Shared data validation
- Network
- Maintaining an adequate test environment

This section contains a discussion of each of these areas.

CF

Since the CF is vital to the integrity of the system, verifying that it is working correctly with new releases of software is critical.

Considerations include how to test all the various links to each of the systems, verifying that recovery works, validating the data sharing capabilities, and insuring that each user of the CF is working correctly.

The same CF cannot be used for both production and test, because some subsystems have hardcoded the names of the CF structures into their code.

Note: This does not mean that multiple CFs cannot be run in separate LPs on a given CPC. However, multiple CFs in separate LPs on a CPC must all run with the same level of CFCC. For this reason, running your test CFs in the same CPC as the production ones does not provide the capability to test new CF levels in the test sysplex before migrating them to the production sysplex.

Dynamic transaction routing

The implementation of transaction routing introduces the necessity to verify that routing is occurring as expected. When new applications are introduced, routing may be impacted. Test cases will need to be run to verify the routing. The best test would be to have all the systems participate. Realistically, only a subset of the systems is validated during a test. An availability or risk analysis is needed to determine the optimum test environment.

During the application testing, tests should be developed to validate or detect transaction affinities. It is far better to learn of the affinity during testing than after the application has gone into production and been propagated across several systems. If affinities are detected, appropriate changes can be made to the dynamic routing tables.

Failure and recovery scenarios

One of the largest causes of problems with CF recovery is a lack of familiarity with the recovery process by those responsible for monitoring and controlling it. Many times, recovery procedures are written, tested, and then never touched again until they are needed, at which point they may be out of date, or the person driving the process has forgotten exactly how the process works. We strongly recommend providing a facility (and an incentive) for system programmers and operators to test their procedures on a regular basis. This allows any

problems to be discovered and addressed and helps maintain familiarity with the process and logic behind the procedures.

Test cases will need to be developed to see whether the new environment will recover appropriately when the following failures occur:

- Connectivity failure, including:
  - VTAM EN/NN/IN
  - CTCs
  - 3745s
  - Couple data sets failure
  - CF link failure
  - CF failure
  - CF structure failure
  - CPC failure
  - OS/390 system failure
- Subsystem failure, including:
  - Individual CICS AOR failure
  - Individual CICS TOR failure
  - Database managers (for example DB2 and IMS DB)
- Sysplex Timer connectivity failure
- Application failure

Failure tests should be used to get a clear understanding of what all the different structure owners *do* during CF failures (for example, VTAM rebuilds its structure, JES2 goes into CKPT RECONFIG dialog, IRLM rebuilds depending on REBUILDPERCENT, and so on).

Recommended reference test document: *OS/390 Parallel Sysplex Test Report,* GC28-1963 is an experience-based document. The document is updated quarterly with new experiences and recent product enhancements.

Variations or unique functions

To successfully test a new system that is propagated to each of the different hardware platforms in the Parallel Sysplex, all variations must be known. Test cases are needed to validate every variation prior to propagation so that errors do not occur upon implementation because the system has some unique feature or application.

Note: Thus, we recommend keeping all the images in a Parallel Sysplex as *identical* as possible.

Stress or performance testing

Many problems, both in application code and system software, only show up under stress. Also, the problems may only surface with a certain workload mix. For these reasons, we strongly recommend that some mechanism to do stress testing should be available, both to the system programmers and also to the application programmers. Further, the stress tests should be set up to mirror the production workload mix as closely as possible. So, if your production workload consists of 50% CICS, 20% DB2, and 10% batch, you should try to match those proportions when running the stress tests.

Shared data validation

With the introduction of enabling shared data, testing needs to ensure that the integrity of the data is being maintained. Testing needs to verify that the expected data is updated correctly and that applications do not break any rules.

Network

Note that VTAM has been changed to allow the specification of its CF structure name. The name is specified as a VTAM start parameter. Before this change, VTAM was one of the subsystems that hardcoded the name of its structure in the CF.

The second consideration is to ensure that there is adequate connectivity to the test systems to have a thorough test of the various terminals, connections, and printers within the network. This consideration also applies to non-Parallel Sysplex systems.

Maintaining an adequate test environment

Determining what is needed for a valid test environment is a function of your configuration and availability requirements. The trade-off of equipment and resources (costs) versus reduction in errors (availability) needs to be made. The greater the availability requirement, the more robust the test environment and test plan need to be.

For example, to functionally test dynamic transaction routing, the following is needed:

- At least two systems with the software
- CF
- CF links and transactions that are to be routed
- Shared data

4.9 How to select applications to exploit Parallel Sysplex

Parallel Sysplex provides enhanced application functions that can improve application value, and thus benefit your business. Some application requirements that exist in most installations include:

- Improved application performance
- Continuous application availability
- Application growth

Refer to the *System/390 MVS Parallel Sysplex Migration Paths*, SG24-2502 for a method to use when actually selecting the workloads to exploit a Parallel Sysplex.

A

Software AG's Adabas in a Parallel Sysplex environment

Note: The text of this Appendix has been provided by Software AG.

Software AG has extended Adabas, its mainframe-based database management system, for use with IBM Parallel Sysplex technology. The combination of Adabas with the Parallel Sysplex technology resulted in the add-on product Adabas Cluster Services, which was released in 2001.

Worldwide, more than 3000 clients use Adabas. These clients, for example, banking, insurance, administration and logistics organizations, typically have large or very large OLTP (Online Transaction Processing) applications.

Adabas is known for its outstanding performance, in particular its ability to process large data volumes without performance degradation, its minimal overhead with respect to system administration, and its high level of stability and reliability (continuous operation, and efficient and reliable error recovery).

Adabas Cluster Services

The main development objectives for Adabas Cluster Services were:

- Increased availability:

  Whenever one of the participating systems must be serviced, all other systems can continue to use the database without interruption. If an Adabas nucleus within a cluster aborts, the other cluster nuclei can continue to run practically interruption-free. The ultimate objective is to render each system component redundant so that it can be taken out of operation without causing a system-wide failure (no single point of failure).

- Improved performance:

  The throughput (Adabas commands per time unit) was to be increased by using CPU resources from several systems. The throughput should be scalable, that is, it should increase as linearly as possible with the number of nuclei in a cluster.

- Application transparency:

  The behavior of an Adabas cluster towards application programs must be compatible to that of a single Adabas nucleus. For an application program, it should be transparent whether it is running with a single nucleus or with an Adabas cluster, and it should also be transparent as to which specific nucleus within a cluster is processing the application program's commands.

The greatest challenge was the attainment of the performance objectives. The main reason for this is that a typical operation against the coupling facility costs approximately one fourth to one third of the average CPU time required by an Adabas command in a well tuned OLTP environment. Although coupling facility operations are typically a factor of 10-100 times faster than read or write operations, they are CPU-intensive

The resulting conclusion for the design of Adabas Cluster Services was to minimize the number of coupling facility operations.

Architecture

With Adabas, one nucleus works on one database. With Adabas Cluster Services, up to 32 nuclei can work on the same database. Each cluster nucleus runs its own process in its own address space, which can be located on different machines connected to the coupling facility. The datasets that comprise the database are accessible from all machines (shared disks).

For synchronization, the cluster nuclei use, via Cross-System Extended Services® (XES), a cache structure and a lock structure in one or two coupling facilities, to which all connect to. In addition, the nuclei exchange messages with one another using Cross-System Coupling Facility Services (XCF).

A single Adabas nucleus uses a Work dataset for storing intermediate results as well as *protection data* that is required for *restart recovery*. In addition, the nucleus writes protection data to up to eight Protection Log datasets, which are used for *archive recovery*. In a cluster, each cluster nucleus has its own Work and Protection Log datasets, which are written to only by that nucleus, but which can be read by all other nuclei.

Figure A-1 illustrates the architecture of Adabas Cluster Services.

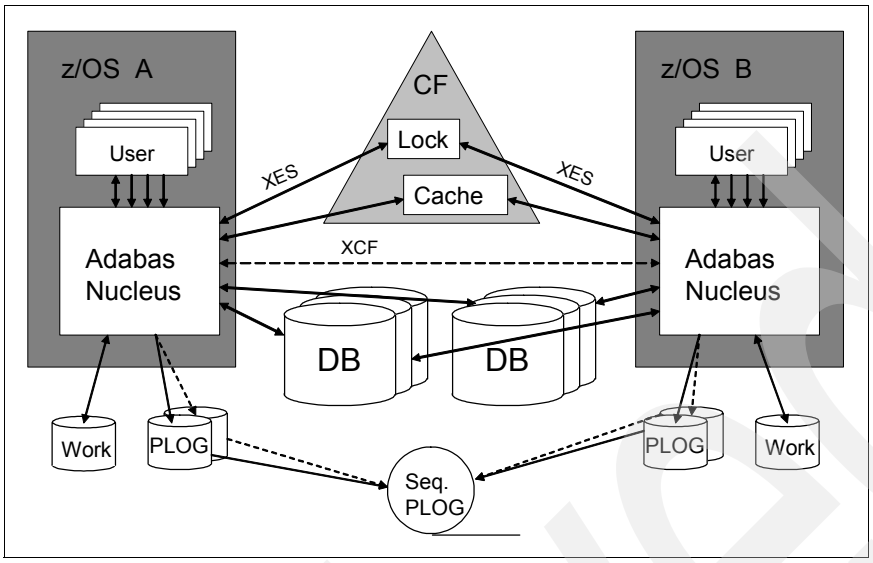


Figure A-1 Architecture of Adabas Cluster Services

The functions of the nucleus in an Adabas cluster are fully symmetrical: each nucleus can execute all operations. However, there are operations that at a given point in time can be executed by only one nucleus. An example of such an operation is a cast-out process, which writes changed blocks from the cache structure to the database. Still, the next cast-out process can be executed by a different cluster nucleus.

Inter-nucleus communication

Nuclei, which participate in an Adabas cluster, must exchange information for a variety of purposes. XCF Signaling Services is used to accomplish this information exchange.

An internal interface is used to send Adabas commands to one, several, or all other nuclei in a cluster. XCF collects the responses and signals the end of the operation when all expected responses have been received.

Database block operations

Adabas Cluster Services uses a cache structure in the coupling facility as provided by XES Cache Services. The cluster nuclei write all database blocks, which they have changed, to this cache structure. Instead of writing the blocks directly to the database, a store-in cache is used. Cast-out processes write the changed blocks regularly from the cache structure to the database. Blocks that have not been changed by the cluster nuclei are not written to the cache structure.

Recovery

Although to a lesser extent than database operations, the Adabas recovery logic was also affected by the extension for clusters. This involved not so much the backing out of individual transactions (transaction recovery) but rather the handling of error situations that cause an abort of an Adabas nucleus, as well as errors that cause damage to or destruction of the database.

Transaction and restart recovery

During operation, Adabas writes protection records for updates and transactions to the so-called Work dataset. If the Adabas nucleus should abort, this protection information is used to return the database to a physically and logically consistent state.

The writing of changed blocks to the database occurs asynchronously to the execution of update commands and the changing of blocks in the buffer pool. Changed blocks can be written to the database before the updates have been confirmed by the end of the corresponding transaction (steal). However, they need not necessarily be written to the database at the end of the transaction (noforce). Therefore, during a *restart recovery* following an abort of the Adabas nucleus, all transactions that have ended but whose updates were not stored in the database prior to the abort, must be repeated (*redo*, consequence of noforce). Conversely, all updates, which were stored in the database as part of a transaction, which was not yet completed at the time of the abort, must be backed out of the database (*undo*, consequence of steal). The protection records, which the nucleus writes to the Work dataset, contain the necessary information for the redo and undo operations.

In Adabas Cluster Services, each cluster nucleus has its own Work dataset, and only the nucleus itself writes protection data to its own Work dataset. However, each other cluster nucleus is able to read the protection records created by another nucleus within the cluster.

The backout of individual transactions (for example, backout (rollback) commands) functions in a cluster nucleus in the same way as with a single nucleus. The nucleus reads the required protection records from its own Work dataset and backs out the updates of the transaction.

If all active nuclei in a cluster should abort, the nucleus that is started next initiates the restart recovery. This is the *offline recovery*, during which the nucleus reads the protection records from the Work data sets of all previously active cluster nuclei, and performs in essence the same recovery logic as for a single nucleus. Before they can be applied, the protection records from the various Work datasets must be placed in chronological order. This is possible since each protection record contains a SYSPLEX time stamp indicating exactly when it was created.

If in an Adabas cluster one or more but not all nuclei should abort, the remaining nuclei perform the *online recovery*. Each nucleus allows a certain time for all currently running transactions to reach normal completion, following which all remaining activities are interrupted and the connections to the cache and lock structure are terminated. This results in the loss of all data in these structures. The nucleus then performs the offline recovery logic, after which all nuclei can connect to new cache and lock structures, and normal operations can continue. During this entire process, the user contexts remain intact in the nuclei that did not abort.

Conclusion

Client tests have already shown that the above described procedures for the improvement of cluster performance have achieved impressive results. During August/September 2001, a test was performed for a client using two z/900 IBM mainframes with 16 processors each. In a test scenario provided by the client, 13 Adabas Cluster Services nuclei together processed per second (elapsed time) approximately 160,000 Adabas commands that were submitted by 15,000 simulated users.

Software AG released Adabas Cluster Services in December 2001.

The development of Adabas Cluster Services continues. The primary emphasis now is concentrated in the following areas: additional performance improvements, improvements in availability, and implementation of additional Adabas functionality not yet provided by Adabas Cluster Services.

Abbreviations and acronyms

ABARS	aggregate backup and recovery support
ac	alternating current
ACB	access control block
ACDS	alternate control data set
ACDS	active control data set (SMS)
ACS	automatic class selection (SMS)
ACTS	automated computer time service
ADM	asynchronous data mover
ADMF	asynchronous data mover function
ADMOD	auxiliary DC power module
ADSM	ADSTAR distributed storage manager
ALA	link address (ESCON)
AMD	air moving device
AMODE	addressing mode
AMRF	action message retention facility
ANR	automatic networking routing
AOR	application owning region (CICS)
APA	all points addressable
API	application program interface
APPC	advanced program-to-program communication
APPLID	application identifier
APPN	advanced peer-to-peer networking
ARF	automatic reconfiguration facility
ARM	Automatic Restart Manager
AS	address space
AS	auto switchable (Tape Sharing)
ASKQ	ask questions (HONE)
ASYNC	asynchronous
ATL	automated tape library
AVG	average
AWM	alternate wait management (z/OS)
AXM	authorized cross-memory
BAN	boundary access node
BBU	battery backup
BCDS	backup control data set (DFSMShsm)
BCP	basic control program
BCU	basic configurable unit
BCS	basic catalog structure
BDAM	basic direct access method
BGP	border gateway protocol
BIX	build index
BL	block (block-multiplexer channel)
BMP	batch message process (IMS)
BNN	boundary network node (SNA)
BOC	battery operated clock
BOF	bill of forms
BP	buffer pool
BRS	bandwidth reservation
BSC	binary synchronous communication
BSDS	bootstrap data set
BSN	bus switch network
BUF	buffer
BUFND	buffer number data (VSAM)
BUFNI	buffer number index (VSAM)
BUFNO	buffer number
BUFSP	buffer space (VSAM)
BWAT	batch workload analysis tool
BWO	backup while open
BY	byte (byte-multiplexor channel)
CA	Computer Associates
CA	continuous availability
CAA	cache analysis aid
CADS	channel adapter data streaming
CAS	catalog address space
CAS	coordinating address space
CAU	CICS affinity utility
CB	Component Broker
CBIC	control blocks in common
CBIPO	custom-built installation process offering
CBPDO	custom-built product delivery offering
CBU	capacity backup
CBY	ESCON byte multiplexer channel
CCU	communications control unit
CCU	central control unit
CDRM	cross-domain resource manager
CDSC	catalog data space cache

CDS central directory server

CDS couple data set

CEC central electronics complex

CF Coupling Facility

CFCC Coupling Facility Control Code

CFCC LP Coupling Facility Logical Partition

CFDT Coupling Facility data table

CFFP Coupling Facility failure policy (z/OS)

CFIA component failure impact analysis

CFR Coupling Facility receiver

CFRM Coupling Facility resources management

CFS Coupling Facility sender

CFVM Coupling Facility Virtual Machine (VM)

CHG changed

CHKP checkpoint

CHKPT checkpoint

CHPID channel path identifier

CI control interval (VSAM)

CICS Customer Information Control System

CIU CICS Interdependencies Utility

CKPT checkpoint

CLI call level interface

CLIST command list

CLO control link oscillator

CMAS CICSPlex SM address space

CMC communications management configuration

CMF CICS monitoring facility

CMOS complementary metal oxide semiconductor

CNC ESCON channel attached to an ESCON-capable device

COB card-on-board

COMDS communications data set

CP control program

CP central processor

CP/SM CICSPlex systems manager (CICSPlex SM)

CPC central processing complex

CPF command prefix facility

CPU central processing unit

CQS common queue server

CR control region (CICS)

CR capture ratio

CRA Configuration Reporting Architecture (S/390)

CDRSC cross-domain resources

CS cursor stability (DB2)

CS central storage

CSA common systems area

CSAR complex systems availability and restart

CSECT control section

CSI consolidated software inventory (SMP/E)

CSS channel subsystem

CSTOR control storage (central storage for CF)

CSU customer setup

CTC channel-to-channel

CTCA channel-to-channel adapter

CU control unit

CUA® common user access (SAA®)

CUoD Capacity Upgrade on Demand

CV Central Version (IDMS/CA)

CVC conversion channel (ESCON)

CVOL control volume

CWA common work area

DADSM direct access device space management

DAE dump analysis elimination

DASD direct access storage device

DB database

DBD database definition (IMS)

DBRC database recovery control (IMS)

DCAF distributed control access facility

DCB data control block

DCCF disabled console communication facility

DCE distributed computing environment

DCFD Dynamic CF Dispatching

DD data definition

DDSR Dynamic Database Session Routing (IDMS/CA)

DDF distributed data facilities (DB2)

DEDB data entry database (IMS)

DEQ dequeue

DFHMS Data Facility Hierarchical Storage Manager

DFR deferred

DFSMS Data Facility Storage Management Subsystem

DFW DASD fast write

DIB data in block

DIM data in memory

DL/1 Data Language 1

DLC data link control

DLISAS DL/1, separate address space

DLM Distributed Lock Manager (Oracle)

DLS data link switch

DLSw data link switching

DLUR dependent LU requester/server

DMA direct memory access

DMA dynamic memory array

DNS Domain Name Server

DP dispatching priority

DPL distributed program link (CICS)

DRDA distributed remote database access

DRDA distributed relational database architecture

DSG data sharing group

DSM distributed security manager

DSNT data set name table

DSP data space

DSR dynamic storage reconfiguration

DTE data terminal equipment

EC engineering change

ECB event control block

ECC error correction code

ECI external call interface (CICS)

ECL emitter coupled logic

ECS Enhanced Catalog Sharing

EDR enqueue residency value

EDT eligible device table

EMC event monitoring control

EMCS extended multiple console support

EMH expedited message handler (IMS)

EMHQ expedited message handler queue (IMS)

EMIF ESCON multiple image facility

EMIF enhanced multiple image facility

EN end node

ENF event notification facility

ENQ enqueue

ENTR Ethernet/Token-Ring

EP Emulation Program (3745)

EPDM Enterprise Data Manager

EPSO Enhanced S/390 Parallel Sysplex Offering

ERS enqueue residency value (GRS)

ES expanded storage

ES Enterprise Systems

ESCA ESCON adapter

ESCD ESCON director

ESCON Enterprise Systems Connection

ESO expanded storage only (Hiperspace)

ESTOR non-control storage (expanded storage for CF)

ETOD extended time-of-day

ETR external time reference

ETS external time source

EXCI external CICS interface (CICS)

EV execution velocity (WLM)

FAX facsimile

FDBR fast database recovery (IMS)

FF fast forward

FF fox fox (hexadecimal)

FF full function

FICON FIbre CONnection

FOR file-owning region (CICS)

FRAD frame relay access device

FRU field replaceable unit

FSS functional subsystem

FTP file transfer program

FTP file transfer protocol

FTS fiber transport services

GA general availability

GAC global access checking

GB gigabyte

GbE Gigabit Ethernet

GBP group buffer pool (DB2)

GDG generation data group

GEM global enterprise manager

GENCB generate control block

GMLC graduated monthly license charge

GMMA Goal Mode Migration Aid

GMT Greenwich mean time

GR generic resource

GRG generic resource group (IMS)

GRECP group buffer pool recovery pending (DB2)

GRG generic resource group

GRS global resource serialization

GSR global shared resources

GTF Generalized Trace Facility

GUI graphical user interface

GW gateway

HCD hardware configuration definition

HDAM hierarchic direct access method

HFS hierarchical file system (UNIX®)

HIDAM hierarchic indexed direct access method

HISAM hierarchic indexed sequential access method

HLQ high level qualifier

HMC hardware management console

HOD host-on-demand

HONE hands on network environment

HPDT high performance data transfer (APPC)

HPR high performance routing (APPN)

HRDW hardware

HSA hardware service area

HSB high speed buffer

HSC hardware system console

HSSI high speed serial interface

HW hardware

HWMCA hardware management console application

Hz hertz

I/O input/output

IBB internal bus buffer

IBF internal battery facility

IBM International Business Machines Corporation

IC internal coupling

ICB integrated cluster bus

ICE I/O controller element

ICF Integrated Catalog Facility

ICF Internal Coupling Facility

ICMF Integrated Coupling Migration Facility

ICP Interconnect Controller Program (IBM program product)

ICRF Integrated Cryptographic Feature

ICS installation control specifications

IFP IMS fast path

IIOP Internet Inter-Object Request Block Protocol

IML initial microcode load

IMS Information Management System

IMS DB Information Management System Database Manager

IMS TM Information Management System Transaction Manager

IOBF I/O buffer (IMS)

IOC I/O count (SRM)

IOCP input output configuration program

IODF I/O definition file

IOP I/O processor

IOQ I/O queue

IOSP input/output support processor

IPA IP assist

IP Internet protocol

IPC initial power controller

IPCS interactive problem control system

IPL initial program load

IPS installation performance specification

IPsec IP security protocol

IPX™ Internet packet exchange

IRC interregion communications

IRLM integrated resource lock manager

ISC inter-system communications (in CICS and IMS)

ISC inter-system coupling (CF link type)

ISD internal disk (channel path)

ISMF Integrated Storage Management Facility

ISPF Interactive System Productivity Facility

ISO integrated services offering

ISR intermediate session routing

ISV independent software vendor

ITR internal throughput rate

ITRR internal throughput rate ratio

ITSC International Technical Support Center

ITSO	International Technical Support Organization
ITU	International Telecommunication Union
JCL	job control language
JECL	JES control language
JES	Job entry subsystem
JMF	JES3 monitoring facility
JOE	job output element (JES)
KB	kilobyte
KGTV	Korhonen George Thorsen Vaupel
km	kilometer
LAN	local area network
LASER	light amplification by stimulated emission of radiation
LCU	logical control unit
LE	language environment
LED	light emitting diode
LFS	LAN file server
LIC	licensed internal code
LOB	large object (DB2)
LOC	locate
LOC	location
LP	logical partition
LPAR	logically partitioned mode
LPCTL	logical partition controls (frame)
LPL	logical page list
LSCD	large scale computing division
LSPR	large systems performance reference
LSR	local shared resources
LST	load module temporary store (SMP/E)
LTERM	logical terminal
LU	logical unit (SNA)
LUPS	local UPS
LX	long wavelength
MAC	medium access control
MAS	multiaccess spool
MAU	multistation access unit
MAX	maximum
MB	megabyte
Mbps	megabits per second
MBps	megabytes per second
MCCU	multisystem channel communications unit
MCDS	migration control data set (DFSMShsm)
MCL	maintenance change level
MCM	multichip module
MCS	multiple console support
MDB	message data block
MDH	migration data host
MES	miscellaneous equipment specification
MIB	management information base (OSI)
MICR	magnetic ink character recognition/reader
MIH	missing interrupt handler
MIM	Multi-Image Manager
MIPS	millions of instructions per second
MLSLEC	maximum list-set-element count
ML1	migration level 1 (DFSMShsm)
ML2	migration level 2 (DFSMShsm)
MODCB	modify control block
MM	multimode
MNPS	multinode persistent session
MP	multiprocessor
MPC+	multipath channel+
MPF	message suppressing facility
MPG	multiple preferred guests
MPL	multiprogramming level
MPNP	multiprotocol network program
MPR	message processing region (IMS)
MPTN	multiprotocol transport networking
MRNS	multiprotocol routing network services
MRO	multiregion operation
MSC	multisystems communication (IMS)
MSDB	main storage database (IMS)
MSGQ	shared message queue (IMS)
MSO	main storage occupancy (SRM)
MSS	multiprotocol switched services
MSU	millions of service units
MTS	macro temporary store (SMP/E)
MTU	maximum transmission unit (Ethernet)
MTW	mean time to wait
MUF	multi-user facility (Datacom/CA)
MULC	measured usage license charge
MVS	Multiple Virtual Storage

N/A not applicable
NAU network addressable unit
NDF non-deferred
NFS network file system
NIB node initialization block
NIP nucleus initialization process
NLP network layer packets
NN network node (SNA)
NNP network node processor
NNS named counter server (CICS)
NO number
NOOVLY no overlay
NSS national system support
NSSS networking and system services and support
NVS nonvolatile storage
OAS OSA address table
OCDS offline control data set (DFSMShsm)
OCF operations command facility
OCR optical character recognition/reader
ODBC open database connectivity
OEMI original equipment manufacturer information/interface
OLDS online log data set (IMS)
OLS offline sequence
OMF operations management facility
ONC™ open network computing
OO object-oriented
OPC operations planning and control
OS operating system
OSA open systems architecture
OSAM overflow sequential access method
OSI open systems interconnection
OSPF open shortest path first
OVLY overlay
PAF Processor Availability Facility
PCE processor controller element
PCM plug-compatible manufacturers
P/DAS peer-to-peer dynamic address switching
PD program directory
PDS partitioned data set
PDSE partitioned data set enhanced
PEL picture element
PEP Partitioned Emulation Program (3745)
PES Parallel Enterprise Server
PET platform evaluation test
PI performance index
PI path in
PI program isolation
PIF program interface
PLET product announcement letter
PLO perform locked operation
PM Presentation Manager (OS/2®)
PMIO performance management input/output
PMOS performance management offerings and services
PO path out
POR power-on reset
PP physical partitioned (mode of operation)
PPP point-to-point protocol
PPRC peer-to-peer remote copy
PPS pulse-per-second
PR/SM processor resource/system manager
PROFS® professional office system
PSB program specification block
PSLC Parallel Sysplex license charge
PSMF Parallel Sysplex Management Facility
PSO Parallel Server Option (Oracle)
PSP preventive service planning
PSP primary support processor
PTF program temporary fix
PTH path
PTS parallel transaction server
PU physical unit (SNA)
PU processing unit (9672)
QDIO Queued Direct Input/Output
QOR queue-owning region (CICS)
RACF Resource Access Control Facility
RAMAC® Random Access Method of Accounting and Control
RAS reliability availability serviceability
RBA relative block address
RBA relative byte address
RCD read configuration data

RDS restructured database (RACF)
REA RAMAC electronic array
REQ required
RETAIN® Remote Technical Assistance and Information Network
RG resource group
RIOC relative I/O content
RIP routing information protocol
RIT RECON initialization time stamp (IMS)
RJP remote job processor
RLL row-level locking
RLS record-level sharing (VSAM)
RLSWAIT RLS wait
RMF Resource Measurement Facility
RMODE residency mode
RNL resource name list (GRS)
ROT rules of thumb
RPC remote procedure call
RPL request parameter list
RPP relative processor performance
RPQ request for price quotation
RR repeatable read (DB2)
RRDF Remote Recovery Data Facility
RRMS Recoverable Resource Management Services (z/OS)
RRS resource recovery services
RRSF RACF Remote Sharing Facility
RSA RAMAC scalable array
RSA ring system authority (GRS)
RSM™ real storage manager
RSR remote site recovery
RSU recommended service upgrade
RTA real time analysis
RTM recovery termination manager
RTP rapid transport protocol
RVA RAMAC virtual array
SA z/OS System Automation for z/OS
SAA systems application architecture
SAE single application environment
SAF System Authorization Facility
SAP system assist processor
SAPR systems assurance product review
SCA shared communication area (DB2)
SCDS save control data set (SMP/E)
SCDS source control data set (SMS)
SCE system control element
SCH subchannel
SCKPF Store Clock Programmable Field
SCP system control program
SCS source control data set
SCSI small computer system interface
SCTC ESCON CTC control unit
SCV software-compatible vendor
SDEP sequential dependent (IMS)
SDT shared data tables (CICS)
SE support element
SEC system engineering change
SECS seconds
SETI SE technical information (HONE)
SFM sysplex failure management
SHISAM simple hierarchic indexed sequential access method
SI single image (mode of operation)
SID system identifier
SIE start interpretive execution (instruction)
SIGP signal processor (instruction)
SIMETR simulated external time reference (z/OS)
SLA service level agreement
SLIP serviceability level indicator processing
SLM system lock manager (XCF)
SLMH single-link multihost (IBM 3174 with ESCON interface)
SLSS system library subscription service
SM single-mode
SMF Systems Management Facility
SMOL sales manual online (HONE)
SMP/E system modification program/extended
SMPLOG SMP/E log
SMPLTS SMP/E load module temporary store
SMPMTS SMP/E macro temporary store
SMPSCDS SMP/E save control data set
SMPSTS SMP/E source temporary store
SMQ shared message queue (IMS)
SMS system managed storage
SMSVSAM system managed storage VSAM

SNA systems network architecture

SNAP/SHOT® system network analysis program/ simulation host overview technique

SNI SNA network interconnect

SNMP simple network management protocol

SP system product

SPE small programming enhancement

SPOC single point of control

SPOF single point of failure

SPOOL simultaneous peripheral operation online

SPUFI SQL processor using file input (DB2)

SQG shared queues group (IMS)

SQL structured query language (DB2)

SQL/DS™ structured query language/data system (VM)

SRB service request block

SRDS structure recovery data set (CQS)

SROD shared read-only database (IMS)

SRM System Resources Manager

S/S start/stop

SSCH start subchannel

SSI single system image

SSI subsystem interface

SSL secure socket layer

SSM secondary space management (DFSMShsm)

SSP subsystem storage protect

STC started task control

STCK store clock (instruction)

STCKE store clock extended (instruction)

STI self-timed interconnect (S/390)

STS source temporary store

STSI Store System Information Instruction (S/390)

SUBSYS subsystem

SC Shared Cache (IDMS/CA)

SVC supervisor call (instruction)

SVS solutions validation services

SW software

SX short wavelength

SYNC synchronous

SYSAFF system affinity

SYSCONS system consoles

SYSID system identifier

SYSPLEX systems complex

SYSRES system residence volume (or IPL volume)

TAI French for International Atomic Time

TCB task control block (z/OS)

TCM thermal conduction module

TCP/IP Transmission Control Protocol/Internet Protocol

TCU terminal control unit

TESTCB test control block

TG transmission group

TKE trusted key entry

TM transaction manager

TME® Tivoli Management Environment®

TMM tape mount management

TOD time of day

TOR terminal-owning region

TP transaction program (APPC)

TPF Transaction Processing Facility

TPNS Teleprocessing Network Simulator

TS temporary storage (CICS)

TS transaction server (CICS)

TSC TOD synchronization compatibility

TSO Time Sharing Option

TSOR temporary storage-owning region (CICS)

TSQ temporary storage queue (CICS)

TTL time-to-live (TCP/IP)

UBF user buffering

UCW unit control word

UD undeliverable (message)

UDB universal database (DB2)

UDF update-time-order (DB2)

UDP user diagram protocol

UOW unit of work

UP uniprocessor

UPS uninterruptible power supply/system

UR uncommitted read (DB2)

URL universal resource locator

UTC universal time coordinate

VF vector facility

VGA video graphics array/adapter

VIO	virtual I/O
VIPA	virtual IP addressing (TCP/IP)
VLF	virtual lookaside facility
VM	Virtual Machine
VPD	vital product data
VR	virtual route (SNA)
VRTG	virtual-route-based transmission group (SNA)
VSAM	Virtual Storage Access Method
VSO	virtual storage option (IMS)
VTAM	Virtual Telecommunications Access Method
VTOC	volume table of content
VTS	virtual tape server
VVDS	VSAM volume data set
WAITRBLD	wait for rebuild (IMS)
WAN	wide area networks
WLM	Workload Manager
WLR	IMS/ESA workload router
WQE	write-to-operator-queue-element
WSC	Washington System Center
WSS	working set size
WTAS	world trade account system
WTO	write-to-operator
WTOR	write-to-operator-with-reply
WWQA	world wide question & answer
WWW	World Wide Web
XCA	external communications adapter
XCF	Cross-System Coupling Facility
XDF	Extended Distance Feature
XES	Cross-System Extended Services
XI	cross-invalidate
XJS	extended job entry subsystem
XRC	extended remote copy
XRF	Extended Recovery Facility

Glossary

Explanations of cross-references

The following cross-references are used in this glossary:

Contrast with.	This refers to a term that has an opposed or substantively different meaning.
See.	This refers the reader to multiple-word terms in which this term appears.
See also.	This refers the reader to terms that have a related, but not synonymous meaning.
Synonym for.	This indicates that the term has the same meaning as a preferred term, which is defined in the glossary.

If you do not find the term you are looking for, see the IBM Software Glossary at the Web site:

http://www.networking.ibm.com/nsg/nsgmain.htm

A

abend. Abnormal end of task.

ACF/VTAM. Advanced Communications Function for the Virtual Telecommunications Access Method. Synonym for *VTAM*.

active IRLM. The IRLM supporting the active IMS subsystem in an XRF complex.

active service policy. The service policy that determines workload management processing if the sysplex is running in goal mode. See *goal mode*.

adapter. Hardware card that allows a device, such as a PC, to communicate with another device, such as a monitor, a printer, or other I/O device. In a LAN, within a communicating device, a circuit card that, with its associated software or microcode, enables the device to communicate over the network.

affinity. A connection or association between two objects.

alternate IRLM. The IRLM supporting the alternate IMS subsystem in an XRF complex.

alternate site. Another site or facility, such as a commercial hot site or a client-owned second site, that will be a recovery site in the event of a disaster.

ambiguous cursor. A database cursor that is not declared with either the clauses FOR FETCH ONLY or FOR UPDATE OF, and is not used as the target of a WHERE CURRENT OF clause on an SQL UPDATE or DELETE statement. The package processes dynamic SQL statements.

architecture. A logical structure that encompasses operating principles, including services, functions, and protocols. See *computer architecture, network architecture*, and *Systems Network Architecture (SNA)*.

asynchronous. Without regular time relationship. Unexpected or unpredictable with respect to the program's instructions, or to time. Contrast with synchronous.

authorized program analysis report (APAR). A request for correction of a problem caused by a defect in a current release of a program unaltered the user.

availability. A measure of how much (often specified as a percentage) the data processing services are available to the users in a specified time frame.

B

base or basic sysplex. A base or basic sysplex is the set of one or more z/OS systems that is given a cross-system coupling facility (XCF) name and in which the authorized programs can then use XCF coupling services. A Base Sysplex does not include a CF. See also *Parallel Sysplex* and *sysplex*.

basic mode. A central processor mode that does not use logical partitioning. Contrast with *logically partitioned (LPAR) mode*.

batch checkpoint/restart. The facility that enables batch processing programs to synchronize checkpoints and to be restarted at a user-specified checkpoint.

batch environment. The environment in which non-interactive programs are executed. The environment schedules their execution independently of their submitter.

batch message processing (BMP) program. An IMS batch processing program that has access to online databases and message queues. BMPs run online, but like programs in a batch environment, they are started with job control language (JCL).

batch-oriented BMP program. A BMP program that has access to online databases and message queues while performing batch-type processing. A batch-oriented BMP does not access the IMS message queues for input or output. It can access online databases, GSAM databases, and z/OS files for both input and output.

batch processing program. An application program that has access to databases and z/OS data management facilities but does not have access to the IMS control region or its message queues. See also *batch message processing program* and *message processing program.*

block level sharing. A kind of data sharing that enables application programs in different IMS systems to update data concurrently.

block multiplexer channel. A channel that transmits blocks of data to and from more than one device by interleaving the record blocks. Contrast with *selector channel*.

BMP program. See *batch message processing program.*

buffer. A portion of storage used to hold input or output data temporarily. A routine or storage used to compensate for a difference in data rate or time of occurrence of events, when transferring data from one device to another.

buffer invalidation. A technique for preventing the use of invalid data in a Parallel Sysplex data sharing environment. The technique involves marking all copies of data in DB2 or IMS buffers invalid once a sharing DBMS subsystem has updated that data.

buffer pool. A set of buffers that contains buffers of the same length. See also *buffer*, *buffer invalidation*, and *group buffer pool*.

byte multiplexer channel. A multiplexer channel that interleaves bytes of data. See also *block multiplexer channel*. Contrast with *selector channel*.

C

cache structure. A CF structure that enables high-performance sharing of cached data by multisystem applications in a sysplex. Applications can use a cache structure to implement several different types of caching systems, including a store-through or a store-in cache. As an example, DB2 uses data sharing group cache structures as GBps. See also *group buffer pool*, *castout*, and *cache structure services.*

cache structure services. z/OS services that enable applications in a sysplex to perform operations such as the following on a CF cache structure:

- Manage cache structure resources.
- Store data into and retrieve data from a cache structure.
- Manage accesses to shared data.
- Determine when shared data has been changed.
- Determine whether a local copy of shared data is valid.

card-on-board (COB) logic. The type of technology that uses pluggable, air-cooled cards.

CAS. Coordinating address space.

castout. The DB2 process of writing changed pages from a GBP to DASD.

catalog. A data set that contains extensive information required to locate other data sets, to allocate and deallocate storage space, to verify the access authority of a program or operator, and to accumulate data set usage statistics.

central processing unit (CPU). The part of a computer that includes the circuits that control the interpretation and execution of instructions.

central processor (CP). The part of the computer that contains the sequencing and processing facilities for instruction execution, initial program load, and other machine operations. See also *central processor complex*, *central electronic complex*, and *PU*.

central processor complex (CPC). A physical collection of hardware that includes central storage, one or more CPs, timers, and channels.

central storage. Storage that is an integral part of the processor unit. Central storage includes both main storage and the hardware system area.

CF. Coupling Facility. See also *Coupling Facility*.

CFCC. Coupling Facility Control Code. See also *Coupling Facility Control Code*.

CFRM policy. A declaration regarding the allocation rules for a CF structure. See also *structure*.

channel. A functional unit, controlled by a S/390 CPC, which handles the transfer of data between processor storage and local peripheral equipment. A path along which signals can be sent. The portion of a storage medium that is accessible to a given reading or writing station. In broadband transmission, a designation of a frequency band 6 MHz wide.

channel subsystem (CSS). A collection of subchannels that directs the flow of information between I/O devices and main storage, relieves the processor of communication tasks, and performs path management functions.

channel-to-channel (CTC). Refers to the communication (transfer of data) between programs on opposite sides of a channel-to-channel adapter (CTCA).

channel-to-channel adapter (CTCA). A hardware device that can be used to connect two channels on the same computing system or on different systems.

CICSPlex. The largest set of CICS systems to be monitored and controlled as a single entity. In a CICSPlex SM environment, the user-defined name, description, and configuration information for a CICSPlex. A CICSPlex can be made up of CICS systems or CICS system groups. See also *CICS system* and *CICS system group*.

CICSPlex SM. CICSPlex System Manager.

CICSPlex SM address space (CMAS). A CICSPlex SM component that is responsible for managing a CICSPlex. A CMAS provides the single system image for a CICSPlex by serving as the interface to other CICSPlexes and external programs. There must be at least one CMAS for each z/OS image on which you are running CICSPlex SM. A single CMAS can manage CICS systems within one or more CICSPlexes. See also *coordinating address space (CAS)* and *managed address space (MAS)*.

CICSPlex System Manager (CICSPlex SM). An IBM CICS systems management product that provides single system image and single point of control for one or more CICSPlexes, including CICSPlexes on heterogeneous operating systems.

classification. The process of assigning a service class and, optionally, a report class to a work request. Subsystems, together with workload management services, use classification rules to assign work to a service class when it enters a sysplex.

classification rules. The rules workload management and subsystems use to assign a service class and, optionally, a report class to a work request. A classification rule consists of one or more of the following work qualifiers: subsystem type, subsystem instance, user ID, accounting information, transaction name, transaction class, source LU, NETID, and LU name.

CMAS. CICSPlex SM address space. See also *CICSPlex SM address space (CMAS)*.

CMAS link. A communications link between one CICSPlex SM address space (CMAS) and another CMAS or a remote managed address space (MAS). CMAS links are defined when CICSPlex SM is configured.

CNC. Mnemonic for an ESCON channel-attached to an ESCON-capable device.

command. An instruction that directs a control unit or device to perform an operation or a set of operations.

commit. In data processing, the point at which the data updates are written to the database in a way that is irrevocable.

compatibility mode. A mode of processing in which the SRM parmlib members IEAIPSxx and IEAICSxx determine system resource management. See also *goal mode*.

complementary metal-oxide semiconductor (CMOS). A technology that combines the electrical properties of positive and negative voltage requirements to use considerably less power than other types of semiconductors.

component. Hardware or software that is part of a functional unit. A functional part of an operating system; for example, the scheduler or supervisor.

computer architecture. The organizational structure of a computer system, including hardware and software.

configuration. The arrangement of a computer system or network as defined by the nature, number, and chief characteristics of its functional units. More specifically, the term *configuration* may refer to a hardware configuration or a software configuration. See also *system configuration.*

connectivity. A term used to describe the physical interconnections of multiple devices/computers/networks employing similar or different technology or architecture together to accomplish effective communication between and among connected members. It involves data exchange or resource sharing.

console. A logical device that is used for communication between the user and the system. See also *service console*.

construct. A collective name for data class, storage class, management class, and storage group.

continuous availability. The elimination or masking of both planned and unplanned outages, so that no system outages are apparent to the user. Continuous availability can also be stated as the ability to operate 24 hours/day, 7 days/week, with no outages apparent to the user.

continuous operations. The elimination or masking of planned outages. A system that delivers continuous operations is a system that has no scheduled outages.

control interval (CI). A fixed-length area of direct access storage in which VSAM creates distributed free space and stores records. Also, in a key-sequenced data set or file, the set of records pointed to by an entry in the sequence-set index record. The control interval is the unit of information that VSAM transmits to or from direct access storage. A control interval always comprises an integral number of physical records.

control region. The z/OS main storage region that contains the IMS control program.

control unit. A general term for any device that provides common functions for other devices or mechanisms. Synonym for *controller*.

coordinating address space (CAS). An z/OS subsystem that provides ISPF user access to the CICSPlex. There must be at least one CAS for each z/OS image on which you are running CICSPlex SM. See also *CICSPlex SM address space (CMAS)* and *managed address space (MAS).*

couple data set. A data set that is created through the XCF couple data set format utility and, depending on its designated type, is shared by some or all of the z/OS systems in a sysplex. See also *Sysplex couple data set* and *XCF couple data set*.

Coupling Facility (CF). A special LP that provides high-speed caching, list processing, and locking functions in Parallel Sysplex. See also *Coupling Facility channel*, *Coupling Facility white space*, and *coupling services*.

Coupling Facility channel (CF link). A high bandwidth fiber optic channel that provides the high-speed connectivity required for data sharing between a CF and the CPCs directly attached to it.

Coupling Facility Control Code (CFCC). The Licensed Internal Code (LIC) that runs in a CF LP to provide shared storage management functions for a sysplex.

Coupling Facility Data Tables (CFDT). CFDT enables user applications, running in different CICS regions that reside in one or more z/OS images, within a Parallel Sysplex, to share working data with update integrity.

Coupling Facility white space. CF storage set aside for rebuilding of structures from other CFs, in case of failure.

coupling services. In a sysplex, the functions of XCF that transfer data and status between members of a group residing on one or more z/OS systems in the sysplex.

CPU service units. A measure of the task control block (TCB) execution time multiplied by an SRM constant that is CPC-model-dependent. See also *service unit*.

CP TOD. In a CPC with more than one CP, each CP can have a separate TOD clock, or more than one CP might share a clock, depending on the model. In all cases, each CP has access to a single clock called a CPC TOD clock.

common queue server (CQS). A server that receives, maintains, and distributes data objects from a shared queue on a CF list structure for its clients.

cross-system coupling facility (XCF). XCF is a component of z/OS that provides functions to support cooperation between authorized programs running within a sysplex.

cross-system extended services (XES). Provides services for z/OS systems in a sysplex to share data on a CF.

cryptographic. Pertaining to the transformation of data to conceal its meaning.

Customer Information Control System (CICS). An IBM-licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

CVC. Mnemonic for an ESCON channel-attached to a IBM 9034 (ESCON Converter).

D

daemon. A task, process, or thread that intermittently awakens to perform some chores and then goes back to sleep (software).

data entry database (DEDB). A direct-access database that consists of one or more areas, with each area containing both root segments and dependent segments. The database is accessed using VSAM media manager.

Data Facility Hierarchical Storage Manager (DFHSM). An IBM-licensed program used to back up, recover, and manage space on volumes.

Data Language/I (DL/I). The IMS data manipulation language, a common high-level interface between a user application and IMS. DL/I calls are invoked from application programs written in languages such as PL/I, COBOL, VS Pascal, C, and Ada. It can also be invoked from assembler language application programs by subroutine calls. IMS lets the user define data structures, relate structures to the application, load structures, and reorganize structures.

data link. Any physical link, such as a wire or a telephone circuit, that connects one or more remote terminals to a communication control unit, or connects one communication control unit with another. The assembly of parts of two data terminal equipment (DTE) devices that are controlled by a link protocol and the interconnecting data circuit, and that enable data to be transferred from a data source to a data link. In SNA, see also *link*. *Note*: A telecommunication line is only the physical medium of transmission. A data link includes the physical medium of transmission, the protocol, and associated devices and programs; it is both physical and logical.

data set. The major unit of data storage and retrieval, consisting of a collection of data in one of several prescribed arrangements and described by control information to which the system has access.

data sharing. In Parallel Sysplex, the ability of concurrent subsystems (such as DB2 or IMS database managers) or application programs to directly access and change the same data while maintaining data integrity. See also *Sysplex data sharing* and *data sharing group*.

data sharing group. A collection of one or more subsystems that directly access and change the same data while maintaining data integrity. See also *DB2 data sharing group* and *IMS DB data sharing group*.

data sharing. A Parallel Sysplex where data is shared at the record level across more than one system, using a CF structure to guarantee cross-system integrity.

database. A set of data, or a part or the whole of another set of data, that consists of at least one file and is sufficient for a given purpose or for a given data-processing system. A collection of data fundamental to a system. See also *Database Control (DBCTL)*, *data entry database (DEDB)*, *data sharing*, and *data sharing group*.

Database Control (DBCTL). An environment allowing full-function databases and DEDBs to be accessed from one or more transaction management subsystems.

DB2 data sharing group. A collection of one or more concurrent DB2 subsystems that directly access and change the same data while maintaining data integrity.

DDF. Distributed Data Facility (DB2). DB2 subsystem running in an address space that supports VTAM communications with other DB2 subsystems and supports the execution of distributed database access requests on behalf of remote users. This provides isolation of remote function execution from local function execution.

DEDB. See *data entry database*.

delay monitoring services. The workload management services that monitor the delays encountered by a work request.

device. A mechanical, electrical, or electronic contrivance with a specific purpose. An input/output unit, such as a terminal, display, or printer.

direct access storage device (DASD). A physical device, such as the IBM 3390, in which data can be permanently stored and subsequently retrieved using licensed products like IMS and DB2, or using IBM-supported access methods like VSAM in operating system environments like z/OS.

directory. A list of files that are stored on a disk or diskette. A directory also contains information about the file, such as size and date of last change.

disaster. An event that renders IT services unavailable for an extended period. Often the IT facilities must be moved to another site in the event of a disaster.

DNS. See *Domain Name System*.

domain name. In the Internet suite of protocols, a name of a host system. A domain name consists of a sequence of subnames separated by a delimiter character. For example, if the fully qualified domain name (FQDN) of host system is ralvm7.vnet.ibm.com, each of the following is a domain name: ralvm7.vnet.ibm.com, vnet.ibm.com, and ibm.com.

domain name server. In the Internet suite of protocols, a server program that supplies name-to-address translation by mapping domain names to IP addresses. Synonymous with *name server*.

Domain Name System (DNS). In the Internet suite of protocols, the distributed database system used to map domain names to IP addresses.

dynamic. Pertaining to an operation that occurs at the time it is needed rather than at a predetermined or fixed time. See also *dynamic connection*, *dynamic connectivity*, *dynamic reconfiguration*, *dynamic reconfiguration management*, and *dynamic storage connectivity*.

dynamic CF dispatching. With dynamic CF dispatching, the CF will monitor the request rate that is driving it and adjust its usage of CP resource accordingly. If the request rate becomes high enough, the CF will revert back to its original dispatching algorithm, constantly looking for new work. When the request rate lowers, the CF again becomes more judicious with its use of CP resource. See also *dynamic ICF expansion*.

dynamic connection. In an ESCON director, a connection between two ports, established or removed by the ESCD and that, when active, appears as one continuous link. The duration of the connection depends on the protocol defined for the frames transmitted through the ports and on the state of the ports.

dynamic connectivity. In an ESCON director, the capability that allows connections to be established and removed at any time.

dynamic ICF expansion. Dynamic ICF expansion provides the ability for a CF LP that is using a dedicated ICF to expand into the pool of shared ICFs or shared CPs. At low request rates, the resource consumption of the shared PU should be 1% to 2%. As the request rate increases, the resource consumption will increase, up to the point where the LP will consume its full share of the shared PU as defined by the LPAR weights. See also *dynamic CF dispatching*.

dynamic reconfiguration. Pertaining to a processor reconfiguration between a single-image (SI) configuration and a physically partitioned (PP) configuration when the system control program is active.

dynamic reconfiguration management. In z/OS, the ability to modify the I/O configuration definition without needing to perform a power-on reset (POR) of the hardware or an initial program load (IPL).

dynamic storage reconfiguration. A PR/SM LPAR function that allows central or expanded storage to be added or removed from a logical partition without disrupting the system control program operating in the logical partition.

E

ECS. Enhanced Catalog Sharing (DFSMS/MVS V1.5).

EMIF. Enhanced multiple image facility (formerly ESCON multiple image facility). A facility that allows the sharing of FICON or ESCON channels between LPs.

emitter. In fiber optics, the source of optical power.

end node. A type 2.1 node that does not provide any intermediate routing or session services to any other node. For example, APPC/PC is an end node.

enhanced catalog sharing. By using a CF cache structure instead of DASD to store catalog sharing control information, shared catalog performance in sysplex environment is improved. This sharing method, called enhanced catalog sharing (ECS), eliminates a reserve, dequeue, and I/O request to the VVDS on most catalog calls.

Enhanced Multiple Image Facility (EMIF). See EMIF.

enhanced sysplex. An enhanced sysplex is a sysplex with one or more CFs. See also *Base Sysplex* and *sysplex*.

enterprise. A business or organization that consists of two or more sites separated by a public right-of-way or a geographical distance.

Enterprise Systems Connection (ESCON). A set of products and services that provides a dynamically connected environment using optical cables as a transmission medium. See also *ESCD*, *ESCM*, and *ESCON channel*.

Environmental Services Subsystem (ESSS). A component of CICSPlex SM that owns all the data spaces used by the product in an z/OS image. The ESSS executes at initialization and remains in the z/OS image for the life of the IPL to ensure that the data spaces can survive the loss of a CICSPlex SM address space (CMAS).

ESA/390. Enterprise Systems Architecture/390.

ESCD. Enterprise Systems Connection (ESCON) Director. See also *ESCD console*, *ESCD console adapter*, and *ESCM*.

ESCD console. The ESCON director input/output device used to perform operator and service tasks at the ESCD.

ESCD console adapter. Hardware in the ESCON director console that provides the attachment capability between the ESCD and the ESCD console.

ESCM. See *ESCON Manager*.

ESCON channel. A channel having an Enterprise Systems Connection channel-to-control-unit I/O interface that uses optical cables as a transmission medium. Contrast with *parallel channel*.

ESCON director (ESCD). A device that provides connectivity capability and control for attaching any two links to each other.

ESCON Extended Distance Feature (ESCON XDF). An ESCON feature that uses laser/single-mode fiber optic technology to extend unrepeated link distances up to 20 km. LPs in an ESCON environment.

ESCON Manager (ESCM). A licensed program that provides S/390 CPC control and intersystem communication capability for ESCON director connectivity operations.

ESCON multiple image facility (EMIF). A facility that allows channels to be shared among PR/SM logical partitions in an ESCON environment.

ESCON XDF. ESCON extended distance feature.

Ethernet. A local area network that was originally marketed by Xerox Corp. The name is a trademark of Xerox Corp.

ETR. See *External Time Reference*.

ETR offset. The time zone offset identifies your system location within a network of other systems. This offset is the difference between your local time and Universal Time Coordinate (UTC). See also *Universal Time Coordinate*.

ETS. See *External Time Source*.

exclusive lock. A lock that prevents concurrently executing application processes from reading or changing data. Contrast with *shared lock.*

expanded storage. Optional integrated high-speed storage that transfers 4 KB pages to and from central storage. Additional (optional) storage that is addressable by the system control program.

extended Parallel Sysplex. This name is sometimes used to refer to Parallel Sysplexes that exploit data sharing and future enhancements for ultra high availability and disaster recovery.

Extended Recovery Facility (XRF). Software designed to minimize the effect of failures in z/OS, VTAM, the S/390 CPC CP, or IMS/VS on sessions between IMS/VS and designated terminals. It provides an alternate subsystem to take over failing sessions.

External Time Reference (ETR). This is how z/OS documentation refers to the 9037 Sysplex Timer. An ETR consists of one or two 9037s and their associated consoles.

External Time Source (ETS). An accurate time source used to set the time in the Sysplex Timer. The accurate time can be obtained by dialing time services or attaching to radio receivers or time code generators.

F

false lock contention. A contention indication from the CF when multiple lock names are hashed to the same indicator and when there is no real contention.

Fast Path. IMS functions for applications that require good response characteristics and that may have large transaction volumes. Programs have rapid access to main-storage databases (to the field level), and to direct-access data entry databases. Message processing is grouped for load balancing and synchronized for database integrity and recovery. See also *MSDB* and *DEDB.*

Fast Path databases. Two types of databases designed to provide high data availability and fast processing for IMS applications. They can be processed by the following types of programs: MPPs, BMPs, and IFPs. See also *main storage database* and *data entry database.*

feature. A part of an IBM product that can be ordered separately by the client.

FICON channel. FIbre CONnection. A S/390 channel that uses industry standard Fibre Channel Standard (FCS) as a base.

file system. The collection of files and file management structures on a physical or logical mass storage device, such as a disk.

format. A specified arrangement of things, such as characters, fields, and lines, usually used for displays, printouts, or files. To arrange things such as characters, fields, and lines.

forward recovery. Reconstructing a file or database by applying changes to an older version (backup or image copy) with data recorded in a log data set. The sequence of changes to the restored copy is in the same order in which they were originally made.

frame. For an S/390 microprocessor cluster, a frame may contain one or more CPCs, support elements, and AC power distribution.

frequency. The rate of signal oscillation, expressed in hertz (cycles per second).

full function databases. Hierarchic databases that are accessed through Data Language I (DL/1) call language and can be processed by all four types of application programs: IFP, MPPs, BMPs, and batch. Full function databases include HDAM, HIDAM, HSAM, HISAM, SHSAM, and SHISAM.

G

generic resource name. A name used by VTAM that represents several application programs that provide the same function in order to handle session distribution and balancing in a sysplex.

gigabytes. One billion (10^9) bytes.

global locking (DB2). For data consistency in a data sharing environment, locks must be known and respected between all members. DB2 data sharing uses global locks that ensure that each member is aware of all members' locks.

Two locking mechanisms are used by DB2 data sharing to ensure data consistency, logical locks, and physical locks.

The two types can be briefly compared as follows:

1. Logical locks

   Logical locks are used to control concurrent access from application processes, such as transactions or batch programs.

2. Physical locks

   Physical locks are used by DB2 members to control physical resources

   – Page set physical locks are used to track the level of interest in a particular page set or partition and thus determine the needed GBP coherency controls.

   – Page physical locks are used to preserve the physical consistency of pages.

See also *P-lock*.

global resource serialization (GRS). A component of z/OS used for sharing system resources and for converting DASD reserve volumes to data set ENQueues.

global resource serialization complex (GRSplex). One or more z/OS systems that use global resource serialization to serialize access to shared resources (such as data sets on shared DASD volumes).

GMT. See *Greenwich Mean Time*.

goal mode. A mode of processing where the active service policy determines system resource management. See also *compatibility mode*.

Greenwich Mean Time (GMT). Time at the time zone centered around Greenwich, England.

group buffer pool. A CF cache structure used by a DB2 data sharing group to cache data and to ensure that the data is consistent for all members. See also *buffer pool*.

group services. Services for establishing connectivity among the multiple instances of a program, application, or subsystem (members of a group running on z/OS) in a sysplex. Group services allow members of the group to coordinate and monitor their status across the systems of a sysplex.

H

Hardware Management Console. A console used to monitor and control hardware, such as the 9672 CPCs.

hardware system area (HSA). A logical area of central storage, not addressable by application programs, used to store Licensed Internal Code and control information.

highly parallel. Refers to multiple systems operating in parallel, each of which can have multiple processors. See also *n-way*.

high-speed buffer. A cache or a set of logically partitioned blocks that provides significantly faster access to instructions and data than that provided by central storage.

HiPerLink. A HiPerLink provides improved CF link efficiency and response times in processing CF requests, compared to previous CF link configurations. With HiPerLinks, current data sharing overheads are reduced and CF link capacity is improved.

host (computer). In a computer network, a computer that provides users with services such as computation and databases and that usually performs network control functions. The primary or controlling computer in a multiple-computer installation.

HSA. See *hardware system area*.

I

IBF. See *Internal Battery Feature*.

IC. See *Internal Coupling Link*.

ICB. See *Integrated Cluster Bus*.

ICF. See *Internal Coupling Facility*.

ICF. Integrated Catalog Facility.

importance level. An attribute of a service class goal that indicates the importance of meeting the goal relative to other service class goals, in five levels: lowest, low, medium, high, and highest.

IMS DB data sharing group. A collection of one or more concurrent IMS DB subsystems that directly access and change the same data while maintaining data integrity. The components in an IMS DB data sharing group include the sharing IMS subsystems, the IRLMs they use, the IRLM, OSAM, and VSAM structures in the CF, and a single set of DBRC RECONs.

IMS system log. A single log made up of online data sets (OLDSs) and write-ahead data sets (WADSs).

in-doubt period. The period during which a unit of work is pending during commit processing that involves two or more subsystems. See also *in-doubt work unit*.

in-doubt work unit. In CICS/ESA and IMS/ESA, a piece of work that is pending during commit processing; if commit processing fails between the polling of subsystems and the decision to execute the commit, recovery processing must resolve the status of any work unit that is in doubt.

indirect CMAS. A CICSPlex SM address space (CMAS) that the local CMAS can communicate with through an adjacent CMAS. There is no direct CMAS-to-CMAS link between the local CMAS and an indirect CMAS. Contrast with *adjacent CMAS*. See also *local CMAS*.

initial microcode load (IML). The action of loading the operational microcode.

initial program load (IPL). The initialization procedure that causes an operating system to start operation.

input/output support processor (IOSP). The hardware unit that provides I/O support functions for the primary support processor (PSP). It also provides maintenance support function for the processor controller element (PCE).

installed service definition. The service definition residing in the couple data set for WLM. The installed service definition contains the active service policy information.

interactive. Pertaining to a program or system that alternately accepts input and then responds. An interactive system is conversational; that is, a continuous dialog exists between user and system. Contrast with *batch*.

interface. A shared boundary. An interface might be a hardware component to link two devices, or it might be a portion of storage or registers accessed by two or more computer programs.

Integrated Cluster Bus channel (ICB). The Integrated Cluster Bus channel uses the Self Timed Interface to perform the S/390 coupling communication. The cost of coupling is reduced by using a higher performing (Approximately 280 MBps) but less complex transport link suitable for the relatively short distances (The cable is 10 meters; the distance between CPCs is approximately 7 meters).

Integrated Offload Processor (IOP). The processor in the interconnect communication element that detects, initializes, and ends all channel subsystem operations.

Integrated Coupling Migration Facility (ICMF). A PR/SM LPAR facility that emulates CF links for LPs (CF LPs and z/OS LPs) running on the same CPC to assist in the test and development of data sharing applications.

internal battery feature (IBF). The internal battery feature (IBF) provides the function of a local uninterruptible power source (UPS). This feature may increase power line disturbance immunity for S/390 CPCs.

Internal Coupling channel (IC). The Internal Coupling channel emulates the coupling facility functions in microcode between images within a single CPC. It is a high performance channel transferring data at up to 6 Gbps Internal Coupling implementation is a totally logical channel requiring no channel or even cable hardware. However, a CHPID number must be defined in the IOCDS. A replacement for ICMF.

Internal Coupling Facility (ICF). The Internal Coupling Facility (ICF) uses up to two spare PUs on selected S/390 CPCs. The ICF may use CF links or emulated links (ICMF). It can be used initially as an entry configuration into Parallel Sysplex and then maintained as a backup configuration in the future.

interrupt. A suspension of a process, such as execution of a computer program caused by an external event, and performed in such a way that the process can be resumed. To stop a process in such a way that it can be resumed. In data communication, to take an action at a receiving station that causes the sending station to end a transmission. To temporarily stop a process.

invalidation. The process of removing records from cache because of a change in status of a subsystem facility or function, or because of an error while processing the cache image of the set of records. When such a cache image is invalidated, the corresponding records cannot be accessed in cache and the assigned cache space is available for allocation.

IOCDS. I/O configuration data set.

IOCP. I/O configuration program.

I/O service units. A measure of individual data set I/O activity and JES spool reads and writes for all data sets associated with an address space.

J

Job Entry Subsystem (JES). A system facility for spooling, job queuing, and managing I/O.

jumper cable. In an ESCON environment, an optical cable, having two conductors, that provides physical attachment between two devices or between a device and a distribution panel. Contrast with *trunk cable*.

L

latency. The time interval between the instant at which an instruction control unit initiates a call for data and the instant at which the actual transfer of data starts.

leap second. Corrections of exactly one second inserted into the UTC time scale since January 1, 1972. This adjustment occurs at the end of a UTC month, normally on June 30 or December 31. Seconds are occasionally added to or subtracted from the UTC to compensate for the wandering of the earth's polar axis and maintain agreement with the length of the solar day. See also *Universal Time Coordinate (UTC)*.

LIC. See *Licensed Internal Code*.

Licensed Internal Code (LIC). Software provided for use on specific IBM machines and licensed to clients under the terms of the IBM Customer Agreement. Microcode can be Licensed Internal Code and licensed as such.

link. The combination of physical media, protocols, and programming that connects devices.

list structure. A CF structure that enables multisystem applications in a sysplex to share information organized as a set of lists or queues. A list structure consists of a set of lists and an optional lock table, which can be used for serializing resources in the list structure. Each list consists of a queue of list entries.

list structure services. z/OS services that enable multisystem applications in a sysplex to perform operations, such as the following, on a CF list structure:

- Read, update, create, delete, and move list entries in a list structure.
- Perform serialized updates on multiple list entries in a list structure.
- Monitor lists in a list structure for transitions from empty to non-empty.

local cache. A buffer in local system storage that might contain copies of data entries in a CF cache structure.

local CMAS. The CICSPlex SM address space (CMAS) that a user identifies as the current context when performing CMAS configuration and management tasks.

local MAS. A managed address space (MAS) that resides in the same z/OS image as the CICSPlex SM address space (CMAS) that controls it and that uses the Environmental Services Subsystem (ESSS) to communicate with the CMAS.

lock resource. Data accessed through a CF structure.

lock structure. A CF structure that enables applications in a sysplex to implement customized locking protocols for serialization of application-defined resources. The lock structure supports shared, exclusive, and application-defined lock states, as well as generalized contention management and recovery protocols. See also *exclusive lock, shared lock,* and *false lock contention*.

lock structure services. z/OS services that enable applications in a sysplex to perform operations, such as the following, on a CF lock structure:

- Request ownership of a lock.
- Change the type of ownership for a lock.
- Release ownership of a lock.
- Manage contention for a lock.
- Recover a lock held by a failed application.

logical connection. In a network, devices that can communicate or work with one another because they share the same protocol.

logical control unit. A group of contiguous words in the HSA that provides all of the information necessary to control I/O operations through a group of paths that are defined in the IOCDS. Logical control units represent to the channel subsystem a set of control units that attach common I/O devices.

logical partition (LP). In LPAR mode, a subset of the processor unit resources that is defined to support the operation of a system control program (SCP). See also *logically partitioned (LPAR) mode*.

logical unit (LU). In VTAM, the source and recipient of data transmissions. Data is transmitted from one logical unit (LU) to another LU. For example, a terminal can be an LU, or a CICS or IMS system can be an LU.

logically partitioned (LPAR) mode. A CPC power-on reset mode that enables use of the PR/SM feature and allows an operator to allocate CPC hardware resources (including CPs, central storage, expanded storage, and channel paths) among logical partitions. Contrast with *basic mode*.

loosely coupled. A multisystem structure that requires a low degree of interaction and cooperation between multiple z/OS images to process a workload. See also *tightly coupled*.

LP. See *logical partition*.

LPAR. See *logically partitioned (LPAR) mode*.

LU. See *logical unit*.

M

m-image. The number (m) of z/OS images in a sysplex. See also *n-way*.

main storage. A logical entity that represents the program addressable portion of central storage. All user programs are executed in main storage. See also *central storage*.

main storage database (MSDB). A root-segment database, residing in main storage, that can be accessed on a field level.

mainframe (S/390 CPC). A large computer, in particular one to which other computers can be connected so that they can share facilities the S/390 CPC provides; for example, an S/390 computing system to which personal computers are attached so that they can upload and download programs and data.

maintenance point. A CICSPlex SM address space (CMAS) that is responsible for maintaining CICSPlex SM definitions in its data repository and distributing them to other CMASs involved in the management of a CICSPlex.

managed address space (MAS). A CICS system that is being managed by CICSPlex SM. See also *local MAS* and *remote MAS*.

MAS. Managed address space.

MAS agent. A CICSPlex SM component that acts within a CICS system to provide monitoring and data collection for the CICSPlex SM address space (CMAS). The level of service provided by a MAS agent depends on the level of CICS the system is running under and whether it is a local or remote MAS. See also *CICSPlex SM address space (CMAS), local MAS,* and *remote MAS*.

massively parallel. Refers to thousands of processors in a parallel arrangement.

mega-microsecond. A carry out of bit 32 of the TOD clock occurs every 1.048576. This interval is sometimes called a "mega-microsecond". This carry signal is used to start one clock in synchronism with another, as part of the process of setting the clocks. See also *time-of-day clock*.

member. A specific function (one or more modules or routines) of a multisystem application that is defined to XCF and assigned to a group by the multisystem application. A member resides on one system in the sysplex and can use XCF services to communicate (send and receive data) with other members of the same group. See *XCF group*, and *multisystem application*.

memory. Program-addressable storage from which instructions and other data can be loaded

directly into registers for subsequent execution or processing. Synonymous with *main storage*.

microcode. One or more microinstructions. A code, representing the instructions of an instruction set, that is implemented in a part of storage that is not program-addressable. To design, write, and test one or more microinstructions.

microprocessor. A processor implemented on one or a small number of chips.

migration. Installing a new version or release of a program when an earlier version or release is already in place.

mixed complex. A global resource serialization complex in which one or more of the systems in the global resource serialization complex are not part of a multisystem sysplex.

monitoring environment. A record of execution delay information about work requests kept by the workload management services. A monitoring environment is made up of one or more performance blocks. See also *performance block*.

monoplex. A one system sysplex with sysplex couple data sets that XCF prevents any other system from joining. See also *multisystem sysplex*.

MP. Multiprocessor.

MSDB. See *main storage database.*

MSU. Millions of Service Units. The unit used in IBM PSLC and WLC pricing as an estimate of CPC capacity within a processor range.

multifiber cable. An optical cable that contains two or more fibers. See also *jumper cable, optical cable assembly,* and *trunk cable*.

multimode optical fiber. A graded-index or step-index optical fiber that allows more than one bound mode to propagate. Contrast with *single-mode optical fiber*.

Multi-Node Persistent Session (MNPS). MNPS extends persistent sessions capability across multiple CPCs connected through the CF. MNPS provides for the recovery of VTAM, z/OS, hardware or application failures by restarting the application on another host in the Parallel Sysplex without requiring users to re-logon.

Multiple Systems Coupling (MSC). An IMS facility that permits multiple IMS subsystems to communicate with each other.

multiprocessing. The simultaneous execution of two or more computer programs or sequences of instructions. See also *parallel processing*.

multiprocessor (MP). A CPC that can be physically partitioned to form two operating processor complexes.

multisystem application. An application program that has various functions distributed across z/OS images in a multisystem environment.

Examples of multisystem applications are:

- CICS
- Global resource serialization (GRS)
- Resource Measurement Facility (RMF)
- z/OS Security Server (RACF)
- Workload manager (WLM)

See *XCF group*.

multisystem environment. An environment in which two or more z/OS images reside in one or more processors, and programs on one image can communicate with programs on the other images.

multisystem sysplex. A sysplex in which two or more z/OS images are allowed to be initialized as part of the sysplex. See also *single-system sysplex*.

N

named counter server (CICS). CICS provides a facility for generating unique sequence numbers for use by applications in a Parallel Sysplex environment (for example, to allocate a unique number for orders or invoices). This facility is provided by a named counter server, which maintains each sequence of numbers as a named counter. Each time a sequence number is assigned, the corresponding named counter is incremented automatically so that the next request gets the next number in sequence. This facility uses a CF list structure to hold the information.

NCP. Network Control Program (IBM-licensed program). Its full name is Advanced Communications Function for the Network Control Program. Synonymous with *ACF/NCP*. Network control program is the general term.

network. A configuration of data processing devices and software connected for information interchange. See also *network architecture* and *network control program (NCP)*.

network architecture. The logical structure and operating principles of a computer network.

node. In SNA, an endpoint of a link or junction common to two or more links in a network. Nodes can be distributed to S/390 CPC CPs, communication controllers, cluster controllers, or terminals. Nodes can vary in routing and other functional capabilities.

n-way. The number (n) of CPs in a CPC. For example, a 6-way CPC contains six CPs.

O

OLDS. *See* online log data set.

online log data set (OLDS). A data set on direct access storage that contains the log records written by an online IMS system.

open system. A system with specified standards that therefore can be readily connected to other systems that comply with the same standards. A data communications system that conforms to the standards and protocols defined by open systems interconnection (OSI). Synonym for *node*.

Operational Single Image. Multiple operating system images being managed as a single entity. This may be a basic sysplex, standard Parallel Sysplex, or extended Parallel Sysplex.

optical cable. A fiber, multiple fibers, or a fiber bundle in a structure built to meet optical, mechanical, and environmental specifications. See also *jumper cable* and *trunk cable*.

optical receiver. Hardware that converts an optical signal to an electrical logic signal. Contrast with *optical transmitter*.

optical repeater. In an optical fiber communication system, an opto-electronic device or module that receives a signal, amplifies it (or, for a digital signal, reshapes, retimes, or otherwise reconstructs it), and retransmits it.

optical transmitter. Hardware that converts an electrical logic signal to an optical signal. Contrast with *optical receiver*.

z/OS image. A single occurrence of the z/OS operating system that has the ability to process work.

z/OS system. An z/OS image together with its associated hardware, which collectively are often referred to simply as a system, or z/OS system.

P

P-lock. There are times when a P-lock must be obtained on a page to preserve physical consistency of the data between members. These locks are known as page P-locks. Page P-locks are used, for example, when two subsystems attempt to update the same page of data and row locking is in effect. They are also used for GBP-dependent space map pages and GBP-dependent leaf pages for type 2 indexes, regardless of locking level. IRLM P-locks apply to both DB2 and IMS DB data sharing.

Page set P-locks are used to track inter-DB2 read-write interest, thereby determining when a page set has to become GBP-dependent. When access is required to a page set or partition through a member in a data sharing group, a page set P-lock is taken. This lock is always propagated to the lock table on the CF and is owned by the member. No matter how many times the resource is accessed through the member, there will always be only one page set P-lock for that resource for a particular member. This lock will have different modes depending on the level (read or write) of interest the member has in the resource. See also *global locking*.

parallel. Pertaining to a process in which all events occur within the same interval of time, each handled by a separate but similar functional unit; for example, the parallel transmission of the bits of a computer word along the lines of an internal bus. Pertaining to the concurrent or simultaneous operation of two or more devices or to concurrent performance of two or more activities in a single device. Pertaining to the concurrent or simultaneous occurrence of two or more related activities in multiple devices or channels. Pertaining to the simultaneity of two or more processes. Pertaining to the simultaneous processing of the individual parts of a whole, such as the bits of a character and the characters of a word, using separate facilities for the various parts. Contrast with *serial*.

parallel processing. The simultaneous processing of units of work by many servers. The units of work can be either transactions or subdivisions of large units of work (batch).

Parallel Sysplex. A Parallel Sysplex is a sysplex with one or more CFs. See also *Base Sysplex*, *sysplex*, *extended Parallel Sysplex*, and *standard Parallel Sysplex*.

partition. An area of storage on a fixed disk that contains a particular operating system or logical drives where data and programs can be stored.

partitionable CPC. A CPC can be divided into two independent CPCs. See also *physical partition, single-image mode, MP*, and *side*.

partitioned data set (PDS). A data set in DASD storage that is divided into partitions, called *members*, each of which can contain a program, part of a program, or data.

performance. For a storage subsystem, a measurement of effective data processing speed against the amount of resource that is consumed by a complex. Performance is largely determined by throughput, response time, and system availability. See also *performance administration*, *performance block*, *performance management*, and *performance period*.

performance administration. The process of defining and adjusting workload management goals and resource groups based on installation business objectives.

performance block. A piece of storage containing a workload management's record of execution delay information about work requests.

performance management. The process workload management uses to decide how to match resources to work according to performance goals and processing capacity.

performance period. A service goal and importance level assigned to a service class for a specific duration. You define performance periods for work that has variable resource requirements.

persistent connection. A connection to a CF structure with a connection disposition of KEEP. z/OS maintains information about the connection so that when the connection terminates abnormally from a CF structure, z/OS places the connection in a failed-persistent state, and the connection can attempt to reconnect to the structure.

persistent session. In the NetView program, a network management session that remains active even though there is no activity on the session for a specified period of time. A LU-LU session that VTAM retains after the failure of a VTAM application program. Following the application program's recovery, the application program either restores or terminates the session.

persistent structure. A structure allocated in the CF with a structure disposition of KEEP. A persistent structure keeps its data intact across system or sysplex outages, regardless of whether any users are connected to the structure.

physical partition. Part of a CPC that operates as a CPC in its own right, with its own copy of the operating system.

physically partitioned (PP) configuration. A system configuration that allows the processor controller to use both CPC sides as individual CPCs. The A-side of the processor controls side 0, and the B-side controls side 1. Contrast with *single-image (SI) mode*.

policy. A set of installation-defined rules for managing sysplex resources. The XCF PR/SM policy and sysplex failure management policy are examples of policies.

power-on reset. The state of the machine after a logical power-on before the control program is IPLed.

preference list. An installation list of CFs, in priority order, that indicates where z/OS is to allocate a structure.

processing unit (PU). The part of the system that does the processing, and contains processor storage. On a 9672 CPC, the PU may be assigned as either a CP, SAP, ICF, or act as a spare PU.

processor. A processing unit, capable of executing instructions when combined with main storage and channels. See also *processor complex*, *processor controller*, and *processor controller element (PCE* and *CPC)*.

processor complex. A physical collection of hardware that includes main storage, one or more processors, and channels.

processor controller. Hardware that provides support and diagnostic functions for the CPs.

processor controller element (PCE). Hardware that provides support and diagnostic functions for the processor unit. The processor controller communicates with the processor unit through the logic service adapter and the logic support stations, and with the power supplies through the power thermal controller. It includes the primary support processor (PSP), the initial power controller (IPC), the input/output support processor (IOSP), and the control panel assembly.

Processor Resource/Systems Manager™ (PR/SM). A function that allows the processor unit to operate several system control programs simultaneously in LPAR mode. It provides for logical partitioning of the real machine and support of multiple preferred guests. See also *LPAR.*

program specification block (PSB). The control block in IMS that describes databases and logical message destinations used by an application program.

PR/SM. See *Processor Resource/Systems Manager*.

PSB. See *program specification block*.

public network. A communication common carrier network that provides data communication services over switched, non-switched, or packet-switching lines.

R

RAS. Reliability, availability, and serviceability.

receiver. In fiber optics, see *optical receiver*.

reconfiguration. A change made to a given configuration in a computer system; for example, isolating and bypassing a defective functional unit or connecting two functional units by an alternative path. Reconfiguration is effected automatically or manually and can be used to maintain system integrity. The process of placing a processor unit, main storage, and channels offline for maintenance, and adding or removing components.

Record Level Sharing (RLS). RLS is an access mode for VSAM data sets supported by DFSMS 1.3 and later releases. RLS enables VSAM data to be shared, with full update capability, between many applications running in many CICS regions across the Parallel Sysplex.

recovery. To maintain or regain system operation after a failure occurs. Generally, to recover from a failure is to identify the failed hardware, to de-configure the failed hardware, and to continue or restart processing.

recovery control (RECON) data sets. Data sets in which Database Recovery Control stores information about logging activity and events that might affect the recovery of databases.

relative processor power (RPP). A unit used to express processor capacity. RPP is a measured average of well defined workload profiles ITR-ratios. ITR (Internal Throughput Rate) is measured in transactions/CPU second. LSPR (Large Systems Performance Reference) measurements predict RPP values for processors running certain releases of operating systems.

remote MAS. A managed address space (MAS) that uses MRO or LU6.2 to communicate with the CICSPlex SM address space (CMAS) that controls it. A remote MAS may or may not reside in the same z/OS image as the CMAS that controls it.

remote operations. The ability to perform operations tasks from a remote location.

remote site recovery. The ability to continue or resume processing of the critical workload from a remote site.

report class. A group of work for which reporting information is collected separately. For example, you can have a WLM report class for information combining two different service classes, or a report class for information about a single transaction.

request. A service primitive issued by a service user to call a function supported by the service provider.

request for price quotation (RPQ). A custom feature for a product.

resource group. An amount of processing capacity across one or more z/OS images, assigned to one or more WLM service classes.

Resource Sharing. S/390 Resource Sharing provides the following functionality:

- XCF Signalling: Providing multisystem signaling with reduced cost/management
- GRS Star: Multisystem resource serialization for increased performance, recoverability, and scalability
- JES Checkpointing: Multisystem checkpointing for increased simplicity and reduced cost
- Shared Tape: Multisystem tape sharing for reduced duplication cost
- Merged Operations Log: Multisystem log for single system image/management
- Merged LOGREC: Multisystem log for single system image/management
- Shared Catalog: Multisystem shared master catalogs/user catalogs for increased performance/simplicity and reduced cost

response time. The amount of time it takes after a user presses the enter key at the terminal until the reply appears at the terminal.

routing. The assignment of the path by which a message will reach its destination.

RPP. See *relative processor power*.

RPQ. See *request for price quotation*.

S

secondary host promotion. Secondary host promotion allows one DFSMShsm system to automatically assume the unique functions of another DFSMShsm system that has failed.

selector channel. An I/O channel that operates with only one I/O device at a time. Once the I/O device is selected, a complete record is transferred one byte at a time. Contrast with *block multiplexer channel.*

serial. Pertaining to a process in which all events occur one after the other; for example, serial transmission of the bits of a character according to V24 CCITT protocol. Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel. Pertaining to the sequential processing of the individual parts of a whole, such as the bits of a character or the characters of a word, using the same facilities for successive parts. Contrast with *parallel*.

serialized list structure. A CF list structure with a lock table containing an array of exclusive locks whose purpose and scope are application-defined. Applications can use the lock table to serialize on parts of the list structure, or resources outside the list structure.

server. A device, program, or code module, on for example, a network dedicated to a specific function.

server address space. Any address space that helps process work requests.

service administration application. The online ISPF application used by the service administrator to specify the workload management service definition.

service class. A subset of a workload having the same service goals or performance objectives, resource requirements, or availability requirements. For workload management, you assign a service goal and, optionally, a resource group to a service class.

service console. A logical device used by service representatives to maintain the processor unit and to isolate failing field replaceable units. The service console can be assigned to any of the physical displays attached to the input/output support processor.

service definition. An explicit definition of the workloads and processing capacity in an installation. A service definition includes workloads, service classes, systems, resource groups, service policies, and classification rules.

service definition coefficient. A value that specifies which type of resource consumption should be emphasized in the calculation of service rate. The types of resource consumption are CPU, IOC, MSO, and SRB.

service policy. A named set of performance goals and, optionally, processing capacity boundaries that workload management uses as a guideline to match resources to work. See also *active service policy*.

service request block (SRB) service units. A measure of the SRB execution time for both local and global SRBs, multiplied by an SRM constant that is CPU model dependent.

service unit. The amount of service consumed by a work request as calculated by service definition coefficients and CPU, SRB, I/O, and storage service units.

session. A connection between two application programs that allows them to communicate. In SNA, a logical connection between two network addressable units that can be activated, tailored to provide various protocols, and deactivated as requested. The data transport connection resulting from a call or link between two devices. The period of time during which a user of a node can communicate with an interactive system; usually, it is the elapsed time between logon and logoff. In network architecture, an association of facilities necessary for establishing, maintaining, and releasing connections for communication between stations.

shared. Pertaining to the availability of a resource to more than one use at the same time.

shared lock. A lock that prevents concurrently executing application processes from changing, but not from reading, data. Contrast with *exclusive lock.*

side. A part of a partitionable PC that can run as a physical partition and is typically referred to as the A-side or the B-side.

single-image (SI) mode. A mode of operation for a multiprocessor (MP) system that allows it to function as one CPC. By definition, a uniprocessor (UP) operates in single-image mode. Contrast with *physically partitioned (PP) configuration*.

single-mode optical fiber. An optical fiber in which only the lowest-order bound mode (which can consist of a pair of orthogonally polarized fields) can propagate at the wavelength of interest. Contrast with *multimode optical fiber*.

single-z/OS environment. An environment that supports one z/OS image. See also *z/OS image*.

single point of control. The characteristic a sysplex displays when you can accomplish a given set of tasks from a single workstation, even if you need multiple IBM and vendor products to accomplish that particular set of tasks.

single point of failure. An essential resource for which there is no backup.

single GRSplex serialization. Single GRSplex serialization allows several HSMplexes, within a single GRSplex, to operate without interfering with any other HSMplex.

single-system image. The characteristic a product displays when multiple images of the product can be viewed and managed as one image.

single-system sysplex. A sysplex in which only one z/OS system is allowed to be initialized as part of the sysplex. In a single-system sysplex, XCF provides XCF services on the system but does not provide signalling services between z/OS systems. See also *multisystem complex* and *XCF-local mode.*

SMS communication data set (COMMDS). The primary means of communication among systems governed by a single SMS configuration. The SMS communication data set (COMMDS) is a VSAM linear data set that contains the current utilization statistics for each system-managed volume. SMS uses these statistics to help balance space usage among systems.

SMS configuration. The SMS definitions and routines that the SMS subsystem uses to manage storage.

SMS system group. All systems in a sysplex that share the same SMS configuration and communications data sets, minus any systems in the sysplex that are defined individually in the SMS configuration.

SSP. See *system support programs*.

standard. Something established by authority, custom, or general consent as a model or example.

standard Parallel Sysplex. A non-data sharing Parallel Sysplex.

STI. Self-Timed Interconnect.

storage. A unit into which recorded data can be entered, in which it can be retained and processed, and from which it can be retrieved.

storage management subsystem (SMS). An operating environment that helps automate and centralize the management of storage. To manage storage, SMS provides the storage administrator with control over data class, storage class, management class, storage group, and ACS routine definitions.

structure. A construct used to map and manage storage in a CF. See *cache structure, list structure*, and *lock structure.*

subarea. A portion of the SNA network consisting of a subarea node, any attached peripheral nodes, and their associated resources. Within a subarea node, all network addressable units, links, and adjacent link stations (in attached peripheral or subarea nodes) that are addressable within the subarea share a common subarea address and have distinct element addresses.

subarea node. In SNA, a node that uses network addresses for routing and whose routing tables are therefore affected by changes in the configuration of the network. Subarea nodes can provide gateway function, and boundary function support for peripheral nodes. Type 4 and type 5 nodes are subarea nodes.

subsystem. A secondary or subordinate system, or programming support, that is usually capable of operating independently of or asynchronously with a controlling system.

support element. A hardware unit that provides communications, monitoring, and diagnostic functions to a central processor complex (CPC).

symmetry. The characteristic of a sysplex where all systems, or certain subsets of the systems, have the same hardware and software configurations and share the same resources.

synchronous. Pertaining to two or more processes that depend on the occurrences of a specific event, such as common timing signal. Occurring with a regular or predictable timing relationship.

sysplex. A set of z/OS systems communicating and cooperating with each other through certain multisystem hardware components and software services to process client workloads. There is a distinction between a Base Sysplex and a Parallel Sysplex. See also *z/OS system, Base Sysplex, enhanced sysplex*, and *Parallel Sysplex*.

sysplex couple data set. A couple data set that contains sysplex-wide data about systems, groups, and members that use XCF services. All z/OS systems in a sysplex must have connectivity to the sysplex couple data set. See also *couple data set*.

sysplex data sharing. The ability of multiple IMS subsystems to share data across multiple z/OS images. Sysplex data sharing differs from two-way data sharing in that the latter allows sharing across only two z/OS images.

sysplex failure management. The z/OS function that minimizes operator intervention after a failure occurs in the sysplex. The function uses installation-defined policies to ensure continued operations of work defined as most important to the installation.

sysplex management. The functions of XCF that control the initialization, customization, operation, and tuning of z/OS systems in a sysplex.

sysplex partitioning. The act of removing one or more systems from a sysplex.

sysplex query parallelism. Businesses have an increasing need to analyze large quantities of data, whether to validate a hypothesis or to discover new relationships between data. This information is often critical to business success, and it can be difficult to get the information in a timely manner. DB2 V4 and later releases lets you split and run a single query within a DB2 subsystem. With sysplex query parallelism, DB2 V5 and later releases extends parallel processing to allow a single query to use all the CPC capacity of a data sharing group.

Sysplex query parallelism is when members of a data sharing group process a single query. DB2 determines an optimal degree of parallelism based on estimated I/O and processing costs. Different DB2 members processes different ranges of the data. Applications that are primarily read or and are processor-intensive or I/O-intensive can benefit from sysplex query parallelism. A query can split into multiple parallel tasks that can run in parallel across all images (up to 32) in a Sysplex. It can run

in parallel on up to 344 CPs within a Parallel Sysplex of 32 systems with 12 CPs each.

sysplex sockets. Socket applications are written generally to communicate with a partner on any platform. This means that the improved performance and scalability Parallel Sysplex is not exploited, unless some application-specific protocol is used; this is not always possible.

The sysplex sockets function provides a standard way to discover information about the connected partner that can then be used to make decisions that can exploit the value of the Parallel Sysplex where applicable.

Sysplex Timer. An IBM unit that synchronizes the time-of-day (TOD) clocks in multiple processors or processor sides. External Time Reference (ETR) is the z/OS generic name for the IBM Sysplex Timer (9037).

system. In data processing, a collection of people, machines, and methods organized to accomplish a set of specific functions.

system configuration. A process that specifies the devices and programs that form a particular data processing system.

system control element (SCE). The hardware that handles the transfer of data and control information associated with storage requests between the elements of the processor unit.

System Support Programs (SSP). An IBM-licensed program, made up of a collection of utilities and small programs, that supports the operation of the NCP.

systems management. The process of monitoring, coordinating, and controlling resources within systems.

S/390 microprocessor cluster. A configuration that consists of CPCs and may have one or more CFs.

S/390 Partners in Development. Membership in S/390 Partners in Development is open to companies and organizations developing or planning to develop commercially marketed software executing in an IBM S/390 environment under z/OS, VM, or VSE operating systems.

Offerings include low-cost application development and porting platforms, answer to technical questions, and information about IBM trends, directions, and latest technology.

- Special offers on development machines and software
- Bulletin-board Q&A support - Free
- Fee-based technical support
- Choice of S/390 platform library (CD-ROM) - Free
- S/390 Rainbow collection CD-ROM - Free
- Porting and technology workshops
- Remote access to IBM systems for application porting and development
- Solution Partnership Centers for test driving IBM Technologies
- Access to early IBM code
- z/OS, VM, and VSE technical disclosure meetings
- Information Delivery - Members get funneled pertinent S/390 information

T

takeover. The process by which the failing active subsystem is released from its extended recovery facility (XRF) sessions with terminal users and replaced by an alternate subsystem.

TCP/IP. Transmission control protocol/Internet protocol. A public domain networking protocol with standards maintained by the U.S. Department of Defense to allow unlike vendor systems to communicate.

Telnet. U.S. Department of Defense's virtual terminal protocol, based on TCP/IP.

throughput. A measure of the amount of work performed by a computer system over a given period of time, for example, number of jobs per day. A measure of the amount of information transmitted over a network in a given period of time.

tightly coupled. Multiple CPs that share storage and are controlled by a single copy of z/OS. See also *loosely coupled* and *tightly coupled multiprocessor*.

tightly coupled multiprocessor. Any CPC with multiple CPs.

time-of-day (TOD) clock. A 64-bit unsigned binary counter with a period of approximately 143 years. It is incremented so that 1 is added into bit position 51 every microsecond. The TOD clock runs regardless of whether the processing unit is in a running, wait, or stopped state.

time-to-live (TTL). In the context of a TCP/IP DNS nameserver, the time-to-live is the time that a DNS nameserver will retain resource records in its cache for resources for which it is not the authoritative name server.

TOD. See time-of-day (TOD) clock.

Token-Ring. A network with a ring topology that passes tokens from one attaching device (node) to another. A node that is ready to send can capture a token and insert data for transmission.

transaction. In an SNA network, an exchange between two programs that usually involves a specific set of initial input data that causes the execution of a specific task or job. Examples of transactions include the entry of a client's deposit that results in the updating of the client's balance, and the transfer of a message to one or more destination points.

transmission control protocol/Internet protocol (TCP/IP). A public domain networking protocol with standards maintained by the U.S. Department of Defense to allow unlike vendor systems to communicate.

transmitter. In fiber optics, see *optical transmitter*.

trunk cable. In an ESCON environment, a cable consisting of multiple fiber pairs that do not directly attach to an active device. This cable usually exists between distribution panels and can be located within, or external to, a building. Contrast with *jumper cable*.

TSO. See *TSO/E*.

TSO/E. In z/OS, a time-sharing system accessed from a terminal that allows user access to z/OS system services and interactive facilities.

TTL. See *time-to-live*

tutorial. Online information presented in a teaching format.

type 2.1 node (T2.1 node). A node that can attach to an SNA network as a peripheral node using the same protocols as type 2.0 nodes. Type 2.1 nodes can be directly attached to one another using peer-to-peer protocols. See *end node*, *node*, and *subarea node*.

U

uniprocessor (UP). A CPC that contains one CP and is not partitionable.

universal time coordinate (UTC). UTC is the official replacement for (and is generally equivalent to) the better known "Greenwich Mean Time".

UP. See *uniprocessor (UP)*.

UTC. See *Universal Time Coordinate*.

V

validity vector. On a CPC, a bit string that is manipulated by cross-invalidate to present a user connected to a structure with the validity state of pages in its local cache.

velocity. A service goal naming the rate at which you expect work to be processed for a given service class or a measure of the acceptable processor and storage delays while work is running.

W

VTS. See *Virtual Tape Server*.

warm start. Synonymous with normal restart.

white space. CF storage set aside for rebuilding of structures from other CFs, in case of planned reconfiguration or failure.

workload. A group of work to be tracked, managed and reported as a unit. Also, a group of service classes.

workload management mode. The mode in which workload management manages system resources on an z/OS image. Mode can be either *compatibility mode* or *goal mode*.

work qualifier. An attribute of incoming work. Work qualifiers include subsystem type, subsystem instance, user ID, accounting information, transaction name, transaction class, source LU, NETID, and LU name.

write-ahead data set (WADS). A data set containing log records that reflect completed operations and are not yet written to an online log data set.

X

XCF. See *cross-system coupling facility*.

XCF couple data set. The name for the sysplex couple data set prior to MVS SP V5.1. See *sysplex couple data set*.

XCF dynamics. XCF Dynamics uses the Sysplex Sockets support that is introduced in z/OS V2R7 IP. Sysplex Sockets allows the stacks to communicate with each other and exchange information like VTAM CPNames, MVS SYSCLONE values, and IP addresses. Dynamic XCF definition is activated by coding the IPCONGFIG DYNAMICXCF parameter in TCPIP.PROFILE.

XCF group. A group is the set of related members defined to XCF by a multisystem application in which members of the group can communicate (send and receive data) between z/OS systems with other members of the same group. A group can span one or more of the systems in a sysplex and represents a complete logical entity to XCF. See *Multisystem application*.

XCF-local mode. The state of a system in which XCF provides limited services on one system and does not provide signalling services between z/OS systems. See also *single-system sysplex*.

XCF PR/SM policy. In a multisystem sysplex on PR/SM, the actions that XCF takes when one z/OS system in the sysplex fails. This policy provides high availability for multisystem applications in the sysplex.

XES. See *cross-system extended services*.

XRF. See *Extended recovery facility*.

Y

Year 2000 test datesource facility. The Year 2000 test datesource facility allows you to define several LPs on a single CPC that can enter a sysplex with a time and date other than that of the production system. This eliminates the need for dedicating an entire CPC to do year 2000 testing in a multi-member sysplex.

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 307. Note that some of the documents referenced here may be available in softcopy only.

- *ABCs of z/OS System Programming Volume 5*, SG24-6985
- *Accessing DB2 for OS/390 Data from the World Wide Web*, SG24-5273
- *Achieving the Highest Levels of Parallel Sysplex Availability*, SG24-6061
- *Architecting High Availability Using WebSphere V6 on z/OS*, SG24-6850
- *Batch Processing in a Parallel Sysplex*, SG24-5329
- *CICSPlex SM Business Application Services: A New Solution to CICS Resource Management*, SG24-5267
- *CICS Transaction Server for OS/390: Version 1 Release 3 Implementation Guide*, SG24-5274
- *CICS Transaction Server for OS/390 Version 1 Release 3: Web Support and 3270 Bridge*, SG24-5480
- *CICS Transaction Server for OS/390: Web Interface and 3270 Bridge*, SG24-5243
- *CICS and VSAM Record Level Sharing: Implementation Guide*, SG24-4766
- *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765
- *CICS and VSAM Record Level Sharing: Recovery Considerations*, SG24-4768
- *Connecting IMS to the World Wide Web: A Practical Guide to IMS Connectivity*, SG24-2220
- *Continuous Availability S/390 Technology Guide*, SG24-2086
- *DB2 on MVS Platform: Data Sharing Recovery*, SG24-2218
- *DB2 UDB for OS/390 and Continuous Availability*, SG24-5486
- *DB2 UDB for z/OS: Design Guidelines for High Performance and Availability*, SG24-7134
- *DB2 UDB for z/OS Version 8 Performance Topics*, SG24-6465
- *Disaster Recovery with DB2 UDB for z/OS*, SG24-6370
- *e-business Cookbook for z/OS Volume I: Technology Introduction*, SG24-5664
- *e-business Cookbook for z/OS Volume II: Infrastructure*, SG24-5981
- *e-business Cookbook for z/OS Volume III: Java Development*, SG24-5980
- *Enhanced Catalog Sharing and Management*, SG24-5594
- *GDPS Family - An Introduction to Concepts and Capabilities*, SG24-6374
- *Hierarchical File System Usage Guide*, SG24-5482
- *IBM System z9 and @server zSeries Connectivity Handbook*, SG24-5444

- *IBM TotalStorage Enterprise Storage Server Implementing ESS Copy Services with IBM @server zSeries*, SG24-5680
- *IBM Web-to-Host Integration Solutions*, SG24-5237
- *IMS/ESA Data Sharing in a Parallel Sysplex*, SG24-4303
- *IMS/ESA Multiple Systems Coupling in a Parallel Sysplex*, SG24-4750
- *IMS/ESA Sysplex Data Sharing: An Implementation Case Study*, SG24-4831
- *IMS/ESA V6 Parallel Sysplex Migration Planning Guide for IMS TM and DBCTL*, SG24-5461
- *IMS Connectivity in an On Demand Environment: A Practical Guide to IMS Connectivity*, SG24-6794
- *IMS e-business Connect Using the IMS Connectors*, SG24-5427
- *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908
- *IMS in the Parallel Sysplex Volume II: Planning the IMSplex*, SG24-6928
- *IMS in the Parallel Sysplex Volume III IMSplex Implementation and Operations*, SG24-6929
- *IMS Version 9 Implementation Guide: A Technical Overview*, SG24-6398
- *JES3 in a Parallel Sysplex*, SG24-4776
- *Merging Systems into a Sysplex*, SG24-6818
- *MVS/ESA HCD and Dynamic I/O Reconfiguration Primer*, SG24-4037
- *OS/390 e-business Infrastructure: IBM HTTP Server V5.1 for OS/390*, SG24-5603
- *OS/390 MVS Multisystem Consoles Implementing MVS Sysplex Operations*, SG24-4626
- *OS/390 Parallel Sysplex Configuration, Volume 1: Overview*, SG24-5637
- *OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook*, SG24-5638
- *OS/390 Parallel Sysplex Configuration, Volume 3: Connectivity*, SG24-5639
- *OS/390 R4 Implementation*, SG24-2089
- *OS/390 Release 5 Implementation*, SG24-5151
- *Parallel Sysplex Application Considerations*, SG24-6523
- *Parallel Sysplex - Managing Software for Availability*, SG24-5451
- *Patterns: Connecting Self-Service Applications to the Enterprise*, SG24-6572
- *Patterns: Direct Connections for Intra- and Inter-enterprise*, SG24-6933
- *Patterns: Service-Oriented Architecture and Web Services*, SG24-6303
- *Patterns: Self-Service Application Solutions Using WebSphere for z/OS V5*, SG24-7092
- *Patterns on z/OS: Connecting Self-Service Applications to the Enterprise*, SG24-6827
- *A Performance Study of Web Access to CICS*, SG24-5748
- *Planning for CICS Continuous Availability in a MVS/ESA Environment*, SG24-4593
- *Planning for IBM Remote Copy*, SG24-2595
- *Revealed! Architecting e-business Access to CICS*, SG24-5466
- *Revealed! CICS Transaction Gateway with More CICS Clients Unmasked*, SG24-5277
- *S/390 Parallel Sysplex: Resource Sharing*, SG24-5666
- *Securing Web Access to CICS*, SG24-5756

- *SNA in a Parallel Sysplex Environment*, SG24-2113
- *System/390 MVS Parallel Sysplex Continuous Availability Presentation Guide*, SG24-4502
- *System/390 MVS Parallel Sysplex Continuous Availability SE Guide*, SG24-4503
- *System/390 MVS Parallel Sysplex Migration Paths*, SG24-2502
- *System/390 Parallel Sysplex Performance*, SG24-4356
- *Systems Programmer's Guide to Resource Recovery Services (RRS)*, SG24-6980
- *System Programmer's Guide to: Workload Manager*, SG24-6472
- *Systems Programmer's Guide to: z/OS System Logger*, SG24-6898
- *Using VTAM Generic Resources with IMS*, SG24-5487
- *WOW! DRDA Supports TCP/IP: DB2 Server for OS/390 and DB2 Universal Database*, SG24-2212
- *z/OS Intelligent Resource Director*, SG24-5952

Other publications

These publications are also relevant as further information sources:

- *BatchPipes OS/390 V2R1 Users Guide and Reference,* SA22-7458
- *CICS for OS/390 and Parallel Sysplex,* GC33-1180
- *CICS Transaction Server for OS/390 V1R2 Release Guide,* GC33-1570
- *CICS Transaction Server for OS/390 V1R3 CICS Intercommunication Guide,* SC33-1695
- *CICS Transaction Server for OS/390 V1R3 Planning for Installation,* GC33-1789
- *CICS Transaction Server for OS/390 V1.3 CICSPlex SM Concepts and Planning,* GC33-0786
- *CICS Transaction Server for OS/390 V1.3 CICS Internet Guide,* SC34-5445
- *CICS Transaction Server for OS/390 V1.3 Migration Guide,* GC34-5353
- *CICS Transaction Server for z/OS V3.1 Installation Guide,* GC34-6426
- *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager,* GG24-4286
- *Coupling Facility Configuration Options,* GF22-5042
- *DB2 for OS/390 V5 Release Guide,* SC26-8965
- *DB2 UDB for OS/390 V6 Data Sharing: Planning and Administration,* SC26-9007
- *DB2 UDB for OS/390 V6 Release Planning Guide,* SC26-9013
- *DB2 UDB for z/OS V8 Administration Guide,* SC18-7413
- *DB2 UDB for z/OS V8 Data Sharing: Planning and Administration,* SC18-7417
- *DB2 UDB for z/OS V8 Release Planning Guide,* SC18-7425
- *DB2 UDB for z/OS V8 What's New,* GC18-7428
- *DB2 Universal Database™ Server for OS/390 V7 What's New?,* GC26-9017
- *DB2 Universal Database™ Server for z/OS and OS/390 V7 What's New?,* GC26-9946
- *DFSMS/MVS V1R5 Planning for Installation,* SC26-4919
- *DFSMS/MVS V1R5 Using Data Sets,* SC26-4922

- *IMS/ESA V5 Administration Guide: System,* SC26-8013
- *IMS/ESA V6 Administration Guide: System,* SC26-8730
- *IMS V9 Release Planning Guide,* GC17-7831
- *OS/390 Parallel Sysplex Recovery,* GA22-7286
- *OS/390 Parallel Sysplex Test Report,* GC28-1963
- *OS/390 V2R10.0 MVS Planning: Global Resource Serialization,* GC28-1759
- *OS/390 V2R10.0 MVS Setting Up a Sysplex,* GC28-1779
- *OS/390 V2R5.0 Parallel Sysplex Systems Management,* GC28-1861
- *OS/390 V2R8.0 Parallel Sysplex Hardware and Software Migration,* GC28-1862
- *OS/390 V2R9.0 Parallel Sysplex Application Migration,* GC28-1863
- *System-Managed CF Structure Duplexing*, GM13-0103
- *System-Managed CF Structure Duplexing Implementation Summary,* GM13-0540
- *z/OS HTTP Server Planning, Installing and Using, V1R7,* SC34-4826
- *z/OS MVS Initialization and Tuning Reference,* SA22-7592
- *z/OS MVS Programming: Sysplex Services Guide,* SA22-7817
- *z/OS MVS Programming: Workload Management Services,* SA22-7619
- *z/OS MVS Setting Up a Sysplex*, SA22-7625
- *z/OS Parallel Sysplex Overview: An Introduction to Data Sharing & Parallelism*, SA22-7661
- *z/OS Resource Measurement Facility Report Analysis,* SC33-7991
- *z/OS V1R1.0 Parallel Sysplex Application Migration,* SA22-7662
- *z/OS V1R5.0 Parallel Sysplex Test Report,* SA22-7663
- *z/OS V1R2 Communications Server: APPC Application Suite Administration,* SC31-8835
- *z/OS V1R7.0 Communications Server: IP Configuration Guide,* SC31-8775
- *z/OS V1R7.0 DFSMS OAM Planning, Installation, and Storage Administration Guide for Object Support,* SC35-0426
- *z/OS V1R7.0 DFSMShsm Implementation and Customization Guide,* SC35-0418
- *z/OS V1R7.0 Hot Topics Newsletter,* GA22-7501
- *z/OS V1R7.0 JES2 Initialization and Tuning Guide,* SA22-7532
- *z/OS V1R7.0 JES3 Initialization and Tuning Guide,* SA22-7549
- *z/OS V1R7.0 MVS Planning: Global Resource Serialization,* SA22-7600
- *z/OS V1R7.0 MVS Planning: Workload Management,* SA22-7602
- *z/OS V1R7.0 MVS Programming: Resource Recovery,* SA22-7616
- *z/OS V1R7.0 MVS Setting Up a Sysplex*, SA22-7625
- *z/OS V1R7.0 MVS System Commands,* SA22-7627
- *z/OS V1R7.0 Security Server RACF Security Administrator's Guide,* SA22-7683
- *z/OS V1R7.0 UNIX System Services Planning,* GA22-7800
- *z/OS and z/OS.e V1R7.0 Planning for Installation,* GA22-7504

Online resources

These Web sites and URLs are also relevant as further information sources:

- About RMF reports: Refer to *z/OS Resource Measurement Facility Report Analysis,* SC33-7991 on the Web at:

  http://publibz.boulder.ibm.com/epubs/pdf/erbzra41.pdf

- Achieving Near Continuous Availability:

  http://www.ibm.com/servers/eserver/zseries/pso/

- Applications by S/390 Technology:

  http://www.ibm.com/systems/z/solutions/isv/

- Best way of leveraging your existing CICS applications and knowledge in an e-business world is available on the Internet at:

  http://www.redbooks.ibm.com/abstracts/sg245466.html?Open

- CF Sizer tool:

  http://www.s390.ibm.com/cfsizer/
  http://www.ibm.com/servers/eserver/zseries/cfsizer/index.html

- CFLevel info:

  http://www.ibm.com/servers/eserver/zseries/pso/cftable.html

- CF Sizer:

  http://www.ibm.com/servers/eserver/zseries/cfsizer/
  http://www-1.ibm.com/servers/eserver/zseries/cfsizer/

- Coupling Facility Configuration, including links:

  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gf225042.pdf

- Dynamic CF Dispatching:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/TD102670

- Find the Availability Pages via the Parallel Sysplex home page, or directly from:

  http://www.ibm.com/servers/eserver/zseries/pso/availability.html

- For information about Adabas Cluster Services, refer to:

  http://www.softwareag.com/Corporate/products/adabas/add_ons/cluster.asp

- For APARs about dataset management in Shared HFS environment, refer to:

  http://www.ibm.com/servers/eserver/zseries/zos/unix/pdf/ow54824.pdf

- For more information about BRLM, refer to:

  http://www.ibm.com/servers/eserver/zseries/zos/unix/apps/brlm.html

- For information about Datacom, refer to:

  http://www3.ca.com/solutions/Solution.aspx?ID=2899

- For more information about WebSphere products, refer to:

  http://publib.boulder.ibm.com/infocenter/wasinfo/v5r1/index.jsp
  http://publib.boulder.ibm.com/infocenter/wasinfo/v6r0/index.jsp

- For the most current HTTP server documentation and information updates, refer to:

  http://www.ibm.com/software/webservers/httpservers/doc53.html

- GDPS home page:

  http://www.ibm.com/systems/z/gdps/

- How VM can assist in Parallel Sysplex testing:

  http://www.vm.ibm.com/os390/
- IBM Global Services High Availability Services, available on the Web:

  http://www.ibm.com/services/tsm/Implementing
- IBM Publications:

  http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi
- IBM Techdocs:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs
- IBM eServer zSeries sysplex software pricing:

  http://www.ibm.com/servers/eserver/zseries/swprice/sysplex
- IBM eServer zSeries:

  http://www.ibm.com/systems/z/
- Improve Your Availability with Sysplex Failure Management, found at:

  http://www.s390.ibm.com/products/pso/availability.html
- IMS:

  http://www.ibm.com/software/data/ims/imswwwc.html#9
- *Leveraging z/OS TCP/IP Dynamic VIPAs and Sysplex Distributor for higher availability*, found at:

  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130165.pdf
- Look at the special flash paper available on the TECHDOC Web site at:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/FLASH10285
- OS/390 Integration Test Site:

  http://www.s390.ibm.com/os390/support/os390tst
- Parallel Sysplex Education and Training:

  http://www.ibm.com/servers/eserver/zseries/pso/education.html
- Parallel Sysplex Configuration Assistant:

  http://www.ibm.com/servers/eserver/zseries/zos/wizards/parallel/plexv1r1/
- Patterns for e-business:

  http://www.ibm.com/developerWorks/patterns
- Primary checkpoint data set:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/W9748B
- Second Data Center Considerations:

  http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS1063
- Shared HFS function:

  http://www.ibm.com/servers/eserver/zseries/zos/bkserv/animations/ussanims.html
- *System-Managed CF Structure Duplexing*, GM13-0103:

  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130103.pdf
- *System-Managed CF Structure Duplexing Implementation Summary,* GM13-0540:

  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gm130540.pdf
- The Sizer Utility Program.doc:

  http://www-1.ibm.com/servers/eserver/zseries/cfsizer/altsize.html

- Use the self-assessment questionnaire available on Resource Link at:

  http://www.ibm.com/servers/resourcelink
- Various white papers, manuals, and Redbooks:

  http://www.ibm.com/software/data/ims/library.html
- White paper to obtain highest possible application availability:

  http://ibm.com/servers/eserver/zseries/library/whitepapers/pdf/availchk_parsys.pdf
- White paper, Value of Resource Sharing:

  http://www.ibm.com/servers/eserver/zseries/library/techpapers/pdf/gf225115.pdf
- XES application interface:

  http://publibz.boulder.ibm.com/cgi-bin/bookmgr_OS390/BOOKS/IEA2I630/CCONTENTS
- z/OS MVS Parallel Sysplex Test Report:

  http://www.ibm.com/servers/eserver/zseries/pso
- z/OS Integration Test:

  http://www-1.ibm.com/servers/eserver/zseries/zos/integtst/

How to get IBM Redbooks

You can search for, view, or download Redbooks, Redpapers, Hints and Tips, draft publications and Additional materials, as well as order hardcopy Redbooks or CD-ROMs, at this Web site:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Index

C

D

E

F

G

H

J

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

z/OS Parallel Sysplex Configuration Overview

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

z/OS Parallel Sysplex Configuration Overview

An update of the Parallel Sysplex Configuration Volume 1

High-level design concepts for Parallel Sysplex

The workloads in Parallel Sysplex

This IBM Redbook will provide you with the information you require to understand what is a Parallel Sysplex. With an understanding of the basics, it then goes on to describe how these components are used to enable the two fundamental capabilities of Parallel Sysplex: namely, the ability to concurrently update a database from two or more database managers (thereby removing single points of failure), and dynamic workload balancing.

The IBM Redbook then moves on to discuss how the Parallel Sysplex-exploiting products enable you to start delivering continuous application availability, a growing requirement or most IBM System z clients. Typical client workloads are discussed, and their exploitation of Parallel Sysplex is described in clear terms.

This IBM Redbook is an excellent starting point for those involved in designing and configuring a Parallel Sysplex. It should also be used by those that already have a Parallel Sysplex when further exploitation is being considered. The IBM Redbook refers throughout to other relevant publications that cover specific areas in more detail.

This IBM Redbook is an update to the first volume of a previous set of three Parallel Sysplex Configuration IBM Redbooks. Most of the information in the second and third volumes is either available elsewhere, or else it is no longer required.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6485-00 ISBN 073849562X