IBM

Exploiting Parallel Sysplex: A Real Customer Perspective

- Quantifiable benefits
- Actual implementation efforts
- Lessons learned in the real world

Frank Kyne
Matthias Bangert
Bernd Daubner
Gerhard Engelkes
Ruediger Gierich
Andreas Kiesslich
Juergen Klaus
Helmut Nimmerfall
Thomas Schlander
Friedhelm Stoehler
Leo Wilytsch

ibm.com/redbooks

Redbooks

International Technical Support Organization

Exploiting Parallel Sysplex: A Real Customer Perspective

October 2006

SG24-7108-00

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

First Edition (October 2006)

This edition applies to IBM CICS, DB2, and z/OS.

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 about 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:

@server®
@server®
AFP™
CICS®
CICSPlex®
DB2®
DFSMShsm™
ESCON®
FICON®
GDPS®
HiperSockets™
IBM®
ibm.com®
IMS™
iSeries™
MQSeries®
MVS™
NetView®
OS/390®
Parallel Sysplex®
PR/SM™
Redbooks™
RACF®
Redbooks (logo) ™
Redbooks™
RMF™
S/390®
Sysplex Timer®
System z™
System z9™
Tivoli®
VTAM®
WebSphere®
z9™
z/OS®
z/VM®
zSeries®

The following terms are trademarks of other companies:

EJB, Java, JDBC, JMX, JSP, JVM, J2EE, Solaris, StorageTek, Sun, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Outlook, 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.

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

Preface

IBM® System z™ is well known for its reliability, availability, and serviceability (RAS). So how can your enterprise obtain all this benefit, along with low total cost of ownership (TCO) and excellent centralized administration? And what other benefits can you realize by implementing a Parallel Sysplex® configuration? This IBM Redbook answers these questions by documenting the experiences of a real life client that undertook this process. Informatik Zentrum Frankfurt und München (IZB) completed a Parallel Sysplex implementation, and it shares its perspective with you in this detailed analysis.

IZB is a large banking service provider in Germany. Five years ago, it had two data centers with differing standards, no sysplexes, underpowered hardware, and an expensive, tape-based disaster recovery strategy. It lacked DASD or tape mirroring; it needed to customize systems for each client; and it could not aggregate CPUs, resulting in elevated software licensing costs. Today, by exploiting Parallel Sysplex features, IZB is achieving maximum service levels and financial value from its System z environment, and looking forward to growing its client base.

This publication provides step-by-step information about how IZB set up its System z environment. Covering the areas of processors, disk, networking, middleware, databases, peripheral output, and backup and recovery, the book was written to demonstrate how other installations can derive similar value from their System z environments.

The team that wrote this redbook

This IBM Redbook was produced by the team of IZB IT specialists responsible for the planning and implementation of the Parallel Sysplex exploiters discussed here.

Frank Kyne is a Senior Certified IT Specialist at the International Technical Support Organization, Poughkeepsie Center. He writes extensively and teaches IBM classes worldwide on all areas of Parallel Sysplex. He is also responsible for the GDPS® product documentation. Before joining the ITSO eight years ago, Frank worked in IBM Global Services in Ireland as an MVS™ Systems Programmer.

Matthias Bangert is a Certified IT Specialist at the zSeries® and System z9™ Field Technical Sales Support organization in Munich, Germany. His areas of expertise are disaster recovery and data center merges. He is a project leader for various client projects, as well as a regular speaker at client briefings and events. Before joining IBM in 1999 Matthias spent seven years as an MVS Systems Programmer, and also worked for six years as a System Engineer for storage.

Bernd Daubner is a System Architect for networking at IZB in Munich, Germany. He is responsible for z/Series Secure Way Communication Server implementation, as well as IP and SNA/APPN data center networking. Before joining IZB in 2001, Bernd spent 10 years working on the design and configuration of Cisco Routers and Cisco/Stratacom Wide Area Networks at AT&T Global Information Systems, later Lucent technologies Netcare.

Gerhard Engelkes is a DB2® Specialist working for IZB in Germany as a DB2 system programmer. He has 18 years of experience with DB2. Before joining IZB in 1998, he worked for 11 years in a large German bank as a DB2 applications DBA, and spent two years as a PL/I and IMS™ developer. His areas of expertise include DB2 performance, database

administration, and backup and recovery. For the last six years, Gerhard has also been responsible for IZB's WebSphere® MQ installation on z/OS®.

Ruediger Gierich is a Senior Systems Programmer who has worked for IZB since 1995. He is the Team Leader of IZB's systems programming team. Ruediger has 15 years of experience working with operating systems and mainframes.

Andreas Kiesslich is a Systems Programmer with IZB. After working as a DB2 and IMS Application Developer at the Barvarian State Bank, he joined IZB in 1993, where he was responsible for the DB2 and IMS system environment. Since 2003, Andreas has been responsible for WebSphere on z/OS.

Juergen Klaus is an IT Specialist who joined IZB in 1997. A Communications Engineer, he has been a member of the Hardware Planning team since 1999. Juergen's responsibilities include capacity planning on the 390 Systems, and hardware planning for backup.

Helmut Nimmerfall is a Systems Programmer at IZB. For the last eight years he has been responsible for the implementation and maintenance of transaction monitor CICSTS. Prior to joining the CICS® team, Helmut worked as an MVS system programmer for 11 years.

Thomas Schlander is an Adabas database administrator with IZB in Munich. He joined IZB in 1994, and is the Team Leader of the entire Adabas team at IZB. The team is responsible for maintaining all Software AG's products, such as Adabas, Natural, and ENTIRE. Prior to joining IZB, Thomas worked for the savings bank organization.

Friedhelm Stoehler is a System Architect at IZB in Munich. He holds a Master's degree in Computer Science, and joined IZB in 1995 as an MVS Systems Programmer. Friedhelm is responsible for the system architecture and design of the mainframe systems.

Leo Wylitch is an Automation Specialist who joined IZB in 1998. His team is responsible for automating and monitoring z/OS LPARs, and for implementing and maintaining all BMC and LeuTek Products.

Thanks to the following people for their contributions to this project:

Anton Müller
CIO Informatik Zentrum Bayern

Manfred Heckmeier
Manager - Department IT-Fabric., Informatik Zentrum Bayern

Ella Buslovich
Mike Ebbers
International Technical Support Organization, Poughkeepsie Center

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/or clients.

Your efforts will help increase product acceptance and customer 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. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Part 1

Overview and configuration

This part introduces the company Informatik Zentrum Frankfurt und München (IZB), and describes its migration history.

1

Introduction to IZB

This chapter introduces Informatik Zentrum Frankfurt und München (also known as IZB), an IT services provider and an IBM client, whose experiences are documented in this IBM Redbook. The chapter begins with a description of the environment that existed at IZB prior to starting the sysplex implementation project, then proceeds to an explanation of where IZB is today, and concludes with a glimpse into IZB plans for the future.

The chapter covers the following topics:

- Overview of IZB
- From 1999 until now
- The solution
- Description of the project
- Complications during project implementation
- The future

1.1 Overview of IZB

IZB is a leading IT provider in Germany. Its biggest client is the association of savings banks of Bavaria. In addition to providing IT services to the 80 savings banks located in Bavaria, IZB also serves Bayerischen Landesbank AG (BayernLB) and Landesbank Hessen Thuringia (HeLaBa).

Previously, the savings banks used the same z/OS applications for their business, that is, groups of savings banks ran in “groups” on one z/OS image. Three groups of savings banks, one in the south of Bavaria and two in the north, were formed, with each group running on a different CPC. This was a carryover from earlier times when sysplex was *not* implemented.

1.1.1 Description of IZB

IZB was founded in 1994 when BayernLB and IZB SOFT were merged. IZB SOFT was founded in 1994 by the Association of Savings Banks of Bavaria. BayernLB, the seventh largest bank in Germany, is owned in equal parts by the Association of Savings Banks of Bavaria and the Bavarian government. Since January 2001, the Hessen-Thuringen Landesbank has also held a share in IZB. While IZB SOFT handles application development for the savings banks, IZB offers services in the client/server, mainframe, and network sectors, ranging from consulting services to setting up operations, as well as after-sales care and maintenance of IT systems. In addition to its savings bank clients, IZB also offers these services to other clients.

In 1994, IZB ran two data centers: one in Munich and one in Nuremberg. These data centers were connected through a network, but DASD or tape mirroring was *not* done. While IZB is responsible for guaranteeing stable operation of all applications on various platforms, the development of applications is not a part of the IZB mission.

Among others, IZB’s current clients include the following organizations:

- The Association of Savings Banks of Bavaria
- The Bavarian State Bank (BayernLB)
- The State Bank Hessen and Thuringen (HeLaBa)
- Landesbauspar Kasse (LBS)
- The Transaction Bank (TXB)
- The Deka Bank

These clients were organized among the sysplexes as shown in Figure 1-1 on page 5. Note that throughout this IBM Redbook, we refer to these sysplexes as Customer A, B, C, and so on.

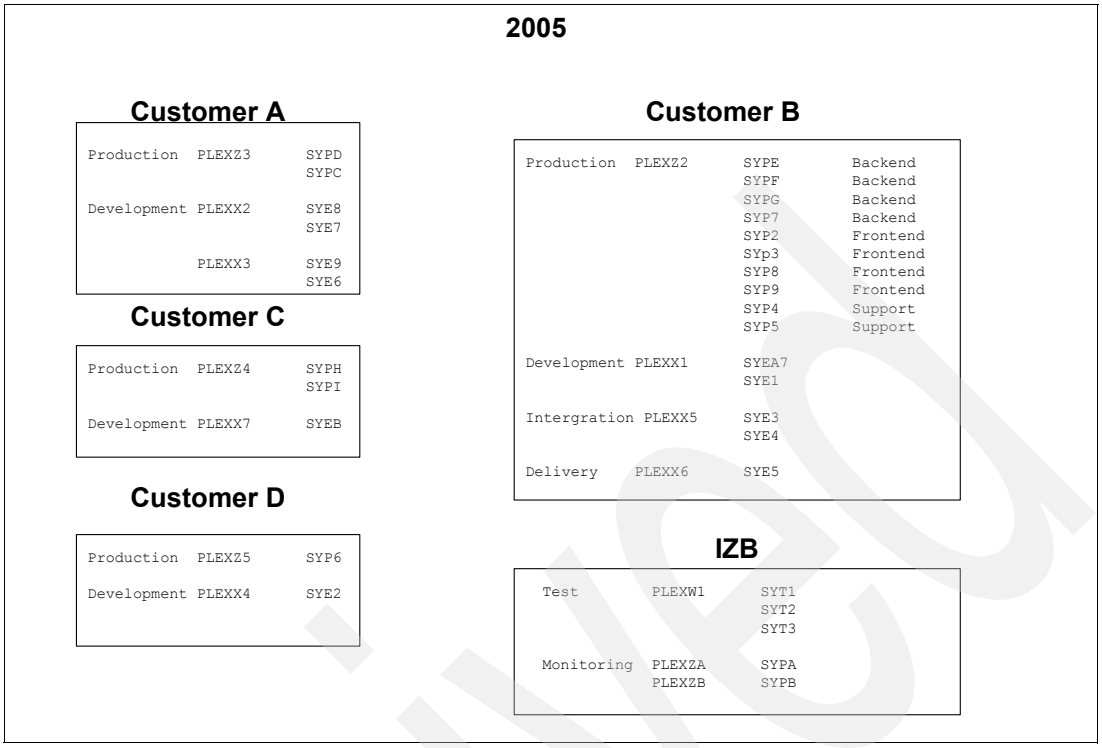


Figure 1-1 Sysplex structure

1.1.2 The IZB mission

IZB sees itself as a dynamic, well-positioned, IT service provider that is able to provide cost-effective services in the area of IT technology to its clients. Its essential capabilities include disaster recovery readiness (which is tested twice a year) and high availability. To reach its high availability goals, IZB has, over the years, exploited nearly all Parallel Sysplex features to the maximum extent.

IZB hopes to grow further in the future by adding new clients. New clients are each run in their own logical partitions (LPARs).

1.2 From 1999 until now

This IBM Redbook describes the project undertaken by IZB over the past seven years. In particular, the past five years is focused on, since the most significant changes occurred during that time frame. This includes the implementation of Parallel Sysplex, relocation of a data center, and the optimization of the installations.

1.2.1 Two data centers, 160 kilometers apart

In 1999 IZB ran two data centers that were 160 km apart, as shown in Figure 1-2 on page 6. These data centers were connected through telecom lines, and not with dark fiber cables. FICON®[1] technology was not available in the marketplace at that time to provide

[1] FICON is a high-speed input/output (I/O) interface for mainframe computer connections to storage devices. As part of the IBM S/390® server, FICON channels increase I/O capacity through the combination of a new architecture and faster physical link rates to make them up to eight times as efficient as ESCON®, or Enterprise System Connection, which is the previous IBM fibre optic channel standard.

Peer-to-Peer Remote Copy (PPRC) mirroring over 160 km. An asynchronous copy technology was not feasible because of IZB's policy of "no data loss".

Additionally, the two sites could not be connected in the same Parallel Sysplex due to Sypslex Timer distance limitations. Even a *BronzePlex* spanning the two sites was not possible with the technology available at that time. This meant that optimization from both a software cost perspective (Parallel Sysplex License Charge (PSLC) pricing) and a hardware perspective (virtualization technology in z/OS servers) had its limitations in that this was possible *within* the data centers, but not *between* the data centers.

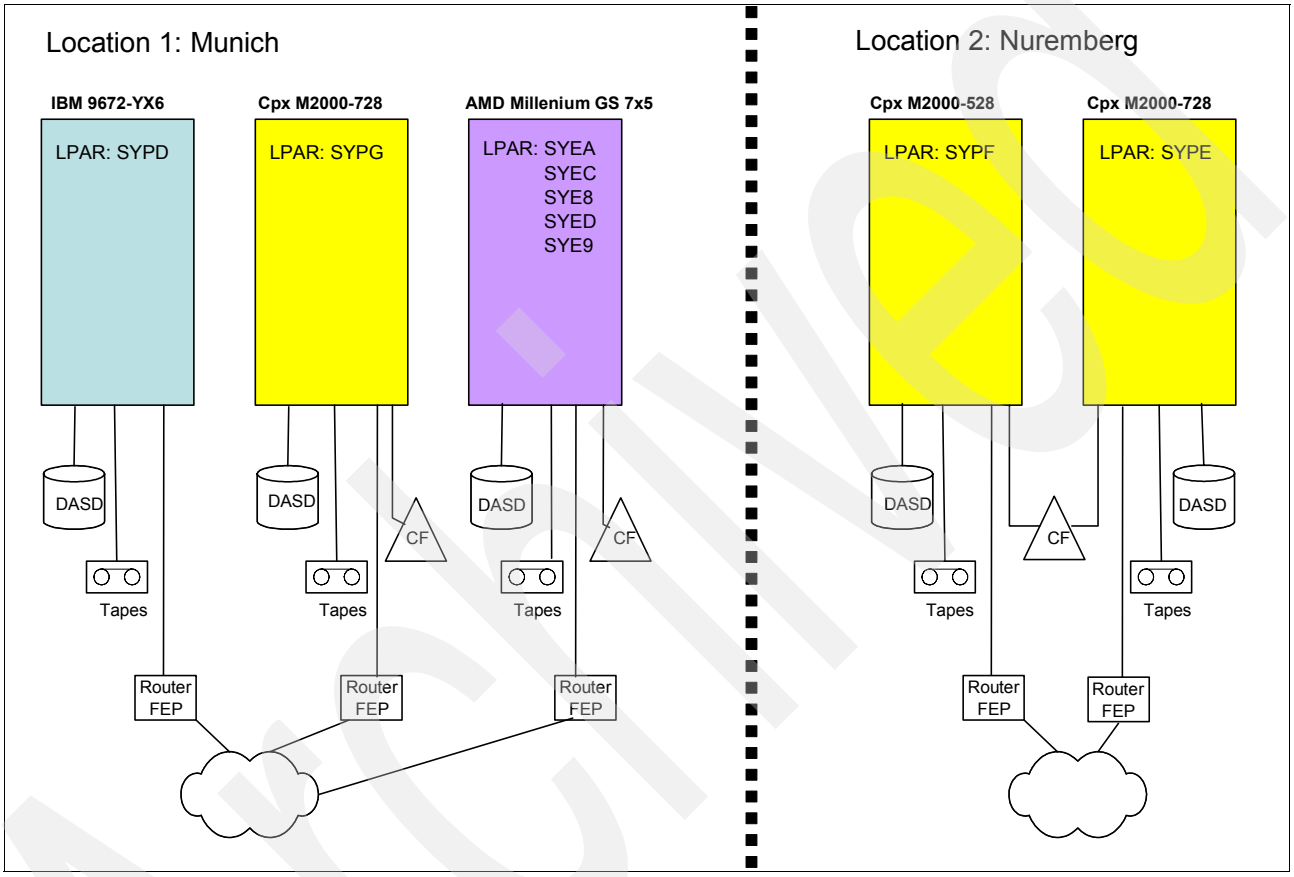


Figure 1-2 The situation in 1999, with one data center in Munich and one in Nuremberg

1.2.2 Expensive disaster recovery strategy based on tapes

At that time, disaster recovery service was provided by a company located hundreds of kilometers distant from the data centers. The concept was to create and maintain tape copies of all relevant data, and store it in a separate location. In the event of a disaster, the tapes had to be selected and carried to the disaster recovery (DR) service provider. However, because of a dynamic increase in IZB's CPU and storage capacity, the DR solution became increasingly expensive. Additionally, the time to recover following a hardware failure was unacceptably long.

1.2.3 No DASD or tape mirroring

IZB could not maintain its zero data loss policy with the DR technology it was using. In a worst case scenario, only the backup tapes from the previous day were available, meaning several hours of data loss.

The other reason why this DR solution was not feasible was because no copies/backups existed for the vital data stored on tapes. That is, batch jobs wrote their output files directly to the tapes without having a copy on DASD, meaning that there was only one copy of that data.

At that time, due to a fall in DASD storage prices, PPRC technology, which provides a zero data loss solution, became feasible. IZB decided to test this technology extensively, resulting in its subsequent implementation.

1.2.4 Systems customized for each client

In 1999, system cloning had not been fully exploited by IZB. Each LPAR had a different setup of software and service levels that had to be maintained by the systems programming department. This was a result of a homegrown situation that had existed since 1994.

There was always a demand for new LPARs that were required at short notice. However, the uniqueness of each client's configuration meant that it was difficult to meet this requirement.

1.2.5 Hardware not powerful enough for largest LPAR

IZB proved to be a good customer for hardware sellers and resellers, since the demand for capacity in a single LPAR was larger than any of the hardware vendors could provide at that time. Additionally, IZB's dynamic growth led to a demand for the newest and the most expensive hardware. However, this was not their preferred position.

1.2.6 Lack of homogeneous software implementation

When IZB was founded in 1994, both parties involved brought their homegrown solutions and products to the "marriage". Not only were different products used by the different companies, but also systems management policies and practices differed. For example, one used DFHSM for space management, while the other used a competitive product from a different vendor. Also, the RACF® policies to secure data were different, the automation and monitoring tools were different, and so on.

The problem with this heterogeneous software landscape was that IZB had to maintain the skills needed to manage all the products. Thus, the definition of software standards within IZB became a huge challenge. An even greater challenge was that of replacing software that did not adhere to the newly-defined standards.

1.2.7 CPUs could not be aggregated

As described earlier, the data centers were 160 km apart from each other. No coupling technology or Sysplex Timer® technology could provide connectivity over this distance. From a pricing point, IZB had two separate sysplexes, one in Munich and one in Nuremberg, with the software licences being calculated based on these two sysplexes which could not be aggregated.

To run the data centers in a more cost-effective manner and ensure optimized software pricing, resource exploitation became the focus of the company during this period.

1.3 The solution

Early in 1999, IZB determined that resources could be more efficiently utilized if the company ran all zSeries-related workload in one data center. From a disaster recovery perspective,

relocating one of the data centers to a distance less than 10 kilometers from the other data center was seen as the best technical solution.

Another driver behind this decision was the benefit it would provide regarding hardware considerations. Consolidating all hardware in one location would ease the installation process because it would mean that only one staff group (rather than two) would be responsible for installing new hardware, thereby increasing consistency and improving conformance to standards. This, in turn, would provide improved manageability and availability.

1.3.1 Shifting a data center

Shifting the Munich data center to Nuremberg was one of the most important projects that IZB ever undertook. The plan was to consolidate all the IBM zSeries-related hardware in Nuremberg. After a long preparation phase, IZB successfully moved the Munich data center to Nuremberg with minimum downtime.

During the move, some positive side-effects were realized. For example, a complete tape mirroring solution was implemented during the preparation phase. Before the move, IZB had shared DASD between the production and development systems. The different dates involved in moving the systems to Nuremberg provided an opportunity to implement a 100% separation of test, development, and production workload, resulting in improved operability, manageability, and availability.

Because of pricing issues involving dark fiber, an active/passive solution was chosen for the two data centers at Nuremberg. This involved the secondary data center providing only DR capabilities, with no production workload running there.

By moving the data center from Munich to Nuremberg, IZB met the following goals:

- PPRC mirroring of all DASD

  All production and development DASD were mirrored to the secondary data center to facilitate optimum recovery time in case of a DASD hardware failure.

- Tape duplexing

  This was implemented for all the tapes. The implementation was done differently, depending on the software using the tapes. IZB classified the tapes into three categories: User; Data Facility Hierarchical Storage Management System (DFHSM); and Beta Systems.

  The User tapes were duplexed with the software product Virtual Tape Support (VTAPE) from Computer Associates, while DFHSM and Beta Systems had their own tape duplexing capabilities. These capabilities were exploited to have one tape in one site and another containing the same content in the other site.

- Disaster management

  Although it is prudent to be prepared for a disaster involving the entire data center, sometimes a problem is isolated to a CPC, a tape drive, a DASD controller or other hardware product. For example, if one CPC fails, only clients running their workload on this CPC suffer from the outage; those running their workloads on any other CPC in the data center are spared.

  IZB decided against impacting the running systems just to recover the failing system, which would have been the case if only a disaster recovery solution had been implemented. Instead, IZB implemented a *component backup* strategy. This meant that disaster involving a single component could be overcome without impacting the other clients.

1.3.2 Managing costs

To manage costs, IZB undertook the following initiatives:

- Leverage the possibility of sharing hardware between the CPCs.

  By setting up a new disaster recovery data center within a 4 km radius of the existing data center, synergy between hardware could be realized. For example, one CPC in the DR data center was defined as the *component backup CPU*. This meant that if any of the other CPCs faced component failure of any kind, this CPU acted as the backup.

- Take the necessary action to qualify for Parallel Sysplex Aggregation pricing.

  This task was challenging and time-consuming. One of IZB's clients had a monolithic, standalone system that consumed a significant amount of MIPS. To quality for PSLC pricing, IZB had to divide this single monolithic system into two systems in a Parallel Sysplex. Additionally, the savings bank sysplex had to be spread across all the existing CPCs so that the criteria could be met. This challenge dominated IZB's activities and decisions for a significant period of time.

- Prepare for Workload License Charging (WLC) pricing.

  In 2001, IBM announced a new software pricing model. At that time, IZB decided that even if the new pricing was not available to it, it would position itself to meet the necessary conditions. It then created new LPARs to serve independent software vendor (ISV) and other specialized products.

  This reduced the cost for those products, because they were running in smaller LPARs, and also reduced the cost for the more expensive products such as Customer Information Control System (CICS), DB2, and so on as the capacity used by the ISV products was moved out of those LPARs.

- Consolidate software.

  As described earlier, the software was heterogeneous during IZB's early days. In order to be more efficient, IZB decided to move to only one product for one function. It then replaced products that did not fit in with that strategy.

1.3.3 Preparing for multi-client support

One of IZB's goals was to reduce its efforts and thus the costs involved in catering to numerous clients. This was primarily a software issue. IZB determined that the different exits present in its clients' z/OS installation, the different product customization, and so on, were creating problems.

IZB invested a considerable amount of time preparing for the following tasks:

- Cloning LPARs
- Cloning z/OS
- Defining a standardized system setup for all clients
- Defining a software stack, including ISV software, as a standard

These efforts were crucial in order to enable the systems programming staff to produce numerous new LPARs within a short time frame.

1.4 Description of the project

Because the systems programming groups had to continue to deal with normal day-to-day activities (taking on new clients, creating new development systems, and so on), migration to

the target configuration could only be completed in a gradual manner. The following section summarizes the different phases involved as IZB moved to achieve its objectives.

1.4.1 Phase 1 (Data center move): 1/1999-9/2000

As mentioned, this document primarily concentrates on the work undertaken by IZB over the past five years. However, the road that led to where IZB is positioned today actually started back in 1999, as described here. This phase consisted of consolidating the systems from the Munich data center into the existing Nuremburg data center.

Motivations

The key motivators for this phase were:

- To improve IZB's software costs by creating one PricingPlex[2] instead of two.
- To implement DASD and tape mirroring technologies, which was imperative because a recovery time of 24 hours had become unacceptable to IZB's clients.
- In addition to mirroring the tape and DASD data, a synchronous mirroring technology such as PPRC was required to address the desire of IZB clients for zero data loss. The requirement for a synchronous mirroring solution placed limits on how far the sites could be located from each other.

Actions taken

Because relocating a data center is a large project, a great deal of preparation was involved. Tasks included testing and selecting a tape mirroring solution, building the new infrastructure in the target data center, and implementing a 100% separation of all test, development, and production data.

Results

Phase 1 produced the following results:

- All three production systems where included in one BronzePlex. Refer to *Merging Systems into a Sysplex*, SG24-6818, for details about the different types of plexes.
- Two data centers in Nuremberg, within a distance of 3.5 km, were involved. While one data center was used for production, the other was a warm standby data center only, relying on capacity backup upgrade (CBU) functions to provide the capacity required in case of a real disaster.

  The infrastructure that was set up by IZB aimed to provide both disaster recovery capabilities and component backup capabilities. The infrastructure's reliability was proven during an extensive disaster recovery test conducted after the data center in Munich was moved to Nuremberg.
- Unfortunately, IZB was only able to spread the Parallel Sysplex License Charge (PSLC) over three of the four CPCs. As a result, they had two PricingPlexes (one with three CPCs, and one with a single CPC), and not the single PricingPlex they had hoped for. This, however, was addressed during some of the follow-on activities undertaken after the move.
- Some availability problems continued to exist in the area of planned CICS outages and DB2. In addition, problems in some ISV products resulted in subsystem or system outages; therefore a later decision was made to move those products to a different LPAR in the same sysplex so that maintenance would be less onerous.

[2] For a description and discussion of the terms related to Sysplex Aggregation, refer to the IBM Redpaper *z/OS Systems Programmers Guide to: Sysplex Aggregation,* REDP-3967.

1.4.2 Phase 2 (PlatinumPlex implementation): 9/2000-9/2001

After having located all their systems in the same data center and implemented a basic Parallel Sysplex configuration, IZB was then positioned to implement an improved infrastructure, thereby enabling the company to react more quickly and efficiently.

Influences during this time frame

The following were influences during this time frame:

- IZB acquired a new client, Customer C, which was another state bank.
- IZB decided to use WebSphere Application Server on IBM zSeries.
- A large LPAR SYSE affected the capacity limits of the largest available servers.

Motivation

The key motivators for this phase were:

- IZB was encountering stability problems with DB2 at the time. These problems were exacerbated by the fact that they were not exploiting data sharing, meaning that a DB2 outage resulted in those applications being unavailable. Therefore, a decision was taken to redesign the DB2 landscape. Refer to Chapter 5, “DB2 data sharing” on page 93 for more details.
- Because the system infrastructure (DASD, security, automation) was not shared in their BronzePlex, adding new LPARs such as WebSphere systems was more time-consuming than would have been the case with a PlatinumPlex, where everything is shared.
- Problems brought about by the BronzePlex implementation; the existing BronzePlex was a dead end from technological and pricing perspectives, with problems, rather than solutions, being the order of the day.

Actions taken

IZB took the following actions:

- March 2001: The DB2 systems were split into three parts and redesigned. For more details, refer to Chapter 5, “DB2 data sharing” on page 93.
- Second quarter of 2001: The existing BronzePlex Z1 was split into single systems again. This was considered necessary in order to build a new data sharing sysplex with new standards and qualities of service.
- July 2001: The first PlatinumPlex was created, consisting of a new WebSphere system and one of the three LPARs that were previously split. This became the basis for the 10-way full data sharing Parallel Sysplex that exists today at IZB.

1.4.3 Phase 3: 9/2001-3/2002

In the months following the implementation of the PlatinumPlex, IZB began to benefit from their new standards and started the process of investigating the value of data sharing.

Influences during this time frame

The following were influences during this time frame:

- IZB acquired another new client, Customer D, which was another bank that needed to be integrated into the IZB environment.
- Because of the client’s totally separated system, software pricing (especially maintaining the PSLC pricing for one sysplex) became a hurdle.

Motivation

IZB's clients were no longer satisfied with the impact of planned downtime on their application availability. Because the only way to avoid the impact that planned shutdowns have on application availability is to implement data sharing and workload balancing, IZB recommended that its clients implement data sharing. This would be a sizeable project, with both the application development and systems programming departments being involved.

Actions taken

IZB took the following actions during this phase:

- The new client, Customer D, became fully integrated into IZB's IT environment. However, it was not integrated into any of the other sysplexes because it had nothing in common with them. Nevertheless, the new z/OS system was built according to IZB's new standards, moving away from the old problem of having an increasing number of unique system images to maintain. Therefore, even though this client was not in a sysplex, it still benefited from all the infrastructure standardization work that IZB had done.
- Because IZB wanted to implement the data sharing concept, it worked on concepts and related aspects which were then presented to high level management. Management subsequently decided that data sharing would be implemented, initially for data warehouse and later for one-third of the 104 savings banks that were currently IZB clients.

1.4.4 Phase 4: 3/2002-10/2003

This phase of the project brings us from the point where IZB made a strategic decision to implement data sharing, through the setup of the infrastructure, up to the time where the first data sharing client application moved to production.

Motivation

The key motivators for this phase were:

- After the decision was taken to implement data sharing, IZB wanted to implement this concept at the earliest, starting with the DB2 data warehouse application.
- IZB identified some supporting activities that could be moved out of the main production systems, delivering a reduction in the software cost of those large LPARs (which were now exploiting sub-capacity Variable Workload Charging). However, to ensure the high availability of these key services, two such systems would be required, sharing the workload between each other, and each able to take over all tasks should the other fail.

Actions taken

IZB took the following actions during this phase:

- In August 2002, a new back-end pair of systems with CICS, DB2, and Adabas was created as the basis for the data sharing projects. The back-end consisted of system P1 (which was one of the three systems split from the earlier Bronze Plex), and a new system, P7. The pilot installation of Adaplex and CICSplex was carried out on these systems.
- Starting in October 2002, the migration of the DB2 data warehouse into the newly created Platinum Plex was carried out.
- In January 2003, IZB implemented a data sharing solution on its WebSphere systems, in the process creating another WebSphere LPAR so that a failover between the two WebSphere LPARs would be possible.
- In October 2003, WebSphere MQ shared queue was implemented. The MQ queue sharing group consisted of the back-end systems SYPE and SYP7.

- Also in October 2003, a client fully exploited the data sharing concept for the first time.
- During this phase, the support software of each LPAR was moved to the newly-introduced SUSY systems. These LPARs took over the following workloads from the production OLTP systems:
  - DASD space and availability management performed by DFSMShsm™.
  - All the output management systems.
  - Management and ownership of the entire network.
  - All printing activities from *all* clients across all IZB sysplexes. For clients in other sysplexes, network job entry (NJE) was used to send the work to the printing systems.

  As a result of these changes, the size of the back-end and WebSphere LPARs were decreased, resulting in overall software cost savings.

1.4.5 Phase 5: 10/2003-8/2004

Building on its positive experiences with their new standardized platform and data sharing and workload balancing, IZB was able to use this technology and experience to further improve its systems management practices and to continue to drive down its software costs.

Motivation

The key motivators for this phase were:

- Due to the growth of some of IZB's clients, there was a risk that IZB would no longer conform to the sysplex aggregation rules, which would have had an adverse affect on software costs. The only option was to split a large LPAR of a client into two smaller LPARs. This would allow the workload to be distributed over two CPCs *and* would deliver the benefits of being in a "real" Parallel Sysplex environment.
- The systems management processes and products needed further optimization. IZB recognized several areas where improvements could be made, including load balancing techniques between the members of a sysplex, migration to larger DASD models such as Model 27s, and so on.
- The LPARs of two savings banks continued to be standalone LPARs (these were the other two systems that resulted from the split of the BronzePlex). Because of the positive experience with full data sharing implementation of some applications, IZB made a decision to investigate merging the two standalone LPARs into the existing eight-way PlatinumPlex.

Actions taken

IZB took the following actions during this phase:

- Customer A's LPAR was split into two LPARs on two different CPCs. Following the split, data sharing and load balancing were implemented for the client. The only problem encountered was that, when using WLM-Managed initiators, the batch workload was not automatically balanced over the two systems as had been expected. This turned out to be the design of WLM, which strives to maximize workflow rather than trying to achieve balanced CPU utilization. To get closer to the desired balanced utilization, IZB developed automation routines to balance the workload.
- After implementing the data sharing capabilities in phase 4 of the project, IZB exploited the following workload balancing capabilities that the different software products provided:
  - CICS TS Temp Storage in CF
  - VSAM Record Level Sharing
  - DB2 data sharing

- VTAM® Generic Resources
- TCP Sysplex Distributor
- GRS Star
- JES Checkpoint structure in CF
- Adabas Cluster Service

► In December 2003, the MNP-Project was started to integrate two standalone systems into the existing eight-way sysplex. It took eight months to prepare all the systems for a merge during a normal maintenance window. The task that consumed the most time was finding and eliminating duplicate data set names.

► After the MNP-Project was completed successfully, the existing z900 CPCs were replaced with six z990 CPCs. During this change, IZB also implemented a multi-site sysplex. For more details, refer to “September 2004: Multi-site operation” on page 48.

Results

At the completion of the project, IZB had the configuration shown in Figure 1-3. All major steps had been carried successfully, and IZB was satisfied that it was well positioned for future challenges.

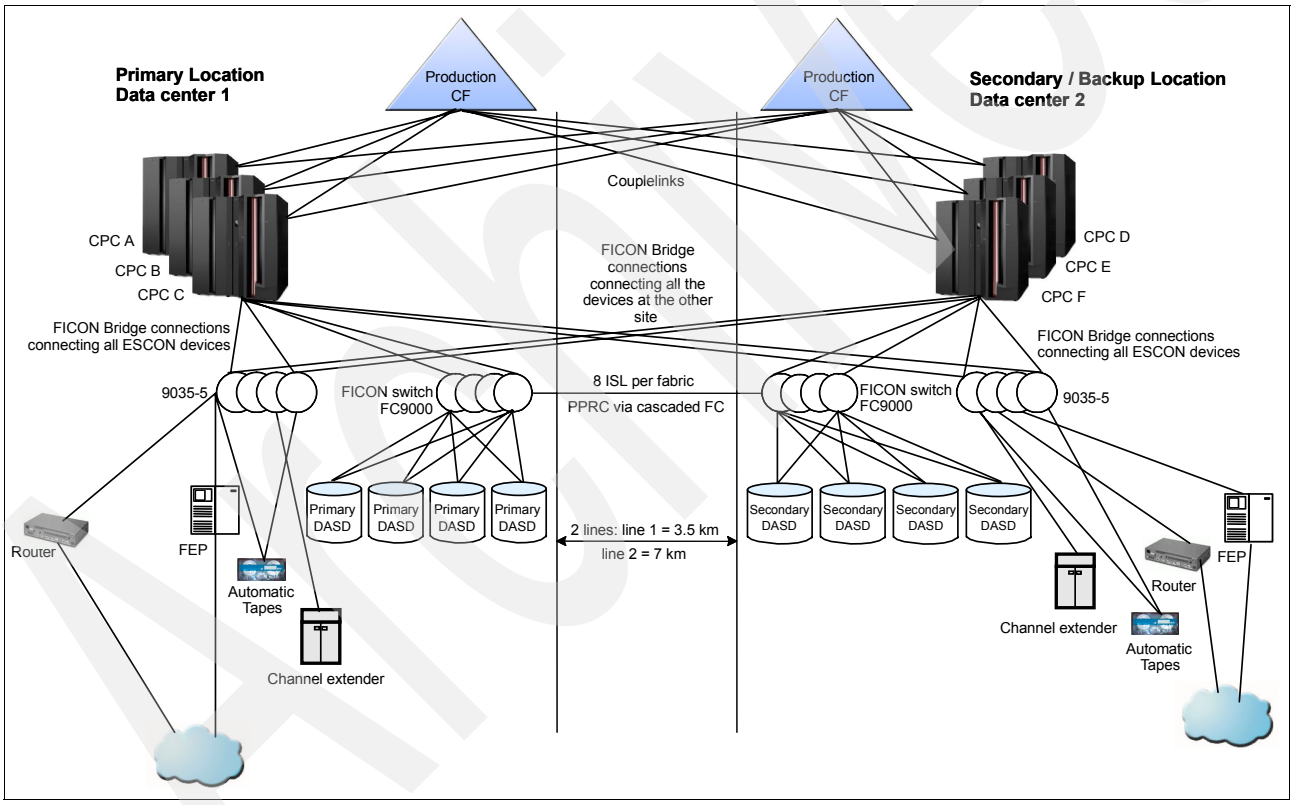


Figure 1-3 IZB’s hardware configuration at the completion of the project - showing two multi-site data centers

1.5 Complications during project implementation

While the overall project was successful and more or less straightforward, some hurdles did complicate the project. During the early days of the project, concurrent with the data center move from Munich to Nuremberg, the Y2K project also had to be addressed. Because the

end date of the Y2K project could not be changed, it understandably had to take precedence over the data center move project.

Another external influence involved the deutsche mark-to-euro conversion. This conversion meant that additional LPARs had to be built and integrated into the existing environment for the systems programming department.

1.6 The future

In addition to the sysplex capabilities described here, z/VM® also plays an important strategic role in IZB. This section summarizes the key enabling technologies used by IZB, and IZB's plans for the future.

1.6.1 zVM installation

In 2005, IZB decided to use z/VM to run all its smaller z/OS systems, especially those systems that were not critical for IZB's business. These systems were moved to guest machines under VM. For example, all systems programming test LPARs were moved under VM. Additionally, two Coupling Facilities were defined under z/VM. Thus, an entire Parallel Sysplex could be created within one z/VM. No extra infrastructure was necessary to build a sysplex under the control of zVM.

Another important aspect was that many of the small systems did not use up all the LPARs available on a z990 processor. IZB also reduced the PR/SM™ overhead associated with running a large number of small systems.

1.6.2 Sysplex features

Despite the benefits and flexibility afforded by z/VM, it is not a viable platform for large z/OS systems or those with high availability requirements. If z/VM goes down, all its guests also go down. All the large IZB z/OS systems run native in LPARs, exploiting most of the Parallel Sysplex features, including data sharing support.

Adaplex

Adaplex is the Parallel Sysplex support of Software AG's Adabas software. IZB is Software AG's largest Adaplex client. Using Adaplex, Adabas databases can be shared over two or more z/OS systems.

VSAM Record Level Sharing

IZB is an extensive user of VSAM Record Level Sharing. VSAM/RLS allows users on different LPARs or different CPCs to share VSAM spheres at the record level with read and write integrity. This also happens to be the first data sharing method implemented at IZB.

DB2

For some applications, IZB uses DB2 data sharing to carry out workload balancing and to increase the availability.

Other sysplex capabilities

in addition to the data sharing features, IZB also exploits sysplex resource sharing capabilities.

The following are some of the features that IZB uses to leverage Parallel Sysplex technology:

- Enhanced Catalog Sharing
- HSM common recall queue
- Resource Access Control Facility (RACF)
- System Logger
- VTAM Generic Resources
- TCP Sysplex Distributor
- Job Entry Subsystem (JES2) checkpoint
- GRS Star

IZB runs its 10-way sysplex as efficiently as possible. With the help of these features, it is possible to optimize the installation. Features such as GRS Star and HSM common recall queue especially contribute to this.

1.6.3 Outlook and planned projects

The following points summarize IZB's future outlook and projects:

- Including new clients into IZB data centers

  IZB is looking forward to offering its service of running data centers to third party clients, that is, to clients outside the state bank and savings bank organization. IZB believes that it can run the data centers efficiently and cost-effectively. Thus, including new clients to its data centers will be beneficial to both the client and IZB.

- Rolling out the use of data sharing for all Customer B's applications

  At the top of the IZB list of priorities is the goal of rolling out data sharing to all of Customer B's applications, instead of to only some, as is the case now. Because IZB now has the requisite experience in implementing data sharing, this is not expected to be a difficult task. With this, IZB aims to reduce the planned and unplanned downtime for Customer B's applications.

2

Developing a multi-site data center

This chapter describes how IZB converted its two data centers into a well-structured multi-site data center providing excellent backup facilities.

The following topics are discussed in this chapter:

- The data centers in earlier times
- The move from Munich to Nuremberg
- The growth path
- Setting up a multi-site Coupling Facility complex
- Planning assumptions
- A new role for z/VM
- Summary

2.1 The data centers in earlier times

As previously mentioned, in 1999 IZB had two fully-developed data centers in two cities, Munich and Nuremberg, at a distance of 160 km from each other. Each data center had its own IT infrastructure, product set, and system management practices, with a heterogeneous architecture. At that time, every LPAR had its own environment, and only a few monoplexes existed.

The two data centers were the result of long-term migration and concentration. Before IZB was founded there were numerous data centers around the country, and each data center was shared by several savings banks. After IZB was founded, all these small data centers were migrated to IZB's two data centers, as shown in Figure 2-1 on page 19. The data center in Nuremberg was responsible for the northern part of Germany, and data center in Munich was responsible for the southern part. Each location was responsible for its own hardware and software.

During its early days, IZB had two technical support groups. The planning group responsible for architecture and capacity was located in Munich, and the installation group responsible for hardware configuration definition (HCD) and hardware installation was located in Nuremberg.

However, this arrangement often led to problems. IZB had always faced some difficulty in installing new hardware, especially in Munich. When the data centers were small, nearly everyone knew where each cable was installed and plugged (and other such minute details), so there was no need to document or label cables.

Furthermore, there were no clear responsibilities assigned with regard to the work to be done in a given center. Thus, numerous people were involved in activities such as cabling, patching, and so on. As a result of these conditions, installing new hardware became difficult as the data center grew. To address these issues, it became necessary to redefine and redistribute the work.

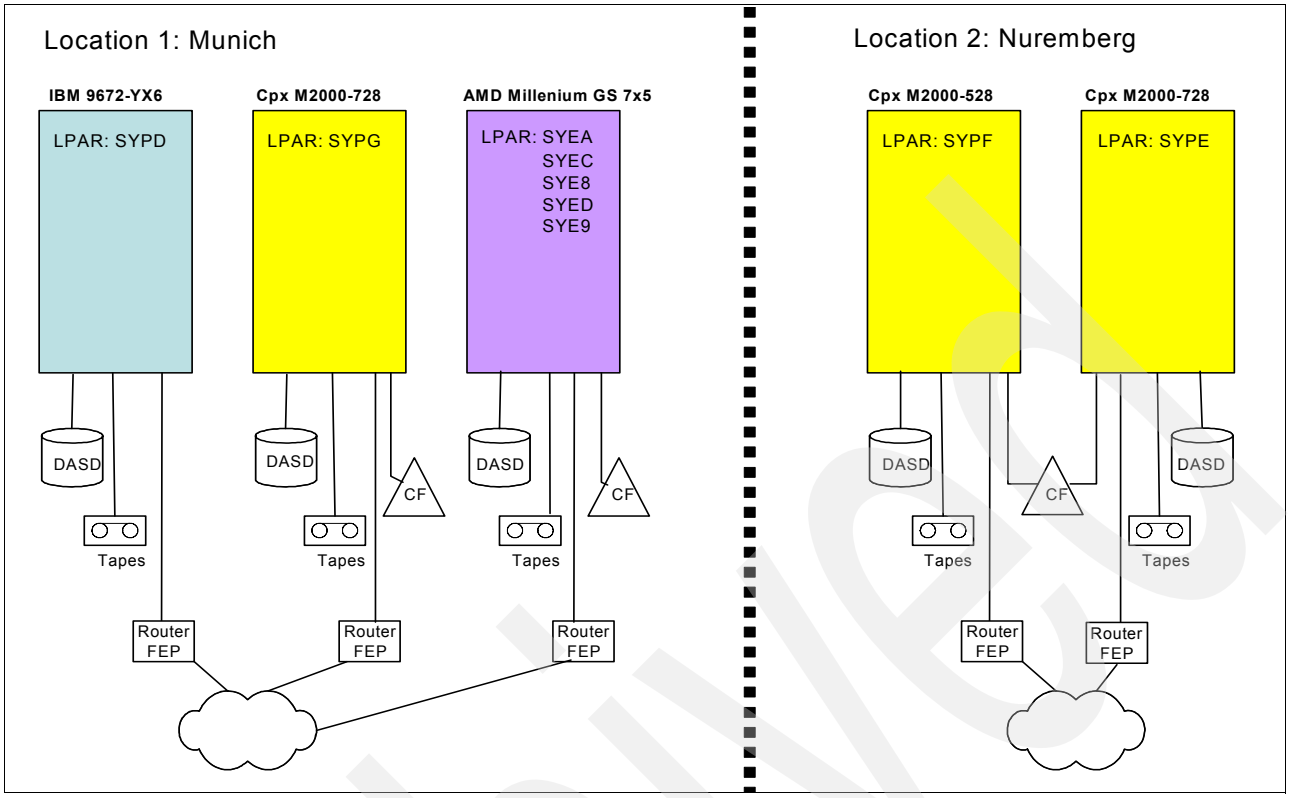


Figure 2-1 IZB data centers in Munich and Nuremberg

Another challenge faced by IZB was in changing its backup solution. For a long time, the IZB backup method involved dumping to tape, delivering the tapes for disaster recovery by car hundreds of kilometers to the vendor recovery site, performing a restore, and hoping that the applications would start and that all the data would be where it was supposed to be.

Not surprisingly, this method proved to be expensive, and eventually IZB realized that it could have its own backup for the same cost. However, two conditions had to be met to enable that to happen: IZB had to build its own backup data center, and the downtime in case of a hardware error had to be reduced from 48 hours to 12 hours.

With this in mind, IZB had to decide where and how the backup data center (or centers) would be located. One possibility was to extend the two existing data centers to provide a backup capability for the other location. However, this would be expensive, and components could not be shared. Besides, there was not enough floor space in the Munich data center for all the disaster recovery infrastructure to be installed there. The one in Nuremberg, however, had enough floor space for installing the necessary infrastructure, with room left for installing the infrastructure from Munich, which meant that components could be shared.

Two other points helped IZB decide to proceed with the move of the Munich data center to Nuremberg:

- Floor space was less expensive in Nuremberg than in Munich,
- It was becoming increasingly difficult to manage the existing data centers. The distance between the two sites meant that it was unrealistic to have one department manage both sites. As a result, separate groups managed each site, resulting in different standards being applied in each site.

Thus, IZB proceeded with its plan of merging the Munich and Nuremberg data centers into a production site and a disaster recovery (DR) site, both located in Nuremberg. With this move, IZB aimed to achieve the following goals:

- Create their own backup location
- Simplify the backup solution to make it more effective
- Reduce the downtime for hardware outages from 48 hours to 12 hours
- Reduce the quantity of expensive floor space being used in Munich
- Build a new team with defined responsibilities for planning, cabling, documentation, and so on, to reduce the problems they previously encountered relating to hardware installation

2.2 The move from Munich to Nuremberg

After investigating and pricing all options, and consulting with all involved parties, IZB was ready to initiate the relocation project. This section describes IZB's objectives and the steps involved in the move.

2.2.1 What IZB aimed to achieve

By consolidating the data centers in Nuremberg, IZB aimed to be eligible for the cost benefits offered by Sysplex Aggregation pricing. It was therefore necessary to bring all the production LPARs of one client into a sysplex, with all the LPARs running in the same location and connected to the same Coupling Facility.

Another objective of IZB was to reduce costs by providing its own backup, with the second data center dedicated purely to backup activities. As a result, the second data center in Nuremberg did not have the infrastructure that the production data center had. For example, the backup location did not have an emergency power system, and the UPS at the time could only provide power for 30 minutes. Also, the air conditioning system in the backup location was meant only for backup and not daily production. Despite these limitations, however, the backup data center turned out to be of great value to IZB because of the reduction in recovery time to just 12 hours from the earlier 48 hours.

A further goal was that IZB wanted all the groundwork to be in place so that when the need for backup arose, the entire process could be carried out without any new hardware changes (for example, no new plugs and cables). Previously, in order for IZB to recover from a CPC failure, time-consuming recabling was required to provide connectivity.

2.2.2 Preparing for consolidation

To prepare for the consolidation, IZB carried out the tasks described here.

Mirroring all DASD data

One of the first things that IZB did to manage the move was to answer the question of how to bring all data to the new location. It decided to mirror all the data in Peer-to-Peer Remote Copy (PPRC) at the Munich data center, and carry the secondary direct access storage devices (DASD) subsystems to the new location in Nuremberg.

To achieve this, the PPRC mirror was suspended so that the downtime would not be calculated during the time the truck moved the DASD. At the new location, the necessary

hardware was installed and the mirror resynchronized with all the latest updates. After these activities were complete, the first initial program load (IPL) could be done at the new location.

The benefit of this method was that the original data was always available. Thus, if something were to happen to the data while transporting it, the original data was still available in good condition. Additionally, the downtime of the LPARs was minimized; it was only the resynchronization time plus the IPL time.

A great deal of preparation also went into getting familiar with the usage of PPRC. When all the LPARs were moved to Nuremberg, the primary DASD were also moved to Nuremberg to reestablish the data mirror with PPRC. This formed the basis for IZB's self-backup.

Introducing tape virtualization

Another problem that IZB had to solve was how to get all the tapes to Nuremberg. They decided to virtualize all the tapes with a software tool called Virtual Tape Support (VTAPE) from Computer Associates. This tool was able to dramatically reduce the number of tapes being transported.

Another advantage of this tool was that it made it possible to separate the tapes for each mandate. This meant that a given tape contained data from only one client. VTAPE was the only tool available at that time to make this possible.

Installing ESCON directors to share devices

In order to get the most out of the installed devices, IZB installed ESCON directors. These switches helped IZB share many kinds of devices among all installed central processor complexes (CPCs), thereby enabling it, for example, to reduce the number of installed tape drives within a sysplex. The switches were also useful for shared DASD data.

Another reason for installing these switches was that all connections that would be required if the need for backup arose could be prepared, because all the devices (including all backup devices such as CPCs, the Coupling Facility (CF), and so on) would be connected to a switch. This advanced the IZB aim of not interfacing with the hardware during backup.

Building the backup guideline

Building the backup guideline was very important for IZB, since it was now handling its own backup. First IZB had to distinguish between backup for a single component and an entire location. Then the company had to decide on the definition of a "component".

For IZB, each component included all devices, such as CPCs, DASD subsystems, switches, routers, tape drives, channel extenders, including every line of cross-site connections, couple links, PPRC links, consoles, cables, and connections. And each device had to have its own backup device. For example, CPCs had to have backup CPCs in DC2, DASDs had to have backup DASDs in DC2, and so on.

Next, IZB decided against considering the guideline of *backup from backup;* that is, if both the primary and the secondary device failed, there would be no further levels of backup. The result of such a decision, for example, was that it was enough to have only one Coupling Facility for location backup.

It was necessary to connect multi-path/multi-channel devices to the CPCs on as many routes as possible. For example, a DASD had eight ESCON channels to a control unit, and IZB had eight ESCON directors at this time. So every channel was connected through a different switch. A faulty switch, cable, or HBA in this configuration had a minimum effect on the working of the systems because only one-eighth of the bandwidth to the DASD was lost, which would be hardly noticeable. For single path devices such as tapes or routers, the

backup was done by duplication; that is, IZB had to buy more devices since they had to have everything backed up.

IZB's clients were happy with this arrangement, because they now knew exactly what to expect. Outage time was reduced to two hours for a component and 12 hours for an entire location (until then, very few hardware faults had a defined outage time).

In order to realize these objectives, IZB found it necessary to install duplicates for many types of devices. For example, backup CPCs (G6 9672 x17) with enough Capacity Backup Upgrade (CBU) power were installed and available. Every DASD subsystem had a mirror at the backup location, the tape roboter was doubled with all the tapes within, and so on.

Figure 2-2 shows the configuration after the move from Munich to Nuremberg, including the backup location.

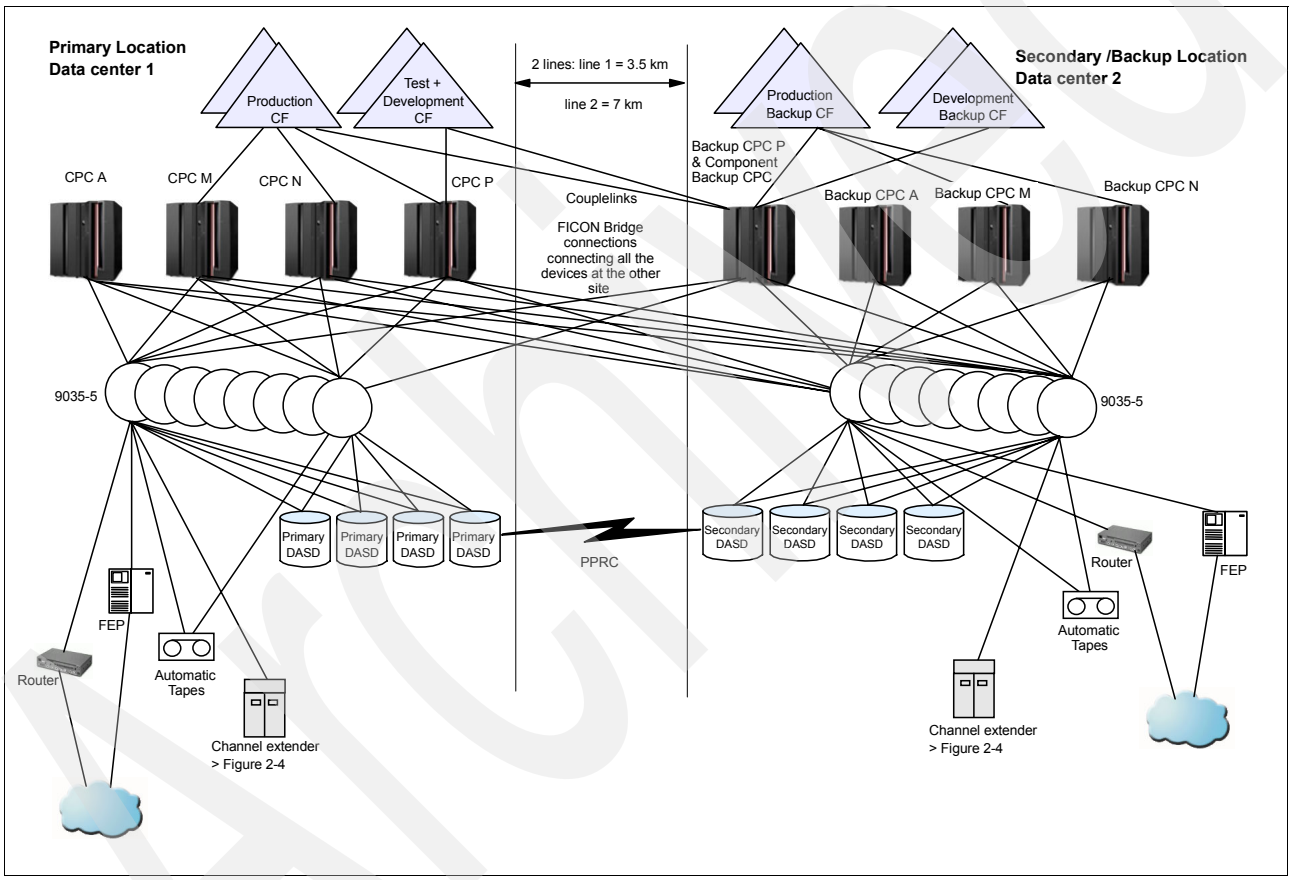


Figure 2-2 Configuration, including backup location - after the move

Next, IZB developed guidelines for procedures to follow if a backup were needed. IZB recommended documenting every detail of the backup plan. This document would also include details about everyone who would be affected by the backup (operations staff, infrastructure staff, systems programmers, and so on), along with the actions they should carry out. Moreover, all these recommendations would be tested periodically and personnel would be trained so that clients could witness the efficiency of the system.

2.2.3 Connecting the two locations

Connecting the two locations was crucial because the second location had to deliver good performance, redundancy, and a high level of security. To achieve this objective, IZB decided

to buy a large quantity of dark fibers: nine micrometers of single mode fibers, directly connected to both data centers on two different routes—between the sites, and within the data centers. The length of the cable on one route was 3.5 km, and on the other route it was 7 km. The distances were suitable for long wave FICON links and ISC3 links, since only ESCON channels (and PPRC links over ESCON) need to be amplified.

IZB continued to work on the backup guideline. First, IZB listed its options. One option, of course, was to build a multi-site data center with active CPCs on every site with enough backup capacity. The other option was to convert one site into a production data center and the other site into a backup data center, at which only backup hardware would be installed.

Each option had its share of both common and unique difficulties. While taking a closer look at the hardware architecture, IZB saw that everything depended on the number of connections between the two data centers. IZB drew up a schematic and began counting the links between the data centers. This turned out to be a problem because of the large number of connections needed between the two locations. For the multi-site data center, 1300 fiber pairs were needed initially. No one was willing to pay for them because there were no existing dark fiber cables between the two sites, thus the telecommunication companies would have to lay them. This was too expensive, so IZB was forced to reduce its link plans.

Likewise, IZB had to make important decisions relating to the standby data center. The first decision concerned *how* a single CPC backup would be done. The second decision concerned *where* the backup CPC would be placed. Placing the component backup CPC at the production site would reduce the number of fibers to about 200, but an additional CPC would have to be installed. Placing that CPC at the backup site would require about 260 fibers.

On the other hand, if one of the existing backup CPCs were to carry out the component backup function for all the CPCs at the production site, an additional CPC could be saved. However, the prerequisite for this was that the additional CPC had to have all the I/O possibilities as the original CPCs in the first site.

After the decision was firmly in place to have a component backup CPC on an available backup CPC, IZB then finalized its backup project. However, it had to pay a high price for taking this decision, not only in terms of money, but also in terms of defining, preparing, and handling the backup.

The other idea that IZB came up with was to use ESCON multiplexer. The devices available at that time could handle only 2:1, 4:1, or 8:1 multiplexing.

All these solutions had one thing in common: they were nearly as expensive as the native links were! A big help to IZB was a new technology from IBM called *FICON bridge*. These new channel cards convert FICON channels (HCD type: FCV) to native ESCON channels (HCD type: CNC) with the help of switches where the bridge cards are placed.

These FICON bridge cards reduce the need for links to an eighth for ESCON channels, so that 200 links could be reduced to 100, and 260 links could be reduced to 160. These bridge cards also made the ESCON environment more flexible, much as simple multiplexers did. (For more information about this topic, refer to *FICON Implementation Guide*, SG24-6497.)

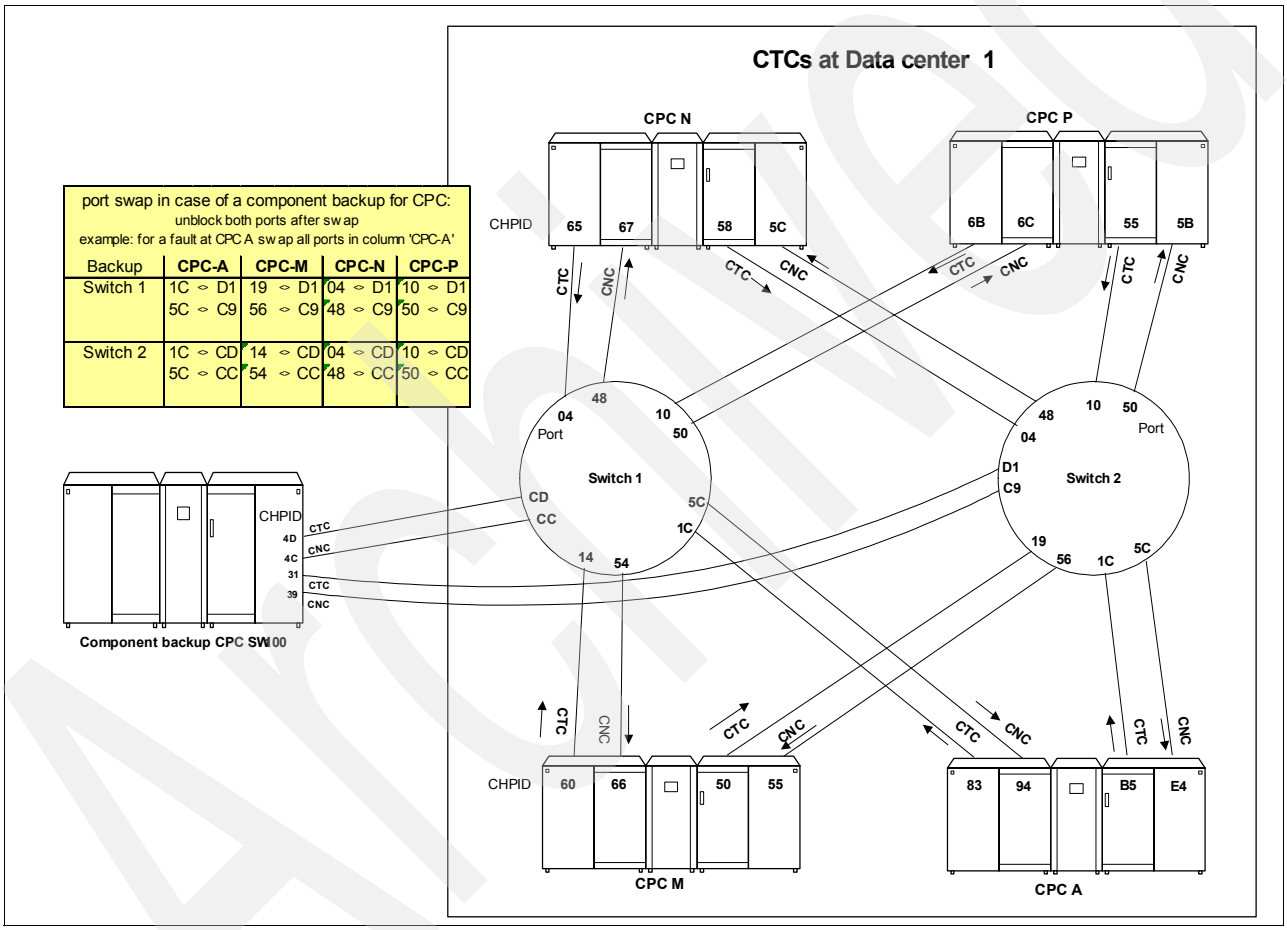
Ultimately, IZB needed only 160 fiber pairs and most of them were used by PPRC. Since there was no way to reduce the links by using a multiplexer, the PPRC links had to be amplified to cover the distance.

2.2.4 Building guidelines

CTCs: VTAM, XCF

IZB now uses numerous CTC connections to connect all LPARs from the sysplex and other systems. There were no existing guidelines, so determining who was connected to whom, using which address, proved to be difficult. IZB also faced the problem of how to make a CTC connection from the backup CPC to the production CPCs.

IZB decided to connect all CTCs only through an ESCON director. This afforded the advantage of the backup CPC being connected to other CPCs with the help of a maintenance function (swap ports) of the ESCON director. Another advantage was that nothing had to be redefined with HCD in case of a backup; only the CTCs had to be restarted. Figure 2-3 illustrates how component backup was performed for CTCs.



port swap in case of a component backup for CPC:
unblock both ports after swap
example: for a fault at CPC A swap all ports in column 'CPC-A'

Backup	CPC-A	CPC-M	CPC-N	CPC-P
Switch 1	1C ⇔ D1 5C ⇔ C9	19 ⇔ D1 56 ⇔ C9	04 ⇔ D1 48 ⇔ C9	10 ⇔ D1 50 ⇔ C9
Switch 2	1C ⇔ CD 5C ⇔ CC	14 ⇔ CD 54 ⇔ CC	04 ⇔ CD 48 ⇔ CC	10 ⇔ CD 50 ⇔ CC

Figure 2-3 Component backup for CTCs

Another aspect of this guideline involved creating a rule about how Multiple Virtual Storage (MVS) addresses would be allocated for CTCs. The goal was for every LPAR to get its own address, so that regardless of which machine or LPAR connected, the receiver always got the same MVS address. In addition, every type (CNC and CTC) and user group (XCF and VTAM) received its own addresses. It was therefore simple to know who was connected, especially if an error occurred and the affected party had to be identified. For more details about this topic, refer to *System z9 and zSeries ESCON and FICON Channel-to-Channel Reference*, SB10-7034, which is available on the Web at:

http://www-1.ibm.com/support/docview.wss?uid=isg29fafd2510f68d17285256ced0051213e

Multiple Virtual Storage addresses

To manage a growing environment, IZB decided to define the guideline of how to use MVS addresses. The objective was for every unit and device to get its own unique address within IZB, as shown in Table 2-1. This would make it easier to find an incident unit or to predefine something by knowing the probable address.

Table 2-1 MVS address guidelines

Device	Address
DASD Location 1	2000 to 7FFF
DASD Location 2	8000 to DFFF
CTC	E000 to EFFF
OSA	0100 to 02FF
Router Location 1	0300 to 03FF
Router Location 2	1300 to 13FF
Virtual devices	0500 to 05FF
HiperSockets	1500 to 15FF
Switches + special devices	0800 to 08FF
Consoles + FEPs	0900 to 09FF
Tape Location 1	0A00 to 0AFF
Tape Location 2	1A00 to 1AFF
Device at Channel extender	F000 to FDFF

Channel extender

Another challenge was to connect the devices that had to remain in Munich with the CPCs at Nuremberg. This involved connecting the environment that was being left behind in Munich to the new location at Nuremberg. The device types that had to be connected were 3420, 3480, 3490, 3745, 3011, and printers 4753 and 3174.

Since some devices “speak” ESCON while others speak bus & tag (B & T), it became difficult to connect all these devices to the new location. There was also an additional challenge of having backup for these devices, since IZB did not want anyone to have to touch the hardware when a need for backup arose.

IZB decided to use special channel extender (XAU) units from CNT. These XAUs would be attached to the host through ESCON. Everything that would be needed (such as B&T, switched ESCON, and WAN) could be connected to the XAU. A four-corner was built with these units, (shown in Figure 2-4 on page 26) so that the units could make the necessary connections equally in case of a need for backup.

The needed connections were predefined within the XAU in a *switch matrix*. Within this matrix were the normal and backup connections to the CPCs, plus connections that would occur if an XAU failed or if a connection between the failed XAUs was also defined.

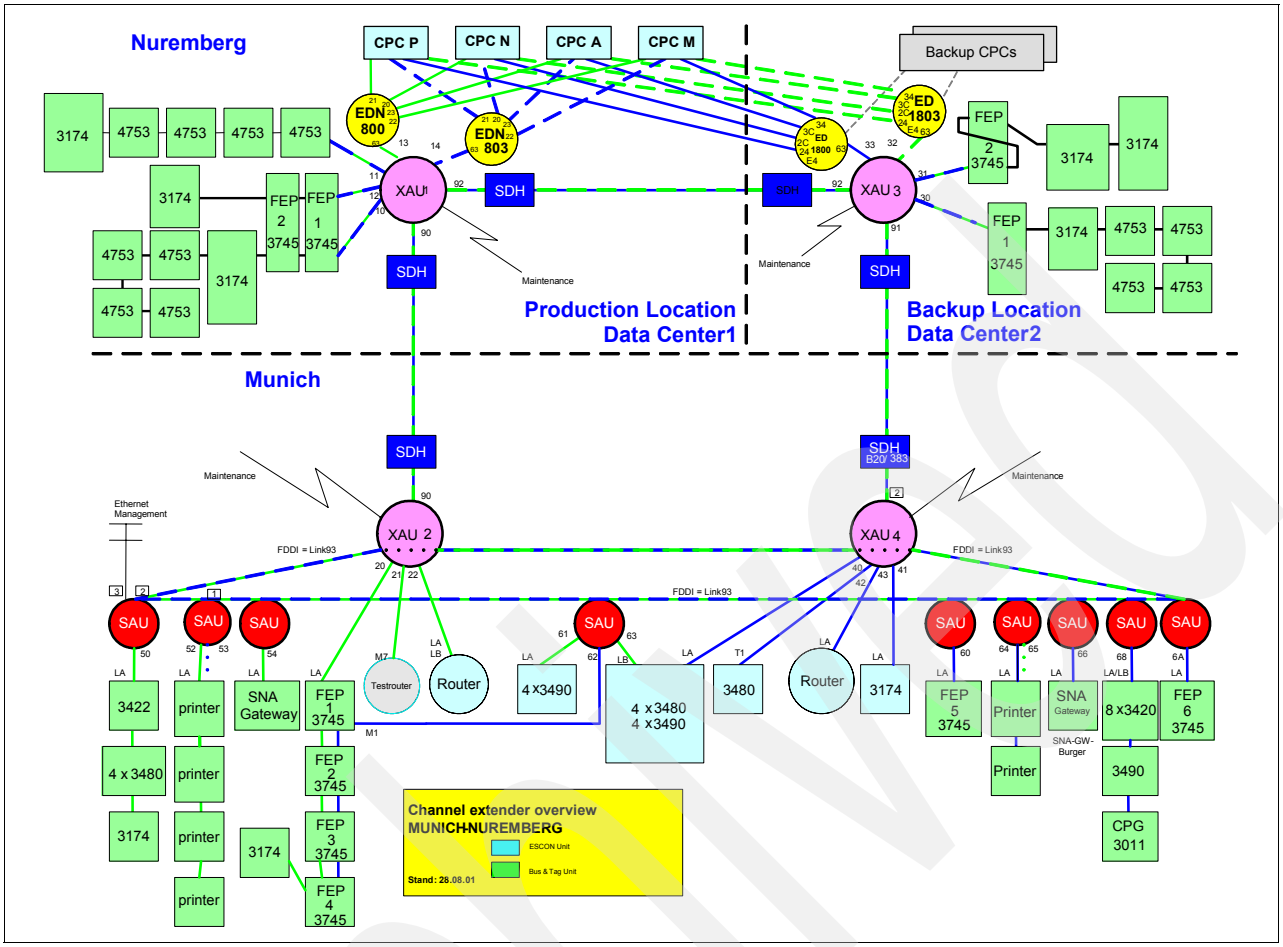


Figure 2-4 The channel extender environment

In short, every device in Munich got four predefined connections: one for everyday use; one if a component of the XAU four-corner failed; one if the normal CPC failed; and one if there was a backup in Nuremberg.

These units had another use, known as *console anywhere*, which made it possible to use one 3174 for up to 32 LPARs as a master console instead of one 3174 per LPAR, which was the norm. IZB was thus able to reduce the number of 3174 control units to three per location. This was useful since IZB had no ESCON connected to 3174, only B&T attachments.

2.2.5 Testing the backup

After these tasks were done, the process of relocating the two data centers to Nuremberg was complete, and it was time to test the backup. IZB planned to undertake the testing with the cooperation of its clients. However, the clients required that their normal production would not be affected by the test, and that all their clients must not realize that a test was being run. Therefore, testing was carried out separately and carefully since the clients were wary of losing data.

In the test, IZB cut all connections to the production environment from the backup machines (that is, CPCs, secondary DASDs, and so on). In this environment, it was possible to start all the applications and carry out updates within six hours. The test worked well and IZB's clients were satisfied.

Note: IZB was the first German client of IBM to activate the CBU feature while a system was running.

2.2.6 Summary of the move

The process of migrating from Munich to Nuremberg took about nine months, and involved two employees of the hardware group. It took such a significant amount of time because all the guidelines as described had to be developed and implemented. In addition, a new site in Nuremberg (including both the actual building and the technical environment) also had to be developed during this time.

The migration produced these benefits:

- PLSC pricing was achieved, which reduced software costs.
- The IZB tech support team got a new structure and a new tool to document all the installed hardware (known as Obtain from Knowledge Flow Cooperation).
- All the data was mirrored (DASD and tape).
- The installed backup location helped IZB provide excellent quality of service.

The migration had its share of challenges, too. The new IT environment worked well at startup, but problems were encountered as time went on, including:

- Initializing the two new Coupling Facilities in case of location backup; refer to 9.3, “Managing Parallel Sysplex” on page 192 for further details about this topic.
- Although all the cables, connections, and plugs were put in place carefully, problems arose relating to incorrect plugging and loss of connection due to the installation of new hardware without an active LPAR using it. The backup test was the only test to check if all the environments and connections worked well. If there was an error in cabling, the time taken for a backup test would increase.
- IZB had to buy six CPCs (four for z/OS and two for CF) with at least one active CPU being used for backup. As a result, there were 750 millions of instructions per second (MIPS) being used for nothing but backup.
- Updating I/O Definition Files (IODFs) and the Input-Output Configuration Data Set (IOCDS) in the backup location involved a significant amount of work and could not be tested. Although the files were created carefully, sometimes problems occurred relating to device definitions (for example, CUAD, switch ports, and so on).

2.3 The growth path

IZB continued to grow by adding new clients and new applications. In the following section, the details of this growth are described.

2.3.1 Growing with a new client

About a year after moving the data center from Munich to Nuremberg, IZB was on a growth path. It provided service to a new client based in Offenbach, about 125 miles (200 km) from Nuremberg. This client had a hardware structure equal to IZB’s. Therefore, it was easy to aggregate the new client with IZB’s systems and backup.

However, there was a special service that IZB provided for this client. For backup of the CPC device, no special CPC or component backup CPC was assigned in the backup location. Instead, for the first time, a *backup couple* was installed at the production location and component backup was carried out in the production location.

A backup couple included two *nearly* equal CPCs (completely equal was not possible at that time due to IBM internals), preferably of the same type. The amount of CPU power used had to be less than or equal to the maximum power of a CPC. Normally, each CPC ran at about 50% of CPU power, with the rest being used in case of backup with the CBU feature. For more details about this topic, refer to *zSeries Capacity Backup (CBU) User's Guide (Level -02h)*, SC28-6823, which is available on the Web at:

http://publibz.boulder.ibm.com/zoslib/pdf/c2868232.pdf

The prerequisite was that every LPAR in the two CPCs was created in both CPCs, including definitions, cabling, and so on. The storage needed for the backup had to always be installed in both machines, with the additional storage being used only for backup. To meet this requirement, the new client was migrated from two 9672-Y46 models to two 2064-1C5 models.

This action also helped to save cost during the occurrence of a fault in the primary location, since only one CPC was needed, as with component backup. Besides, IZB needed only a few of the expensive crossover connections to attach the new client, since most of the I/O did not leave the site.

Another important point was that, at that time, IZB's backup concept used four CPCs. No additional CPCs could be involved without a dramatic increase in backup effort. This was based on the fact that one CPC could be ready for four IOCDSs (A0 to A3), but when there were five or more CPCs to manage, there were some instances you could not prepare for.

The only way in which the backup procedure could be carried out was by writing all the needed configurations as an Input/Output Configuration Program (IOCP) file at a Hardware Management Console (HMC), and if needed, building an IOCDS, as when a new CPC was installed. But with that approach, a backup became a gamble. Figure 2-5 on page 29 illustrates the situation after migrating the new client.

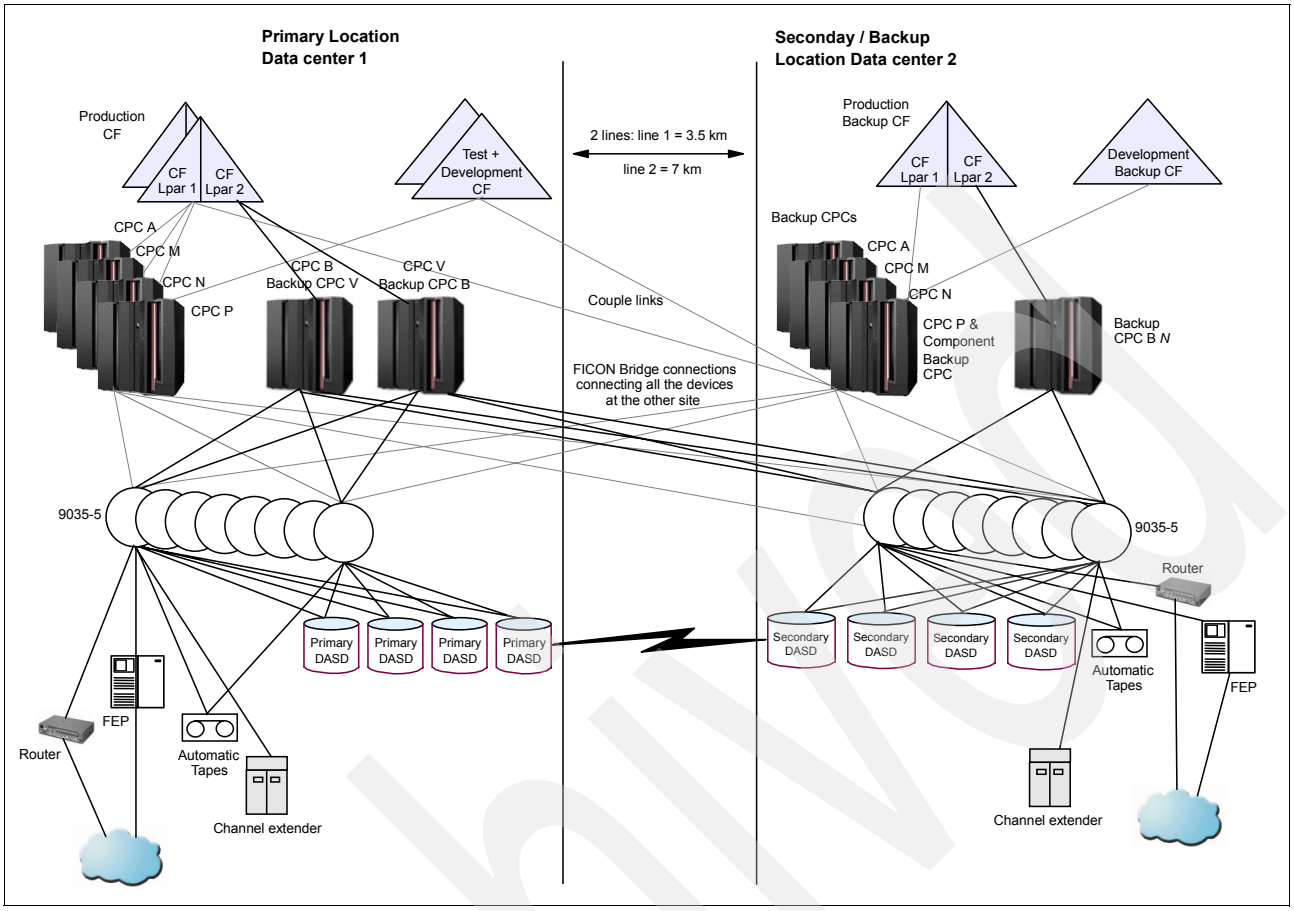


Figure 2-5 The situation after migrating the new client

Results

The migration project for the new client lasted for six months and involved one employee from the hardware group. The only new concept that the hardware group had to work on was expanding the backup concept regarding the backup couples; the remaining aspects were unchanged.

The advantages of this migration was that a backup test carried out three months after the move showed that the backup couples concept worked well. With this, there were no more surprises relating to component backup. The disadvantage of the migration was that in case of a fault in the primary location, the challenges were still the same (refer to 2.2, “The move from Munich to Nuremberg” on page 20).

2.3.2 Growing by releasing new applications

Almost a year later, two new LPARs for WebSphere were built for IZB’s main clients. This, combined with the normal LPAR growth, contributed to an increase in the need for CPU power, with the installed CPCs reaching their limit. IZB was forced to change all the 9672 machines to 2064 machines. The two 2064s that were already installed were too small to carry out mutual backup. Therefore, the backup concept had to be reworked.

For location backup, couples would be built. This meant that every CPC at the primary site got its own backup CPC at the secondary site. This resulted in the installation of an additional CPC for backup at the secondary site for location backup of CPC-B and CPC-V. However,

IZB's credo remained the same, that is, saving money for its clients. The clients, for their part, lowered their backup demands. The most important change was that, for of an outage at the primary location, 93% of the normal CPC power for production LPARs would be enough. In addition, for this backup, case development and test LPARs could be reduced to minimum CPU power, which made it possible to have different machine types for production and backup for the first time. The well-equipped and high-performing machines were located in the primary location, and the budget machines were in the secondary site.

However, there was a challenge to be met in the component backup for a single CPC—in this case, the clients wanted 100% CPU power. To achieve this, IZB had to rework its backup concept. The solution involved installing a 2064 CPC for component backup in the primary location, to save cross-site connections. But it was difficult to manage and customize this machine.

Then IZB did something it would not have done a year previously: it decided that this machine would hold up to six real production machines. Building six machines meant the hardware group had to prepare six IOCDS files for this machine, when the activation profiles could handle only four IOCDS (A0 to A3).

This machine had to be handled like a brand-new machine. For example, in case of a component backup, an IOCP file had to be loaded into the machine from the HMC every time an IOCDS had to be built. Then the activation profile had to be loaded from the disk to SE and the machine would be ready for IML.

This complicated procedure became susceptible to errors. At the same time, though, it helped save money; since no additional expensive FICON bridge cards were needed, there was no need for additional cross-site connections. (IZB did not have any connections left and would have had to dig for new ones if they were needed, and the old CPCs could be used for another year.) IZB therefore went ahead with its new plan.

Results

For component backup, this was only a patched-up solution. So to complete the plan, the involvement of one member of the hardware group was needed for about three months. The backup was altered accordingly.

The advantages were as follows:

- This solution helped to save money because it was cheaper to install two additional CPCs for backup than to buy expensive FICON bridge cards, dig for new cross-site connections, and so on.
- Building backup couples helped to simplify location backup.

The disadvantages were as follows:

- Preparing component backup became very difficult and could not be tested, which have led to some unpleasant surprises.
- Additional CPCs had been bought and could be used only for backup, so about 1400 MIPS could not be used.

Figure 2-6 on page 31 illustrates the situation after the upgrade.

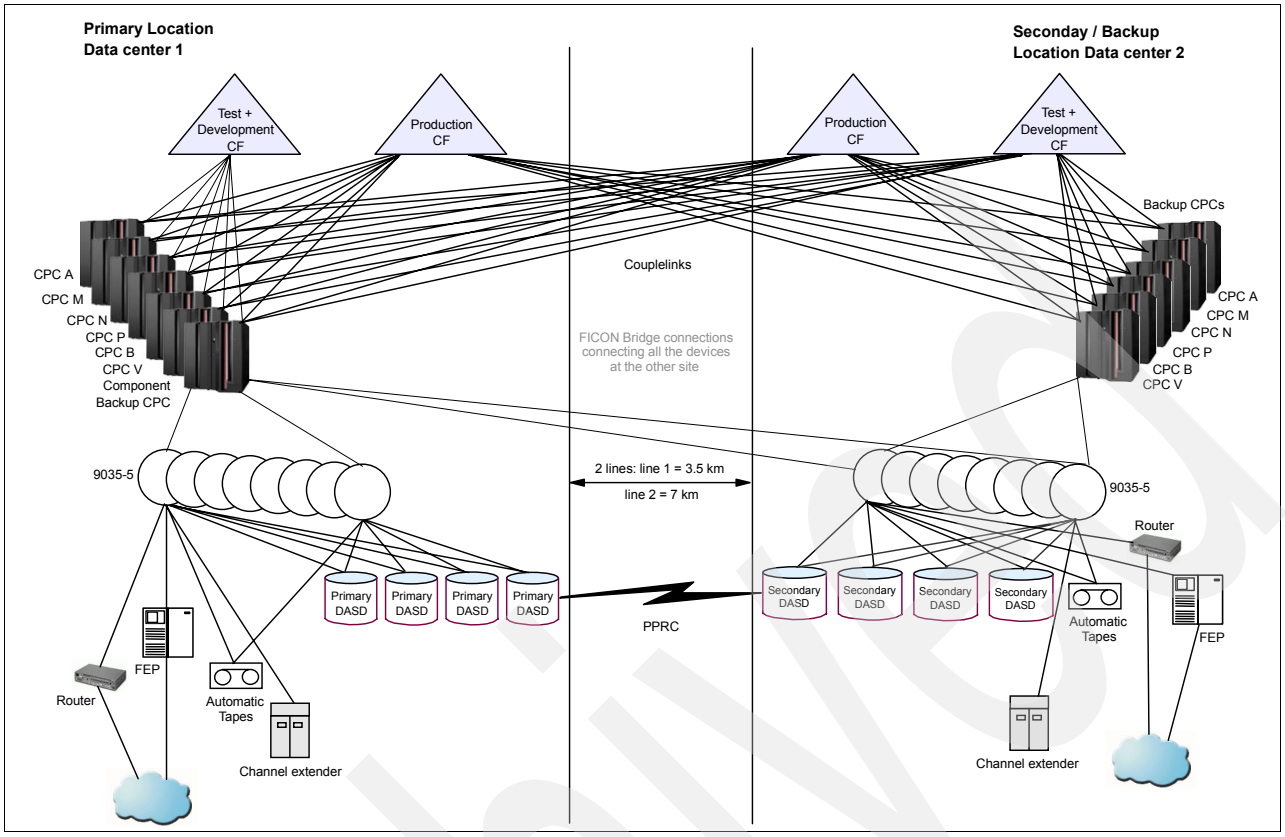


Figure 2-6 The environment after the upgrade

2.4 Setting up a multi-site Coupling Facility complex

Almost a year later, IZB's growth increased on all operating system platforms. With more and more servers being installed, a decision was made to expand or upgrade the secondary location to a *real* data center, with the following components:

- An emergency power system
- An expansion of the infrastructure to meet the new needs
- Fitting the air-conditioning system to meet the needs of the new machines

This was an important change for the IZB mainframe hardware group, as it had been earlier decided that the secondary site would be used for only backup. The new plan involved a step-by-step transformation into a multi-site data center, which was desirable so that CPU power would not be wasted on only backup. (IZB had about 1400 MIPS sitting on its backup machines without being used, which was equal to a fully equipped Generation 6 CPC with 12 CPUs (9672-ZZ7), at that point.) The transformation into a multi-site center would also simplify backup handling.

2.4.1 Upgrading the backup location

One of the greatest challenges that z/OS systems programmers faced during backup was in handling the Coupling Facilities (see 9.3, "Managing Parallel Sysplex" on page 192). IZB hoped to meet this challenge by using a multi-site Coupling Facility (CF) complex; when all Coupling Facilities are used by a sysplex, backup should become easier. All that was needed

was to alter the backup concept for Coupling Facilities and move one of the two production Coupling Facilities to the secondary site.

Additionally, the existing Coupling Facilities had to be upgraded from G5 to G8 machines. The upgrade was useful for achieving this, since the backup CFs could be disassembled and the new one installed and brought into production without any outage. For more details about this topic, refer to *z/OS V1R4.0-V1R5.0 MVS Programming Sysplex Services Reference,* SA22-7618, which is available on the Web at:

http://publibz.boulder.ibm.com/epubs/pdf/iea2i731.pdf

After the CF in the secondary site was working well, then two CFs were disassembled and a new one installed in the primary site. Since all this could be done “on the fly” without any impact on the sysplex, it dramatically increased the chance for getting back online quickly from an outage, since all CFs were always in use and z/OS knew them and knew what to do if a CF was not reachable.

This helped IZB reduce the costs relating to the data centers, since two CFs were not needed anymore. Increasing the length of the connections between the CFs from 10 m to 7 km led to longer response times—but at that point, this was not a problem. On the contrary, the data throughput was better because, with the new Generation 8 CFs and Generation 7 CPCs, it was possible to change the mode on the coupling links from send/receive (HCD pathtype: CFR/CFS) to peer mode (HCD pathtype: ICP). With that, the speed of the links was doubled and the number of subchannels that z/OS could use for communication increased from two to seven per link.

Another action that helped reduce costs and made life easier was also carried out: in the case of component or location backup, all development and test LPARs got only one CF. This reduced the number of links used by the LPARs by about half, besides the channel cards within the CF and the backup CPC.

Figure 2-7 on page 33 illustrates the first step in the transformation to a multi-site data center.

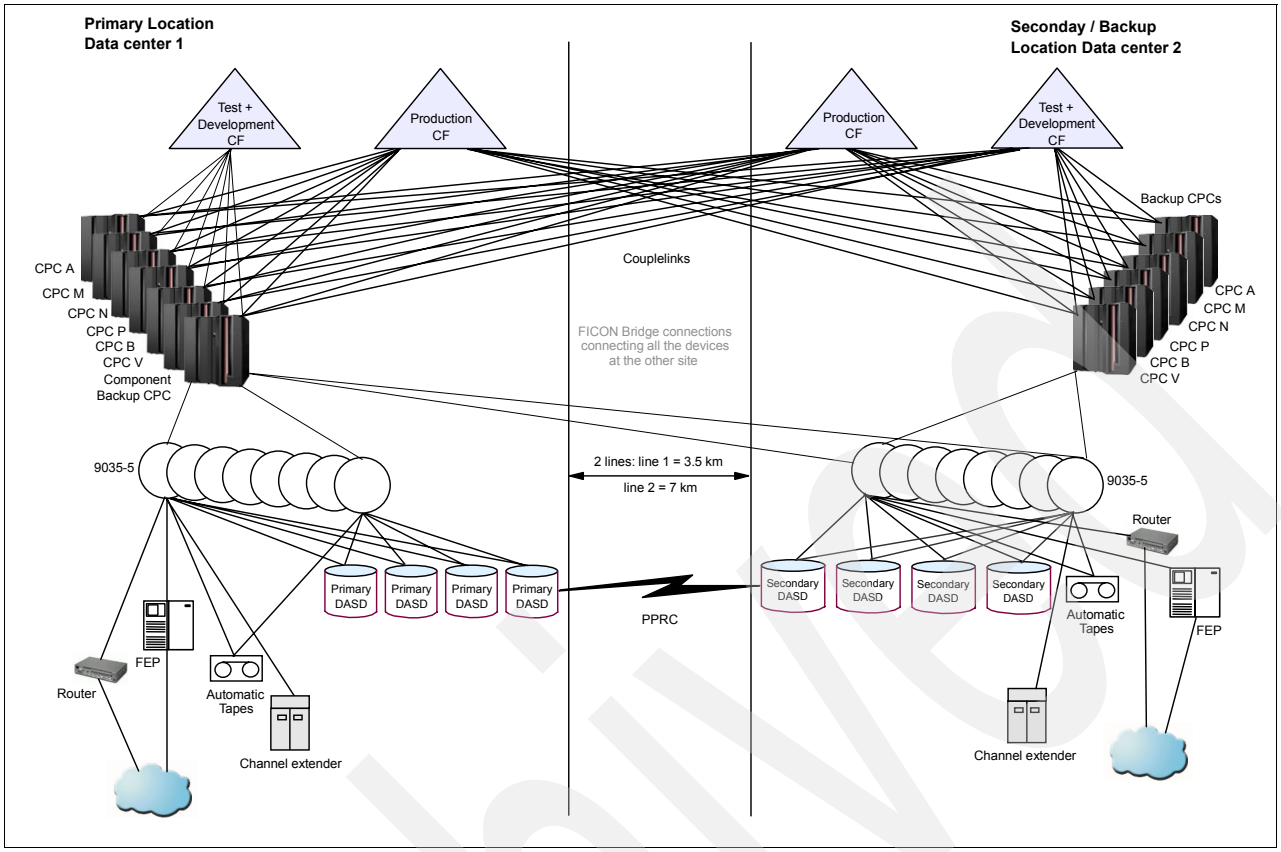


Figure 2-7 The first step in the transformation to a multi-site data center

Results

IZB had simplified its backup concept, which benefitted the z/OS systems programmers, who had not been able to start the LPARs easily during backup.

The advantages of this modification were:

- It helped save money due to reduced hardware needs.
- It simplified the backup process by keeping all installed CFs always active.

The disadvantage of this modification was the higher response time at the coupling links. However, this did not affect applications.

2.4.2 Implementing FICON cascading

Building a multi-site data center within a framework of cost, maintenance, and efficiency was enabled by implementing FICON cascading. This feature was released in early 2003, and IZB became one the first clients worldwide and the first in Germany to successfully use this new technology in the field.

Another reason for IZB to consider implementing FICON cascading quickly was because all the DASD had to be changed the same year. From that point on, all new DASD would be connected with FICON to achieve greater bandwidth at the channels level and to reduce the expensive ESCON channels. Based on this strategy, and with the goal of developing a multi-site data center in mind, a decision was made to test the new technology.

IZB analyzed two vendors, each with a fabric of two switches. The tests analyzed topics such as connectivity, handling, performance, and maintenance. The results were close, thus paving the way for pricing to become the deciding factor. The tests lasted for about two weeks and resulted in the conclusion that this would be the technology that would lead IZB to a multi-site data center. After the tests, IZB bought four FICON switches for each site to get at least four FICON fabrics; (for details, refer to *FICON Native Implementation and Reference Guide*, SG24-6266).

The first “user” of this new architecture was a new type of channel-to-channel connection known as FCTC. This channel type made CTC connections much faster and for purposes of backup, these connections could be predefined in a better way; no “tricks” (such as port swap in ESCON Director) were needed to connect CTCs during backup.

The change from ESCON-driven DASD subsystems to FICON-driven ones made it possible to reduce the number of ESCON Channels and Directors in each site by half. Moreover, the number of DASD subsystems decreased from 14 to five because of the facility to store more data in a single unit. The high bandwidth from FICON helped. This technology and the ability to carry out PPRC through fibre channel over the same switch equipment made it possible to dramatically reduce the amount of time needed for crossover connections. For example, the number of PRRC links over ESCON was reduced from 150 to 32 (-79%) in the case of a cascaded FICON architecture. The number of ISLs within all the FICON fabrics was 32. These ISLs were also used at the same time for other I/Os; for example, to connect the secondary DASDs to the CPC, if needed.

The free crossover connections gave IZB the chance to simplify the backup concept by building backup couples for components, as well. A backup couple consisted of two CPCs, one active with all the z/OS work on it at the primary site, and the other a standby CPC at the backup site.

The connections needed from the secondary site CPCs to the ESCON Directors at the primary site were made with the released FICON bridge cards of the redundant component backup CPC. With that, IZB was able to connect every CPC—both production and backup—in a proper and profitable manner with all of its equipment. With backup couples, backup turned out to be rather easy to manage.

At this point, IZB was very close to becoming an IT services provider with a multi-site data center. By implementing FICON cascading, IZB built the base for a multi-site data center and had to once again develop its backup concept further. Figure 2-8 on page 35 illustrates the situation after the migration to FICON cascading.

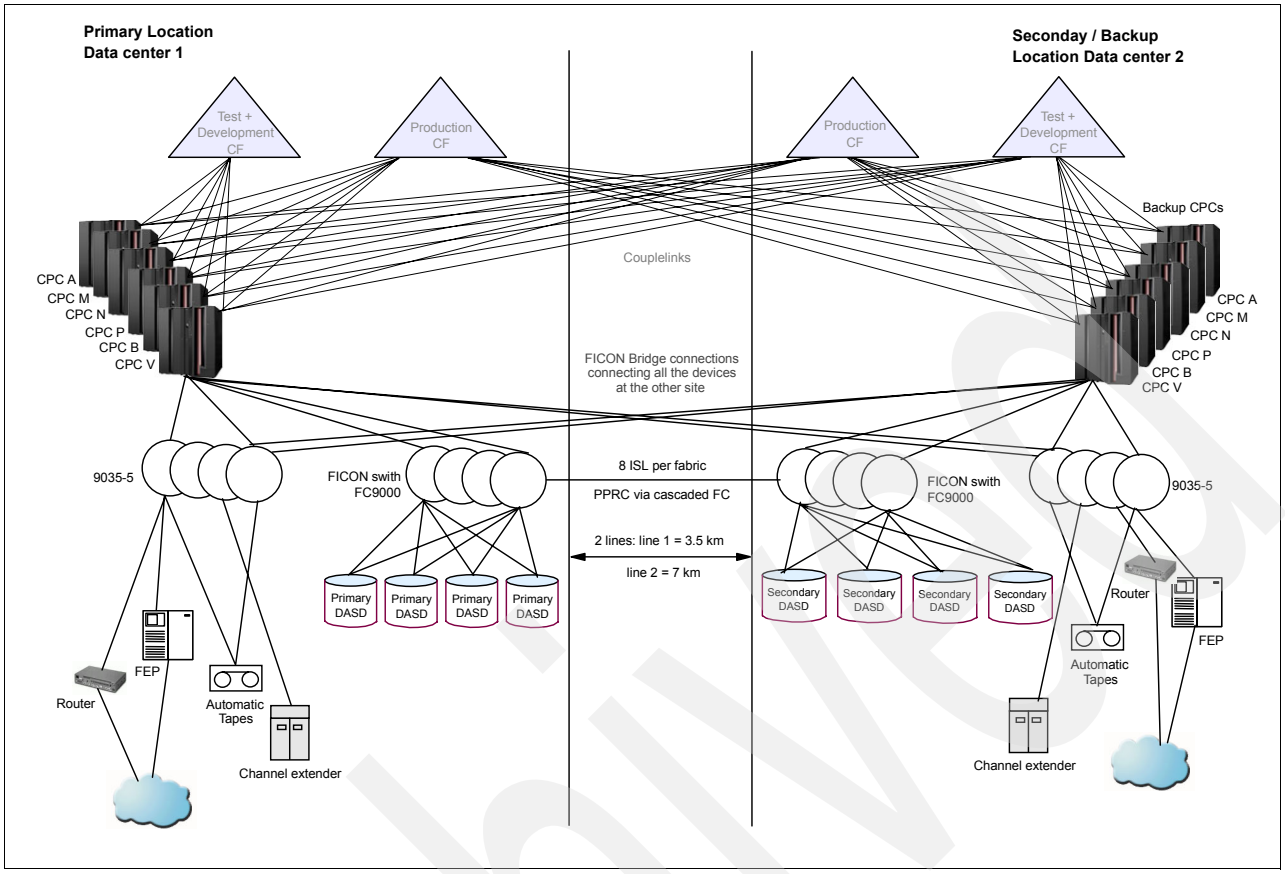


Figure 2-8 After the migration to FICON cascading

Results

These activities further simplified IZB's backup concept, and HCD definitions became much easier for the hardware group.

The advantages were as follows:

- The reduced hardware needs (four fewer ESCON Directors and one less component backup CPC) helped to reduce costs.
- The new DASDs performed well.

There were no disadvantages.

2.4.3 Completing the multi-site data center

By the end of 2004, the CPU power of the 2064 CPC was not enough to carry the load. IZB had to replace them with the new CPC Generation, using z990 model 2084s. This machine type had high flexibility with regard to configuration. Thus, for the first time, it was possible to create CPC couples with an arrangement entirely similar to I/O cards and Channel Path Identifiers (CHPIDs). The new CHPID concept allowed this to work.

Furthermore, the huge CPU power of the new machines made it possible to build new backup couples. This was not done with an active and a standby CPC as IZB had done earlier, but rather with two active CPCs, since the sum of the CPU power used for two older production CPCs was much lower than the maximum available CPU power in the z990. Two 2064 CPCs

could be migrated to one z990, with enough room for growth. This reduced the number of CPCs needed by half.

As a result, changes were made to the backup concept and the HCD definitions. With every couple getting its own IODF, it was easy to build a new IODF for production as well as backup. In every IODF, the backup configuration was included automatically. The backup definitions were also verified with an IODF activation. There were only three IOCDS to take care of. They were tested through an activation of the IODF (writing to HSA), and in case of an unplanned or planned POR, all the necessary definitions were present.

Before migrating, IZB had to answer the following questions:

- How many CPCs are needed?
- Which LPAR will run on which CPC?
- Which two CPCs will build a backup couple?
- Are there any software requirements (for example, WLC pricing)?
- Can the software licence charges be reduced by shifting an LPAR to another CPC?
- What about connectivity (that is, ESCON, FICON, OSA, ISC3, or Crypto cards)?
- How many CPUs are normally needed?
- How many CBUs are needed in the case of backup?
- How much storage has to be purchased?
- What preparations are required for the new *on demand* world?

After answering these questions, IZB concluded that it needed three couples of 2084 machines and wanted to use the new ooCod feature for peaks. The ooCoD feature helped reduce the number of processors needed for normal backup. For peaks, some CPUs could be activated concurrently within a few minutes, which would produce savings because now, the monthly peak did not need to be considered for capacity planning.

However, what did have to be considered were the approximate number of maximum non-peak days which, in the case of IZB, was only about 80% of peak days. For peak days, some CPUs could be activated dynamically with the ooCod feature, at a nominal price. This would be less expensive than if CPUs had to be paid for throughout the month. For more information about this topic, refer to *System z9 and zSeries Capacity on Demand User's Guide*, SC28-6846, available on the Web at:

http://www-1.ibm.com/support/docview.wss?uid=isg2f9dfc895a8a02302852570130054a44e

During planning, IZB made a new assumption for developing and testing a sysplex: all LPARs in a sysplex, including the Coupling Facility LPARs, had to be on the same CPC. This helped save on external coupling links, since the fast and free internal coupling links (HCD type ICP) could be used. After installation, IZB was able to modify its backup concept into a *concept of operations*.

Figure 2-9 on page 37 illustrates the multi-site data center environment.

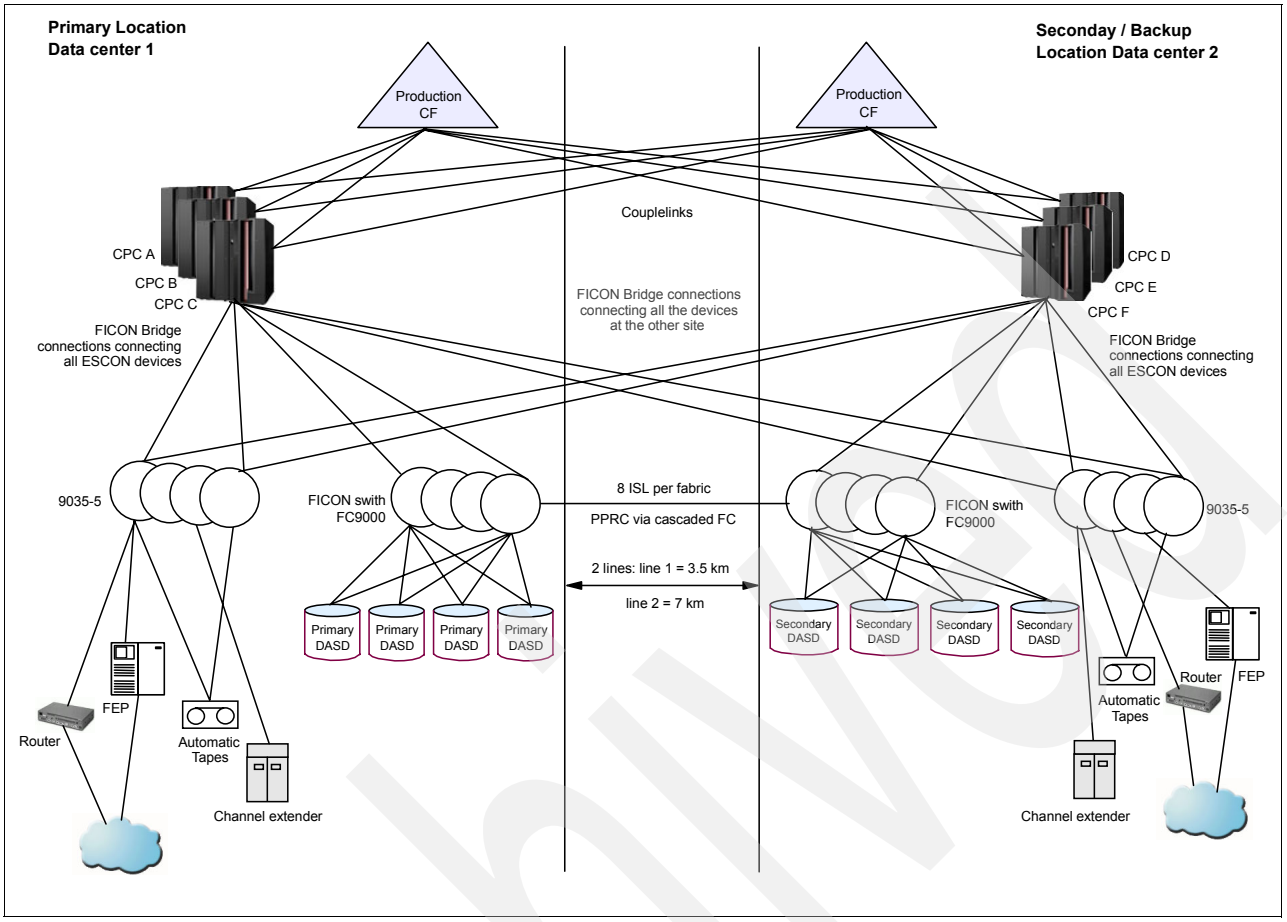


Figure 2-9 The multi-site data center environment

Summary

IZB completed its multi-site data center, and HCD definitions became much easier for the hardware group.

The advantages were as follows:

- Reducing hardware needs (six fewer CPCs) saved money.
- Using ooCod saved money.
- Hardware designated exclusively for backup became a thing of the past because of the concept of operations.
- Tests showed that it worked. The problem of whether an LPAR was started on its normal CPC or on its backup CPC no longer existed. Even performance was the same, and backup was only an IPL away.
- Every machine was an active part of the data center.
- Because every cable, every plug, and everything else was being used, all connections were tested and monitored. There were no more surprises, such as a connection breaking during backup.
- It no longer mattered whether the backup type was a component backup or a location backup. Regardless of the type, the incident LPARs had only to be “IPLed” on the other site.
- Taking care of I/O definitions became easier.

- Using all installed CPUs saved money; there were no more machines sitting idle.
- The development and test sysplexes resided on the same CPC using internal CFs, which saved links and external CFs.

There were no disadvantages.

2.5 Planning assumptions

Documenting all tasks and activities throughout implementation proved helpful in framing rules and assumptions for planning. The documentation also expedited decision-making and helped in meeting project deadlines.

2.5.1 CPU

When planning for CPCs, IZB followed the assumptions listed in Table 2-2.

Table 2-2 Assumptions for CPCs

Assumptions	Aim
Distribution of production sysplex “PlexZ2” on all CPCs	Reduce SW license charges Spread workload over other CPCs for applications using data sharing in case of a CPC outage
Equal hardware equipment for all CPCs	Retain the backup concept
Sysplex for development and testing to be on the same CPC	Save money Reduce couple links and channels
Where possible, production and development LPARs from a client to be on the same CPC	Reduce SW license charges Develop LPARs as load reserves for production LPARs
Number of LPARs to be equal on all CPCs	Reduce LPAR overhead
Ratio of logical-to-physical CPUs to be two or more	Reduce LPAR overhead

2.5.2 Coupling Facility (CF)

While planning for CFs, IZB followed the assumptions shown in Table 2-3.

Table 2-3 Assumptions for CFs

Assumption	Production CF	Development/Test CF	Aim
Usage of CPU	CF LPARs get at least one dedicated CPU.	CF LPARs use shared CPUs.	Reduce hardware costs
Usage of couple links	CF LPARs get at least two couple links.	CF LPARs get at least one internal couple link.	Achieve duplication for production Reduce hardware cost for non-production sysplex

Assumption	Production CF	Development/Test CF	Aim
CF LPAR	A production sysplex always gets two external[a] CF LPARs.	Basically, every sysplex gets one internal CF LPAR. But for special tests, it is possible to start a second one for a period of time. The CF LPAR is in the same CPC as the sysplex LPARs. In the future, all z/OS and CF LPARs of the sysplex should be run under the control of z/VM.	Backup care and performance for production Reduce hardware cost and LPAR overheads

a. In this case, “external” means a CF LPAR on a CPC not running an LPAR of this sysplex.

2.5.3 DASD

While planning for DASD, IZB followed the assumptions shown in Table 2-4.

Table 2-4 Assumptions for DASD

Assumption	Aim
One LCU will be connected to only one sysplex.	Ensure safety Separate clients
For low performance requirements, drives with high capacities of 144 GB should be used.	Save money
For normal and high performance requirements, drives with capacities of 72 GB should be used.	Maintain quality
The response time should be under 4 ms.	Maintain quality
A DASD subsystem should be connected with eight FICONs to the mainframe and with four fibre channels for PPRC to the secondary DASD subsystem. Expansions should be done in steps by four FICONs and two fibre channels.	Standardize connection
New physical control units will be tested 30 days before production.	Improve quality
New devices should be model 27 with PAV.	Save MVS addresses Acquire more storage
New CUs should be planned for 70% of I/O and space capacity.	Enable growth

2.5.4 FICON

While planning for FICON, IZB followed the assumptions shown in Table 2-5.

Table 2-5 Assumptions for FICON

Assumption	Aim
DASD subsystems will be spread over at least 4 FICON Switches.	Maximal 25% lower connectivity by an outage of a switch

Assumption	Aim
A switch in each datacenter always builds a fabric.	Provide a cascaded environment
DASD subsystems get at least two FICON ports per switch, growth in steps of one.	Provide enough bandwidth to back-end
CPCs get at least two FICON ports per switch, growth in steps of one.	Provide enough bandwidth to back-end
Every switch gets at least 4[a] + 4[b] ISLs, growth in steps by 1[c] + 1[d].	Provide enough bandwidth between switches

a. First cableway between the data centers
b. Second cableway between the data centers
c. First cableway between the data centers
d. Second cableway between the data centers

2.6 A new role for z/VM

At IZB, with the growth in the number of CPCs and the corresponding increase in the power of a single processor, z/VM had a new role: it helped IZB by saving LPAR overhead, since there were many small sysplexes containing two or three z/OS LPARs, and an internal CF.

The internal Coupling Facility LPARs running with shared z/OS CPUs, in particular, had a negative influence on the entire performance of the CPC. The problem was that the weight and amount of logical CPUs in an internal CF LPAR had to be defined as much larger than needed because of timing problems between the z/OS LPARs and the CF. A CF request can get good performance results when both LPARs (z/OS and CF) get service at the same time, which hardly ever happens.

A solution to this dispatch problem is to use a special CPU type known as an internal Coupling Facility processor (ICF). However, because ICFs are expensive and IZB's need for ICF power was so small, it was uneconomical to purchase one.

On the other hand, z/VM appeared to be a good solution because it could handle Coupling Facility requests if all parts of a sysplex ran under its control. Another advantage is that small LPARs (each <200 MIPS) can be collected in a z/VM LPAR. So the number of logical processors within a CPC can be reduced, thereby reducing LPAR overhead.

Note: The ratio of physical-to-logical CPUs (RoT) should be less than or equal to two.

All guests (LPARs) running under the control of z/VM profit from z/VM's effective dispatching algorithm, especially the Coupling Facility, which is why IZB put its test sysplex onto z/VM, and followed that with all the development sysplexes of their clients. Figure 2-10 on page 41 illustrates z/VM advantages.

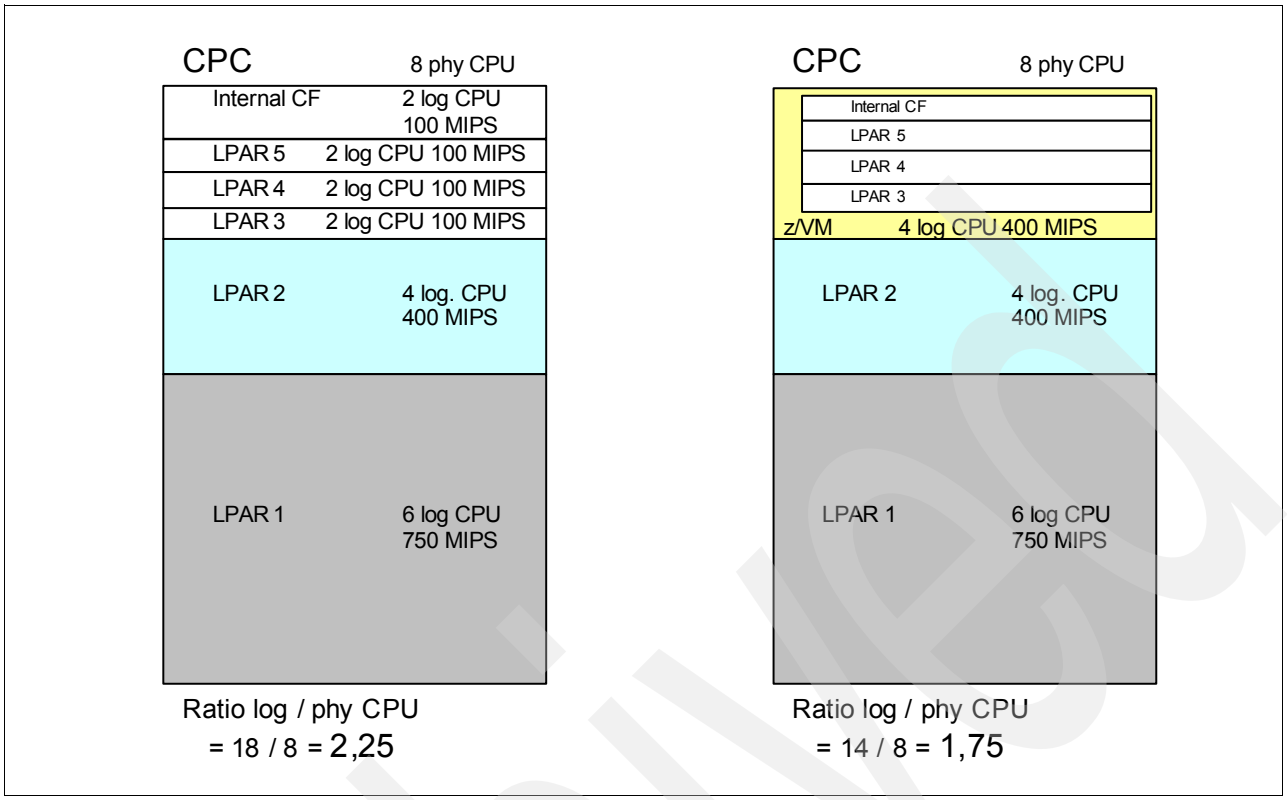


Figure 2-10 Test sysplex running in z/VM

A traditional LPAR distribution is shown on the left side of the figure; here PR/SM has to dispatch six LPARs. This produces some dispatching overhead, besides having a poor logical-to-physical CPU ratio.

The right side of the figure shows the sysplex (LPARs 3 to 5 and CF) under the control of z/VM. Here, PR/SM has to dispatch only three LPARs (LPAR 1, 2, and z/VM) instead of six. A desirable logical-to-physical CPU ratio can be achieved, thereby reducing LPAR overhead and improving the performance of the sysplex.

2.7 Summary

The methodology used by IZB to develop a multi-site data center included many features and took five years to complete. A major contributor to the lengthy implementation was that the technology needed had not yet been developed in the late 1990s, when the initiative began at IZB. But year after year, IBM came out with new and intelligent features that helped IZB reach its goal of developing a multi-site data center with a cost-efficient infrastructure and a simple and secure architecture. The architecture reduced the need for administration to a minimum.

Following are some key technologies that contributed to IZB's goal achievement:

- FICON bridge mode
- FICON cascading
- Coupling Facility
- Peer links
- Powerful CPCs
- PCHIDs
- CBUs

- ooCoD
- PPRC over fibre channel

Overall, the development effort proved to be both profitable and worthwhile for IZB, for the following reasons:

- CPU power is no longer wasted.
- There are no unpleasant surprises during backup because of badly connected cables.
- There are no surprises regarding IODFs, since all IODFs are always active and therefore verified.
- New LPARs can be integrated very quickly.
- Existing LPARs can be moved from one CPC to another using an IPL.
- Hardware backup is only an IPL away.
- Backup is a thing of the past; the present focus is on operations.

Tip: If you decide to follow IZB's path with regard to hardware architecture, advise your clients that you can only do production, not backup.

Previously, IZB clients were used to testing their backup ability each year in a separate environment. Now IZB carries out a hardware backup test that is transparent to most clients. Sometimes clients ask IZB, "When will you carry out our backup test this year?" and IZB replies, "We did it last month!" So inform your clients of this changed way of testing.

3

Network considerations

This chapter describes the central mainframe communication structure at IZB for System Network Architecture (SNA), Advanced Peer to Peer Networking (APPN), and Internet Protocol (IP). IZB's backbone provides secure connectivity between mainframe networks for multiple clients.

Beginning in 1999, requirements such as systems growth, expanded backup capabilities, increased application availability, maintenance without service interruption, exponential growth in IP communication, reduction in complexity, and costs led to a network redesign at IZB.

The implementation of Parallel Sysplex technology also required additional networking functions, such as automatic failover, load distribution, and intelligent load balancing.

This chapter highlights the major design criteria involved and describes the selected Parallel Sysplex high availability networking functions:

- VTAM Generic Resource
- Dynamic VIPA
- Sysplex Distributor

Finally, the chapter illustrates example applications to clarify the implementation of these functionalities and the benefits to IZB.

3.1 Central mainframe network

In 1999, the central mainframe network interconnection consisted of a channel-to-channel (CTC) network that used the ESCON Multiple Image Facility (EMIF) as transport between the Central Processing Complexes (CPCs) or attached networking peripherals. All communication was served by VTAM SNA/APPN. Network connection to other server platforms or to outside the processing center was provided by Front End Processors 3745 (FEP), Cisco Channel Attached Routers (CIP), or locally connected cluster controllers.

The redesign of IZB's mainframe network was driven by several factors and by upcoming requirements, as detailed in the following section. The implementation of Parallel Sysplex technology was just one part of the redesign, but it had major impacts on other parts. Parallel Sysplex networking functions interact or depend on services provided by the network infrastructure in front of the mainframe system complexes.

3.2 Factors and requirements behind the network redesign

The SNA/APPN network redesign and the implementation of IP communication on the mainframe was influenced by the following factors.

3.2.1 Construction of the backup processing center

As mentioned in Chapter 1, "Introduction to IZB" on page 3 and in Chapter 2, "Developing a multi-site data center" on page 17, IZB installed a second backup processing location. From the network perspective, the primary site and the backup site were now redundant and both infrastructures were in full production. External network links are load balanced connected to both processing centers. If one of the outages (disaster case "K-Fall") occurs, then approximately 50% of the network load will be shifted or rerouted to the remaining links and location. And regardless where the production LPARs are located, the dynamically routed networking infrastructure provides the connectivity at both processing centers without any changes. This flexibility also guarantees a smooth migration from the primary and standby backup processing model to the final multi-site processing mode.

3.2.2 Systems/LPAR growth

By 2002, the establishment of systems programming, testing, engineering, production and so forth on multiple Parallel Sysplex systems had raised the system count by 13 LPARs. High availability CTCs for SNA/APPN and IP systems interconnection on an "any-to-any" network layout did not represent a scaling approach, because each new system would increase the network definitions and links exponentially. Assigning networking responsibilities to selected systems and members in a Parallel Sysplex made it possible to design a new and scalable backbone; refer to 3.6.3, "Current design: Scalable SNA/APPN backbone" on page 51 for more details about this topic.

3.2.3 Increased application availability

Multiple and distributed application authorities within the Parallel Sysplex requires common network access to applications like VTAM Generic Resources and IP Sysplex Distributor. Load distribution between the application entities and automatic failover are features of VTAM Generic Resource and Sysplex Distributor, easing service and maintenance on a sysplex application and expanding the time frame for planned maintenance beyond the agreed service hours.

3.2.4 Increasing IP applications

All new applications provide IP interfaces or transport capabilities. Existing subsystems have been extended with IP communications facilities. The explosive growth of Internet and intranet applications, like internal user portals and Internet home banking, have increased IP connection counts and bandwidth consumption dramatically.

3.3 Multi-client support

The ability to run multiple clients with separated system environments in one processing center or Parallel Sysplex requires special measures in security planning and network planning (for example, providing exclusive network connectivity to a client's own corporate networks, and so on). And shared network entities also grant multiple clients access to secure network infrastructures. This produces synergistic results for Internet and business partners access.

3.3.1 Reduced complexity and cost

The following factors led IZB to decisions designed to reduce complexity and cost in its operations.

Front-end processor (FEP) replacement

The former home banking application was accessible via a public X.25 packet switched network. 3745 front-end processors with NPSI software served the network access. However, continuous growth in the home banking users count led to massive transaction peaks on target days (monthly allowance), and reached the CPU or LIC scanner bandwidth limits. High maintenance fees and high software fees, combined with a lack of high availability support in a Parallel Sysplex scenario, confirmed the IZB decision to replace its 3745 FEPs.

Token ring-to-Ethernet migration

The development of more efficient Ethernet variants and the continuing improvement in Ethernet switching are leading to the replacement of token ring in networks and are also impacting token ring hardware availability. But central SNA Logical Link Control/Source Route Bridging (SRB) structures still depend on token ring media like FEP Token Ring Interface Couples (TICs), Cisco CIP and so on. Therefore, IZB's migration strategy had to take into account converting communications from SNA to IP, or converting SRB to Source Route Translational Bridging (SRTB), which enables Ethernet to act as a media provider.

The following section describes the steps IZBs took to accomplish the network migration. Subsequent sections detail the structural networking design for SNA/APPN and IP.

3.4 Network migration steps and evolution

This section provides details of the phases that IZB passed through when migrating its Parallel Sysplex, with a particular emphasis on the milestones in mainframe networking as highlighted in 1.4, "Description of the project" on page 9.

1999: Setting up a new data center

IZB's Datacenter 1 contained a fully redundant network infrastructure, all within one building complex. IZB then opened a new backup processing center, known as Datacenter 2, which was located about 3.5 km from Datacenter 1.

In order to include Datacenter 2 in the existing backbone network, some components were moved to the Datacenter 2 location. Interconnection was established by dark fiber links. Dual-homed WAN links located in Datacenter 1 were also split up to both locations.

1999/2000: Data center consolidation

During the preparation phase for moving mainframe systems from Munich to Nuremberg, IZB built a channel extension network between all locations. This network existed primarily to connect channel-based printers, tape drives, or point-of-service devices to the new mainframes in Nuremberg.

However, function and performance tests with channel-attached routers connected by the extended ESCON network showed massive throughput problems. Bulk file transfers affected all online sessions and transactions by causing unacceptable response times. Overall throughput was also limited because internal channel protocol handshaking caused added network delay by WAN links and ESCON extenders.

This situation reinforced IZB's decision to move the channel-attached routers to Nuremberg and cover the distance between Nuremberg and Munich with standard IP LAN extensions using redundant routers and multiple 34-Mbit/sec WAN links. Time-sensitive logical link control sessions (SNA) were handled by using Data Link Switching (DLSW) over the WAN and adapted to the source route bridging infrastructure in Munich.

1999/2000: Y2K preparations

An additional challenge during this period was the preparation of an isolated test system environment for Year 2000 (Y2K) tests. The test mainframe LPARs needed network read-only access to the production systems. Various VTAM and IP application exits provided the logical access to and from the Y2K test bed. For security reasons, IZB decided to implement additional IP stacks for Y2K testing, and provided synchronized time stamps by Simple Network Time Protocol (SNTP) to decentralized servers and network devices.

2000: Securing internal IP networking

The implementation of multiple IP stacks (common INET logical file system) with dedicated IP services bound by stack-affinity application listeners matched the security expectations of IZB's Y2K requirements. The experience of running multiple IP stacks in a system also simplified the connection setup to networks with different security levels for their individual clients.

2000 end: WebSphere

The decision to implement WebSphere on z/OS required Internet access to mainframe front-end systems. Several projects began to use WebSphere on zSeries with Internet applications for Bavarian savings banks such as Internet home banking. Using multiple IP stacks, IZB created a WebSphere communication platform to grant secure user access from the Internet, while separating back-end connectivity within the Parallel Sysplex to back-end mainframe systems. This platform was later extended by additional intranet network connectors.

June 2000: A new client

Similar to the mainframe movement from Munich to Nuremberg, IZB relocated the mainframe systems of Customer C from Offenbach into its processing centers. Mainframe channel extensions and LAN-to-LAN interconnections covered the distance.

July 2000: New networking capabilities through z/900 systems

Starting with the z/900 hardware, the new Open Systems Adapter Express (OSA-E) delivered simplified access to local area networks and boosted I/O rates into the gigabit range. The new DMA Access method for IP traffic, known as Queued Direct I/O (QDIO), together with the ability to share an OSA Express feature among multiple LPARs in a CPC, enabled the relocation of IP network traffic from a CIP router to OSA Express.

After connecting the test sysplex PlexW1 to three systems (SYT1-SYT3) using internal XCF communication and shared OSA Express Connectivity to external networks, the WebSphere production system SYP2 joined the OSA Express transport method.

April 2002: Expanding the production sysplex by supporter systems

Freeing the production systems from “housekeeping” applications, IZB expanded the production sysplex by two LPARs, SYP4 and SYP5. Both systems were “support systems” and hosted the consolidated networking functions. SYP4 and SYP5 act as VTAM network nodes, so all external communication was directed to them. For IP communications, the support systems are represented by the distribution stacks (for more information about this topic, refer to 3.8.3, “Sysplex Distributor” on page 58.

Consolidate TPX session managers on support systems

The session managers for TPX were relocated to the support systems and were allocated a common application control block (ACB) from VTAM generic resources.

Splitting user interface programs from back-end processing

In addition, a graphical user interface (GUI) to the Beta output management solution based on HTTP was an ideal candidate to run on the support systems, so HTTP servers and the accompanying sysplex distributor were placed there.

September 2002: Layer 3 LAN switching

Static or dynamic virtual IP addresses (VIPAS) and the sysplex distributor function require a dynamically routable network topology. The Communications Servers OMPROUTE routing tasks, together with Open Shortest Path First (OSPF) dynamic routing protocol, interface with routers in front of the mainframe systems. But the VTAM/Communications Server and the network infrastructure are often managed by different departments.

To address this, IZB started a joint project called “switched LAN access to centralized hosts” to set up a flexible and powerful layer 3 switching network front-end. This platform provides necessary networking functions to the systems, and also defines clear interfaces and responsibilities between mainframe networking and the networking infrastructure department (refer to 3.7, “IP network” on page 54 for more information about this topic).

2003: FEP replacement

At this point, IZB prepared its migration strategy to remove the FEPs. SNA Network Interconnection (SNI) leased line connections between business partners and IZB were replaced by LAN-to-LAN connections with Enterprise Extender Links on top. Packet-switched X.25 and ISDN lines moved from FEP LICs to Cisco routers acting as XOT (X.25-to-TCP) converters. Together with the mainframe software HostNAS, the XOT sessions are terminated inside the HostNAS mainframe application. For reasons of availability and security, Enterprise Extender and HostNAS Service run on multiple LPARs in different Parallel Sysplexes.

Multiple HostNAS Services are offered through a Sysplex Distributor configuration. By contrast, Enterprise Extender Links are fixed to system VTAMs. Using dynamic VIPA or Sysplex Distributor to present highly available EE Service is not a feasible solution. However,

using multiple fixed EE links provided by VTAM border nodes in different Parallel Sysplexes provides equal availability.

September 2004: Multi-site operation

The multi-site operation has production systems in both locations that are served by the networking structure without any changes. From the beginning, the network was designed and sized for this final step. All network devices on both locations are in production, and the dynamic OSPF routing guarantees consistent routing and reach ability. All systems and applications are ready to serve, independently of where their LPARs are activated.

Network virtualization (VFR)

In order to run more than one network instance within the layer 3 switch in front of the systems, IZB implemented the LAN switch software feature known as Virtual Routing Facility (VRF). This enables the switching hardware to split the routing tables in separate and independent tablespaces. Along with the ability to start more than one routing process concurrently, each tablespace got full independent routing capabilities. This virtualization offered great flexibility and shortened network infrastructure delivery time for upcoming projects or new client installations.

March 2005: System virtualization (z/VM)

After installing z/VM 5.1, IZB moved the sysproc and test Parallel Sysplex over to VM as guest systems. In addition, the z/OS systems SYT1 through SYT3 and the Coupling Facility were moved to z/VM. From the networking perspective, the virtualization was accomplished by OSA adapters. The OSA feature delivered shared access. Each VM guest received dedicated connects through z/VM onto the OSA Express feature.

z/VM system access was installed through direct console access using either 2074 controllers or the z/VM TCPIP guest function. To participate in the layer 3 LAN switch network infrastructure, the z/VM guest OMROUTE delivers the necessary OSPF dynamic routing function. This network authority is implemented as an independent demilitarized management zone (DMZ) with access for authorized administrators only.

3.5 Technical migration conclusions

z/VM installation before LPAR growth

Before the LPAR growth in 2002, IZB considered the implementation of VM guests. The former VM version did not provide necessary functions such as crypto support or implementation of internal coupling facilities. A test in 2004 with z/VM V4.4 yielded the same result. However, z/VM 5.1 delivered the expected virtualization. In the future, all upcoming test and engineering systems will be installed as z/VM guests.

HiperSockets

HiperSockets™ have not been implemented. The chosen LPAR distribution by IZB on z/900 CPCs is optimized for WLC pricing and tuned for optimal CPC redundancy. Corresponding systems within a common sysplex are distributed to separate CPCs. Also, the production systems are spread over all available CPCs.

However, HiperSockets provided by zSeries microcode are high speed communication paths between LPARs within a CPC only. Because of today's LPAR distribution across CPCs, there is no useful scenario that will benefit by implementing HiperSockets.

OMPROUTE tuning

For virtual IP addressing, correct routing updates are essential. IP Stack VIPA takeovers, processes, and sysplex distributors depend on a working OMPROUTE dynamic routing process. To reduce the time gap for inconsistent tables at IP Stack and OMPROUTE startup, some timers can be tuned.

Default values for “Hello” and “dead interval” are 10 and 40 seconds. Specifying the OMPROUTE environment variable OMPROUTE_OPTIONS=hello_hi and reducing the timers on all participating OSPF interfaces at OMPROUTE and corresponding routers interfaces to 2 and 10 seconds limits the time gap from 40 seconds to less than 10 seconds. This solves problems at IP stack start with immediate dynamic VIPA takeover. Restarting OMPROUTE at normal IP stack production will not interrupt established sessions.

CIP router limitations in large backup scenarios

Channel-attached routers with attachment to multiple LPARs require EMIF configuration entries for each system location. EMIF addressing consists of LPAR numbers, ESCON director port numbers, and control unit addresses. To cover normal operations plus all possible disaster cases, the IZB configuration matrix includes 30 entries (LPARs) for normal operations multiplied by all possible backup destinations. The selectable LPAR ID is limited between 0x0-0xF, and the whole Identifier with LPAR ID, ESCON Director port number, and CUAD must be unique. There is also a maximal limit on predefined EMIF paths in a channel interface processor (CIP) of 64 entries.

Support systems

Centralizing the network access inside a Parallel Sysplex on two members (support systems) simplifies network setup, network operations, and troubleshooting. Moving “housekeeping” software such as session managers and file transfer products to this system combines network entry and applications inside support systems without traversing XCF links. The IZB support systems reduced complexity and made available more resources for production LPARs. This improved and stabilized the overall production process significantly.

VTAM IP stack bind limitations

VTAM can be bound to a single IP stack with the VTAM startup options. Binding VTAM to more than one IP stack is not possible. This limits the use of Enterprise Extender functions per system to one IP stack authority.

Multiple IP stacks

Operating multiple IP stacks simplified the connection setup to networks with different security levels or accompanying individual clients within a system. Splitting up communication on different IP stacks limits the production impact in the case of an IP stack recycle or outage.

If more than one IP stack uses XCF communication, there currently is no way to partition the IP stacks on XCF interfaces. To separate IP stacks from each other by using common XCF network transport, individual IP subnets must be defined on each stack at dynamic XCF transport. Additional filter lists prevent IP stack inter-communications across XCF.

3.6 SNA/APPN network

The challenges previously described resulted in a mainframe backbone redesign. This section discusses the structural changes that were implemented, starting from eight systems in 1999 up to the scalable design of today. IZB’s APPN backbone implementation delivers reliable transportation for Parallel Sysplex functions such as VTAM generic resources.

3.6.1 The starting point

The previous APPN network layout was based on eight monoplex systems. Only four production systems (SYPE, SYPG, SYPF, and SYPDSYPD) were connected in a redundant manner. Test and engineering systems used one of the production systems for network interconnection. All systems participated in the APPN network with a common NETID.

The individual APPN node types were selected to fill needed roles: endnode or networknode, with or without SNA subarea support. Each production system acted as a network/interchange node to handle FEP subarea connections. Others acted as APPN network or end nodes. Figure 3-1 shows the network layout in 1999.

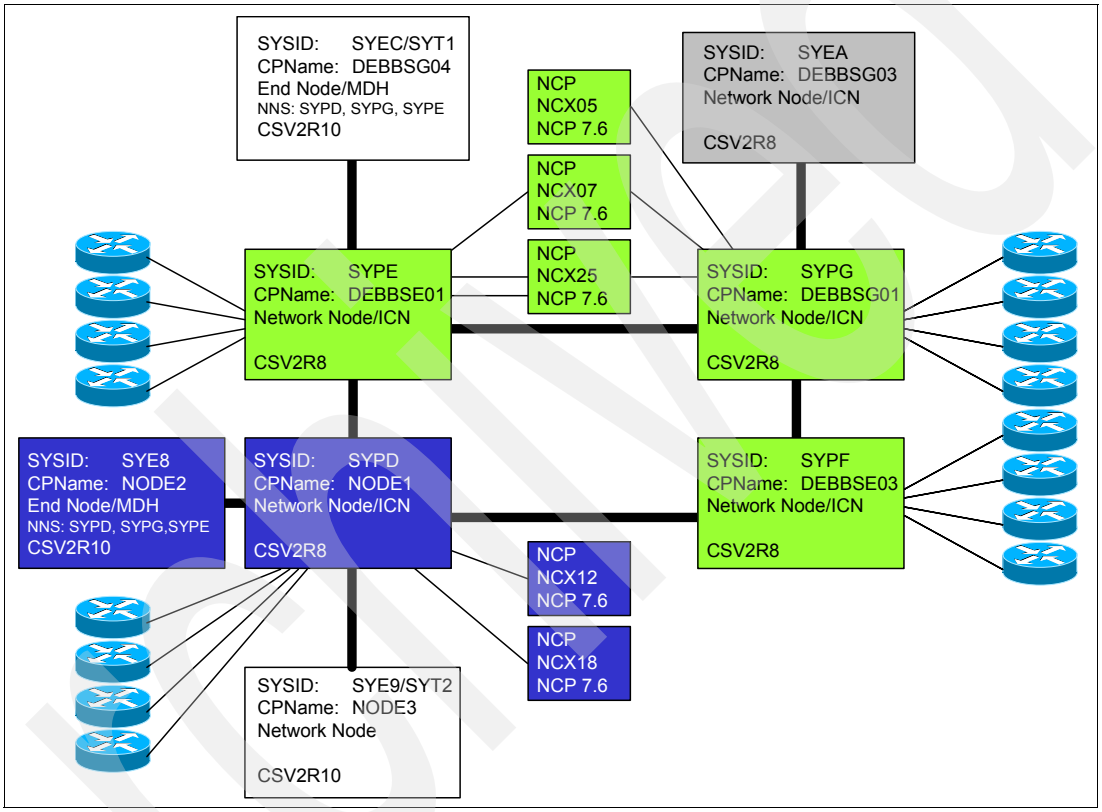


Figure 3-1 Former network layout (1999)

Channel attached Routers (CIP) and FEP/NCPs provided external network access.

3.6.2 Considerations

IZB designed an efficient, redundant, and scalable APPN backbone to minimize hops between APPN nodes and APPN pipes on multiple communications links. Border nodes on different Parallel Sysplexes guarantee highly available network connections to other APPN networks within (or outside of) IZB's processing center.

Redundancy

The APPN link design reduced the hop counts between all systems to two or less. Each link includes four multipath channels (MPCs) per APPN transmission group (TG). Equal paths provide symmetrical links for load balancing within the MPCs in a TG, and also between the redundant system-to-system interconnection TGs.

Border node functionality is implemented on selected systems in different Parallel Sysplexes, which are spread over multiple CPCs (SYPD,SYP4, and SYP5). These border nodes are used to interconnect APPN networks among hosted clients at IZB without changing their NETIDs.

Two extra border node systems (SYP6 and SYP8) are used for external APPN/EE connections. SYP6 and SYP8 are running a dedicated VTAM IP stack belonging to a separate demilitarized zone (DMZ) for external IP Enterprise Extender links. All external APPN partner networks run EE links to both border nodes (SYP6 and SYP8) simultaneously.

Availability

To avoid dependencies between IP and SNA, Enterprise Extender is not used for internal system-to-system communication. Instead, multiple autonomic MPC connections are selected. To avoid single points of failure, each MPC has at least two different addresses, for read and write. These I/O address pairs are served by FICON cascaded switches. Outages in the ESCON/FICON network infrastructure affect performance, but do not interrupt all MPCs and TGs.

Security

Selective access is granted by predefined cross-domain resources and the use of session management exits.

3.6.3 Current design: Scalable SNA/APPN backbone

Within a Parallel Sysplex, all communication is provided by dynamic XCF links. Sysplexes containing more than two systems centralize all communications outside of their own plex on Support Systems such as SYP4 or SYP5. Inter-sysplex communication is done through multi-path channels (MPCs).

SYP4/5 and SYPD belong to different sysplexes and act as star points for all MPC connections. Each system is reachable within two hops. Three parallel MPC connections prevent single points of failure. If all other points are down, one remaining star point can provide the whole backbone interconnection.

The APPN border nodes for external Enterprise Extender connectivity are located in different Parallel Sysplexes. These VTAM nodes are connected by dedicated IP stacks to a demilitarized zone. Through a secured IP network infrastructure, external business partners have redundant Enterprise Extender links to both border nodes.

Figure 3-2 on page 52 displays the scalable SNA/APPN backbone design.

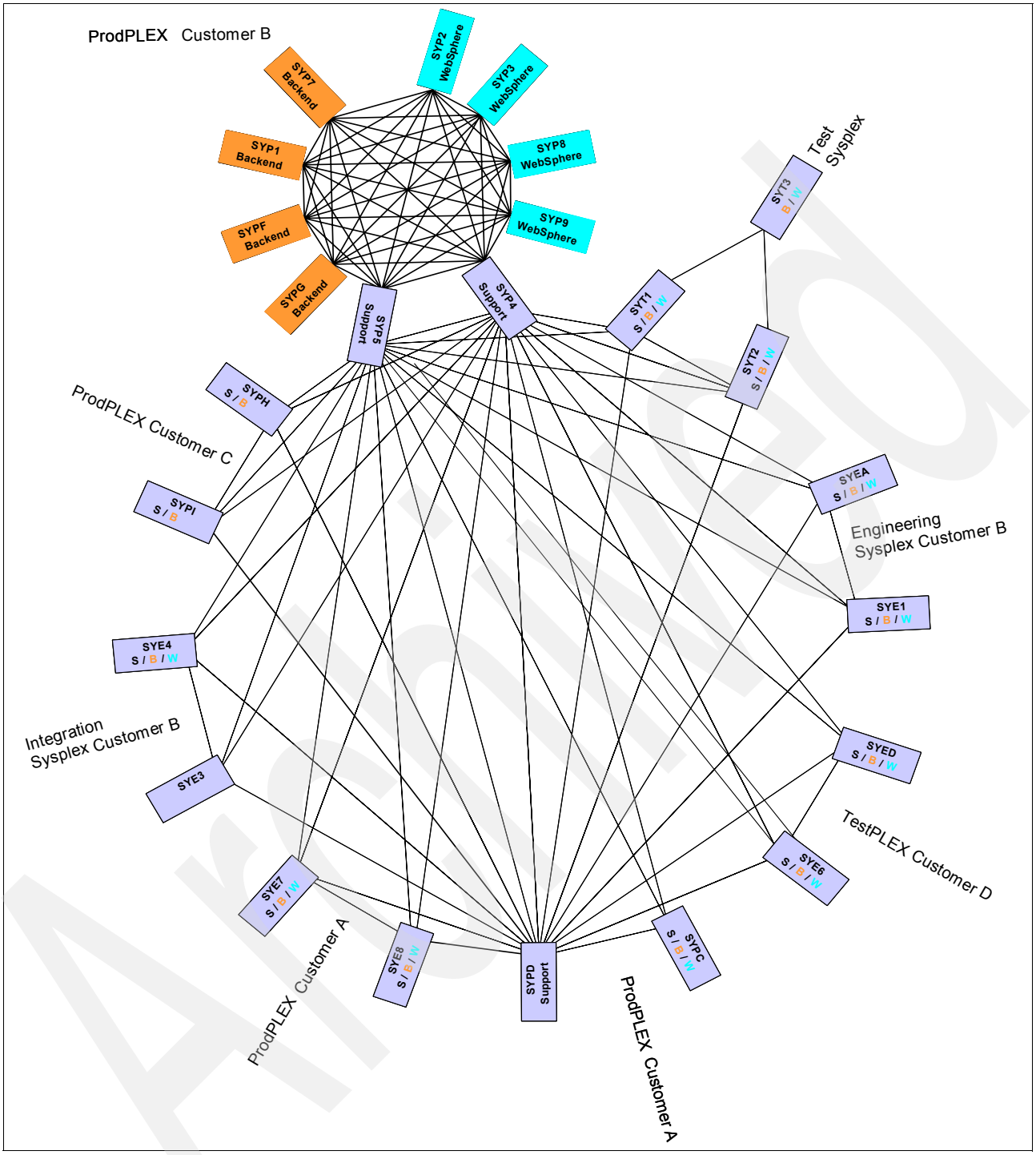


Figure 3-2 Scalable SNA/APPN backbone design

3.6.4 Implementation project

The IZB hardware department developed a CTC matrix for all the MPCs that are illustrated in Figure 3-2. This was generated using redundant paths through the ESCON/FICON infrastructure. All I/O addresses were included in the corresponding IODF systems configuration.

The following tasks were carried out by the Communications Server planning group:

- Generating CTC address network matrix and IODF (hardware department)
- Changing APPN node type
- Expanding VTAM TRL MPC definitions
- Defining APPN local major nodes
- Activating all MPCs in addition to existing network connections
- Deactivating former network links

Changing the VTAM APPN node type required a network restart or IPL. All other tasks were done dynamically. Because of the need to schedule a NET restart, the new backbone implementation was completed in four weeks.

MPC sample configuration

Example 3-1 illustrates an MPC sample definition. The address ranges E500-E503 and E510-E513 are used for normal operation. If a backup condition occurs, the system is started in the alternate processing center and address ranges E508-E50B and E518-E51B are activated.

Hardware failures in the ESCON/FICON infrastructure should interrupt no more than one of the address ranges, E50x or E51x. At least two of the four MPCs would remain.

Example 3-1 MPC sample definition

```
VBUILD TYPE=TRL
MPC1P5   TRLE  LNCTL=MPC,                                    X
               READ=(E501,E509),                             X
               WRITE=(E500,E508),                            X
               MAXBFRU=16
*
MPC2P5   TRLE  LNCTL=MPC,                                    X
               READ=(E503,E50B),                             X
               WRITE=(E502,E50A),                            X
               MAXBFRU=16
*
MPC3P5   TRLE  LNCTL=MPC,                                    X
               READ=(E511,E519),                             X
               WRITE=(E510,E518),                            X
               MAXBFRU=16
*
MPC4P5   TRLE  LNCTL=MPC,                                    X
               READ=(E513,E51B),                             X
               WRITE=(E512,E51A),                            X
               MAXBFRU=16
*
```

For every MPC, a local major node definition is required; see Example 3-2.

Example 3-2 LOCAL MAJOR NODE definition for APPN connections

```
VBUILD TYPE=LOCAL
*
APU1P5   PU    TRLE=MPC1P5,                                  X
               XID=YES,                                      X
               DYNLU=YES,                                    X
               CPCP=YES,                                     X
               TGP=MPC,                                      X
               CONNTYPE=APPN
```

3.6.5 Lessons learned

In order to monitor and restart more than 50 MPC connections, IZB implemented an automated CLISTin NetView®. This tracks the operation of the network and discovers MPC hangups in order to keep the full redundant backbone up and running. Service and maintenance on single links can be done during normal business hours.

Routes within the highly redundant APPN backbone are selected dynamically. From an operational point of view, this provides IZB with a very stable and highly available network. For troubleshooting or diagnostics, IZB can modify COS values or block link stations to get a more predictable route selection. But as yet there have been no major problems which would required disabling the dynamics.

3.7 IP network

The IP network devices and functions at the front of the mainframe systems are important in providing all expected functions for Parallel Sysplex networking. Well-designed dynamic IP routing is vital for implementing dynamic VIPA support for sysplex distributor services.

To this end, in 2002 IZB began an internal project known as “Switched LAN access to centralized hosts”. Team members from three groups (network planning, network operations, and mainframe planning) built a common layer 3 network switching platform to connect mainframes to internal and external customer IP networks.

The project helped the groups to develop a common understanding, then defined detailed planning and operation responsibilities for networking and mainframe engineers. Today all three departments are able to extend or change their environment without holding large and involved meetings on change control.

3.7.1 Starting point

IZB began implementing Internet Protocol (IP) on MVS systems using a single IP stack per system, addressed over physical interfaces and static routing entries. Selected CIP routers also provided IP communication links to outside networks. EMIF was used to share the ESCON CIP Router interfaces for IP transport using the Channel Link Access Workstation (CLAW) protocol.

To guard against all upcoming disaster scenarios, the CIP router and IP stack configurations changed from static network definitions to virtual IP addressing (VIPA) and dynamic RIPV2 or OSPF routing. But attempting to cover all possible disaster situations resulted a very large number of predefined EMIF definitions among all possible system locations (ESCON directors) and all participating router interfaces.

Limitations in Cisco CIP maximum EMIF path definitions lowered CIP IP throughput. This, together with massive EMIF configuration overhead, called for CIP replacement. IZB determined that OSA Express Gigabit Ethernet features delivered the next step in IP connectivity.

3.7.2 Considerations

Building a network structure for mainframes with highly virtualized servers (LPARs and z/VM guests) requires great flexibility in network components and design. Before proceeding, IZB examined various factors such as systems growth, expanded backup capabilities, increased application availability, maintenance without service interruption, exponential growth in IP

communication, reduction in complexity, and costs. IZB then initiated the “Switched LAN Access to centralized Hosts” projects. Modular Cisco IOS Switches provided the expected interface types and counts, switching throughput, software functionality, and virtualization levels. The next section describes how this interacted with z/OS Communications Server facilities.

Dynamic routing

For dynamic routing, z/OS Communications Server (CS) and Cisco provide a common link state routing protocol known as Open Shorted Path First or OSPF. OSPF routing is a very fast converting protocol. Like APPN, each node computes its own topology table for its network area. A change in the network, such as a network link failure, a moving dynamic VIPA, or a Sysplex Distributor switched to another node, is recognized immediately.

Additional tunable routing protocol timers on z/OS CS and Cisco’s routing processes provide a consistent routing table within two to 10 seconds. Established IP sessions are not affected by rerouting of dynamic sysplex resources at all. To keep the z/OS OMPROUTE processing time for OSPF at a minimal level, an area numbering requirement and the proper area operation mode (total stub area) are required.

Multi-site operation

Both processing centers provide an identical networking infrastructure. Virtual network implementations are mirrored at both locations. If a system is moved to its backup position, the OSPF neighbor selection will discover the Cisco switches and all other LPARs belonging to its network domain. The system can now propagate its VIPAs and owning Sysplex Distributors, and become automatically reachable to hosts and clients in its network.

IP network redundancy

z/OS CS supports load balancing between multiple gateways or default gateways. In a redundant network layout, the use of symmetrical links and link costs allows the traffic to be balanced between all available links. A link outage will reroute the IP sessions to the remaining links without interruption. This allows IZB to provide service and maintenance on links, switches, and IP stack interfaces during business hours.

Multiple IP stack design

Separation of networks with different security levels is achievable by implementing the z/OS CINET filesystem and multiple autonomous IP stacks. Each IP stack operates its own OMPROUTE OSPF process. Virtual networking connections through OSA port sharing or virtual LAN (VLAN) implementations, together with the virtual routing facility (VRF) at the Cisco switches, provide full virtualized networks which are independent from each IP stack’s security level.

Security considerations

So far, only security on z/OS CS and the networking infrastructure have been discussed. To secure IP connectivity, z/OS CS provides a security access facility (SAF) interface. SAF class Servauth is a user-based access control for network resources. Permits can be granted for IP stack, IP port, IP network, or terminal accesses.

3.7.3 The final design: Scalable IP backbone

Figure 3-3 on page 56 illustrates the IP and OSPF routing implementation of clients inside IZB’s virtualized mainframe network. The virtual routing facility is identified by the client’s name (yy) and a free OSPF Area ID xx. All VLAN numbers also contain the client’s OSPF Area ID.

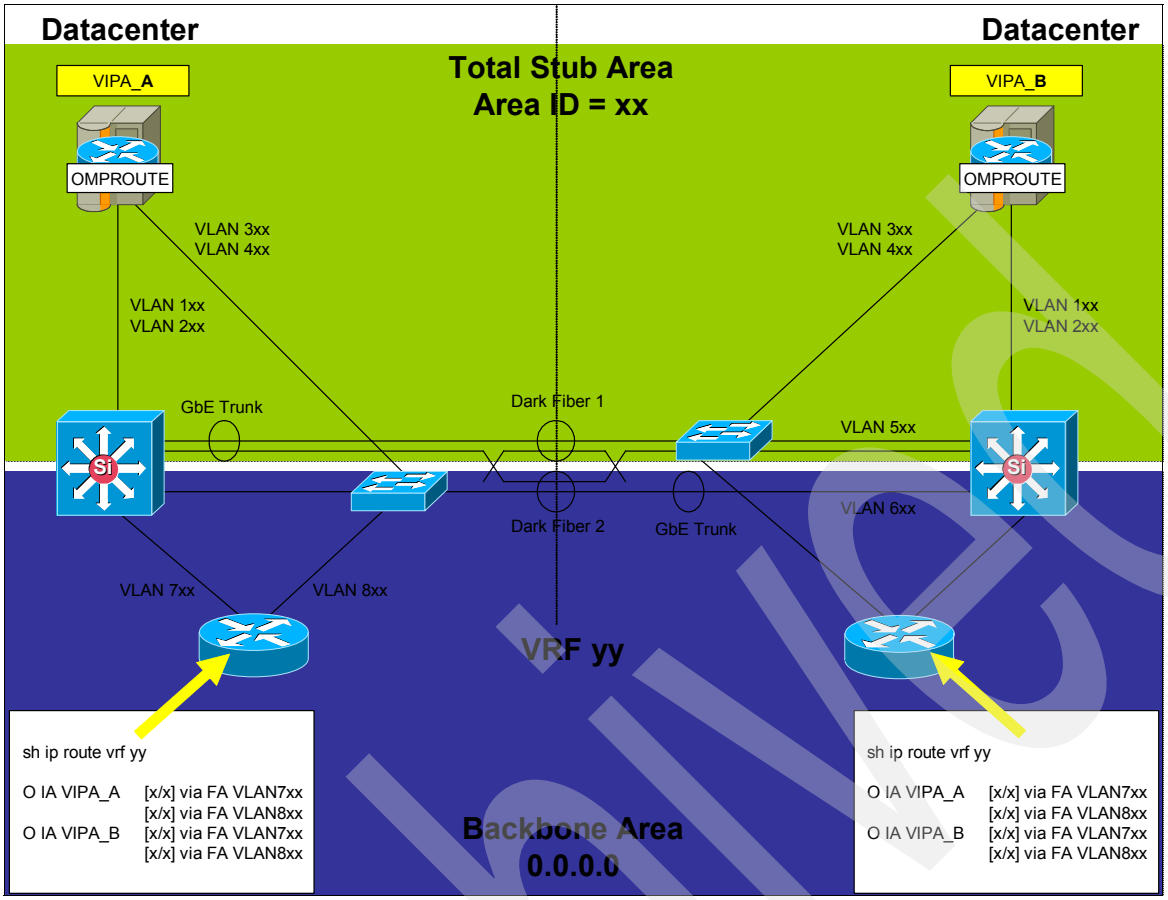


Figure 3-3 Scalable IP backbone

As shown, the z/OS CS IP stack is connected to the Total Stub Area VLANs 1xx and 3xx. One VLAN is terminated at the local layer 3 switch. The other VLAN uses a layer 2 switch to reach the layer 3 switch at the opposite datacenter.

If OMPROUTE is started and there is a match on Total Stub Area ID and a predefined security string, the OSPF process will automatically establish a neighborhood to both layer 3 switches. The layer 2 switch acts just as a port concentrator to save dark fibers between both centers. Layer 2 switches are invisible to OSPF.

Inside the Total Stub Area, all OMPROUTE processes receive only area internal routing information. Additionally, the default routes from the backbone area are propagated by both OSA Express interfaces. All routes show equal costs and the IP stack will balance outgoing traffic between the OSA ports for both data centers.

OMPROUTE also propagates its VIPAs, dynamic VIPAs, or responsible Sysplex Distributor IP addresses to the layer 3 switches. The clients' own networks are connected at routers in the backbone area, and can access VIPAs by automatically balancing traffic on multiple links with equal costs.

To avoid a single point of failure or prevent OSPF from a suboptimal routing, both dark fiber interconnections include a backbone and total stub area VLAN interconnection. Each network device or link can be deactivated without losing mainframe connectivity.

3.7.4 Implementation project

IZB's network implementation project started in July 2002. A detailed test plan covered Cisco's IOS software and z/OS CS interoperability and all functions, plus backup disaster case testing. The test phase was passed at the end of August. The first pilot clients were migrated in September 2002.

For each migration, the new infrastructure was implemented in parallel with the existing network connections. Through simple IP rerouting, traffic was moved from CIP to OSA Express interfaces. The migration project was successfully completed by the end of December 2002.

3.7.5 Lessons learned

Switched LAN access to centralized hosts delivered excellent flexibility in planning and operation to network or mainframe engineers. New virtualized customer networks could be set up within hours. Also, maintenance is possible during normal working hours.

Modifying a running OMPROUTE configuration by operator command **`/f mproute, reconfig`** is not applicable for all configuration changes. Just a restart of OMPROUTE executes all configuration statements the right way.

Due to the tuned OSPF timers (see "OMPROUTE tuning" on page 49), OMPROUTE restarts can be done during full production. This results in a short freeze (up to 12 seconds), but all communication resumes without any loss of connectivity.

3.8 Load distribution and failover techniques

Load balancing or automatic failover can be implemented at different stages. They can start at external load balancing devices located at the network site, and progress up to communication server functions such as Parallel Sysplex dynamic virtual IP addressing (DVIPA), Sysplex Distributor (SD), or VTAM Generic Resource (VGR).

External Load Balancers (LB) represent a virtual server entity. They distribute the workload rather intelligently, directly to the associated real target systems. Connectivity to the target systems can be implemented by methods like MAC address-based forwarding or using Network Address Translation (NAT).

The network infrastructure needed to operate external LB devices differs in comparison to system internal VIPA or SD. DVIPA and SD require additional dynamic routing. In order to enjoy all the benefits of SD, a proper architecture, with multiple distributed and target IP stacks, is required.

3.8.1 External load balancing solutions without Parallel Sysplex awareness

External LB devices are normally planned and operated by network administrators across multiple server platforms. These devices treat systems belonging to a Parallel Sysplex like any other server system or server cluster.

The balancing process is not aware of current and changing workload conditions inside a Parallel Sysplex. Other than that, external balancing solutions can support functions beyond the scope of Sysplex Distributor. Content-based load balancing on URL, session ID, and cookies are additional features of external LBs. For secure SSL sessions, this requires decryption of SSL at or in front of the load-balancing device prior to the content inspection.

IZB used external Alteon Load balancers for WebSphere Application Server V3.5 failover; refer to 7.4.1, “How it works under WebSphere Application Server V3.5 on MVS” on page 159 for information about this topic.

3.8.2 External network load balancing awareness with Parallel Sysplex

Sysplex Distributor and Cisco multi-node load balancing (MNLB)

Sysplex Distributor (SD) and Cisco multi-node load balancing (MNLB) was the first available sysplex-aware load balancing solution. SD and MNL use a Cisco proprietary protocol for dynamic feedback (DFP) between external Cisco Load Balancer and a Service Manager task located in the z/OS Communications Server. For details about the Cisco MNLB approach, refer to section 5.5 of IBM Redbook *Networking with z/OS and Cisco Routers: An Interoperability Guide*, SG24-6297.

z/OS load balancing advisor for z/OS Communications Server

The z/OS Load Balancing Advisor is an application which communicates with external load balancers and one or more z/OS systems Load Balancing Agents using a vendor-independent SASP protocol.

The purpose of the Advisor and Agents is to provide information to a load balancer about the availability of various resources/applications and their relative ability to handle additional workloads with respect to other resources that have the ability to handle the same workload.

The outboard load balancer will take data that the Advisor passes to it and make a determination about where to route new workloads. This load balancing solution is different from existing load balancing solutions such as Sysplex Distributor and CISCO Multi-node Load Balancing (MNLB) because in this implementation, the actual decision of where to route work is made outside of the sysplex.

3.8.3 Sysplex Distributor

Load balancing for IP applications was introduced at IZB soon after the establishment of the supporter systems in 2002. The Supporter Systems concentrate all external network connections on their IP stacks, so they are ideal candidates for holding the Sysplex Distributor role Distribution Stack. SD Target Stack roles are located in the back-end systems SYPG, SYPF, SYPE, and SYP7. As mentioned in 3.7.3, “The final design: Scalable IP backbone” on page 55, all systems participate in dynamic OSPF routing.

Due to the fulfilled requirements for implementing Sysplex Distributor, IZB started balancing TN3270 access. For guidance during in the configuration process, IZB used the IBM Redbook *TCP/IP in a Sysplex*, SG24-5235, and *CS IP Configuration Guide,* SC31-8725.

Figure 3-4 on page 59 illustrates the first Sysplex Distributor implementation distributing TN3270 connections to back-end systems.

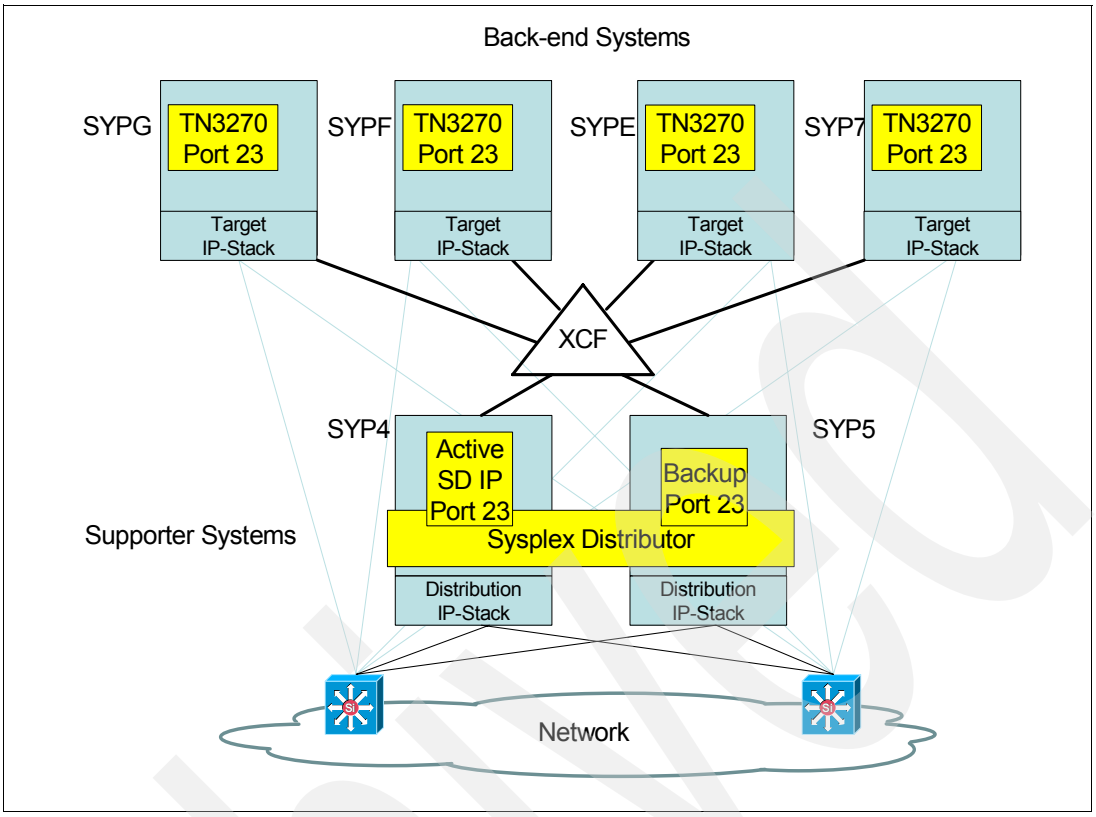


Figure 3-4 Sysplex Distributor for back-end application access

Sysplex Distributor configuration steps

The steps to configure Sysplex Distributor were as follows:

1. Activate XCF (VTAM Startoption XCFINIT=YES).
2. Assign a network address to XCF on each distribution and target IP stack.
3. Include Sysplex Distributor IP Network to OMPROUTE dynamic OSPF routing on all distribution and target IP stacks.
4. Add active and backup Sysplex Distributor definitions to the supporter systems' TCP/IP profile.

Figure 3-4 shows Sysplex Distributor for back-end application access.

Today, more than 20 application Sysplex Distributor IP Addresses at SYP4 and SYP5 provide a WLM-balanced access to back-end applications; read 5.1, "The Customer B DB2 configuration - then and now" on page 94 to learn more about this topic.

Approximately half of the Sysplex Distributor IP addresses are active on SYP4. SYP5 acts as a backup distribution stack to SYP4, and vice versa. Maintenance or outages on a supporter system do not affect the back-end communication. All back-end application sessions traversing a distribution stack can be taken over automatically without any interruption to the remaining support system distribution stack.

The client IP session points to a Sysplex Distributor address. This address is active on SYP4 or SYP5. A new session request reaches the responsible supporter system and is checked against the VIPA Dynamic Port Table.

Depending on the selected distribution method (round robin or WLM), a back-end system with the active application is selected. The distribution stack holds, for each active session, the back-end affinities in its virtual connection routing table. The packet is passed over the XCF network to the back-end systems application.

When routing back to the client, the back-end system can directly access the network switches, and the packet bypasses XCF and the corresponding distribution stack.

Sysplex Distributor balanced access to WebSphere V5

A more sophisticated design was chosen for the WebSphere V5 implementation. Figure 3-5 shows the Sysplex Distributor layout for WebSphere V5.

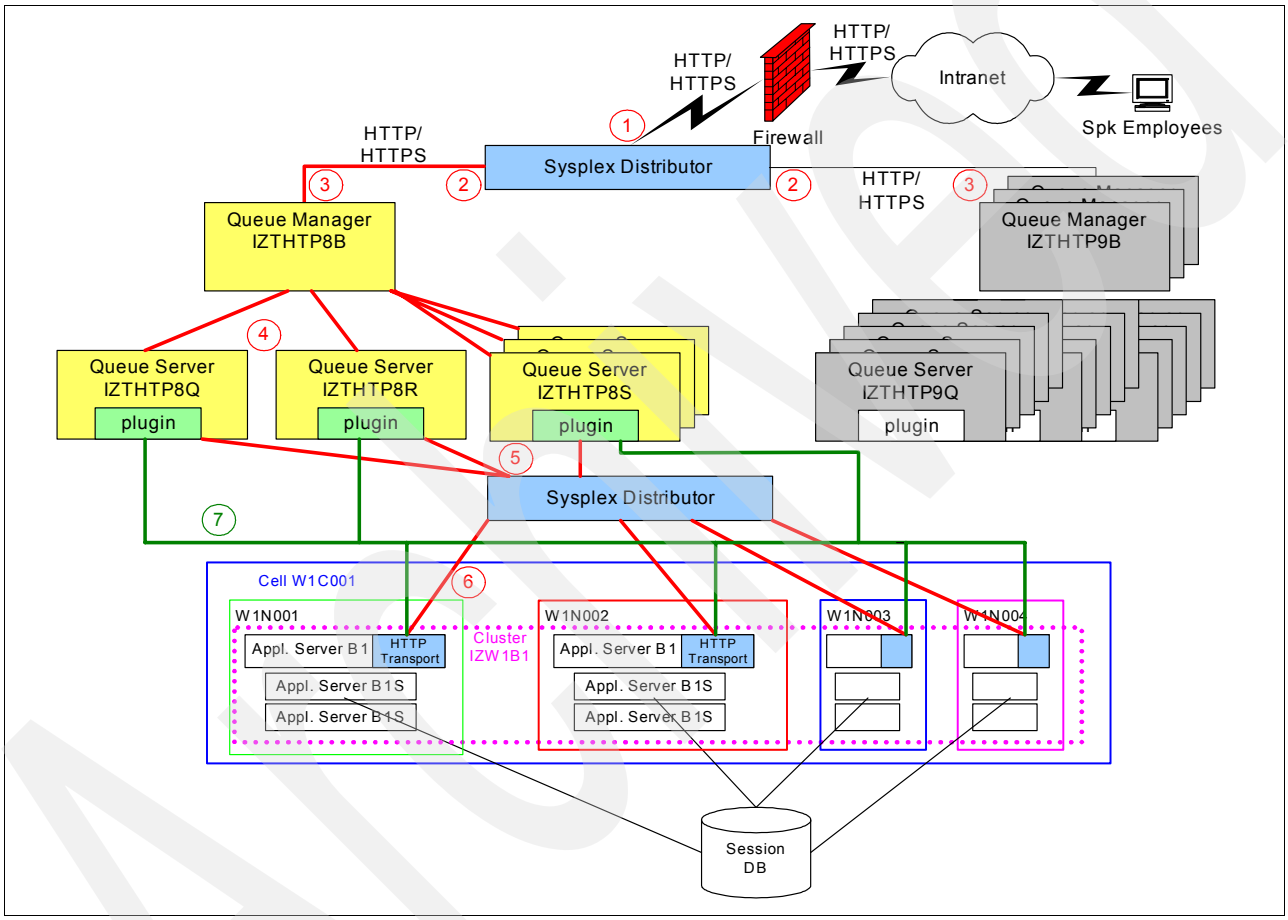


Figure 3-5 Valid system configuration - provides high availability to IZB clients even with load fluctuations

Two Sysplex Distributor instances provide load balancing at different levels. A client request addresses the first Sysplex Distributor. One of the four systems is the active Distribution Stack for SD1, and the network forwards the request to it. All others are backup systems to take over in case the active SD1 fails.

Based on the VIPA distribution port list, WLM might route the request for the active SD1 path over to an available HTTP Server Queue Manager. It could be the HTTP Queue Manager at the same system, or one of the three other target systems. The included plug-in forwards a stateless request to the next Sysplex Distributor level for final balancing by the WebSphere Application Server. A transaction on a back-end system can occur. The result takes the direct way back to the client.

Figure 3-6 illustrates an example of Sysplex Distributor session routing.

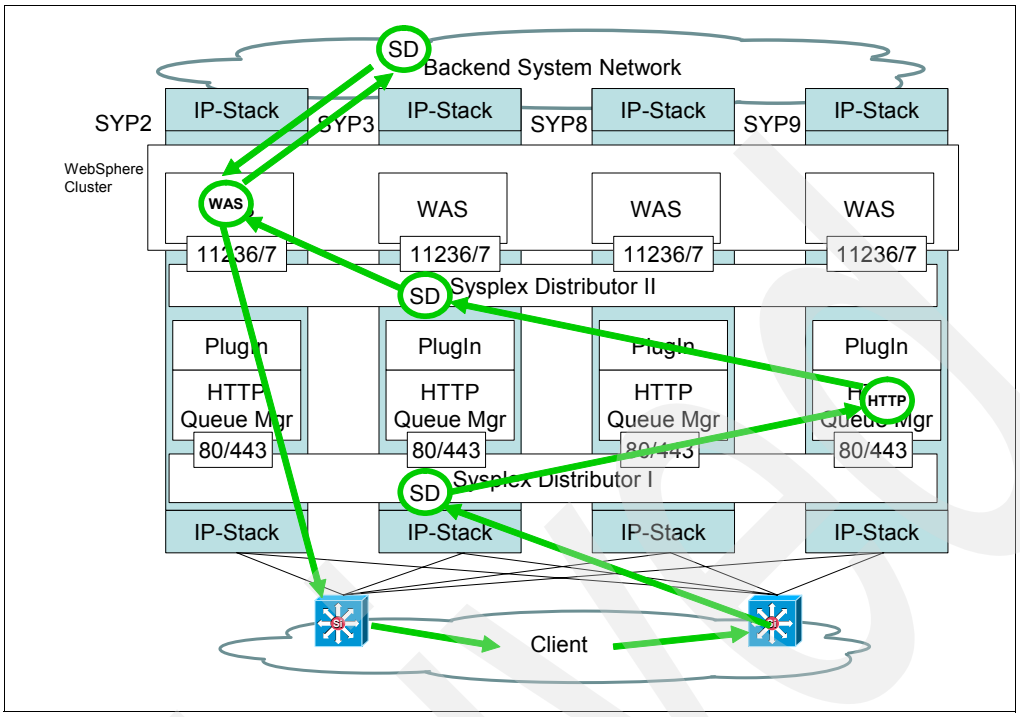


Figure 3-6 Sysplex Distributor session routing

For more information about this topic, refer to 7.4.1, “How it works under WebSphere Application Server V3.5 on MVS” on page 159.

3.8.4 Dynamic VIPA

Almost all applications are implemented using Sysplex Distributor. However, in rare situations, the use of application-specific dynamic VIPAs is preferred. For instance, a WTO started task is started on different systems in a sysplex for high availability. Only one of those STCs should be the collector of all WTO requests. If the active WTO STC fails, the dynamic VIPA is handed over to one of the remaining WTO STCs.

Dynamic VIPA configuration steps

(Note that steps 1, 2, and 3 are identical to Sysplex Distributor configuration.)

1. Activate XCF (VTAM Startoption XCFINIT=YES).
2. Assign a network address to XCF on each distribution and target IP stack.
3. Include dynamic VIPA IP Network to OMPROUTE dynamic OSPF routing on all distribution and target IP stacks.
4. Add active and backup dynamic VIPA definitions to the supporter systems’ TCP/IP profile, as shown in Example 3-3 and Example 3-4 on page 62.

Example 3-3 Sample TCP/IP profile definitions at active DVIPA supporter system

```
VIPADynamic
VIPADEFine move whenidle 255.255.255.0 172.20.143.10
VIPARange DEFINE MOVE NONDISRUPTIVE 255.255.255.0 172.20.143.0
ENDVIPADynamic
Port
```

```
3002 TCP ISTWTO5S BIND 172.20.143.10 ; WTO STC
```

Example 3-4 Sample TCP/IP Profile definitions at backup DVIPA supporter system

```
VIPADynamic
VIPABackup 100 172.20.143.10
ENDVIPADynamic
Port
3002 TCP ISTWTO5S BIND 172.20.143.10 ; WTO STC
```

3.8.5 VTAM generic resources

IZB uses VTAM generic resources (VGRs) to provide a single VTAM application control block (ACB) for multiple application instances. From a VTAM point of view, the implementation of VGRs is limited to setting up a Coupling Facility structure (default ISTGENERIC), as shown in Example 3-5. To complete the VGR setup, IZB ensured that the VTAM start option STRGR=ISTGENERIC pointed to the default STRNAME ISTGENERIC.

Example 3-5 D XCF,STRUCTURE,STRNAME=ISTGENERIC

```
IXC360I  19.26.43  DISPLAY XCF 482
STRNAME: ISTGENERIC
 STATUS: ALLOCATED
 TYPE: SERIALIZED LIST
 POLICY INFORMATION:
  POLICY SIZE    : 96000 K
  POLICY INITSIZE: 60000 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : DISABLED
  PREFERENCE LIST: CFNH1    CFNG1
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 01/22/2006 03:34:25
  CFNAME         : CFNH1
  COUPLING FACILITY: 002084.IBM.83.00000007C27F
                     PARTITION: 01   CPCID: 00
  ACTUAL SIZE    : 63744 K
  STORAGE INCREMENT SIZE: 256 K
  ENTRIES:  IN-USE:      11914 TOTAL:     166753,   7% FULL
  ELEMENTS: IN-USE:         16 TOTAL:       9809,   0% FULL
  LOCKS:     TOTAL:          4
  PHYSICAL VERSION: BE3F9A30 E4FAA54E
  LOGICAL  VERSION: BE3F9A30 E4FAA54E
  SYSTEM-MANAGED PROCESS LEVEL: 8
  XCF GRPNAME    : IXCLO008
  DISPOSITION    : DELETE
  ACCESS TIME    : 1800
  MAX CONNECTIONS: 12
  # CONNECTIONS  : 10

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 DEBBSMON_SYP2    05 00050044 SYP2     NET      0029 ACTIVE
```

```
IXC360I  19.26.43  DISPLAY XCF 482
STRNAME: ISTGENERIC
  DISPOSITION    : DELETE
  ACCESS TIME    : 1800
  MAX CONNECTIONS: 12
  # CONNECTIONS  : 10

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 DEBBSMON_SYP2    05 00050044 SYP2     NET      0029 ACTIVE
 DEBBSMON_SYP3    04 00040056 SYP3     NET      0027 ACTIVE
 DEBBSMON_SYP4    03 0003003F SYP4     NET      002C ACTIVE
 DEBBSMON_SYP5    07 0007001E SYP5     NET      003F ACTIVE
 DEBBSMON_SYP7    09 00090015 SYP7     NET      0028 ACTIVE
 DEBBSMON_SYP8    06 0006003D SYP8     NET      0028 ACTIVE
 DEBBSMON_SYP9    02 00020045 SYP9     NET      003F ACTIVE
 DEBBSMONDEBBSM01 0A 000A0019 SYPG     NET      003A ACTIVE
 DEBBSMONDEBBSP01 01 0001004C SYPE     NET      0027 ACTIVE
 DEBBSMONDEBBSP03 08 0008001D SYPFSYPFSYPD     NET      0027 ACTIVE
```

It is up to the application customization to make use of VGRs. For example, IZB uses VGRs for common access to session managers' TPX running on both supporter systems. Example 3-6 shows session manager IZTPSYP4 located at system SYP4, and IZTPSYP5 at SYP5, requesting the VGR IZTTPIZ.

Example 3-6 Session Manager TPX requests the VGR

```
IXL014I IXLCONN REQUEST FOR STRUCTURE TPXGENERIZTPIZ  112
WAS SUCCESSFUL.  JOBNAME: IZTTPIZ ASID: 004B
CONNECTOR NAME: IZTPSYP4 CFNAME: CFNG1
TPX5555 CA-TPX SYSPLEX SERVICES FOR NODE(IZTPSYP4) are enabled
```

Also CICS, MQSeries® and other file transfer products within the Parallel Sysplex shared DASD environment make use of VGRs; see 4.4.6, "Using VTAM Generic Resource (VGR)" on page 82 for more information. The TOR CICS VGR shown in Example 3-7 provides access to ISTCAT01. It also provides access to ISTCAT02, as shown in Example 3-8.

Example 3-7 CICS VGR (d net,rsclist,id=ISTCAT0,idtype=generic)*

```
NETID    NAME     STATUS     TYPE              MAJNODE
DEBBSMON ISTCAT0# ACT/S      GENERIC RESOURCE  **NA**
```

Both TOR CICSes are located at system SYPGSYPGSYPF.

Example 3-8 TOR CICS VGR ISTCAT0#

```
C CNMP4    DISPLAY NET,ID=ISTCAT0#,SCOPE=ALL
  CNMP4    IST097I  DISPLAY  ACCEPTED
' CNMP4
IST075I  NAME = ISTCAT0#         , TYPE = GENERIC RESOURCE
IST1359I  MEMBER NAME        OWNING CP   SELECTABLE  APPC
IST1360I  DEBBSMON.ISTCAT02  DEBBSM01       YES       NO
IST1360I  DEBBSMON.ISTCAT01  DEBBSM01       YES       NO
IST1393I  GENERIC RESOURCE NAME RESOLUTION EXIT IS ISTEXCGR
IST924I  -------------------------------------------------------------
IST075I  NAME = DEBBSMON.ISTCAT0#, TYPE = DIRECTORY ENTRY
IST1186I  DIRECTORY ENTRY = DYNAMIC    LU
IST1184I  CPNAME = DEBBSMON.DEBBSM01 - NETSRVR = ***NA***
IST484I  SUBAREA = ****NA****
IST1703I  DESIRED LOCATE SIZE =   1K  LAST LOCATE SIZE =   1K
```

```
IST1402I  SRTIMER =    60  SRCOUNT =    10
IST314I  END
```

Other CICS VGRs combine TOR CICS regions on different systems. See ISTCIT7# or ISTCIT0# in Example 3-9. By default, Workload Manager (WLM) will balance the session distribution between both systems. However, the size ratio of system SYPE and SYP7 is approximately 2:1, so WLM favors SYPE.

Example 3-9 TOR CICSes on different systems

```
C CNMP4    DISPLAY NET,ID=ISTCIT7#,SCOPE=ALL
  CNMP4    IST097I  DISPLAY  ACCEPTED
' CNMP4
IST075I  NAME = ISTCIT7#         , TYPE = GENERIC RESOURCE
IST1359I  MEMBER NAME        OWNING CP   SELECTABLE  APPC
IST1360I  DEBBSMON.ISTCIT03  DEBBSP01       YES       NO
IST1360I  DEBBSMON.ISTCIT07  SYP7           YES       NO
IST1393I  GENERIC RESOURCE NAME RESOLUTION EXIT IS ISTEXCGR
IST314I  END
C CNMP4    DISPLAY NET,ID=ISTCIT0#,SCOPE=ALL
  CNMP4    IST097I  DISPLAY  ACCEPTED
' CNMP4
IST075I  NAME = ISTCIT0#         , TYPE = GENERIC RESOURCE
IST1359I  MEMBER NAME        OWNING CP   SELECTABLE  APPC
IST1360I  DEBBSMON.ISTCIT01  DEBBSP01       YES       NO
IST1360I  DEBBSMON.ISTCIT05  SYP7           YES       NO
IST1393I  GENERIC RESOURCE NAME RESOLUTION EXIT IS ISTEXCGR
```

To achieve a round robin session distribution between SYPE and SYP7, the VTAM generic resource Exit ISTEXCGR was customized.

Exit ISTEXCGR

A description of how to customize this exit is provided in 8.5 “Use of Generic Resources Resolution Exit ISTEXCGR”, in the IBM Redbook *SNA in a Parallel Sysplex Environment*, SG24-2113. IZB disabled the WLM flag GRRFWLMX. Example 3-10 shows the ISTEXCGR generic resource exit.

Example 3-10 ISTEXCGR generic resource exit

```
GENRSCRS DS    OH                     BEGIN RESOLUTION
         MVC   GRREXIT,NULL           SET EXIT CHOICE TO NULL
         OI    GRRFLAG1,(GRRFUVX+GRRFNPLA)
*                                     TURN ON CALL GEN.RESOURCE EXIT
*                                     DO NOT PREFER APPL GENERIC RES
* ********************************************************************
*        OI    GRRFLAG1,GRRFWLMX       TURN ON CALL WORK LOAD MANAGER
*                                     BIT AND DEFAULT OFF CALL
*                                     GENERIC RESOURCE EXIT BIT @N1C
* /*******************************************************************
* /*  The following is sample code to turn ON all GRRFLAG1 bits       *
* /*  It can be modified to set the preferred bits ON                 *
* /*  OI    GRRFLAG1,GRRFUVX+GRRFWLMX+GRRFNPLA+GRRFNPLL               *
* /*                                                                  *
* /*  The following is sample code to turn OFF all GRRFLAG1 bits      *
* /*  It can be modified to set the preferred bits OFF                *
* /*  NI    GRRFLAG1,255-(GRRFUVX+GRRFWLMX+GRRFNPLA+GRRFNPLL)    @N1C *
* /*******************************************************************
```

3.9 Conclusions

Implementation of a full dynamic and redundant backbone design for SNA and IP, together with the virtualization of network entities, provided important benefits to IZB such as the following:

- High availability
- Scalability
- Easy implementation of new features
- Central control of operations

As a result, IZB can immediately supply mainframe networking services for new applications or clients.

Part 2

Middleware

This part describes the middleware used by IZB.

4

Migration to CICSPlex

This chapter describes the IZB migration from a single CICS environment to a multi-site CICSPlex on a Parallel Sysplex. It covers the following topics:

- IZB CICS, then and now
- Why IZB changed the environment
- The implementation project
- Conclusions

4.1 Overview

IZB serves as a data processing center for about 80 Bavarian savings banks which are spread over three production systems (SYPG, SYPF, and SYPE). On these three systems there were 28 CICS regions running. One-third of the savings banks were connected to each system.

Today, IZB runs about 65 CICS regions on systems SYPG, SYPF, and SYPE for Customer B, and about 360 CICS regions for all its clients.

The complete migration took a long time. The first conceptual plans concerning CICS and data sharing were drawn up in 2001. A roadmap for migration was validated in 2002, and the actual migration began in 2003 on IZB's test environment. Using several steps, IZB was able to do full data sharing within a CICSPlex on the production systems in 2004. The main work in the CICS environment was to establish a CICSPlex.

While it is not difficult to build a CICSPlex with CPSM on one LPAR, it is challenging to determine all affinities. In 2002, for example, Customer B began to analyze its applications and eliminate affinities, and it took about six months.

Splitting a CICSPlex over more than one system is no problem. The prerequisite is that all data can be shared and is available on all members of the sysplex.

Many companies use features such as function shipping or distributed program link (DPL) within CICS. To do this, connections must be defined between CICS regions. CICS regions can be connected by multi-region operating (MRO) connections (if they are on the same system) or intersystem connections (ISC) (if the CICS regions are on different systems). IBM refers to this as a *CICS complex*.

In this sense, a CICSPlex consists of at least one terminal owning region (TOR), one or more application owning regions (AOR), and perhaps file owning or queue owning regions that are connected with each other. This CICSPlex is managed with IBM software known as CICSPlex System Manager (CPSM).

4.2 CICS in IZB - then and now

Figure 4-1 on page 71 illustrates the CICS environment that existed at IZB in 1999. There were many single CICS regions. Each savings bank was connected to a specific region; these included CICSM1 to CICSM4, CICSN1 to CICSN4, and CICSP1 to CICSP4. CICSPN was the OLTP environment. The distribution of the savings banks over the three systems was almost equal, so about one-third of the transactions ran on each system.

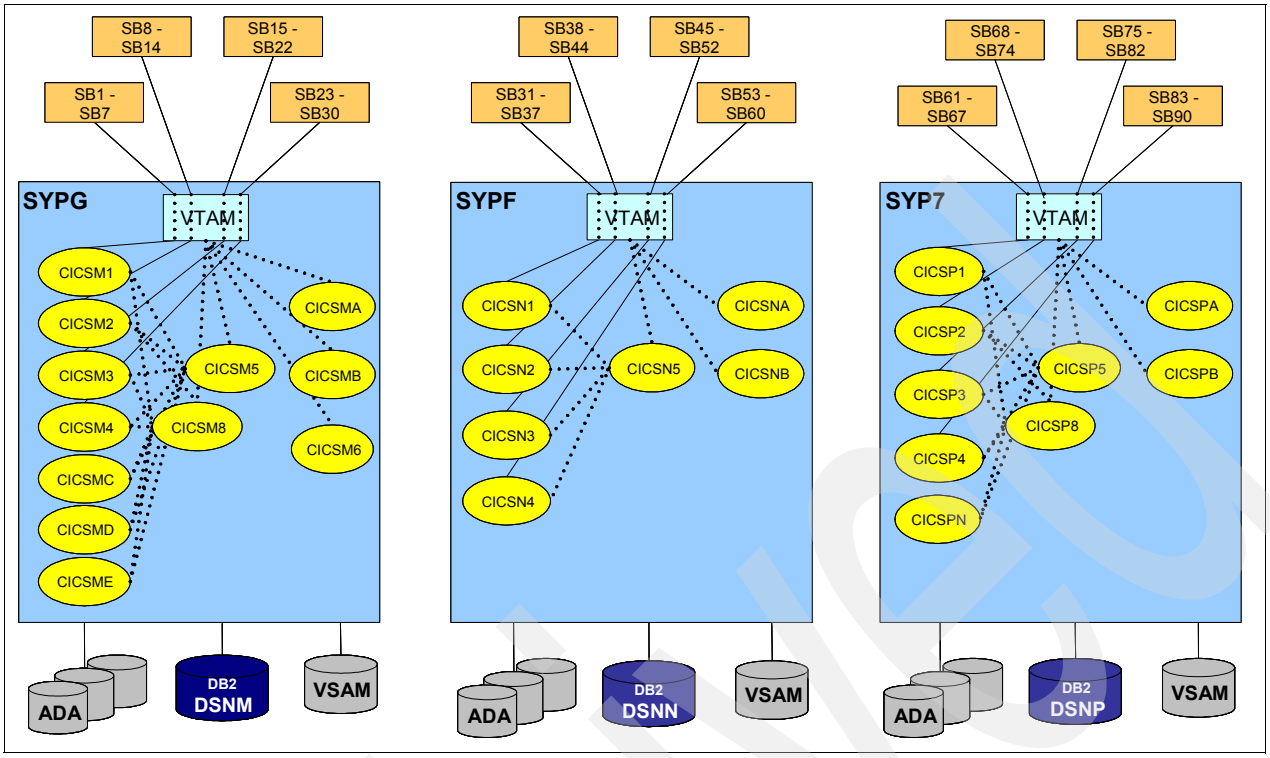


Figure 4-1 IZB's CICS configuration in 1999

System SYPE is where the configuration changed to a multi-site CICSPlex.

CICSP1 through CICSP4h and CICSPN ran the applications of the Bavarian savings banks. A group of savings banks were served by a particular CICS region.

CICSP5 hosted an application known as BTX, which provided the ability to make an inquiry about an account balance and to transfer money at public terminals or using a telephone modem. This service was offered by a telecommunication provider, German Post. The connections were based on X.25 technology. This application was the predecessor of home banking.

All transactions originated at a teller machine and ran in region CICSP8. This region also had the connectivity necessary to do authentication of the client, including checking the PIN or verifying that the account held sufficient funds. There was also an application to log the working time of each employee of the savings banks. This ran in region CICSPB.The applications "MARZIPAN" and "savings bank data warehouse (SDWH)" ran in CICSPA.

The same architecture existed on SYPG and SYPF, with one exception: there was no CICSN8 on SYPF. This meant that all requests for this CICS were routed to CICSP8, and from there to the external partners. This was because one of the partners did not want to have connections to all three IZB systems. All CICS regions were terminal owning, application owning, and resource owning regions in one.

IZB had redundant data sets. All temporary storage queues were either in main storage or auxiliary storage, and they were non-shared.

Figure 4-2 on page 72 illustrates the current CICS environment at IZB.

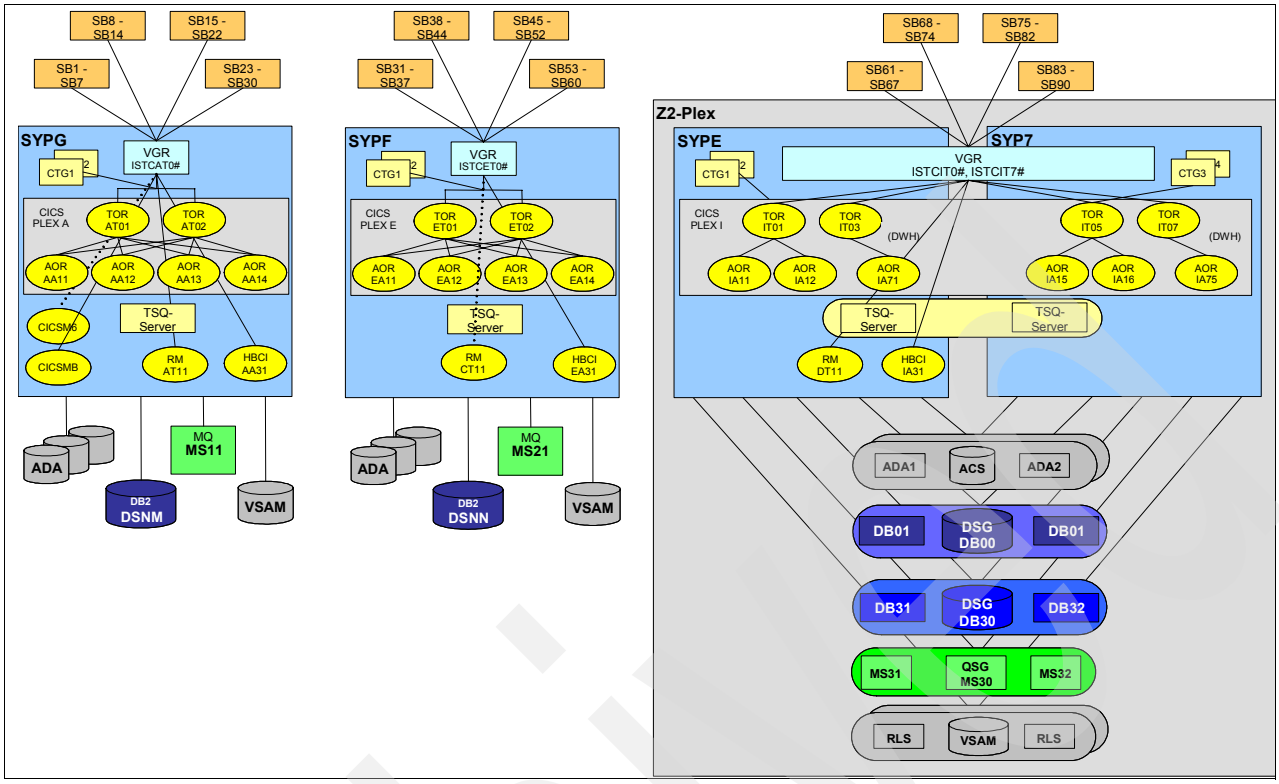


Figure 4-2 IZB's CICS configuration in 2005

Today, IZB has a Parallel Sysplex between systems SYPE and SYP7. There is a production CICSPlex (EYUPLXI in CPSM) which spans both members of the sysplex. ISTCIT01 and ISTCIT05 are the TORs and members of the VTAM Generic Resource (VGR) Group ISTCIT0#. The corresponding AORs are named ISTCIA11, ISTCIA12, ISTCIA15, and ISTCIA16. This group of the CICSPlex represents the former environment of the applications of the savings banks, including some additional applications.

Another group of CICSes consists of ISTCIT03, ISTCIT07 (TORs), and ISTCIA71 and ISTCIA75 (AORs). The S-Data Warehouse application is running in this CICS group.

Formerly, this application ran in CICSMA (SYPG), CICSNA (SYPF), and CICSPA (CICSP1). It was separated into its own CICS regions ISTCAA71 (SYPG), ISTCEA71 (SYPF), and ISTCIA71 (SYPE).

After all the DB2 data was on SYPE, the LOGON of the users of SDWH was routed to TOR ISTCIT03 on SYPE. SDWH was the first application migrated to full data sharing. This step is described in detail in 4.4.8, "Multi-system CICSPlex" on page 88, and in Chapter 5, "DB2 data sharing" on page 93.

As shown in Figure 4-2, there are also single CICSes with specific applications:

- ISTCIA31: Home Banking Computer Interface (HBCI)
- ISTCDT11: Routing Manager CICS (the interface to external partners, part of the former CICSP8)

4.3 Why IZB changed the environment

IZB decided to change its environment as a result of the problems and considerations that are detailed in this section.

CPU constraints

At the beginning and the end of each month, operating system SYPE showed CPU constraints which resulted in high response times. As previously noted, there were three production systems. When a savings bank from system SYPG merged with a savings bank on system SYPE, the load grew in that region. Using two systems in a Parallel Sysplex environment relieves this constraint.

Growth (scalability) constraints

The CICS loads were very different, so IZB wanted to balance them. The only way to reduce the load of a CICS system was to start a new CICS region. There was a great deal of work to be done to achieve this; for example:

- Create a new CICS with all necessary resource definitions
- Allocate new data sets
- Adapt databases (DB2 and Adabas)
- Change batch jobs, and so on

The CICS regions on SYPE were almost at their limits.

CICS availability

If a CICS region failed, availability suffered. For example, all savings banks that were served by that region would go down and no one could withdraw funds from affected ATMs. This was also true if the whole LPAR went down.

4.4 The implementation project

IZB and its client undertook this implementation project for the following reasons:

- To separate logins from application programs in CICS
- To make every transaction executable in all CICS (application) regions
- To shorten application outages
- To avoid savings banks outages
- To achieve growth without risk
- To ease the administration of CICS resources

This section explains the steps that IZB followed to implement a multi-site CICSPlex in a Parallel Sysplex environment. (An IBM reference was used for planning purposes; the current version is *z/OS V1R1.0 Parallel Sysplex Application Migration*, SA22-7662.) The following topics are discussed here:

- Implementing a shared temporary storage queue server (TSQ server)
- Building a CICSPlex
- Analyzing application programs
- Introducing a terminal owning region (TOR)
- Introducing application owning regions (AORs)
- Using VTAM Generic Resource (VGR)
- Using VSAM RLS
- Multi-system CICSPlex

IZB moved to the multi-system CICSPlex in two phases. In the first phase, IZB established a CICSPlex. In the second phase, IZB spread this CICSPlex over two systems.

4.4.1 Implementing a shared temporary storage queue server (TSQ server)

Shared TS queues (TSQs) allow shared access to non-recoverable TSQs. This enables the sharing of temporary storage between many applications running in different CICS regions. TSQs are stored in a Coupling Facility (CF) structure named DFHXQLS_poolname.

At IZB, this structure was defined by the z/OS systems programmers and was activated dynamically. It took just a few minutes to define and activate the definitions (a useful formula for this task is provided in *CICS System Definition Guide*, SC34-6428).

Using shared TS queues

There is an additional MVS address space that has access to this structure and that must be started before a CICS region is started. In a SYSIN data set, parameters are defined that control the server address space. One of the parameters is the name of the pool.

Though shared temporary storage queues are not recoverable and are not backed out as part of normal CICS backout operations, they are preserved across CICS restarts and even MVS IPLs, as long as the CF does not stop or fail. Starting with CICSTS 2.1, CF structures can be rebuilt for shared temporary storage queues. (With prior releases, the structures had to be dumped and reloaded. If that happens, temporary storage queues are lost. If needed, a way to get them back must be found, for example, using SM Duplexing.)

IZB's client is unconcerned with temporary storage queues. If a failure occurs, the application is invoked once again. There is no difference when using shared temporary storage queues.

Analyzing temporary storage queues

It is very important to analyze temporary storage queues, whether they are recoverable or not recoverable. Only non-recoverable temporary storage queues are stored in CF structures.

Naming conventions need careful consideration, as explained here. Because storage in a CF is restricted, it is necessary to delete shared temporary storage queues to avoid out-of-space conditions. IZB's client discovered that not every GET STORAGE is followed by a FREE STORAGE. Therefore the client wrote a program that runs every night and deletes TSQs. This program looks for shared temporary storage with certain names which must *not* be deleted, and it looks for TSQs that meet particular criteria (such as, if it was used within a certain time period). However, using this technique demands well-defined naming conventions.

Using temporary storage in auxiliary storage avoids this restriction, for the following reasons:

- A VSAM cluster has much more space available than a CF does.
- Only one CICS region has access to this temporary storage.
- A CICS region can be started with the option TS=COLD; with this option, the temporary storage is cleared.

The temporary storage data sharing server

Access to the structure DFHXQLS_poolname is through an address space running the program DFHXQMN. The poolname is defined in a SYSIN data set. For further information about how to set up, define, and start a TSQ server address space, refer to *CICS System Definition Guide*.

Example 4-1 shows an extract of the joblog of the TSQ server.

Example 4-1 An extract of the joblog - TSQ server

```
J E S 2  J O B  L O G  --  S Y S T E M SYPE --  N O D E N O D E C

01.43.29 STC41733 ---- SUNDAY,   06 NOV 2005 ----
01.43.29 STC41733  IEF695I START IZTCQIAP WITH JOBNAME IZTCQIAP IS ASSIGNED TO USER
IZTCQIAP, GROUP IZ$STASK
01.43.29 STC41733  $HASP373 IZTCQIAP STARTED
01.43.29 STC41733  IZB001I RACF-KLASSE §BATACCT NICHT AKTIV, KEINE ACCOUNT-PRUEFUNG
01.43.29 STC41733  IEF403I IZTCQIAP - STARTED - TIME=01.43.29   DATE=06.11.2005
01.43.29 STC41733  DFHXQ0101I Shared TS queue server initialization is in progress.
01.43.34 STC41733  DFHXQ0401I Connected to CF structure DFHXQLS_ISIQAPP.
01.43.34 STC41733  IXL014I IXLCONN REQUEST FOR STRUCTURE DFHXQLS_ISIQAPP  969
  969          WAS SUCCESSFUL.  JOBNAME: IZTCQIAP ASID: 00D4
  969          CONNECTOR NAME: DFHXQCF_SYPE CFNAME: CFNG1
01.43.34 STC41733  AXMSC0051I Server DFHXQ.ISIQAPP is now enabled for connections.
01.43.37 STC41733  DFHXQ0102I Shared TS queue server for pool ISIQAPP is now active.
```

How CICS and the data server work together

To use shared temporary storage, define an entry in DFHTSTxx with TYPE=SHARED, POOL=poolename and SYSIDNT=sysid. In the application program, TSQ writing is done with this command:

```
EXEC CICS WRITEQ TS SYSID(sysid)
```

In this case, the application program needed to be changed. But there is another way to use shared temporary storage without changing an application: add an entry in DFHTSTxx with TYPE=REMOTE and SYSIDNT=sysid. The line in the application program looks as follows:

```
EXEC CICS WRITE TS QUEUE(any)
```

Figure 4-3 on page 76 illustrates the definitions and components at IZB.

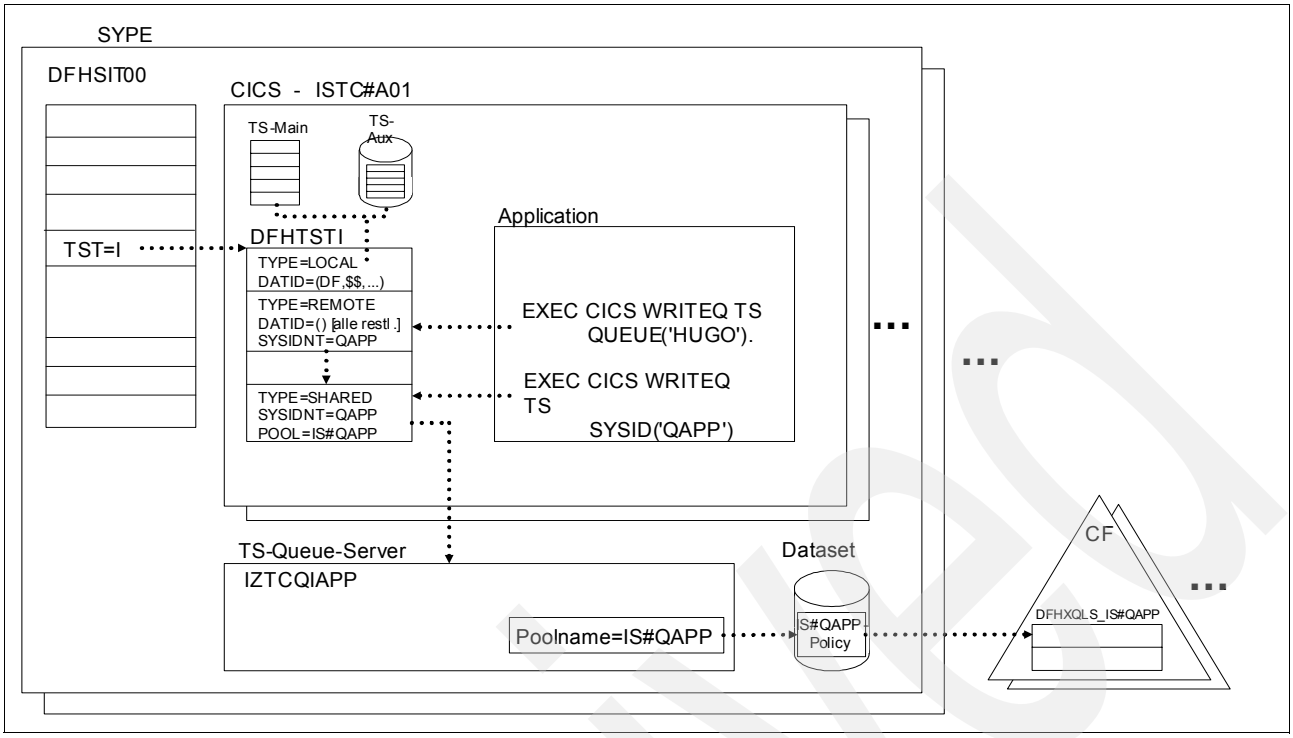


Figure 4-3 A summarized overview of TSQ definitions

Example 4-2 shows an example of IZB's DFHTSTIA.

Example 4-2 IZBs DFHTST member

```
DFHTSTIA DFHTST TYPE=INITIAL,
               SUFFIX=IA,
               TSAGE=2
         SPACE 1
         DFHTST TYPE=LOCAL,
               DATAID=(CEBR,$$,DFH,**,§RM)
         SPACE 1
         DFHTST TYPE=LOCAL,
               DATAID=($TABPROG,$JCAUFTR,RN805TLG,TESTT006,ELCAS811,
               ELGBS811,#ZPRO/,RNDB2,TS261,R356G,RN452,-,<,>,
               HKDCV2RO,§HB,§DS)
         SPACE 1
         DFHTST TYPE=REMOTE,
               DATAID=(),
               SYSIDNT=QAPP
         DFHTST TYPE=SHARED,
               SYSIDNT=QAPP,
               POOL=ISIQAPP
         SPACE 2
         DFHTST TYPE=FINAL
          END
```

4.4.2 Building a CICSPlex

Why build a CICSPlex

As mentioned, both IZB and its client wanted to increase availability, improve performance, use resources more effectively, and handle increased load without any risk. To achieve these goals, it was decided to build a CICSPlex.

One of the characteristics of a CICSPlex is the separation of user logon and the execution of an application transaction in a different CICS region. The idea is to create a terminal owning region (TOR) and one or more application owning regions (AOR) that are clones.

As many transactions as possible should be executable in more than one CICS region, when dynamic routing is possible. The distribution of transactions over many AORs minimizes restart time of an AOR, because fewer recovery actions are necessary. Instead of affecting an entire savings bank if an AOR goes down, such an outage will affect only the transactions that were in flight at that time. Taking into account that all AORs are cloned CICSes and that all transactions can be executed in them, there is no problem if workload increases—simply start a new AOR.

IZB found the IBM product CICSPlex SM to be very helpful in achieving these goals. CICSPlex SM also made it easier to administer all the CICSes.

Planning

There are several possibilities for migrating to a CICSPlex; the following resources provide useful planning advice.

- IZB used CPSM documentation for guidance:
  - *CICSPlex SM Concepts and Planning,* GC33-0786
  - *CICSPlex SM for CICS TS z/OS, Administration,* SC34-6256
  - *CICSPlex SM for CICS TS z/OS, Managing Workloads,* SC34-6259
- The CICS InfoCenter describes an installation path in great detail, including all necessary steps.

IZB and its client developed the following roadmap for the CICSPlex implementation:

1. Analyze all application programs.
2. Remove as many affinities as possible.
3. Put each application with affinities into its own CICS region.
4. Build a CICSPlex topology: TOR - AORs - CAS - CMAS.
5. Use CPSM for administration.
6. Make workload definitions.
7. Change the LOGON processing of the savings banks.
8. Activate dynamic routing.
9. Use VGR.

Preparing for a CICSPlex

Along with the new architecture, IZB decided to introduce new naming conventions. For CICS started tasks, the result was: `ccTCptnn` (for example, `ISTCIT01`), where:

`cc`	This is the abbreviation of the client name.
`TC`	This is a fixed part of the name: **T**ask **C**ICS.
`p`	This is the CICSPlex identifier.
`t`	This is the type of MAS: A for AOR, T for TOR, and so on.
`nn`	This is an ascending number.

VTAM APPLID: the same as the STC name.

The name of the system data sets have the form IZS.CICSTSvr.SP000.lq; for example:

```
IZS.CICSTS22.SP000.SDFHLOAD
```

Region-specific and customer data sets are named ISR%CIC%.**; for example: ISRWCICP.SYS.IA110220.DFHDMP.

The IBM publication *z/OS V1R1.0 Parallel Sysplex Application Migration*, SA22-7662, offers useful tips about what to consider when planning naming conventions. This publication is available at:

http://publibfi.boulder.ibm.com/cgi-bin/bookmgr/BOOKS/e0s2p400/CCONTENTS

4.4.3 Analyzing application programs

Searching for affinities

Transaction affinities prevent the exploitation of CICSPlex, and make one AOR a single point of unavailability for applications. Affinities also limit the benefits of dynamic transaction routing—and dynamic transaction routing provides continuous availability by routing transactions away from an unavailable AOR to another AOR that can run the transaction. For these reasons, transaction affinities must be removed from application programs.

At IZB, one of the major tasks was to discover all affinities in application programs. (IBM documentation offers many hints and tips to help with this task.) IZB also used the IBM CICS Affinity Utility. Information about how to use this utility is provided in *CICS Transactions Affinities Utility Guide,* SC34-6013, which is available on the Web at:

http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC34-6013

This publication contains a detailed explanation of how to run the analyzer programs to get reports about the affinities in application programs. This effort results in an overview of how many programs needed to be changed. (Estimates about how long it takes to implement a CICSPlex can be given only after a comprehensive analysis.) Another IBM publication, *z/OS V1R1.0 Parallel Sysplex Application Migration*, SA22-7662, also provides useful information about this topic.

At IZB, this effort helped to locate all CICS commands that imply affinities in application programs. The reports were large, so IZB's client had to check each program very carefully to find real affinities. But there were other affinities too; for example, checks of the right LPAR, using equal data set names, using equal connection names, and so on.

Types of affinities

There are various types of affinities:

Inter-transaction affinity	Occurs when transactions exchange information between themselves, or try to synchronize their activities
Transaction-to-system affinity	An affinity between a transaction and a system; can be a particular CICS region or LPAR

For a detailed description of these affinities, refer to *CICS Application Programming Guide,* SC34-5993, available on the Web at:

http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC34-5993

As mentioned, there were Bavarian savings banks distributed across three LPARs, so there were many programs that had an affinity to a certain LPAR. The savings banks were connected to dedicated CICS regions, which was another reason why many affinities existed.

Having INQUIRE and SET commands in a transaction are reasons for a transaction-to system affinity, because these commands cannot be executed on a remote site. Hence transactions using those commands must be routed to the CICS region that owns the resources to which they refer. IBM Redbook *J.D. Edwards OneWorld XE Implementation on IBM @server iSeries Servers*, SG24-6529, contains helpful information about dealing with affinities.

IZB's client created a table with all affinities the programmers could think of. They considered the following points:

- APPLID
- CSA
- CWA
- CWADRTAB
- CWADRWTO
- EIBTRMID
- GETMAIN SHARED/FREEMAIN
- Global User exits
  - EXTRACT/ENABLE EXIT (does not work remotely)
  - DISABLE PROGRAM (does not work remotely)
- INQUIRE
- SET
- LOAD defined with RELOAD=YES
- LOAD HOLD
- PERFORM STATISTIC (does not work remotely)
- PERFORM SHUTDOWN (does not work remotely)
- RELEASE pgm
- START TERMID
- START w/DATA
- START w/REQUID
- SYSID (CICS SYSID)
- SYSID MVS (MVS SYSID)
- Synchronization/serialization
  - WAIT EVENT
  - WAIT EXTERNAL
  - WAITCICS
  - ENQ/DEQ
  - CANCEL REQID
  - DELAY / REQID
  - POST / REQID
  - RETRIEVE WAIT/START
  - START/CANCEL REQID
- TS Queues
  - READQ/WRITEQ/DELETEQ
- TD Queues
  - READQ/WRITEQ/DELETEQ

> **Note:** This list can be used as an example checklist, but every environment is different and should be very carefully examined for its own affinities.

In IZB's case, it took about a month to analyze all programs. The most essential step came next.

Changing application programs

After having analyzed the applications, IZB's client divided the programs into four groups:

- Programs that are already dynamically routable
- Programs that have to be changed
- Programs in which affinities cannot be removed
- Programs that are developed outside of the company

The plan was to build a CICSPlex environment with a terminal owning region (TOR) and several application regions (AOR) that were clones. Then all applications could run unchanged, meaning that each savings bank ran its application in the old CICS region. The only change was that the LOGON connected the session to a TOR. From there the transaction was routed to the target CICS. All new applications would run in the new AORs.As IZB analyzed and changed the programs, it implemented a CICSPlex. This provided the capability to do a smooth migration. The whole activity took about six months.

4.4.4 Introducing a terminal owning region (TOR)

Planning for a TOR

In 1999, each savings bank logged on to a specific CICS region. Each server in a savings bank had specific information about the environment on which it ran, such as the target CICS name (for example, CICSP1).

To remove this dependency, IZB introduced *interpret tables* in VTAM which had the same information as the servers. The benefit of this was that a change of the target CICS region was transparent for the savings bank. It only had to remove the information file in the server once. Afterwards, all changes could be done by IZB's VTAM systems programmers. This change was not critical, because the savings banks could decide when to remove the information file. IZB offered an eight-week time period to the savings banks within which they could do the change. The first step was only a one-to-one change, meaning that savings banks which ran in CICSP1 remained running in CICSP1.There was no dynamic routing yet. This new TOR ISTCIT01 had MRO connections to CICSP1 to CICSP4.

4.4.5 Introducing application owning regions (AORs)

Why to clone AORs

If dynamic routing is used, it is very useful to have cloned AORs. Cloned AORs can execute every transaction that is dynamically routed, and there is no interruption if a CICS region fails. CICSPlex System Manager (CPSM) routes new transactions to the available AORs. Even if a whole LPAR goes down there is little impact.

Applications that are not designed for Parallel Sysplex (and thus cannot fail over) can be started on the other LPAR. The others can run on the alternate LPAR. At IZB, if the load is too heavy for the AORs on this system, the automation team can easily start one or more spare CICS regions. This is also true if workload increases in the CICSPlex.

Planning for an AOR

To use the dynamic routing of transactions feature, IZB built AORs; IZB's CICS systems programmers used the information provided in CPSM documentation and in *OS/390 V2R5.0 Parallel Sysplex Application Migration,* GC28-1863. IZB decided to build four AORs, plus two additional spare CICS regions that could be started if load was very heavy. The names of the started tasks were ISTCIA11, ISTCIA12, ISTCIA13, and ISTCIA14.

Figure 4-4 on page 81 shows IZB's first CICSPlex configuration.

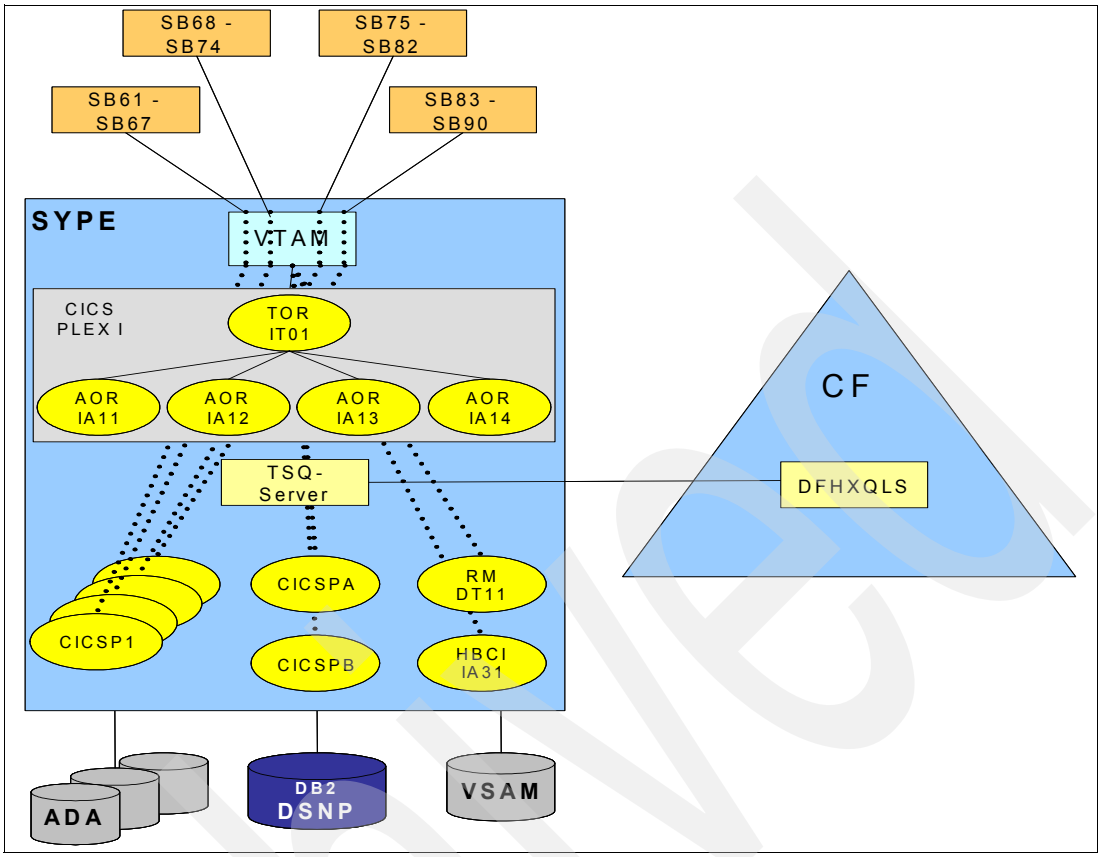


Figure 4-4 IZB's first CICSplex configuration

The same configuration was implemented on SYPG with CICSPlex A, and on SYPF with CICSPlex E.

CPSM allows definition of the topology of a CICS environment. CPSM knows about the TOR, the AORs, and which transactions are able to be routed dynamically; for more information about this topic, see the following references:

- *CICSPLEX SM Concepts and Planning*, which is available on the Web at:

  http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC34-6469

- *CICSPlex SM for CICS TS z/OS, Managing Workloads*, which is available on the Web at:

  http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC34-6465

IZB defined the TOR ISTCIT01, the AORs ISTCIA11 to ISTCIA14, and CICSP1 to CICSP4. Because all the terminals of all the savings banks went to ISTCIT01, there had to be a way for the transactions to be executed in their original CICS.

That was achieved with a workload definition in CPSM, by defining that transactions that come from a terminal with a certain LU name have to run in a particular CICS region. So IZB built a workload definition that reflected the previous environment. All transactions from new applications with no affinities were routed dynamically to the new AORs.

Following is an explanation of what happened to CICSP5, CICSP8, CICSPA, and CICSPB.

What happened to CICSP5, CICSP8, CICSPA, and CICSPB

CICSP5 This was the IZB BTX CICS, as described earlier. This application was replaced by a new application known as Home Banking Computer Interface (HBCI). Unfortunately HBCI cannot use dynamic routing and data sharing, but it is a standard product that is used by every savings bank in Germany. HBCI was developed outside of IZB. IZB put this new application in a new CICS: ISTCIA31

CICSP8 This CICS ran all transactions for teller machines; it was the CICS region with the connections to external partners. CICSP8 was split. Teller machine applications were made dynamically routable and were run in ISTCIA11 to ISZCIA14. The connections to external partners were defined in a new CICS, ISTCDT11, the IZB Routing Manager CICS. Unfortunately this application is also not routable and cannot use data sharing. It was developed outside of IZB. This application will be replaced by a solution using WebSphere MQ in the near future.

CICSPA As described earlier, there were two applications in this region: MARZIPAN and S-Data Ware House (SDWH). These applications were also split. MARZIPAN was integrated in ISTCIA11 to ISTCIA14. SDWH got a region of its own because this application uses a DB2 that is different from the main OLTP applications. This is CICS region ISTCIA71. A separate TOR was implemented for SDWH: ISTCIT03. To make a difference in CPSM, we defined two different CICS system groups. One was for OLTP applications in ISTCIA11 to ISTCIA14: CSGIA10. The CICS system group for SDWH was called CSGIA70.

CICSPB The application of this CICS was transferred to SYPG (CICSMB) from SYPE.

At this point, the initial steps in exploiting a CICSPlex were completed. The client worked hard to remove affinities so IZB could move ahead.

Being able to distribute applications is useful. However, IZB had only one terminal owning region and it wanted to remove this single point of failure. Therefore, a decision was made to build a second TOR. But in order to arrive at a distribution of the sessions between these two TORs, IZB had to introduce VTAM Generic Resources (VGR).

4.4.6 Using VTAM Generic Resource (VGR)

Why to use VGR

VTAM generic resource registration allows multiple TORs to share the same generic VTAM APPLID. This reduces the dependency on a single TOR. If a TOR is unavailable, users can log on to any other TOR that is member of the same VGR.

An additional benefit is that the session workload can be balanced between several TORs, by coding a VTAM exit.

Definitions in VTAM

In VTAM, a logical name is defined for a group of VTAM applids. The end user knows only this logical name, and gets a session to one of the members of the group. CICS registers automatically in the VGR (see 3.1, “Central mainframe network” on page 44 for more information about this topic).

Changing the SIT parameter

It is necessary to add the following parameter:

GRNAME=*vgr name*

> **Note:** The XRF parameter must be specified as XRF=NO.

Some useful commands

The following commands proved useful:

CICS commands

- To inquire about the type and state of the VTAM connection for the CICS:

  ```
  CEMT INQUIRE VTAM
  ```

  This example shows the output:

  ```
  F ISTCIT01,CEMT I VTAM
   +
     Vtam
     Openstatus( Open )
     Psdinterval( 000000 )
     Grstatus( Registered )
     Grname(ISTCIT0#)
     RESPONSE: NORMAL TIME:  13.06.51  DATE: 09.12.05
     SYSID=IT01 APPLID=ISTCIT01
  ```

- To remove a TOR from VGR:

  ```
  CEMT SET VTAM DEREGISTERED
  ```

  This example shows the output:

  ```
  F ISTCIT01,CEMT S VTAM DEREGISTERED
  +
  Vtam
   Openstatus( Open )
   Psdinterval( 000000 )
   Grstatus( Deregistered )
   Grname(A6IZWT1X)
   NORMAL
   RESPONSE: NORMAL TIME:  13.52.17  DATE: 09.12.05
   SYSID=WT11 APPLID=A6IZWT11
  ```

- This shows a message in CICS Joblog:

  ```
  DFHZC0172I ISTCIT01 CICS deregistered successfully from VTAM generic resource name
  ISTCIT0#
  ```

  This command is useful if you want to shut down a TOR without forcing a logoff of users or sessions. Active sessions remain active until they become inactive. The new logon goes to the remaining TORs of the VGR.

- To bring this TOR back to VGR, you have to restart the CICS region or use the following commands:

  ```
  F stcname,CEMT SET VTAM CLOSED
  F stcname,CEMT SET VTAM OPEN
  ```

  This example shows the output.

  ```
  DFHZC0170I ISTCIT01 CICS registered successfully to VTAM generic resource name
  ISTCIT0#
     235
   Vtam
  ```

```
Openstatus( Open )
Psdinterval( 000000 )
Grstatus( Registered )
Grname(ISTCIT0#)
NORMAL
RESPONSE: NORMAL TIME:  14.10.03  DATE: 09.12.05
SYSID=IT01 APPLID=ISTCIT01
```

VTAM commands

- To list statistics of the structure:

  ```
  D NET,STATS,TYPE=CFS,STRNAME=ISTGENERIC
  ```

  This example shows the output:

  ```
  D NET,STATS,TYPE=CFS,STRNAME=ISTGENERIC
  IST097I DISPLAY ACCEPTED
  IST350I DISPLAY TYPE = STATS,TYPE=CFS
  IST1370I DEBBSMON.SYPE IS CONNECTED TO STRUCTURE ISTGENERIC
  IST1797I STRUCTURE TYPE = LIST
  IST1517I LIST HEADERS = 4 - LOCK HEADERS = 4
  IST1373I STORAGE ELEMENT SIZE = 1024
  IST924I -------------------------------------------------------------
  IST1374I                              CURRENT     MAXIMUM  PERCENT
  IST1375I STRUCTURE SIZE                63744K      96000K    *NA*
  IST1376I STORAGE ELEMENTS                  30        9809       0
  IST1377I LIST ENTRIES                   41586      166753      24
  IST314I END
  ```

- To determine which TORs are connected to the VGR, as well as the status of the sessions:

  ```
  D NET,SESSIONS,LU1=vgr name
  ```

  This example shows the output:

  ```
  D NET,SESSIONS,LU1=ISTCIT0#
  IST097I DISPLAY ACCEPTED
  IST350I DISPLAY TYPE = SESSIONS
  IST1364I ISTCIT0# IS A GENERIC RESOURCE NAME FOR:
  IST988I ISTCIT01 ISTCIT05
  IST924I -----------------------------------------
  IST172I NO SESSIONS EXIST
  IST314I END
  ```

Starting a second TOR

At this point, IZB started a second TOR ISTCIT02, making a clone CICS of ISTCIT01 and adjusting the definitions in CPSM. For more information about this topic, refer to *CICSPlex SM Concepts and Planning*, SC34-6469, which is available on the Web at:

http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC34-6469

Restrictions to VGR

IZB overcame some challenges of using VGR. If an LU6.2 connection is bound at synclevel 2 to a specific member of a generic resource group, it is reconnected to that specific APPLID every time it is re-bound. This enables the VTAM generic resources function. If, for some reason, the specific APPLID is not available, then connection to the generic resource as a whole is denied.

IZB defined an LU 6.2 connection from ISTCA31 (HBCI CICS) to VGR ISTCIT0#. So IZB wanted to set up the connection such that, if a TOR failed, the connection would automatically

be established to the second member of the VGR. However, an affinity between one TOR and the HBCI CICS persisted, even across CICS restarts, and messages such as DFHRS2118, DFHRS2110, and so on were received.

4.4.7 Using VSAM RLS

A very useful description of implementing VSAM RLS is provided in Chapter 4 “Workloads in a Parallel Sysplex” of *OS/390 Parallel Sysplex Configuration, Volume 1: Overview*, SG24-5637. *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765, also contains important information.

Why to use VSAM RLS

VSAM record level sharing (RLS) allows multiple address spaces, such as CICS application owning regions (AORs), to access VSAM data simultaneously. Each CICS region can update data while data integrity is guaranteed. CICS regions that share VSAM data can reside in one or more z/OS images.

There is no need for a file owning region (FOR) to share data. The IZB client ran performance tests with FOR and VSAM RLS, and it turned out that VSAM RLS provides better performance. With VSAM RLS, only records are locked, not a whole control interval (CI). This improves performance and reduces deadlock situations. Also, in case of a backout or failures, there is only a lock for a record and not for the entire CI.

Without VSAM RLS, all data sets of a savings bank were related to a certain CICS region. In case of a planned or unplanned outage of a CICS region, the application for this savings bank was unavailable until the CICS region had been restarted.

With RLS, all VSAM data sets can be accessed by all CICS regions. All savings banks can continue working, so higher availability is achieved.

How VSAM RLS works

For data sets accessed in non-RLS mode, VSAM control blocks and buffers are located in each CICS address space and therefore are not available for batch jobs and other CICS regions. Simultaneous writing to the data set is prohibited by SHAREOPTIONS for the data set. Under normal circumstances a data set is open for writing only for one CICS (or batch job). Simultaneous writing can be allowed with SHAREOPTIONS, but then the application must control the access with ENQ/DEQ.

There is a new z/OS address space called SMSVSAM. VSAM control blocks are stored in its data spaces. For RLS accesses to VSAM files, CICS gives control to SMSVSAM. It is responsible for synchronizing and data integrity. SMSVSAM allocates the data set.

To enable CICS/VSAM RLS, one or more CF cache structures must be defined and added to the 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, the CF lock structure must also be defined (a lock structure is needed as protection against overwriting).

To calculate the size of the structures, IZB used the formula provided in the IBM Redbook *OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook*, SG24-5638. For more information about this topic, refer to Chapter 9, “System” on page 177 in this publication.

Figure 4-5 on page 86 illustrates how VSAM RLS is implemented.

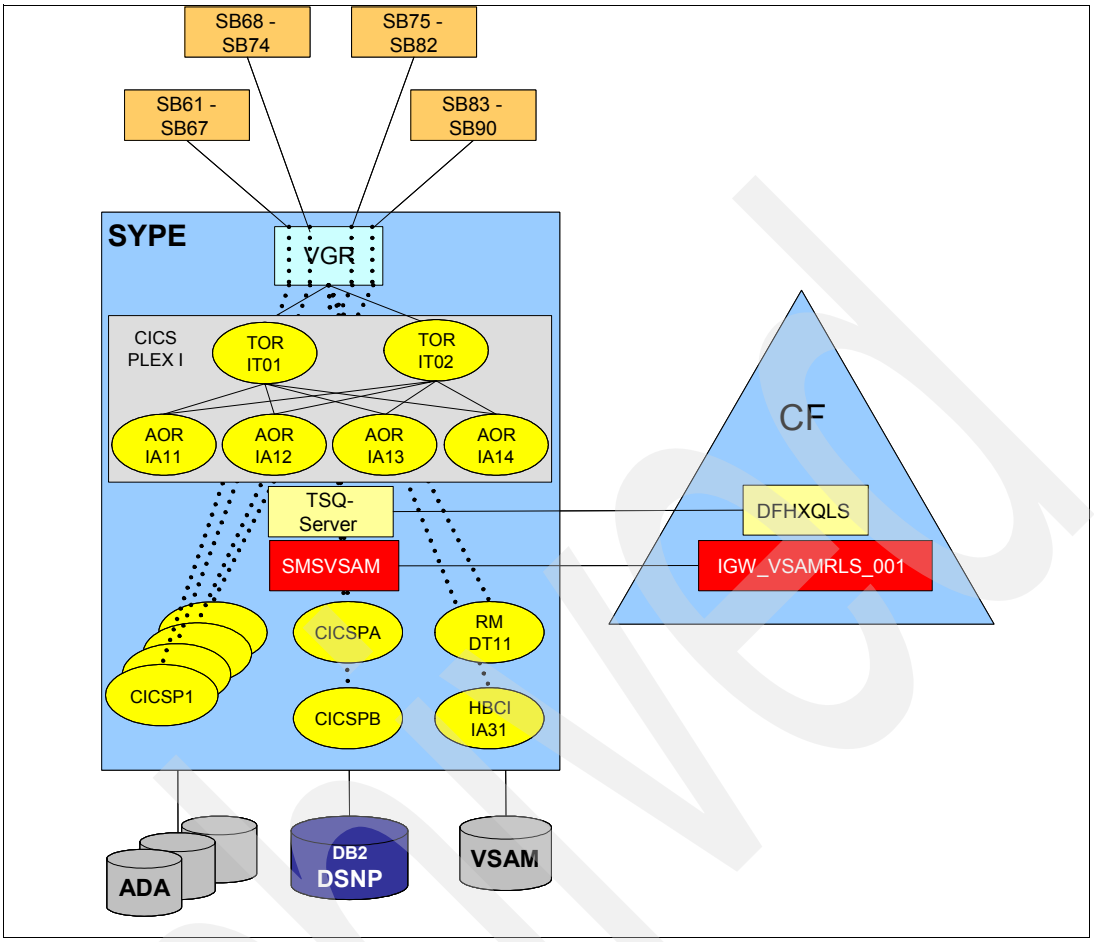


Figure 4-5 CICSPLEX with VSAM RLS

IZB had some considerations concerning data sets accessed by CICS and batch jobs:

- Batch update
  - Recoverable data

    Batch update is only possible in non-RLS mode. A data set can be accessed either in RLS or in non-RLS mode. Because there should be an availability of 24 hours a day, batch update had to be removed.

    During a short batch window VSAM data sets are quiesced, updated by batch jobs, and unquiesced for use in CICS.
  - Non-recoverable data

    There is one exception to the prohibition of updating RLS data sets by batch jobs: data sets defined as nonrecoverable (LOG=NONE). There were only a few data sets. On these data sets there is no BACKOUT or they are opened by CICS only with READ.
- Reading batch

  A VSAM data set can be opened by CICS in RLS mode for writing and at the same time by a batch job in non-RLS mode for reading. This is possible if SHAREOPTIONS are (2 3).

  For IZB, this meant that all batch programs that had read access could be used without any changes. This is known as *inconsistent read* because in the SMSVSAM buffer there could be a changed or deleted record. DASD still has the unchanged record, so the application gets the unchanged old record. This technique is still used.

To implement *consistent read*, there would have been numerous changes to batch application programs. Software from ISVs can be used to handle this task, but IZB did not pursue this action.

Attributes that allow a VSAM data set to be accessed in RLS mode

- The data set must be SMS-managed.
- VSAM cluster definition without IMBED.
- Logging requests are defined in the ICF catalog and not in CICS CSD: LOG(NONE), LOG(UNDO), LOG(ALL).

Most of IZB's VSAM data sets were defined with IMBED. Therefore, many data sets had to be copied in a newly allocated data set. This copy was done, along with the introduction of new data set naming conventions.

Necessary changes in CICS

In CICS there were just two minor changes to do:

- SIT parameter RLS= YES
- Change FCT entries: RLSACCESS(YES)

Summary

- Analyze the data sets (recoverable, non-recoverable, share options, and so on).
- Carefully consider the naming conventions for data sets.
- Implement SMSVSAM address (see Chapter 9, "System" on page 177).
- Define SHCDS (see Chapter 9, "System" on page 177).
- Change the SIT parameter to RLS=YES.
- Change the FCT entries to RLSACCESS(YES).

Benefits of using VSAM RLS

RLS provides many benefits to CICS applications, improving the way CICS regions can share VSAM data. Using RLS can result in the following:

- Improved availability
  - The FOR is as a single point of failure is not needed.
  - 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.
- Improved integrity

  Integrity is improved in RLS mode for both the reading and writing of data. RLS uses 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 "CICS/VSAM RLS Structures" in *OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook*, SG24-5638, and to the CF Sizer tool, available on the Web at:

  http://www.s390.ibm.com/cfsizer/
- Reduced lock contention

  For files opened in RLS mode, VSAM locking is at the record level, not at the control interval level. This can improve throughput and reduce response times.

- Improved 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.

- Improved performance

  Multiple CICS application-owning regions can directly access the same VSAM data sets, thus 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.

4.4.8 Multi-system CICSPlex

At the start of its move to a CICSPlex, IZB wanted to achieve a multi-site CICSPlex with full data sharing, high availability, load balancing and better performance. All planning and activities focused on this goal; in order to achieve it, the following prerequisite tasks needed to be completed.

Prerequisites

- Additional LPAR SYP7 (refer to Chapter 9, “System” on page 177, for more information)
- Building a sysplex (refer to Chapter 9, “System” on page 177)
- Implementing SMSVSAM (refer to Chapter 9, “System” on page 177 and to 4.4.7, “Using VSAM RLS” on page 85)
- Implementing DB2 data sharing (refer to Chapter 5, “DB2 data sharing” on page 93)
- Implementing Adabas data sharing (refer to Chapter 6, “Adabas data sharing guideline” on page 111)
- Implementing MQ sharing

A new TSQ server also had to be started. Only a new started task had to be started, which used the same TS pool in the Coupling Facility as the one on SYPE.

Installation steps

IZB followed these steps to install the CICSPlex:

1. To arrive at the final architecture, IZV chose a specific application: SDWH. SDWH was a single application and so all results could be used for further actions in our OLTP environment. A second reason was that this environment had no Adabas access.

   As previously mentioned, SDWH was an application running on systems SYPG, SYPF and SYPE in CICS regions ISTCAA71, ISTCEA71 and ISTCIA71. On SYPE, there was a TOR ISTCIT03 which routed SDWH transactions to ISTCIA71. A clone TOR ISTCIT04 was built, and both TORs became members of VGR ISTCIT7#.

   A clone CICS region ISTIA72 was also built. After that, all requests for the SDWH application were routed to SYPE and VGR ISTCIT7#.

2. After SYP7 was available, IZB prepared for CPSM on that system. For detailed information about implementing CPSM, refer to *CICSPLEX SM Concepts and Planning*, which is available on the Web at:

   http://www.elink.ibmlink.ibm.com/public/applications/publications/cgibin/pbi.cgi?CTY=US&FNC=SRX&PBL=SC34-6469

   IZB started a coordinating address space (CAS) in order to have an ISPF facility to administer CPSM. A new CMAS was started on SYP7. In order to have all information of the CPSM available on SYP7, a CMAS-to-CMAS connection had been defined between the CMASes on SYPE and SYP7. Following the instructions in the CPSM documentation, IZB defined a maintenance CMAS.

3. IZB then cloned TOR ISTCIT03 and ISTCIT04 on SYP7. The SDWH TOR CICSes on SYP7 were named ISTCIT07 and ISTCIT08, and both became a member of VGR ISTCIT7#.
4. For the clone AORs ISTCIA75 and ISTCIA76, IZB only had to redefine the DB2CONNECTION. Instead of a DB2ID, a DB2GROUP name was defined for DB2 group attach.

At this point, all definitions on SYP7 could be made without any interruption. Even the definition of the new TORs and AORs on SYP7 in CPSM were made dynamically. The transactions immediately began running on SYP7. They were distributed almost equally between SYPE and SYP7.

IZB had a successful experience, so it continued exploiting a multi-system CICSPlex®. In the meantime, most of the applications were ready for dynamic routing. The old CICS regions (CICSP1, CICSP2 and so on) had been stopped and all work was done by the new ones (ISTCIA11, ISTCIA12, and so on). The necessary steps were similar to those for distributing the SDWH application:

1. Cloning TORs (ISTCIT05 und ISTCIT06).
2. Integrating TORs in VGR ISTCIT0#.
3. Cloning AORs ISTCIA15, ISTCIA16, ISTCIA17.
4. Updating CPSM with new routing and target systems.
5. Redefining DB2CONN: DB2 group attach.

However, some additional tasks were also needed:

6. Defining new AORs to routing manager cics ISTCDT11.

   ISTCDT11 was the IZB routing manager CICS (RM) and all connections to external partners are defined in it. In order to distribute requests from outside, IZB had to define the new AORs to this application.
7. Adjusting WLM definitions.

   IZB held many discussions with its systems programmers about the definitions in WLM. A rule of thumb is: all server address spaces (in the view of CICS AOR) have to be defined in a higher service class. CPSM also has to be defined in a very high service class. It is not very busy but if it needs more CPU, it must get it. Otherwise poor response times, abends of transactions, SOS, and MAXTASK conditions may be encountered.

4.5 Conclusions

To obtain the benefits of data sharing, the following prerequisites were needed:

- Preparing the system environment
- Preparing hardware (for example, the Coupling Facility)
- Introducing DB2 data sharing
- Introducing Adabas data sharing
- Introducing MQ shared services
- Eliminating all kinds of affinities by the client and the batch environment
- Introducing VGR
- Using VSAM RLS

What IZB achieved with this migration

With data sharing in a Parallel Sysplex, IZB achieved more effective load balancing. By running one TOR on SYPE and another TOR on SYP7, CICS transactions are distributed equally. LOGINs to TORs are routed with a round robin algorithm. CPSM prefers local connections to AORs, so nearly half of the transactions runs on SYPE and the other half runs on SYP7. Using WLM criteria does not lead to an equal distribution. WLM tries to use hardware most efficiently and does not attempt to balance.

Because all AORs are cloned, there is no interruption if a CICS region fails. CPSM routes new transactions to the available AORs. Even if a whole LPAR fails, there is little impact. Applications that are not designed for Parallel Sysplex do not fail over, but can be started on the other LPAR. The others can run on the alternate LPAR. If load is too heavy for the AORs on this system, the IZB automation team can easily start one or more spare CICS regions. This is also true if the workload increases in the CICSPlex.

All this results in higher availability and fewer outages.

Having a CICSPlex on a Parallel Sysplex with data sharing is beneficial when performing maintenance. Application changes can be put into production in a controlled manner. A new version of the application can be tested. Using CPSM, a set of transactions can be routed to a certain AOR. When the result is checked, and if it works successfully, the new version can be spread dynamically to the remaining AORs. If there are abends or other failures, the changes can be removed.

Lessons learned

- IZB found, in the environment of DB2 and Adabas data sharing, that transactions need about 10% more CPU time. This is due to both IZB's mixed environment and to the fact that some Coupling Facilities and the structures inside are remote.
- There are some restrictions for using VTAM Generic Resources, especially for LU6.2 devices bound at synclevel 2.
- Today, a DB2 data sharing group can be defined in the DB2CONN definition. It would be useful if all members of a sharing group did not have to be on the same LPAR. CICS always makes the attach to the local DB2. If this DB2 fails, then CICS does not work, either. In this case, IZB must shut down all CICSes on this system, TORs as well as AORs, because otherwise there will be MAXTASK conditions and transaction abends.
- The implemented topology is very complex. IZB uses VSAM RLS, Adabas data sharing, and DB2 sharing groups. If one of these components suffers from poor performance, CICS also suffers and the applications have poor response times. Often it is not easy to determine where the problems are, so having a detailed understanding of the applications is very important.
- Unfortunately there are still two applications that cannot be routed dynamically: Internet home banking (HBCI), and the application that handles the connections to external partners (Routing Manager). Both applications were written by an ISV, which prevents IZB from performing maintenance without outages. IZB's Customer B is on the way to getting rid of these applications. HBCI will be replaced by a new release, and Routing Manager will migrate to WebSphere MQ.
- WLM definitions need to be carefully reviewed. It is very important to find the right priority sequence. CICS is a requestor for data, so Adabas and DB2 need to be defined in a higher service class than CICS. The CMAS address space also should have a higher priority than CICS AORs in order to fulfill requests for CPSM services as quickly as possible.

By implementing this architecture, IZB has moved closer to 24x7 availability. Even though not all applications can be routed dynamically, IZB has realized many benefits. Outages have become shorter and, in case of LPAR failures, all applications can be run on the other LPAR.

5

DB2 data sharing

IZB operates the DB2 for z/OS installations for six different clients. Each client has separate environments for development, integration, and production. Four clients also use Adabas, and one client also uses IMS/DB and DB2 for its operational data.

Today three DB2 clients use Parallel Sysplex technology with DB2 Data Sharing. The remaining three DB2 clients still use standalone DB2 environments. Overall there are 85 DB2 subsystems on 28 LPARs. 23 of them are standalone, and 62 are in Data Sharing Groups.

In 2000, IZB had two clients, who had together 25 standalone DB2 systems on eight LPARs. At that time one of these clients, known as Customer B and which also uses Adabas, decided to implement Parallel Sysplex technology together with DB2 Data Sharing. Customer B is the development company for all Bavarian savings banks,

This chapter describes the major steps that IZB followed to achieve DB2 Data Sharing for this client in its production environment. The chapter and covers the following topics:

- The Customer B DB2 configuration - then and now
- The implementation project:
  - Separating the data warehouse
  - Implementing the DWH data sharing group DB30
  - The data warehouse merge
  - Implementing the front-end environment - part 1
  - Implementing the OLTP data sharing group DB00
  - Implementing the front-end environment - part 2
- Outlook and planned projects
- Conclusions

5.1 The Customer B DB2 configuration - then and now

In 2000, the applications of the Bavarian savings banks were on three separate systems: SYPE, SYPF, and SYPG. Each system served about 30 savings banks and the applications for those were identical.

There was one DB2 system per LPAR (DSNM, DSNN, DSNP), which contained the data of all savings banks in this system. In contrast, every savings bank had its own Adabas database; see 6.1, “Adabas at IZB” on page 112 for more information about this topic.

There was also one CICS-environment (called OLTP) per LPAR. These environments contained several single CICS regions with different applications running in them (see 4.2, “CICS in IZB - then and now” on page 70). In this OLTP environment, nearly every CICS transaction connected to DB2, because some control tables used for the CICS dialog were stored in DB2.

In general, DB2 and Adabas transactions, which executed updates in both database systems, were separated from each other, because there is no two-phase commit protocol implemented between those different database managers.

Another important DB2 application was a large data warehouse (DWH), which contained the major part of DB2 objects at that time (about 30,000 tablespaces, 80,000 packages, and 600 GB DASD storage). One part of the DWH application ran on the mainframe and therefore there were separate CICS regions (DWH-CICS) for this effort.

The second part of the DWH application ran on UNIX® servers with an Oracle database, while each of the 90 savings banks had its own server. The connection to the DB2 systems was implemented with an Oracle Transparent Gateway (OTGW), which is a started task running on SYPE, SYPF, and SYPG. It connects on the one hand to DB2 using CAF (DB2 Call Attach Facility), and on the other hand through SNA and TCP/IP to the UNIX servers.

All three DB2 systems were connected to each other with DB2 Distributed Data Facility (DDF) using VTAM connections. At that time DB2 Version 6 was installed for all DB2 clients of IZB.

Figure 5-1 illustrates the DB2 configuration as it existed in 2000.

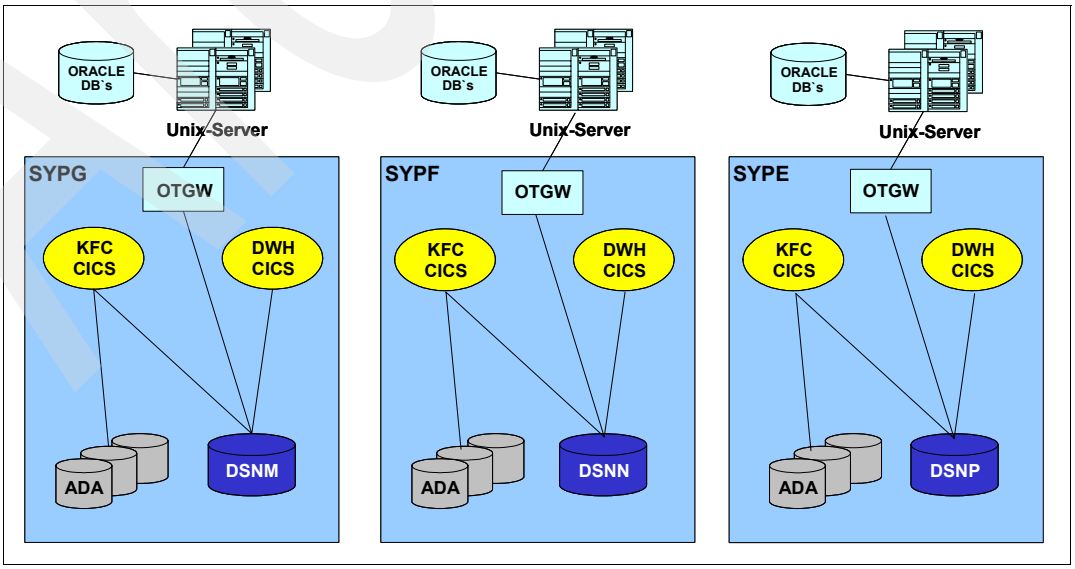


Figure 5-1 DB2 configuration in 2000

Today, the production environment of Customer B has grown to 10 systems, all of them in Parallel Sysplex Z2-Plex. One additional LPAR (SYP7) serves the applications OLTP and DWH. These LPARs (SYPG, SYPF, SYPE, and SYP7) are called “back-end systems”.

The DWH application runs separately in the Data Sharing Group DB30 for all savings banks on SYPE and SYP7 (refer to 5.2.1, “Separating the data warehouse” on page 97 and to 5.2.3, “Data warehouse merge” on page 102 for further details). The third DB2 system DSNP on SYPE was renamed to DB01 (see “Renaming DSNP” on page 105) and ran in the 2-way Data Sharing Group DB00. Today, all the DB2 systems of Customer B are running on DB2 Version 8.

All CICS regions are now consolidated into three CICS-Plex environments. On systems SYPE and SYP7, this CICS-Plex environment also spans two LPARs (see 4.4, “The implementation project” on page 73). Also, all Adabas databases on the third system are now implemented with “Adabas Cluster Services” across SYPE and SYP7 (see 6.4, “Implementation and migration” on page 119). Additionally, four front-end systems (SYP2, SYP3, SYP8, SYP9) are installed to provide the Internet and intranet access for the savings banks (see 7.1, “Choosing WebSphere on MVS” on page 138).

Several new Java applications were implemented, first by using standalone Java STCs on SYP2 and SYP3, and later by also using WebSphere Application Server on z/OS Version 3.5 and Version 5.

Therefore, the four-way Data Sharing Group DB40 was installed to enable the JDBC™ access to DB2 for these applications. For some of these Java applications, DB40 also acts as a gateway for DDF connections to the DWH environment (DB30). For details about this topic, refer to 5.2.4, “Implementing the front-end environment - part 1” on page 103. For details about the RDS application, refer to 7.3, “Implementation of WebSphere Application Server V5” on page 145.

DB40 contains the session database for savings bank users, who access servers that are centralized in IZB’s data center.

The new configuration is illustrated in Figure 5-2 on page 96.

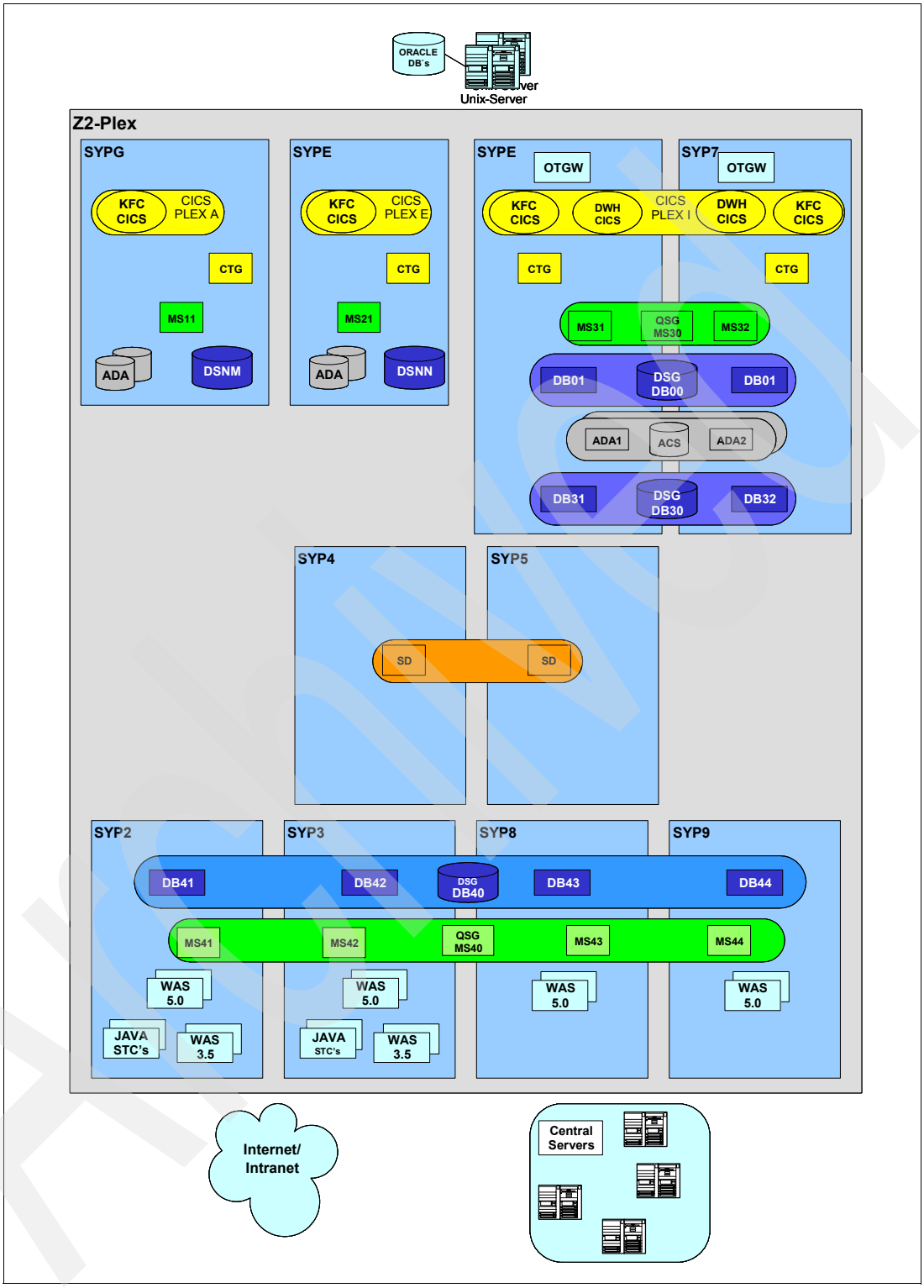


Figure 5-2 DB2 configuration in 2005

WebSphere MQ (illustrated by the green boxes in Figure 5-2) is installed on all systems. Originally it was installed to connect the front-end to the back-end applications, but at this point it also serves many new MQ applications.

For the same reason, CICS Transaction Gateway (CTG) is installed on each of the back-end-systems. The majority of the WebSphere Application Server transactions connect to the back-end systems through CTG; see 4.2, “CICS in IZB - then and now” on page 70 for more details.

Finally, there are now two “supporter systems” (SYP4 and SYP5), which serve some common work that was originally run on each of the other systems; refer to 3.4, “Network migration steps and evolution” on page 45. These two systems also provide several network connections for the entire Z2-Plex, especially the Sysplex Distributor (SD) which runs on SYP4, SYP5, and SYP5; see 3.8.3, “Sysplex Distributor” on page 58 for more information.

The following section describes in detail how IZB achieved this configuration.

5.2 The implementation project

Before the decision to implement DB2 Data Sharing was made, Customer B decided in mid-2001 to migrate the DWH application into separate DB2 systems (DB11, DB21, and DB31). In retrospect, this step was important for the later decision, because the DWH application at this time was not as critical as the applications in the OLTP environment. So DB2 Data Sharing could be implemented first in one of these new DB2 systems to gain experience before the OLTP environment was changed. The prerequisites for the later DWH merge were also put in place, as explained in the following section.

5.2.1 Separating the data warehouse

The main motivation for this separation in 2001 involved some critical DB2 crashes forced by problems in the DWH environment. This led to the outage of the complete OLTP environment.

Also, the major part of all DB2 objects belonged to DWH (see 5.1, “The Customer B DB2 configuration - then and now” on page 94) and were still growing, so the three DB2 systems had reached their limits.

Figure 5-3 on page 98 illustrates the DB2 configuration at IZB after the data warehouse separation step.

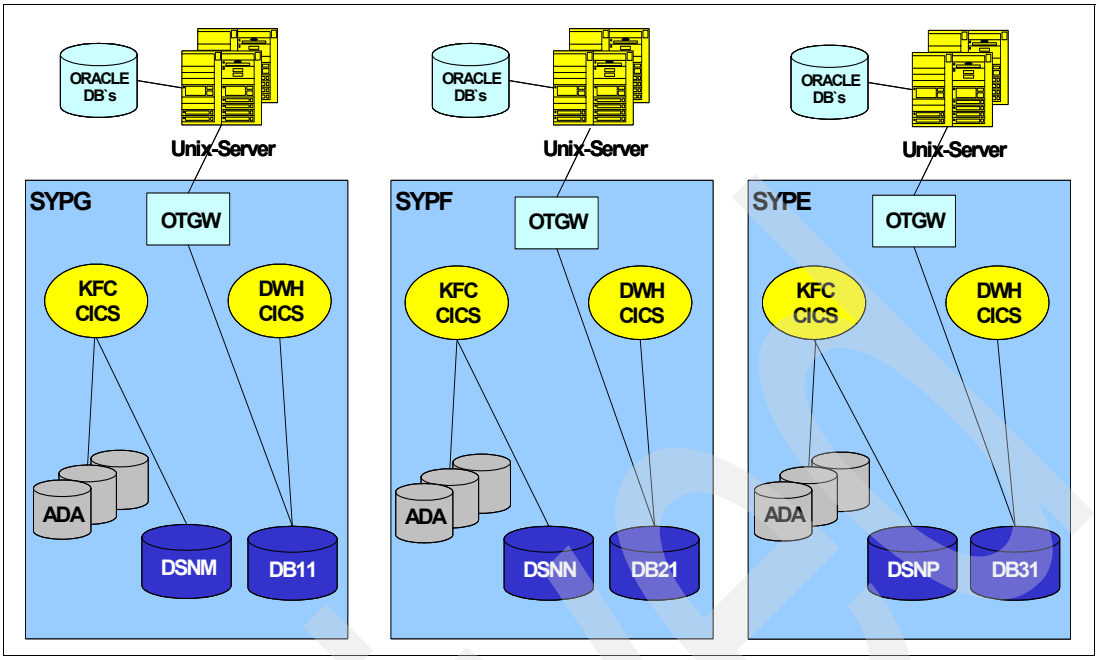


Figure 5-3 DB2 configuration - separating the data warehouse

The migration process

Because of the separation, it became necessary to migrate more than 80% of the DB2 data to the new DB2 systems. All corresponding packages and authorizations also had to be transferred. Using conventional migration techniques such as UNLOAD, RELOAD, DSN1COPY, BIND, GRANT, and so on would have taken too long and the outage time of the DWH application would have been unacceptable.

For this reason, IZB decided for the first time to use the DB2 Subsystem Cloning technique. Figure 5-4 illustrates the major steps of this approach for the DB2 systems DSNM and DB11.

Migration Part 1

Install DB11 (Catalog alias is in the same usercatalog as catalog alias DSNM
ZPARM analogous to DSNM
RACF-definitions analogous to DSNM)

Stop all CICS-Regions, DB2-Batch etc. ➔ System discontinuation

Stop DSNM, DB11

DFDSS DUMP CATALOG
DIRECTORY
BSDS
1 ACTIVE LOG (STATUS=NOTREUSABLE)

DFDSS DUMP some special Userdata (Control-Tables etc.)

Start DSNM

Stop all DWH-Databases in DSNM

Start all KFC CICS-Regions, DB2-Batch etc. ➔ System (without DWH) available

Figure 5-4 DB2 migration process - part 1

At this point the OLTP environment became completely available. The time for the outage depended on the size of the DB2 catalog. In the IZB case, it took approximately one hour. The next step affected only the DWH environment; see Figure 5-5.

Migration Part 2

DFDSS RESTORE	CATALOG, DIRECTORY, BSDS, LOG, special userdata with new catalog alias (DB11)
VSAM ALTER	all DWH TS/IX to new catalog alias (30.000 datasets)
BSDS-Update:	VCAT DDF Delete all “old” ACTIVE LOG entries Define ACTIVE LOG from DSNM with START RBA and END RBA Define new “empty” ACTIVE LOGS

Start DB11

DROP/CREATE WORK-Database (DSNDB07) using new catalog alias

FULL-IC CATALOG, DIRECTORY

Switch the Storage Group for all DWH TS/IX to new catalog alias in the DB2 catalog:
CREATE STOGROUP STGnew using new Catalog alias (DB11)
ALTER all DWH TS/IX using STGnew
DROP old STOGROUP STGold
CREATE STOGROUP STGold using new Catalog alias (DB11)
ALTER all DWH TS/IX using STGold
DROP STOGROUP STGnew

Accommodate connections (DWH-CICS, DWH-BATCH, OTGW, DDF)

Start DWH-CICS, DWH-BATCH, OTGW ➔ System completely available

Figure 5-5 DB2 migration process - part 2

Now the DWH environment also became completely available. The outage during the DB2 migration process - part 2 was approximately two to three hours. The most expensive step was VSAM ALTER.

Finally, both DB2 systems had to be cleaned up, as illustrated in Figure 5-6.

Migration Part 3

DROP DWH-Databases in DSNM

FREE DWH-Packages and Plans in DSNM

REORG CATALOG, DIRECTORY in DSNM (smaller)

Adjust ZPARM DSNM (DSMAX, EDMPOOL, …)

DROP KFC-Databases in DB11

FREE KFC-Packages und Plans in DB11

REORG CATALOG, DIRECTORY in DB11 (smaller)

Adjust ZPARM DB11 (DSMAX, EDMPOOL, …)

Figure 5-6 DB2 migration process - part 3

5.2.2 Implementing the DWH data sharing group DB30

By 2002 the configuration, which was using three systems, each serving 30 savings banks, had reached its limit. System SYPE in particular grew extremely large, and even a hardware change using the biggest machine available could not provide the necessary capacity for the suggested workload.

A major reason for this extraordinary growth was a series of consolidations of savings banks, where the “surviving” institute remained on SYPE. Additionally, the Parallel Sysplex aggregation rules were retained for all machines at IZB’s data center (see 1.3.2, “Managing costs” on page 9 for more information about this topic). As a consequence, SYPE had to be downsized in order to enable the installation of a second productive system of another IZB client on the same machine.

Furthermore, an increasing number of 24x7 applications, including new Java applications on z/OS, required uninterruptible operations. The existing configuration did not provide any opportunity to take over the workload of a failing system. In addition, system maintenance without an outage was not possible.

Consequently, IZB decided to install a fourth system (SYP7) in Customer B’s environment (see 1.4, “Description of the project” on page 9 for more details). In contrast to the standalone systems (SYPG, SYPF, and SYPE), this new system had no dedicated connection to a part of the 90 savings banks. It was implemented with SYPE in the new Parallel Sysplex Z2-plex. The aim was to move parts of the workload of SYPE to SYP7 without changing the established three-part architecture of SYPG, SYPF, and SYPE. To achieve this aim, a decision was made to implement DB2 Data Sharing between SYPE and SYP7.

The first DB2 system that would be enabled was the Data Warehouse DB2 DB31, with the aim of gaining experience with DB2 Data Sharing in this production environment.

Meanwhile, IZB had migrated all of Customer B’s DB2 systems to DB2 Version 7.

Prerequisites

As the first part of the implementation, IZB changed its DB2 naming conventions for Customer B. This meant, essentially, that the DB2 subsystem name was changed from DSNx (x = M,N,P,T...) to DByz (y=number of DS Group, z=number of a member in this group).

As a result, the GROUPNAME (DSNDByz), IRLM XCF NAME (DXRDByz) and the name of the WORKFILE DB (WRKDByz) were derived from this naming convention. These new names were already taken into account for the new DB2 systems DB11, DB21, and DB31. So this prerequisite was fulfilled for DB31 and no special action had to be taken.

Next, the DB2 CF structures had to be defined. For the sizing, IZB used the ROT and appropriate formulas as described in *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935, which is available on the Web at:

http://www-1.ibm.com/support/docview.wss?uid=pub1sc26993503

All Group Buffer Pool structures were defined with DUPLEX(ENABLED).

Enabling DB31

Example 5-1 on page 101 gives an overview of the implementation process to enable DB31.

Example 5-1 Enable DB31

```
Run the DB2 installation dialog (DATA SHARING FUNCTION: 3 (ENABLE))
Stop CICS and Batch Connections
Stop DB31
Run Job DSNTIJIN - ALTER LOG data setS (SHAREOPTIONS(2,3))
Run Job DSNTIJUZ - DSNZDB31, DSNHDECP and Update BSDS
Adjust procedure DB31MSTR - GROUP(DSNDB30),MEMBER(DB31)
Adjust procedure DB31IRLM - IRLMGRP(DXRDB30),SCOPE=GLOBAL
Start DB31
Run Job DSNTIJTM - Create Work-DB
Run Job DSNTIJTC - Image Copy catalog and directory
Start CICS and Batch Connections
```

The outage for this process was approximately 30 minutes. DB30 was now a one-way Data Sharing Group with member DB31. The system overhead was minimal; GROUP Bufferpools were not allocated in this phase and IZB did not detect any measurable performance degradation in its DWH applications.

Adding second member DB32

Now the member DB32 on system SYP7 had to be installed; see Example 5-2. This task was performed without an outage.

Example 5-2 Adding second member DB32

```
Define DB32 to RACF, VTAM, TCP/IP, ...
Run the DB2 installation dialog (DATA SHARING FUNCTION: 2 (ADDING A MEMBER))
Run Job DSNTIJIN - Define BSDS and LOG data sets
Run Job DSNTIJID - Add LOGs into BSDS and NEWCAT DSNDB30
Run Job DSNTIJUZ - DSNZDB32 and Update BSDS
Adjust procedure DB32MSTR - GROUP(DSNDB30),MEMBER(DB32)
Adjust procedure DB32IRLM - IRLMGRP(DXRDB30),SCOPE=GLOBAL,IRLMID=2
Start DB32
Run Job DSNTIJTM - Create Work-DB
```

The new System Configuration with DSGroup DB30 is illustrated in Figure 5-7 on page 102.

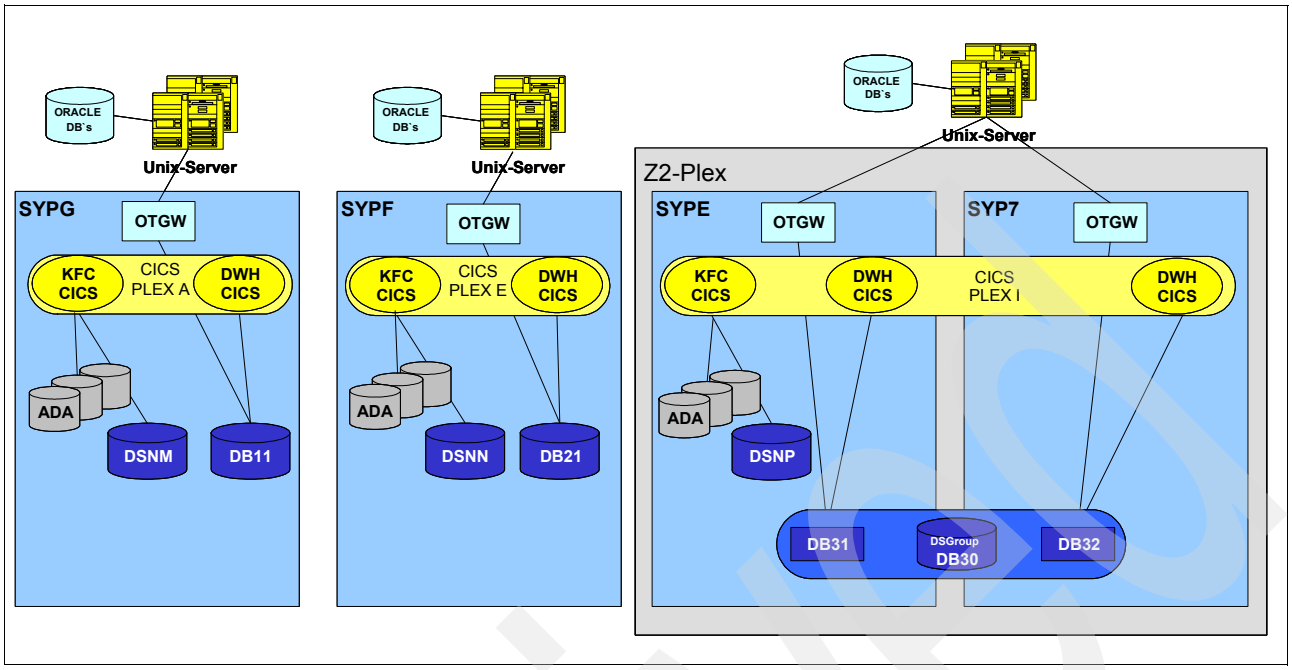


Figure 5-7 DB2 configuration with DWH data sharing group DB30

Adapting connections to DB30

After the installation of DWH DSGroup DB30, the external connections had to be expanded.

Meanwhile the CICS regions on SYPG, SYPF, and SYPE were consolidated to CICSPlex environments (CICSPlex A, CICSPlex E, CICSPlex I). Additionally CICSPlex I on Z2-PLEX was now expanded to a multi-site CICSPlex using a second DWH CICS on SYP7, which was connected to DB32. (The installation of CICSPlex I is described in 4.4.2, “Building a CICSPlex” on page 77.

A second Oracle Transparent gateway (OTGW) was also installed on SYP7. The network connections from the corresponding UNIX servers were changed using VTAM Generic Resources for the SNA Connections and Sysplex Distributor for the TCP/IP connections.

Next, all DDF connections (Customer B uses only SNA connections) from the other five standalone DB2 systems (DSNM, DB11, DSNN, DB21, and DSNP) into DB30 were changed to DDF Member Routing. Consequently this workload is distributed between DB31 and DB32. Refer to *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935, for details regarding how to implement DDF Member Routing. It is available at:

http://www-1.ibm.com/support/docview.wss?uid=pub1sc26993503

Finally, the DWH Batch Jobs on Z2-Plex were adapted. Here the DB2 Group Attach name was used, instead of the DB2 subsystem name. For the dynamic routing of these jobs between SYPE and SYP7, IZB used WLM-managed initiators and specific DWH schedule environments (see 9.4, “Managing workload” on page 196, for more information).

5.2.3 Data warehouse merge

With the installation of the data sharing group DB30, the DWH environment became nondisruptive and scalable for one-third of the savings banks. In order to provide the same advantages for the remaining 60 savings banks, Customer B decided in late 2002 to merge the standalone DB2 systems DB11 and DB21 into the data sharing group DB30.

The migration process

Because SYPG and SYPF were not part of the Z2-Plex at that time, IZB installed a shared DASD pool between SYPG, SYPF, and Z2-Plex. Then, during two separate weekends (the DWH application was stopped for three days in each case), the DWH data was transferred to DB30 using conventional DB2 utilities such as DSNTIAUL, UNLOAD, DSN1COPY, and LOAD.

At the same time the definitions for the OTG connections (SNA and TCP/IP) of the savings banks running on SYPG and SYPF had to be changed. After the merge, these requests were completely transferred to SYPE and SYP7, respectively.

The same was done for all DWH CICS transactions running on SYPG and SYPF. After the merge those were transferred from CICSPlex A and CICSPlex B to the remaining DWH CICS regions in CICSPlex I, and the DWH CICS regions on SYPG and SYPF were uninstalled.

Finally, the DWH Batch Jobs on SYPG and SYPF had to be transferred to Z2-Plex. Here the DB2 subsystem name had to be replaced by the Group Attach Name (DB30). For the dynamic routing of these jobs between SYPE and SYP7, the corresponding WLM-managed initiators and schedule environments had to be adjusted (see “Adapting connections to DB30” on page 102).

The resulting system configuration is illustrated Figure 5-8.

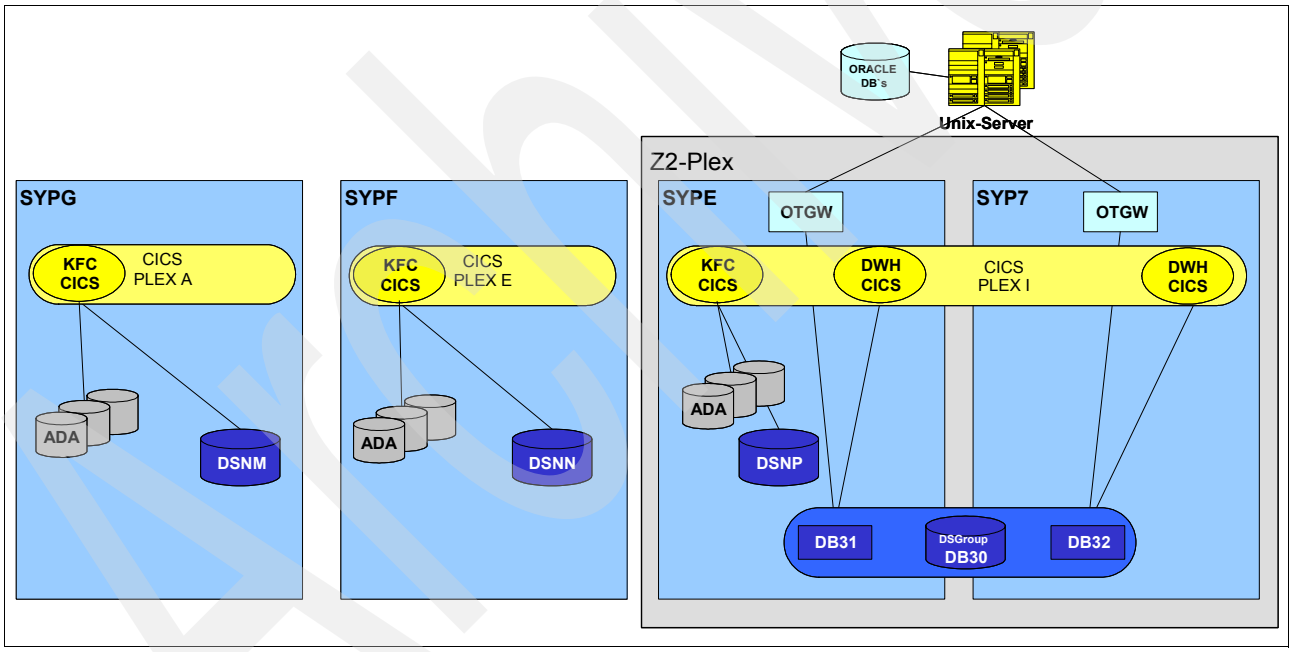


Figure 5-8 DB2 configuration after data warehouse merge

5.2.4 Implementing the front-end environment - part 1

Within the scope of the installation of Z2-Plex and the building of the new back-end environment between SYPE and SYP7 in 2002, two more LPARs (SYP2 and SYP3) were installed into Z2-Plex. These new systems provided the Internet and intranet access for the savings banks.

The first applications used WebSphere Application Server Version 3.5 and some standalone Java STCs, as described in 7.2, “Implementation of WebSphere Application Server V3.5” on page 138. The communication between those Java applications and the back-end

environment was done mainly using CICS Transaction Gateway (GTG), but other applications used WebSphere MQ; see 4.2, “CICS in IZB - then and now” on page 70.

The access to the back-end data (DB2 and Adabas) was done locally on SYPG, SYPF, and SYPE by CICS transactions. The idea was that, for security reasons, no operational data would be stored in the front-end environment.

The first Java application using WebSphere Application Server 3.5 was implemented in early 2003, and it used a third way of communicating to the back-end environment. This application used the DB2 JDBC Legacy Driver (also called the JDBC Version 2.0 driver) to connect to a local DB2 system, and then connected to the back-end DB2 Data Sharing Group DB30 (DWH). Therefore, a second DB2 Data Sharing Group DB40 was installed on SYP2 and SYP3, which at first acted only as a gateway to DB30. To exploit the sysplex capabilities for this new connection, DDF Member Routing was used for the corresponding DDF definitions (see “Adapting connections to DB30” on page 102).

The access to the back-end environment using this third way remains restricted to the DWH environment DB30. Access to the OLTP data was still done by using CTG or WebSphere MQ.

The new System Configuration is illustrated in Figure 5-9; the MS11, MS21, MS31, MS41 and MS42 shown are the new WebSphere MQ systems.

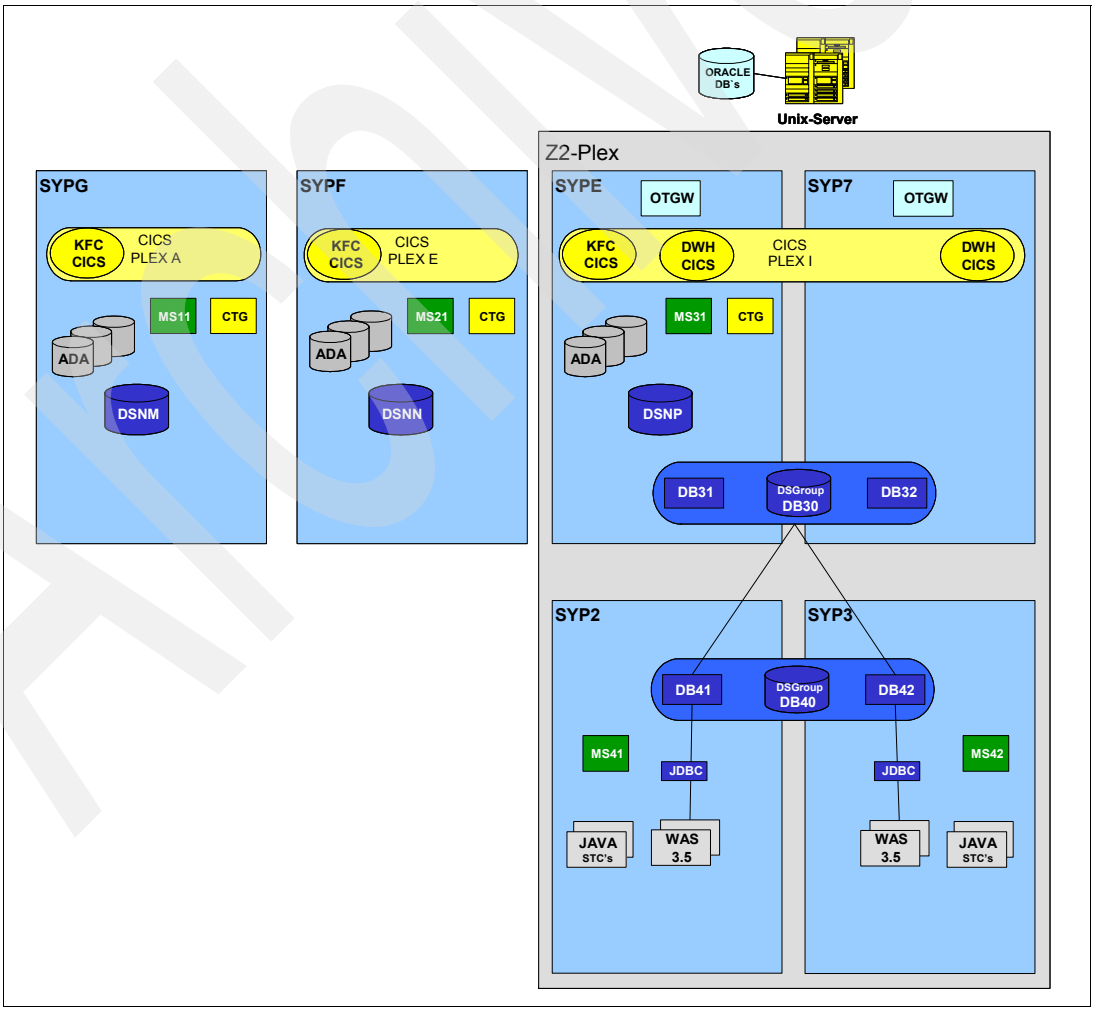


Figure 5-9 DB2 configuration with the first front-end environment

5.2.5 Implementing OLTP data sharing group DB00

After the successful installation of the Data Sharing Groups DB30 (DWH) and DB40 (front-end), this new environment ran without significant problems. Therefore, in mid-2003, Customer B decided to implement the Parallel Sysplex technology in their OLTP environment.

If one were following the steps for the DWH environment, the first step in this project should have been the migration of the OLTP DB2 system DSNP on SYPE. However, DSNP did not match the Customer B naming conventions for DB2 Data Sharing (see “Prerequisites” on page 100). So IZB decided to rename this DB2 system to DB01.

Renaming DSNP

As the first step in this process, IZB modified the IEFSSNxx member for this subsystem in order to define the existing subsystem name as the group attachment name. The definition in this member was originally:

```
DSNP,DSN3INI,‘DSN3EPX,-DSNP,M’
```

This was changed to:

```
DSNP,DSN3INI,‘DSN3EPX,-DSNP,S,DSNP’
```

This change could only be activated by an IPL. Also, the new Data Sharing Group DB00 was defined accordingly in IEFSSNxx:

```
DB01,DSN3INI,‘DSN3EPX,-DB01,S,DSNP’
DB02,DSN3INI,‘DSN3EPX,-DB02,S,DSNP’
```

With these definitions, IZB avoided changing any existing application that used the DB2 subsystem name DSNP (with the exception of CICS).

The second step, renaming DSNP to DB01, was done together with the process of enabling Data Sharing for DB01 because, for both actions, DB2 must be down. For details of this process, see “Enabling DB31” on page 100.

The checklist shown in Figure 5-10 on page 106 gives a brief description of this rename process.

Rename DSNP

Install DB01	Catalog alias is in the same usercatalog as catalog alias of DSNP		
Stop all KFC CICS-Regions, DB2-Batch etc.		➔	System discontinuation
Stop DSNP			
RENAME	DB2 Catalog, Directory, Work-DB, BSDS, 1 ACTIVE LOG (STATUS=NOTREUSABLE) to new catalog alias		
Update BSDS	Delete old LOGs, Define new LOGs		
Adjust procedure	DSNPMSTR (define new BSDS-names)		
Run DSNTIJUZ	define new catalog alias in DSNZPARM of DSNP		
Start DSNP	DSNP with new catalog alias available		
DROP/CREATE	Tablespaces in Work-DB using the new catalog alias		
STOP DSNP			
VSAM ALTER	all user TS/IX to new catalog alias (ca. 10.000 datasets)		
DB2 ALTER	all user TS/IX to new catalog alias (using STOGROUP)		
START DB01			
Adjust CICS	change DB2 attach to DB01		
Start all KFC CICS-Regions, DB2-Batch etc.		➔	System available

Figure 5-10 Rename DSNP to DB01

The process of renaming a DB2 system is described in detail in *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935, which is available on the Web at:

http://www-1.ibm.com/support/docview.wss?uid=pub1sc26993503

The major task for this maintenance window was the VSAM ALTER of all user tablespaces and indexes; Figure 5-5 on page 99 shows details of this task. As mentioned, DB01 was also enabled for data sharing in the same maintenance window. So overall, the outage in this case was approximately two hours.

Finally, the second member DB02 was installed, as well as the first Adabas Cluster Services Installation (ACS). The first WebSphere MQ “Queue Sharing” implementation was also introduced on Z2-Plex. The new system configuration is illustrated in Figure 5-11 on page 107.

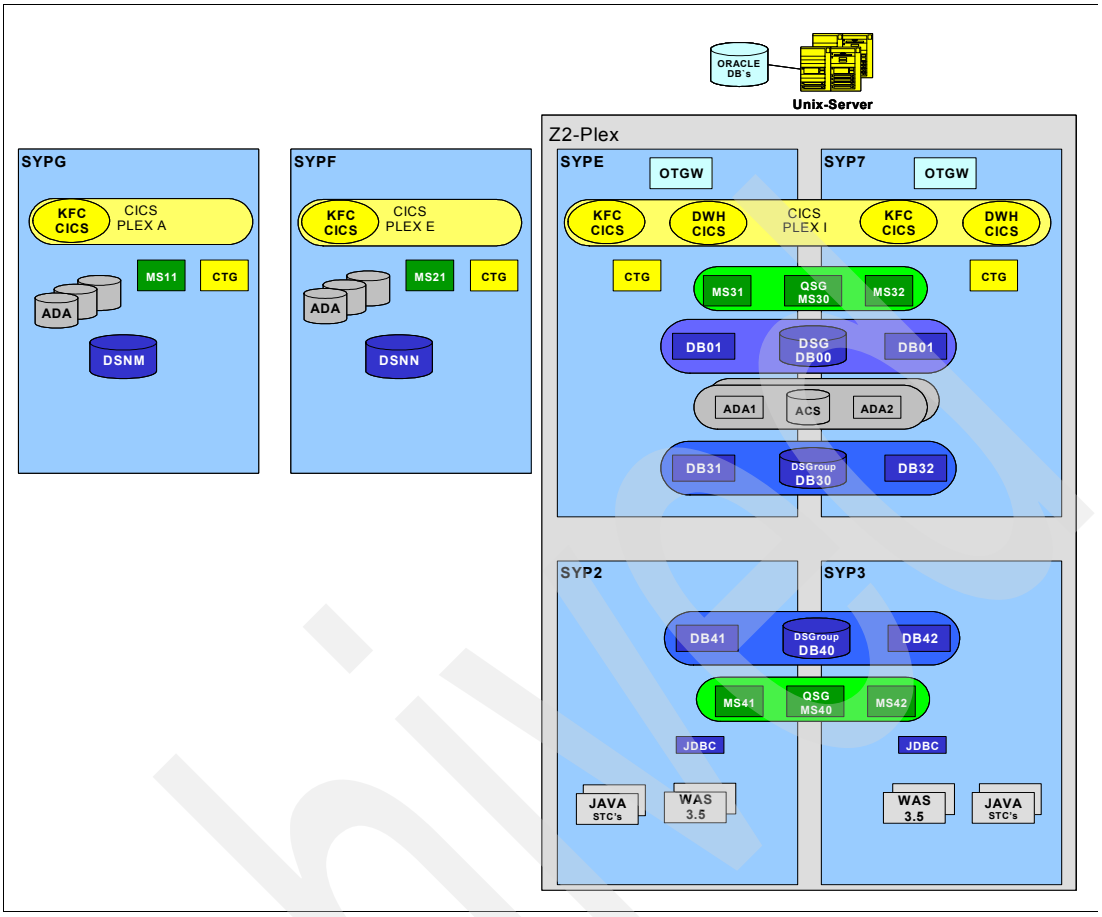


Figure 5-11 DB2 configuration with OLTP data sharing group DB00

As described in “Adapting connections to DB30” on page 102, the connections to DSNP (CICS, DDF and Batch) were adapted accordingly, meaning that a second OLTP CICS environment was implemented on SYP7, all DDF connections into DSNP were changed to DDF Member Routing, and the batch workload accessing DSNP was changed to use WLM-managed initiators and specific OLTP schedule environments.

From this point onward, IZB was able for the first time to completely exploit the advantages of Parallel Sysplex technology for systems SYPE and SYP7 in the Z2-Plex.

Now it became possible to do the following:

- Take over the work of a failing system by the remaining system
- Install system maintenance (z/OS, CICS, Adabas or DB2) without an outage
- Install a third system in the case of a strongly growing workload (scalability)
- Use WLM to spread the workload between SYPE and SYP7

The process of introducing sysplex technology for the back-end environment of Customer B stopped at this point. The remaining systems SYPG and SYPF are still standalone today. In mid-2004, both systems were merged into Z2-Plex. Nevertheless, Customer B had other priorities that prevented the completion of converting the whole back-end environment. One reason for this delay is that IZB determined that this environment, with DB2 and Adabas data sharing CICS-transactions, needs about 10% more CPU time.

5.2.6 Implementing the front-end environment - part 2

After the introduction of the first Java™ applications on systems SYP2 and SYP3 using WebSphere Application Server 3.5, Customer B continued in this environment. Meanwhile, several important applications were implemented on this platform, for instance the main “Home banking” application of Customer B runs in this area. The implementation of another application (RDS, using WebSphere Application Server Version 5) made it necessary to install a third and a fourth LPAR (SYP8 and SYP9) in the front-end environment. This application is a gateway to several servers, which are centralized in IZB’s data center. For this purpose, the front-end Data Sharing Group DB40 had to be expanded to include SYP8 and SYP9, so today this is a 4-way Data Sharing Group.

With the introduction of WebSphere Application Server V5, the new DB2 Universal JDBC Driver (here called JCC), which was introduced in an APAR for DB2 V7, is used in the corresponding applications. Both JDBC Drivers (the Legacy Driver in WebSphere Application Server 3.5 applications, and the JCC Driver in WebSphere Application Server V5 applications) operate independently of each other. The JCC Driver is used only local (Type 2 connection), and the Legacy Driver is only used local, as well. Most of these applications use JDBC, while some new applications also use SQLJ.

In general, the front-end environment today reflects the environment of Customer B, where most of the changes and new applications were implemented.

The actual DB2 system configuration is illustrated in Figure 5-12.

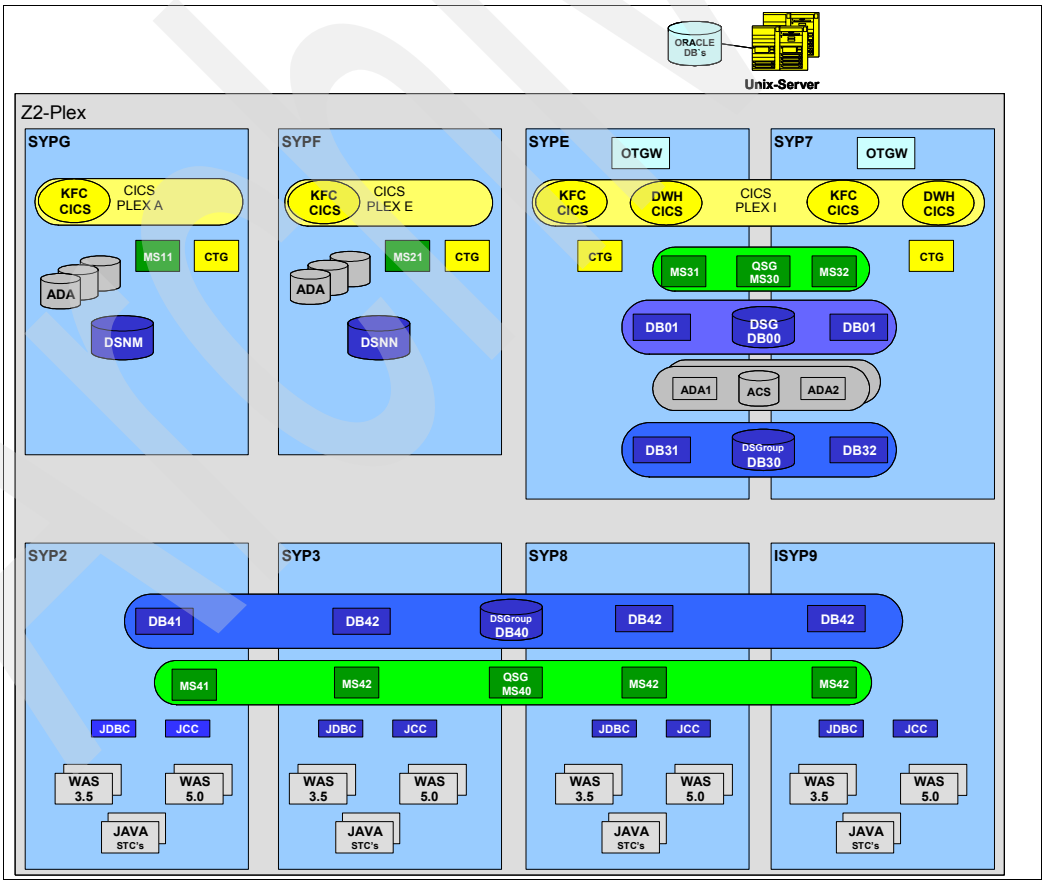


Figure 5-12 DB2 configuration with complete front-end environment

5.3 Outlook and planned projects

The back-end systems SYPG and SYPF remain standalone, from the perspective of DB2 Data Sharing. The plan is to change over the OLTP environment on these systems to data sharing in the 2006 - 2007 time frame. In the meantime, both systems are merged into Z2-Plex and the intention is to retain the established separation of the savings banks into three systems.

Therefore, the existing Data Sharing Group DB00 will probably not be expanded into a 4-way group with two new members on SYPG and SYPF. The DB2 systems DSNM and DSNN will be renamed (DB11 and DB21) instead and then installed into two further 2-way data Sharing Groups DB10 and DB20. The same thing is planned for the CICSPlexes, Adabas databases, and WebSphere MQ systems on SYPG and SYPF.

On the other hand, the workload out of the front-end environment into the DWH Data Sharing Group DB30 is still growing, and therefore the plan is to expand DB30 to a 4-way group with two new members on SYPG and SYPF. In this case the front-end workload will be directed into these two new members in order to separate this from the DDF workload that comes from the OLTP environment. Later, the plan is to implement the JDBC Type 4 Connection (using the JDBC Universal Driver) for this effort. This will result in a significant performance improvement, because the roundabout way, using the gateway DB2 Data Sharing Group DB40, will not be necessary.

The planned DB2 configuration for the 2006 - 2007 time frame is shown in Figure 5-13.

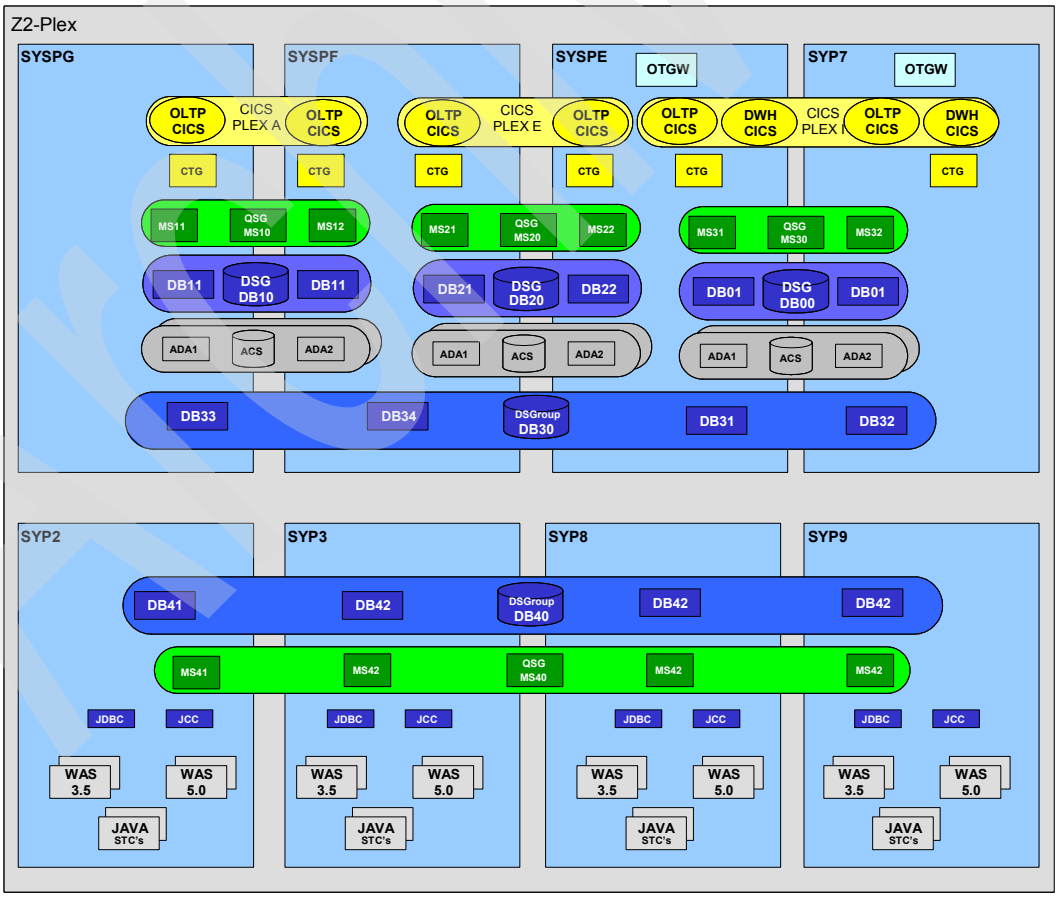


Figure 5-13 Planned DB2 configuration

5.4 Conclusions

DB2 Data Sharing was introduced for IZB's Customer B without any significant problems. However, the whole migration took quite a while, beginning in 2001 with the first plans. And although it was not difficult to install the new Data Sharing Groups or rename the existing DB2 systems, prior to these activities all affinities in the applications had to be removed, which took about six months in 2002.

The installation today is quite complex. In order to profit from the benefits of Parallel Sysplex technology, all components had to be changed. Nevertheless, at this point all systems up to systems SYPG and SYPF can be expanded if necessary (improving scalability), maintenance can be installed without an interruption, workload can be transferred in the case of an outage (improving availability), and workload in general can be spread across several systems.

Tip: The decision to use the old DB2 subsystem name DSNP (instead of DB00) for the group attachment name was not a good choice, because today this "old" name often makes the administration more complex and the environment not clearly arranged.

For more information about this topic, see 5.2.4, "Implementing the front-end environment - part 1" on page 103.

6

Adabas data sharing guideline

IZB operates several database management systems: Adabas, DB2, and IMS DB. This chapter documents the IZB experiences when enabling the sysplex capabilities of Adabas, which is a product from Software AG (SAG).

IZB is one of the world's largest users of Adabas for database management. As a result, IZB enjoys a good relationship with Software AG's support and development team. This is one reason why IZB was willing to accept the risk of installing this new, non-IBM-product.

IZB was the first client in the world to use the Software AG product Adaplex. (Adaplex was the product name in 2000, when IZB began to test this product. Today the official name is Adabas Cluster Services, and the Adabas version is 742.) An IZB customer reference for Software AG is available at this site:

http://www1.softwareag.com/ch/Customers/References/D_IZB_page.asp

In this chapter, the IZB Adabas database administration team provides a guideline for introducing Adabas Cluster Services into a z/OS sysplex environment. The focus is on the configuration and implementation of IZB's largest client Customer B, the company that is responsible for the application development for all Bavarian savings banks.

6.1 Adabas at IZB

Adabas is the largest and most important database management system (DBMS) at IZB. All data concerning accounts, addresses, exchanges and so on is stored in Adabas. The first implementation was in 1974; at that point Adabas was the only DBMS at IZB. Since then, IZB's clients have been very satisfied with this product regarding availability and response times.

At IZB, the Adabas database administration team administers the data in Adabas for about 80 savings banks. This team is also responsible for installing and maintaining all Software AG products. Since March 2001, each savings bank has had its own Adabas database. Each LPAR contained one-third of the savings banks with their dedicated Adabas databases. All the Adabas databases have the same physical structure with the same number of Adabas files and the same file description. Adabas files are comparable to DB2 tables.

In a normal environment, there is one Adabas address space for one Adabas database using only one processor. Figure 6-1 illustrates a simple Adabas, CICS, and batch environment, along with its communication structure.

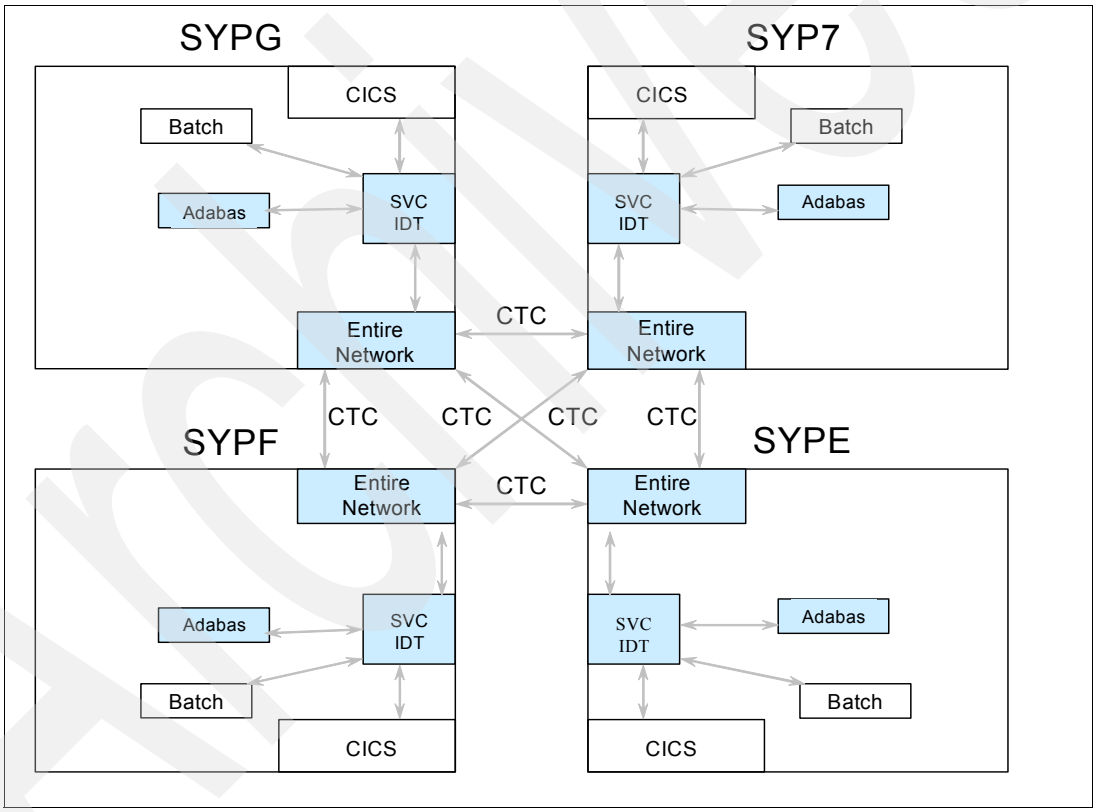


Figure 6-1 Adabas communication paths

The Adabas calls are made over an SVC using cross-memory-services. The Adabas SVC holds a piece of memory in ECSA named ID-table (IDT) to route the user's Adabas calls against the desired Adabas address space. The routing criteria always is the DBID. This is the unique Database Identification number in an Adabas environment.

Adabas uses cross-memory-services for communication. Therefore, there is no requirement for where the user has to be located (as is necessary for CICS, batch, or TSO). The Entire Network product is described in 6.5.2, "SAG's Entire Network product" on page 124.

To provide an overview of the applications using Adabas at Customer B, Table 6-1 lists the daily workload using Adabas and the quantity structure of the Adabas environment. Even with a large number of Adabas databases, Adabas files, and transactions, IZB can provide very good response times without any failures.

Table 6-1 Adabas operating figures

Operating figures	Production	Test development integration
Number of batch jobs using Adabas per day	25,000	6,000
DASD space used for Adabas-DBs (BG)	3,500	200
Number of Adabas files	7,000	2,500
Number of Adabas databases	94	32
Number of Adabas calls per day	630 M	200 M
Number of Natural DDMs	2,000	3,100
Number of Natural objects	5,000	8,000
Number of CICS transactions per day	12,000,000	350,000
Number of MIPS	7,700	1,000
Number of LPARs using Adabas	4	4
Total DASD space used (GB)	29,500	4,000

Natural is a 4GL language and a product of SAG. Many daily batch jobs and TSO applications access the Adabas databases by using Natural, but most online transactions at Customer B that access Adabas use Assembler programs.

Natural has a runtime environment. Using this programming language, applications can be developed or changed and rolled out into production very easily. At Customer B, Natural is used to access Adabas through batch and TSO. Another customer of IZB has a very large application written in Natural running under CICS. Natural is also able to access DB2 and there are many applications which use this feature. There are also many Natural applications accessing both Adabas and DB2.

6.2 Why implement database cluster services

There were several reasons behind the IZB decision to implement Parallel Sysplex and Adabas Clustering Services in its System z mainframe environment. First of all, IZB wanted to save money. This could be achieved by connecting all production LPARs to a Parallel Sysplex and reducing software license costs. But setting up a Parallel Sysplex required the installation of Coupling Facilities due to the IBM pricing policy (Priceplex). So IZB obtained Coupling Facilities and moved its LPARs into a Parallel Sysplex environment.

In CICSPlex, Adabas could still operate in noncluster-mode using the Entire Network product. But the response time would be increased by 50% for users not connected locally to the system. This was caused by heavy network traffic over CTCs and VTAM. (For more details about Entire Network, refer to Chapter 3, “Network considerations” on page 43.) To avoid this problem, IZB decided to use Adabas Cluster Services.

Another reason was based on the project Permanent Online Processing (POP). The goal of this project was to provide higher availability for the applications. Using Adabas Cluster

Services, it is possible to apply service packs without any application outage. Also it is possible to distribute workload dynamically across multiple LPARs.

In a normal Adabas environment, only one Adabas address space operates for one Adabas database using only one physical processor. In an Adabas Cluster Services environment, however, it is possible to start up to 32 Adabas address spaces residing on different LPARs within a Parallel Sysplex accessing one Adabas database. In order to achieve these benefits for the IZB environment, though, it was planned to have only one Adabas address space per LPAR and two address spaces accessing one Adabas database.

6.3 Requirements for introducing Adabas Cluster Services

The following prerequisites needed to be met before IZB could introduce the use of Adabas Cluster Services.

6.3.1 Coupling Facility

The main prerequisite needed to implement data sharing using Adabas Cluster Services was the installation of an external Coupling Facility. The Adabas address spaces use structures in the Coupling Facility to exchange information. For each Adabas database, one Lock and one cache structure are needed.

In order to achieve good performance, Coupling Facility processors should have the same speed (or higher) as the LPAR processors. There must also be sufficient memory space in the Coupling Facility to allocate all structures for the 30 Adabas databases. (Refer to Software AG documentation for the formula to calculate the required space in the Coupling Facilities.)

Figure 6-2 on page 115 illustrates the actual amount of space needed in the IZB production Adabas Cluster environment.

Example for DBID 209

CACHE structure

```
STRNAME: ADA209CACHE
 STATUS: ALLOCATED
 TYPE: CACHE
 POLICY INFORMATION:
  POLICY SIZE    : 65536 K
  POLICY INITSIZE: 32768 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : DISABLED
  PREFERENCE LIST: CFNG1    CFNH1
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

LOCK structure

```
STRNAME: ADA209LOCK
 STATUS: ALLOCATED
 TYPE: LOCK
 POLICY INFORMATION:
  POLICY SIZE    : 65536 K
  POLICY INITSIZE: 32768 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : DISABLED
  PREFERENCE LIST: CFNG1    CFNH1
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY
```

Figure 6-2 Example of lock and cache structure

IZB classified its Adabas Databases into three groups: large, medium, and small. The example shown in Figure 6-2 represents a large database. Here the amount of storage needed in the Coupling Facility is 65536 KB for the lock structure and 65536 KB for the cache structure.

The total amount of space in the Coupling Facility for all IZB Adabas databases running in cluster mode is approximately 2600 megabytes. The IZB Coupling Facility is described in more detail Chapter 9, “System” on page 177.

6.3.2 More space

Each Adabas database, as illustrated in Figure 6-3 on page 116, uses a set of sequential data sets. If two Adabas nuclei are used to access one Adabas database, then the amount of space needed will increase about 15% because each Adabas address space needs its own WORK data set and its own PLOG (protection log) data sets. Additionally, INTERMEDIATE data sets to consolidate recovery information are required.

Example of Adabas Database DBID 057

```
IZTADA.IZ057.PINTERI          additional (same size as PLOG)
IZTADA.IZ057.PINTERO          additional (same size as PLOG)
IZTADA.IZ057.PLOG01.X01
IZTADA.IZ057.PLOG01.X02       additional
IZTADA.IZ057.PLOG02.X01
IZTADA.IZ057.PLOG02.X02       additional
IZTADA.IZ057.WORK.X01
IZTADA.IZ057.WORK.X02         additional
```

Adabas DBID
is the unique database identifier

Figure 6-3 Example - space requirements

6.3.3 New naming conventions

As shown in Figure 6-4, new naming conventions had to be defined. The most important task was to create unique data set names. IZB completed this part of the project in 2000.

Data Set Names

Operation in cluster mode

IZPADA.ISnnn.ASSO
IZPADA.ISnnn.DATA
IZPADA.ISnnn.WORK.X01
IZPADA.ISnnn.WORK.X02
IZPADA.ISnnn.PLOG.X01
IZPADA.ISnnn.PLOG.X02

Operation in non-cluster mode

IZPADA.ISnnn.ASSO
IZPADA.ISnnn.DATA
IZPADA.ISnnn.WORK.X00
IZPADA.ISnnn.PLOG.X00

nnn = unique DBID (1-254)
within the entire network

It is possible to switch from cluster mode to noncluster mode and vice versa.

Figure 6-4 New naming conventions for data sets

Using these new naming conventions made it easier and more feasible to install and handle the a Adabas Cluster Environment.

6.3.4 A separate Adabas version

In 2000, few Software AG clients were interested in running its Adabas environment with Cluster Services; IZB was one of the first. Therefore, Software AG decided to develop a separate release in parallel to the normal release. This caused additional effort to maintain the Adabas environment for a long period of time. The consolidation of these two releases, illustrated in Figure 6-5 on page 117, was done with Adabas version 742 in 2004. Refer to SAG documentation for information about how to install and implement Adabas Cluster Services.

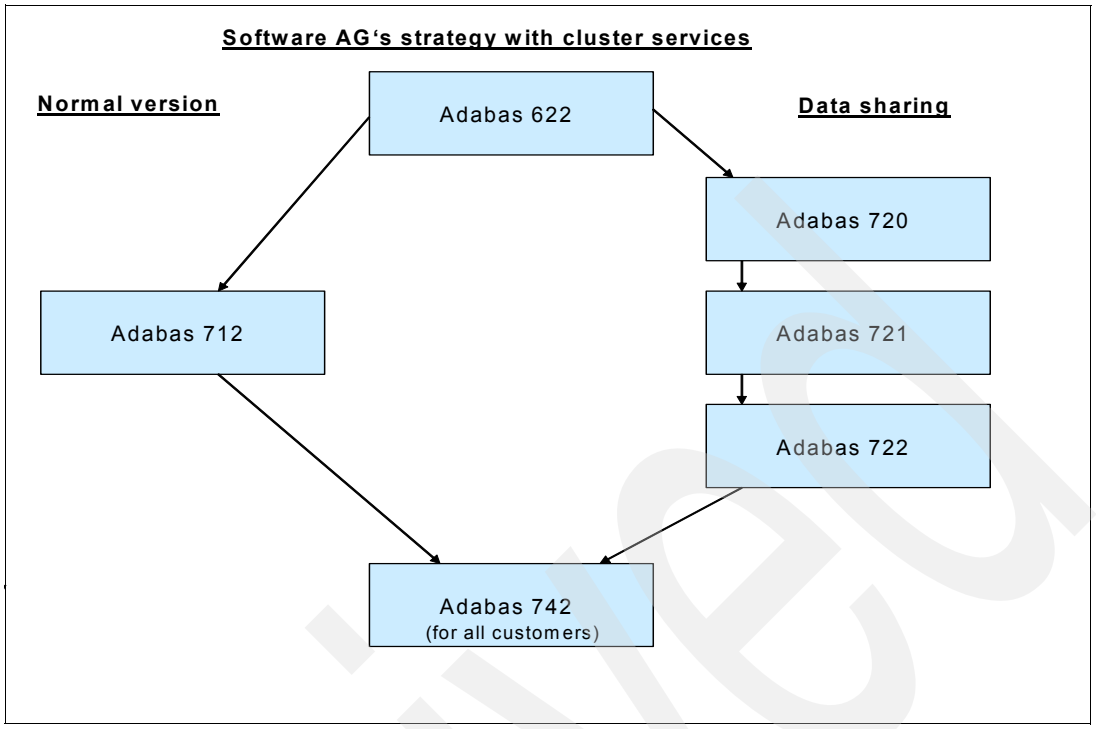


Figure 6-5 Software AG's release strategy

6.3.5 A new LPAR

The existing system environment consists of three production LPARs. The 90 Adabas databases were equally distributed to these LPARs. IZB decided to take one of these LPARs to start implementing data sharing. In August 2002 an additional LPAR SYP7 was created to build a two-way data sharing environment; refer to Chapter 9, "System" on page 177 for more information about this topic.

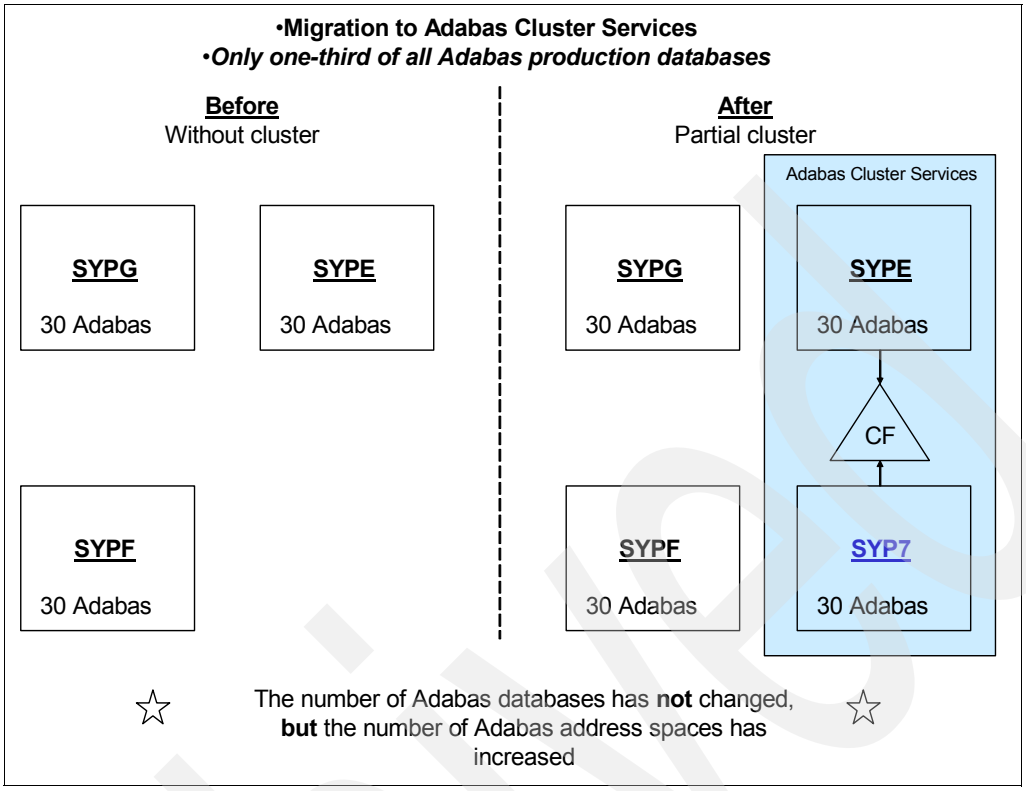


Figure 6-6 Creating a new LPAR

6.3.6 Adabas communicator task

With the introduction of Adabas Cluster Services, Software AG provided a new address space “Adacom”, which manages the Adabas Cluster nuclei. There is only one Adacom per LPAR. For further details refer to 6.5.1, “SAG’s Adabas communicator “Adacom”” on page 123.

6.3.7 CICSplex and scheduling environments

To run a data sharing environment with the goal of high availability and workload balancing, CICSPlex and batch scheduling environments are required. Both features are implemented in the IZB sysplex. In z/OS 1.5, the goal of high availability was reached, but WLM did not balance the workload as IZB had expected. For more information about these topics, refer to Chapter 9, “System” on page 177 and to Chapter 4, “Migration to CICSPlex” on page 69.

6.3.8 Test environment

Finally, an adequate test environment is absolutely necessary. When starting to implement a sysplex at IZB, a new test sysplex with internal Coupling Facilities was created. It was used to test functionality and space requirements. However, performance testing was not carried out for the following reasons:

- The LPARs in the test environment were connected to an internal Coupling Facility, rather than to a remote external CF.
- It was difficult to produce peaks, as in production.
- There was only one physical processor in the test sysplex.

6.4 Implementation and migration

The following sections document the IZB implementation and migration experiences.

6.4.1 The environment in 1999

In 1999, IZB had three production LPARs for its largest customer, Customer B. The LPARs SYPF and SYPE were located in Nuremberg, and LPAR SYPG was located in Munich; see Figure 6-7. Every production LPAR hosted about 30 Adabas databases.

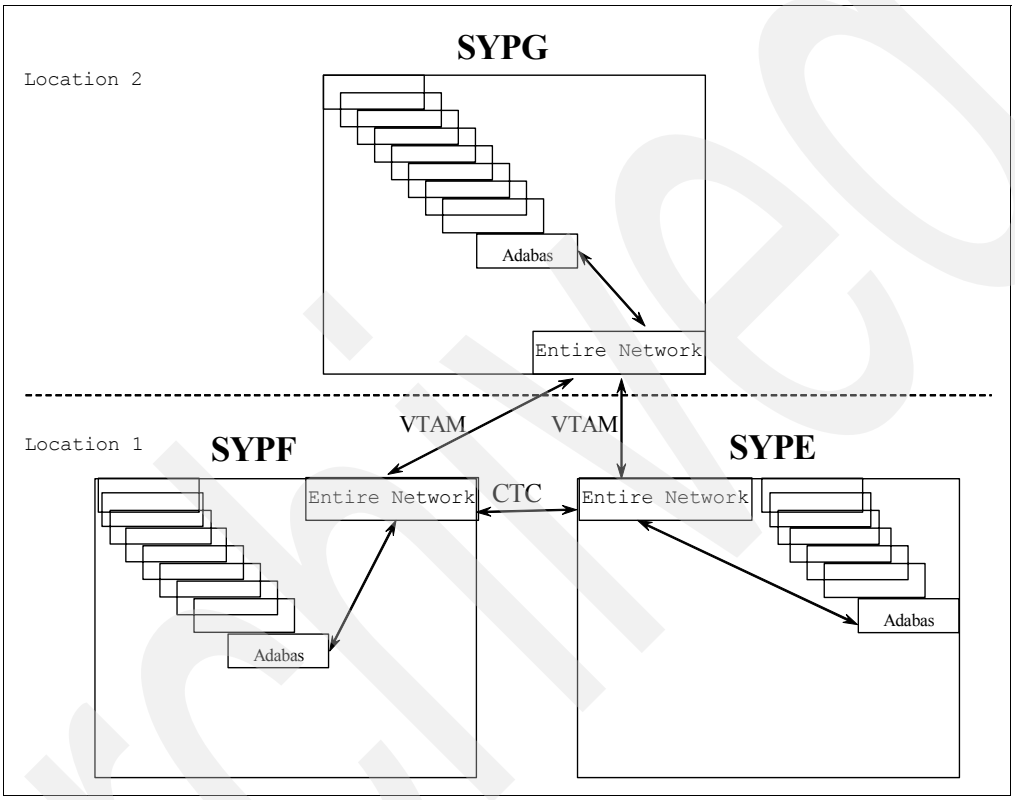


Figure 6-7 Environment in 1999

IZB already had the ability to access all Adabas databases from every LPAR using Software AG's Entire Network (as described in more detail in 6.5.2, "SAG's Entire Network product" on page 124). The physical connections were established using CTCs and VTAM. The Adabas version at this time was 612 and there were no shared DASD, no data sharing, and no Coupling Facility. Refer to Chapter 1, "Introduction to IZB" on page 3, for further details about the system environment at that time.

6.4.2 Time line and efforts

Figure 6-8 on page 120 illustrates the different Adabas system environments.

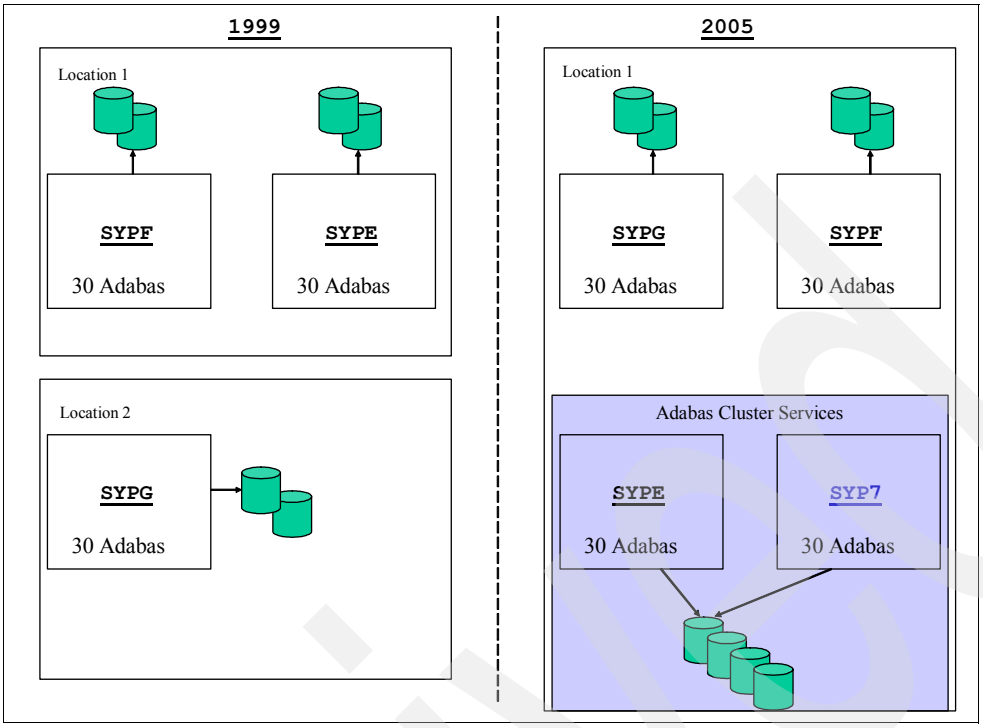


Figure 6-8 Change in configuration over time

In October 2000, IZB began to test the new Adabas data sharing feature. Because it was one of the first customers adopting this feature, IZB had to invest a great deal of time and effort over the next two years in locating and correcting errors before reaching the current configuration. Figure 6-9 illustrates the implementation effort over time.

History

Version	date	Implementation
➢7.2.0	October 2000	Alpha Version, no PLOG, no DELTA SAVE
➢7.2.1	February 2001	development system IZB
➢7.2.1	September 2001	development system Customer B
➢7.2.2	April 2002	development system IZB
➢7.2.2	October 2002	development system Customer B
➢7.2.2	October 2003	6 savings banks production
➢7.4.2	April 2004	30 savings banks production

Figure 6-9 History of implementation

The first release of an Adabas data sharing version was 720 with the name Adaplex. This version was delivered without the features PLOG and DELTASAVE. Without these features, IZB was unable to save and recover Adabas databases. In addition, this version contained many errors.

In December 2000, two Software AGs developers from the USA worked at IZB for a week. They corrected errors on the IZB development system online. The result was a new Adabas version 721, which became available in February 2001. The Adabas version 720 was never used in production at IZB.

Adabas version 721 contained PLOG and DELTASAVE, and many errors were removed. But when in installing, implementing, and testing this version, new errors occurred. Great effort was invested to achieve a relatively robust version. After the removal of many errors between April 2001 and September 2001 in version 721, IZB was able to migrate the test environment at Customer B.

In the next period, beginning in April 2002 with Adabas version 722, IZB addressed the performance aspects of data sharing. IZB considered 722 to be a version for its production environment, since it contained a set of improvements.

The migration of the test environment of Customer B took place in October 2002. One year later the pilot production, including six savings banks, was migrated. During this period, IZB invested a great deal of effort in tuning the Adabas system. External changes such as machines, processors, or the operating system influenced the performance; for more information about this topic, refer to 6.6.6, “Performance” on page 134 and to Chapter 2, “Developing a multi-site data center” on page 17.

Until the availability of Adabas release 742 in December 2003, IZB experienced no outages in the Adabas pilot production environment. Software AG’s Adabas release 742 is a consolidated version for all customers, both cluster- and non-cluster users, as shown in Figure 6-10 on page 122.

In early 2004,the test environment was migrated from version 722 to version 742. The conversion of all 30 Adabas databases residing on the LPAR SYPE from version 712 to version 742 cluster mode occurred in April 2004. Currently all test and production Adabas databases now run under Version 742, approximately 30% in cluster mode distributed on LPARs SYPE and SYP7.

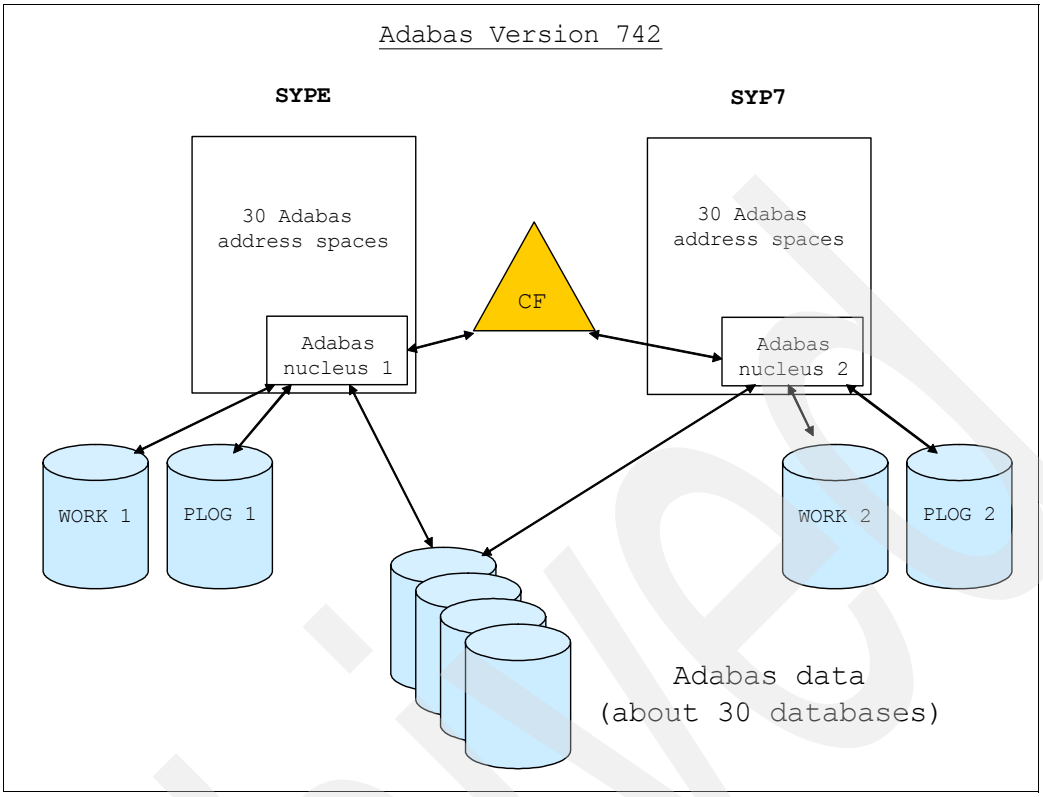


Figure 6-10 Adabas version 742

Adabas version 742 supports important features concerning the Coupling Facility, such as “System managed rebuild” and “duplexing”. Therefore, it is possible to move the Adabas related structures from one Coupling Facility to another without any interruption.

Because IZB is an important customer of SAG, it enjoys a good relationship with local support and with the SAG lab. Usually errors are corrected very quickly, which is one reason why IZB can offer excellent performance and high availability of the Adabas databases to its clients.

All Adabas databases run in version 742, with one-third of them in cluster mode and the rest of them in non-cluster mode. As mentioned, in the first stage of the project, IZB expended a great deal of effort in removing errors. In the second stage, beginning with version 722, the IZB efforts were focused on tuning the Adabas Cluster databases.

6.4.3 The environment in 2005

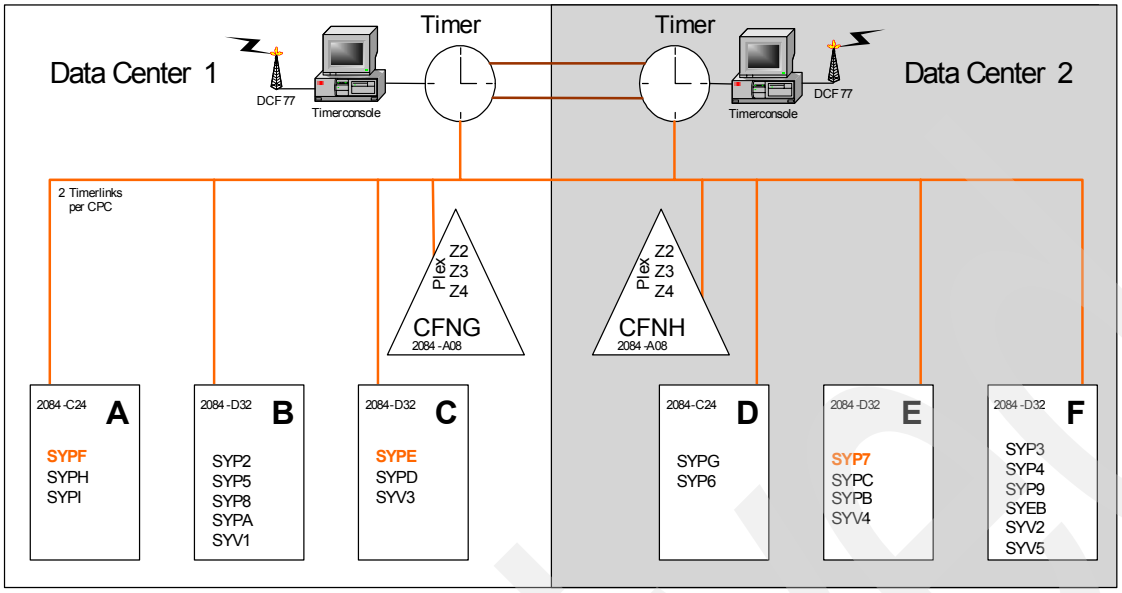


Figure 6-11 System configuration in 2005

Today's IZB z/OS configuration is very different from six years ago. Now IZB has a multi-site platform distributed across two operation centers. Both are in Nuremberg separated by a distance of 3.5 kilometers. IZB is running under z/OS 1.5 on a zSeries z990. Each of the two external CFs have one dedicated processor configured, and are used to host the structures of Customer B.

All Adabas databases are running in version 742. The databases residing on systems SYPE and SYP7 are running in cluster mode; the others, on SYPG and SYPF, are running in non-cluster mode. Every Adabas database running in cluster mode uses one Adabas nucleus per LPAR, so the incoming workload from batch and CICS can always be handled by a local Adabas nucleus. The current Adabas version supports features such as 64-bit addressing and flashcopy.

6.5 Other Software AG products and tools

This section describes the other Software AG products and tools used at IZB.

6.5.1 SAG's Adabas communicator "Adacom"

Along with Adabas Cluster Services, a new started task known as Adacom is required. Only one Adacom task is needed to manage all Adabas address spaces on one LPAR. This task must be started before any Adabas comes up. Adacom communicates with the other Adabas cluster address spaces, and stores all information about the cluster nuclei in the system area ECSA.

For example, it is possible to stop the activity of one or more local Adabas nuclei using operator commands without an outage of the whole application; see Figure 6-12 on page 124. The update transactions can commit correctly, and then a shutdown of Adabas is possible.

Next, the incoming workload will be routed through Entire Network to the remaining Adabas on the other LPAR. At that point, it is possible to apply maintenance without impacting the availability of the application. After starting this Adabas address space again, the data will be processed automatically by the local nucleus. Afterwards, the same procedure can be done on the other LPAR.

Example for Adacom

```
PLI063 PROCESSING:ADACOM SVC=251,DBID=00209,NU=300,CMDMGR=YES
PLI002 INITIALIZING DBID=00209 SVC=251
       ACQUIRING NEW PLXCB
       MAX USERS FOR IMAGE 00000300
       PLXCB IS LOCATED AT 1EC9FE70  <----  area in ECSA
DSP001 INITIALIZING DBID=00209 SVC=251
PLI063 SVC=251,DBID=00209 INITIALIZATION COMPLETE
DSP002 DATASPACE ACQUISITION AUTHORITY ACQUIRED
PLI063 PROCESSING:ADACOM SVC=251,DBID=00210,NU=300,CMDMGR=YES
PLI002 INITIALIZING DBID=00210 SVC=251
       ACQUIRING NEW PLXCB
       MAX USERS FOR IMAGE 00000300
       PLXCB IS LOCATED AT 1EC87E70  <----  area in ECSA
DSP001 INITIALIZING DBID=00210 SVC=251
PLI063 SVC=251,DBID=00210 INITIALIZATION COMPLETE
DSP002 DATASPACE ACQUISITION AUTHORITY ACQUIRED
```

Figure 6-12 Adacom

6.5.2 SAG's Entire Network product

Entire Network has been installed at IZB since 1990. With this product, it is possible to access each Adabas database from every user address space, including CICS, batch, and TSO. Entire Network is a basic requirement for operating Adabas Cluster Services. Additionally, the communication between multiple Adabas address spaces is handled with special Adabas calls over Entire Network. Figure 6-13 on page 125 illustrates the Entire Network configuration.

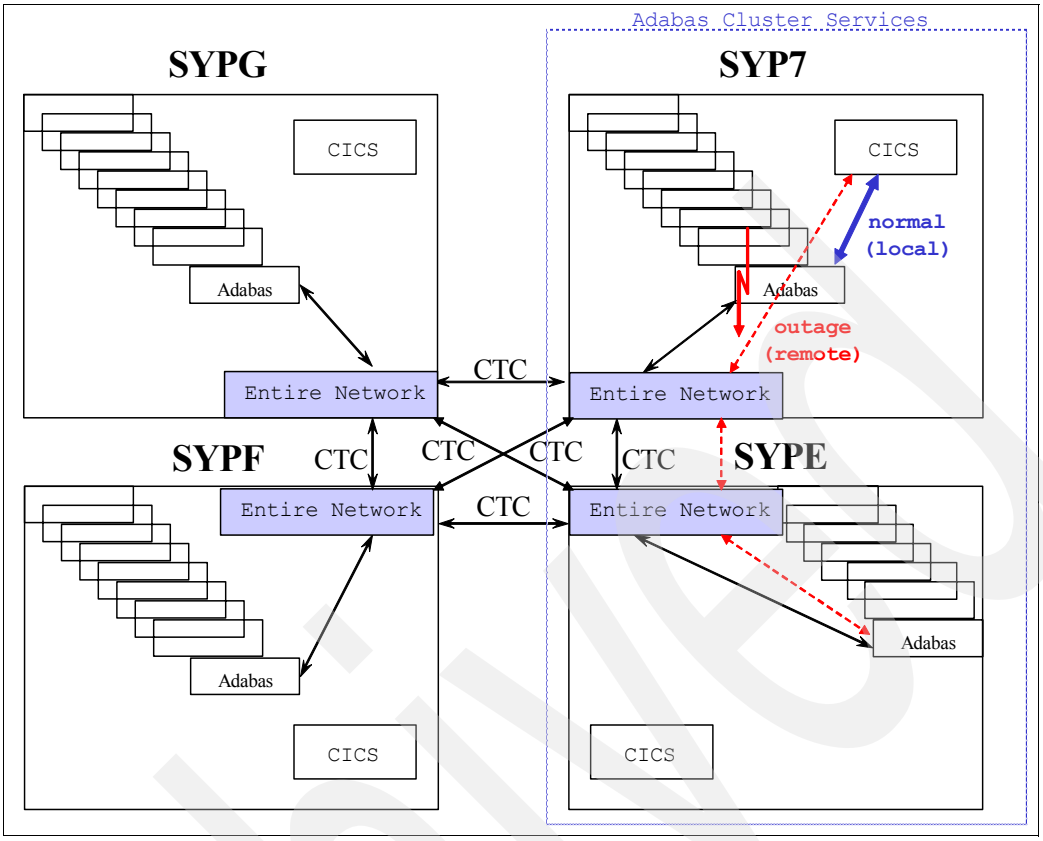


Figure 6-13 Entire Network configuration

To connect the Entire Network, IZB used FICON-CTCs (channel-to-channel) and VTAM as automatic fallback. All LPARs in the sysplex are connected, therefore all user address spaces can access all Adabas databases, no matter where they are located; see Figure 6-14.

Output from Entire Network

```
11:10:10 NET0124I SYPE: TARGET 24502 (L-N)  ACTIVE  ON NODE SYP7
11:10:10 NET0124I SYPE: TARGET 00245 (I-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 24501 (L-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 24402 (L-N)  ACTIVE  ON NODE SYP7     NUCID 2
11:10:10 NET0124I SYPE: TARGET 00244 (I-N)  ACTIVE  ON NODE SYPE     DBID
11:10:10 NET0124I SYPE: TARGET 24401 (L-N)  ACTIVE  ON NODE SYPE     NUCID 1
11:10:10 NET0124I SYPE: TARGET 24302 (L-N)  ACTIVE  ON NODE SYP7
11:10:10 NET0124I SYPE: TARGET 00243 (I-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 24301 (L-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 24002 (L-N)  ACTIVE  ON NODE SYP7
11:10:10 NET0124I SYPE: TARGET 00240 (I-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 24001 (L-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 23202 (L-N)  ACTIVE  ON NODE SYP7
11:10:10 NET0124I SYPE: TARGET 00232 (I-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 23201 (L-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 22202 (L-N)  ACTIVE  ON NODE SYP7
11:10:10 NET0124I SYPE: TARGET 00222 (I-N)  ACTIVE  ON NODE SYPE
11:10:10 NET0124I SYPE: TARGET 22201 (L-N)  ACTIVE  ON NODE SYPE
```

Figure 6-14 Targets in Entire Network

Figure 6-14 on page 125 shows Adabas DBIDs and their corresponding NUCIDs. Entire Network has information about all Adabas databases using the same SVC. The blue marked targets (DBID and NUCIDs) of DBID 244 reside at system SYPE and SYP7. Adabas calls from outside the Adabas cluster will be routed to system SYPESYPF, because it has the first started nucleus, which is the owner of the DBID.

6.5.3 SAG's Adabas Online Services "Sysaos"

Sysaos is the online monitor system of Software AG. If you operate an Adabas Cluster environment, the online monitor supports it. Figure 6-15 shows the Sysaos primary panel.

```
                    Sysaos  -  Primary panel

11:23:33          ***** A D A B A S  BASIC  SERVICES *****          2005-11-25
                          -  Resource Utilization  -                PACU002

Code  Service                            Code  Service
----  ------------------------------     ----  ------------------------------
  C    Command usage                       S    System status
  F    File usage                          T    Thread usage
  H    High water marks (pools/queues)     W    WORK status
  L    Workpool (LWP) usage                X    Cluster usage
  N    Nucleus File Status                 Y    Display PPT table
  P    PLOG status
  ?    Help
 .    Exit
----  ------------------------------     ----  ------------------------------


        Code ......... X
        File Number .. 0
        Database ID .. 219    (P219xxxxxxxx-ST)    NucID .. 21901

Command ==>
PF1----- PF2------ PF3------ PF4------ PF6----- PF7----- PF8----- PF12-----
Help               Exit                                          Menu
```

Figure 6-15 Sysaos cluster panel

The following panels in Sysaos provide information about cluster usage, response times, locks and further details. Figure 6-16 on page 127 illustrates the relationship of synchronous to asynchronous Adabas calls. Sysaos also offers many other functions that help to analyze and tune the Adabas databases.

Statistics of DBID 237

```
13:02:27          ***** A D A B A S  BASIC  SERVICES *****          2005-10-24
DBID  237                  - File 0 Statistics -                    PACUX22
NucID 23701
  Reads                                     Writes
  --------------------------------          ----------------------------------
  Total ............      24,751           Total ..............      27,163
     Sync ..........         109              Sync ............         109
     Async .........      24,642              Async ...........      27,054

     In cache ......      16,162              Written .........      21,574
     Not in cache ..       8,589              Not written .....       5,589
     Struc. full ...           0              Struc. full .....           0

  Cast-out Reads                            Other
  --------------------------------          ----------------------------------
  Total ............      20,251           Validates ..........   3,622,116
     Sync ..........          88              Invalid .........       7,571
     Async .........      20,163           Deletes ............           0
                                              Timeouts ........           0
                                           Redo processes .....       5,635
```

Figure 6-16 Example of Sysaos asynchronous versus synchronous

Figure 6-17 shows the different locks that Adabas uses. This function is needed to analyze problem situations.

Sysaos – Lock statistics

```
09:37:23          ***** A D A B A S  BASIC  SERVICES *****          2005-11-28
                            - Lock Statistics -                     PACUX32

Code   Service                          Code   Service
----   ---------------------------      ----   ------------------------------
  A    Buffer flush lock                  J    Global update command sync lock
  B    Cancel lock                        K    Hold ISN lock
  C    Checkpoint lock                    L    New-Data-RABN lock
  D    DSF lock                           M    Online save lock
  E    ETID lock                          N    Parameter lock
  F    File-lock-table lock               O    Recovery lock
  G    FST lock                           P    RLOG lock
  H    GCB lock                           Q    Security lock
  I    Global ET sync lock                R    Spats lock
  .    Exit                               S    Unique descriptor lock
  ?    Help
----   ---------------------------      ----   ------------------------------
       Code ......... _
       Database ID .. 218    (P218xxxxxxxx-ST)     NucID .. 21801


PF1----- PF2----- PF3------ PF4------ PF6----- PF7------ PF8----- PF12-----
Help              Exit      Refresh                                Menu
```

Figure 6-17 Sysaos lock statistics

Figure 6-18 shows functions used to analyze performance aspects of an Adabas Cluster environment.

```
                    Sysaos  -  Cache blocks

09:52:18          ***** A D A B A S  BASIC  SERVICES *****          2005-11-28
DBID  218                   - All Cache Blocks -                    PACUX12
NucID 21801
  Reads                                 Writes
  --------------------------------      ----------------------------------
  Total ............  67,797,956        Total ..............  32,540,311
     Sync ..........  54,656,220           Sync ............     137,397
     Async .........  13,141,736           Async ...........  32,402,914

     In cache ......   5,741,008           Written .........  32,530,359
     Not in cache ..  62,056,948           Not written .....       9,952
     Struc. full ...           0           Struc. full .....           0

  Cast-out Reads                        Other
  --------------------------------      ----------------------------------
  Total ............   6,263,297        Validates ..........   1,713,179 K
     Sync ..........      27,780           Invalid .........   1,989,241
     Async .........   6,235,517        Deletes ............           0
                                           Timeouts ........           0
                                        Redo processes .....      15,927

PF1----- PF2----- PF3----- PF4----- PF7----- PF8----- PF9-------  PF12-----
Help     Repos    Exit     Refresh  PrevBlk  NxtBlk   Detail      Menu
```

Figure 6-18 Sysaos cache blocks

6.5.4 BMC's monitor product "Mainview"

Mainview is another tool which helps to analyze an Adabas Cluster Services system environment; for example, it can monitor Coupling Facilities. For more detailed information about Mainview, consult the appropriate BMC documentation or online help functions. The highlighted values shown in Figure 6-19 on page 129 illustrate key data for the entire sysplex environment.

Coupling facility overview

```
25NOV2005  10:12:14 ------ MAINVIEW WINDOW INTERFACE (V4.2.04) ----------------
COMMAND  ===>                                                SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =CFINFO===========SYPE======*========25NOV2005==10:12:13====MVMVS====D====1
 CF Name.....    CFNG1 --Requests--        -CF Storage--          %
 CF Id.......        2 Total CF....   1.7M Total Alloc..   3.7M  100.0
 CU Id.......     FFF2 #Req /Sec... 8599.5  Available...   2.3M   61.6
 CF Seq#.....        2 %Conv Async.    0.0  Structure...   1.4M   38.0
 #Structures.       87 -Sync Reqs--         Dump........  12800    0.4
 -Processor-.           #Req.......   3.4M   Used.......      0    0.0
 CPU Util%...     28.3  SvcTm Avg..     36   Max Req....    128    1.0
 #Processors.        2  Std Dev....     60
 ---Config---          -Async Reqs-        Control Def..   3.7M  100.0
 #Paths......        4  #Req.......   3.5M Control Alloc   1.5M   38.4
 #SubCh Def..       28  SvcTm Avg..    157 Data Defined.      0    0.0
 #SubCh Used.       28  Std Dev....    390 Data Alloc...      0    0.0
 #SubCh Max..       28 -Queued Req-
 --Channel---           #Req.......  51539
 Subch Busy..        2  % Queued...    0.8
 Path Busy...        0  QTime Avg..    578
                        Std Dev....   2057
```

Figure 6-19 Mainview overview of Coupling Facility

Figure 6-20 shows the activity and size of the Adabas lock and cache structures in the IZB production environment. The red marked structures of DBID 219 are located in the remote Coupling Facility; the dark blue marked structures of DBID 218 are located in the local Coupling Facility. The structure sizes, the number of requests, requests per second, and the service time of synchronous and asynchronous requests are also shown.

Samples of Adabas structures - Lock and cache

Structure Name	CF Name	Struc Type	Struc Size	Struc Stor%	#Of Reqs	Req/ Sec	Sync SvcTm	Async SvcTm	%Conv Async
ADA212CACHE	CFNG1	Cache	16384	0.4	13391	55.8	52	124	0.1
ADA212LOCK	CFNG1	Lock	16384	0.4	6574	27.4	38	120	0.0
ADA214CACHE	CFNG1	Cache	16384	0.4	14193	59.1	52	116	0.1
ADA214LOCK	CFNG1	Lock	16384	0.4	6413	26.7	35	80	0.0
ADA215CACHE	CFNG1	Cache	8192	0.2	6765	28.2	51	122	0.0
ADA215LOCK	CFNG1	Lock	8192	0.2	3818	15.9	35	127	0.0
ADA216CACHE	CFNG1	Cache	8192	0.2	1882	7.8	53	149	0.1
ADA216LOCK	CFNG1	Lock	8192	0.2	1468	6.1	37	90	0.0
ADA218CACHE	**CFNG1**	**Cache**	**16384**	**0.4**	**16038**	**66.8**	**50**	**134**	**0.1**
ADA218LOCK	**CFNG1**	**Lock**	**16384**	**0.4**	**10251**	**42.7**	**34**	**102**	**0.0**
ADA219CACHE	**CFNH1**	**Cache**	**16384**	**0.4**	**43346**	**181**	**114**	**153**	**0.1**
ADA219LOCK	**CFNH1**	**Lock**	**16384**	**0.4**	**20292**	**84.6**	**77**	**130**	**0.0**
ADA221CACHE	CFNG1	Cache	8192	0.2	1670	7.0	48	133	0.1
ADA221LOCK	CFNG1	Lock	8192	0.2	812	3.4	39	78	0.0
ADA222CACHE	CFNG1	Cache	8192	0.2	3309	13.8	55	130	0.1
ADA222LOCK	CFNG1	Lock	8192	0.2	1944	8.1	35	97	0.0
ADA224CACHE	CFNG1	Cache	8192	0.2	4651	19.4	51	142	0.2
ADA224LOCK	CFNG1	Lock	8192	0.2	2335	9.7	38	109	0.0
ADA226CACHE	CFNG1	Cache	8192	0.2	9228	38.4	52	123	0.1
ADA226LOCK	CFNG1	Lock	8192	0.2	5002	20.8	37	130	0.0
ADA227CACHE	CFNG1	Cache	8192	0.2	10782	44.9	50	119	0.1
ADA227LOCK	CFNG1	Lock	8192	0.2	4636	19.3	37	119	0.0
ADA228CACHE	CFNG1	Cache	16384	0.4	31760	132	48	123	0.2
ADA228LOCK	CFNG1	Lock	16384	0.4	13998	58.4	36	121	0.0

Figure 6-20 Mainview samples of Adabas structures

In addition, Mainview offers many functions to help analyze the Adabas Cluster environment, especially obtaining performance data.

6.5.5 IBM RMF and SMF

Figure 6-21 shows the panel of the RMF™ CF Activity Report, illustrating the relationship between synchronous and asynchronous CF requests and their service times.

```
 RMF V1R5   CF Activity      - PLEXZ2       Line 13 of 188
Command ===>                                                Scroll ===> CSR

Samples: 92      Systems: 10   Date: 11/24/05  Time: 11.45.00  Range: 100   Sec

CF: ALL          Type  ST System      --- Sync ---     --------- Async --------
                                        Rate   Avg       Rate   Avg   Chng   Del
Structure Name                                Serv             Serv   %      %

ADA212CACHE      CACHE    *ALL         23.3     45      60.2    142   0.1    0.1
ADA212LOCK       LOCK     *ALL         29.9     38      16.0    142   0.0    0.1
ADA214CACHE      CACHE    *ALL         13.4     49      50.0    126   0.1    0.1
ADA214LOCK       LOCK     *ALL         34.0     36       6.9    148   0.0    0.4
ADA215CACHE      CACHE    *ALL         11.7     50      26.0    131   0.2    0.2
ADA215LOCK       LOCK     *ALL         13.6     33       7.2    147   0.0    0.3
ADA216CACHE      CACHE    *ALL          2.3     67       9.2    134   0.2    0.2
ADA216LOCK       LOCK     *ALL          5.7     33       3.3    135   0.0    0.9
ADA218CACHE      CACHE    *ALL         33.4     51     104.0    134   0.2    0.2
ADA218LOCK       LOCK     *ALL         46.5     35      28.0    147   0.0    0.4
ADA219CACHE      CACHE    *ALL         30.8     49     213.9    139   0.1    0.1
ADA219LOCK       LOCK     *ALL         50.0     35     100.6    120   0.0    0.3
ADA221CACHE      CACHE    *ALL          3.3     55      15.8    147   0.1    0.1
ADA221LOCK       LOCK     *ALL          5.2     31       4.1    140   0.0    0.0
ADA222CACHE      CACHE    *ALL          7.1     50      12.8    135   0.1    0.1
ADA222LOCK       LOCK     *ALL          8.5     34       3.8    147   0.0    0.0
ADA224CACHE      CACHE    *ALL          3.2     55      10.5    140   0.0    0.0
ADA224LOCK       LOCK     *ALL          5.2     32       2.2    131   0.0    0.0
ADA226CACHE      CACHE    *ALL         17.6     47      45.6    139   0.2    0.2
ADA226LOCK       LOCK     *ALL         27.7     33      11.7    156   0.0    0.1
```

Figure 6-21 Example of RMF Monitor III

RMFIII allows one to view many other performance values of the Adabas Cluster environment. For more details, refer to Chapter 9, “System” on page 177.

6.6 Conclusions

If you plan to migrate your existing Adabas environment to data sharing, the work will not be as involved as when IZB migrated, as long as you have already implemented Adabas Version 742 or higher. At IZB, the largest amount of time and effort was expended in setting up the prerequisites as described in 6.3, “Requirements for introducing Adabas Cluster Services” on page 114.

The following list represent an overview of considerations to keep in mind when implementing Adabas Cluster Services:

- A set of new parameters
- Increase the block sizes of database containers (PLOG and WORK) if necessary

- Allocate new data sets
- Monitor he merging technique of PLOG
- A set of new operator commands
- New messages
- The order of starting the tasks
- Performance
- System components like locations, Coupling Facility, coupling links, processors

Detailed information can be found in *Adabas Implementation Guide for Cluster Services*, available from Software AG. Or contact IZB's Adabas Database Administration Team or Software AG's support.

6.6.1 Disadvantages

IZB considered the following points to be disadvantages when implementing Adabas Cluster Services.

- CPU overhead and space (costs) of using Adabas Cluster Services with two instances:
  - The additional consumption of CPU is about 20%.
  - The additional space required is 15%.

  If more than two nuclei operate against one database, space and CPU consumption increase accordingly.
- Additional hardware required

  When implementing Adabas Cluster Services, the installation of Coupling Facilities is required. For availability reasons, single points of failure have to be avoided, so that two Coupling Facilities are necessary. This adds complexity to the system infrastructure. Operations staff have to be educated to manage this new environment and to take dependencies into account. For system administrators, there is additional work to maintain and tune this complex configuration.

  Adding Coupling Facilities to the sysplex also increases the total costs.

  Note: Adabas Cluster Services offers one benefit compared to DB2: after being migrated to data sharing mode, DB2 needs at least one Coupling Facility active. It is quite labor-intensive to switch DB2 back to non-sharing mode. For Adabas Cluster Services it is rather simple (but involves outage time) to switch back to non-cluster mode and to work without a Coupling Facility.

6.6.2 Benefits today

- Availability

  The greatest benefit offered by Adabas Cluster Services is higher application availability; it is possible to maintain the software level and apply fixes without any application outage.

  There are two ways to maintain Adabas:
  - Close one active Adabas address space using Adacom operator commands. The second one remains active and will take over the workload through Entire Network. Active transactions can end normally and new transactions will be routed to the other Adabas address space. You only must wait the defined transaction time (TT), and then you can shut down the Adabas address space.
  - Shut down the active CICS AOR regions in an LPAR. Then the incoming Adabas calls on that LPAR will be routed through CPSM to the CICS AOR regions on the other LPAR. Because Adabas prefers local Adabas calls, all workload will be processed by

the local Adabas address space. Now it is possible to shut down the other Adabas address space to apply maintenance.

If you are finished with one LPAR and Adabas has been restarted, you can do the same with the other LPAR.

- No more single point of failure

  As mentioned in 6.6.1, “Disadvantages” on page 131, the IZB Adabas Cluster Services has no single point of failure. If there is a problem with the Coupling Facility, it is possible to change the Adabas operation from cluster mode to non-cluster mode.

- Workload balancing with WLM

  The definitions in the WLM policy determine the workload balancing and priority within one LPAR or over many LPARs. Using the velocity statistics, Adabas Cluster nuclei get a dispatching priority to allow them to obtain their goal. See Chapter 9, “System” on page 177 for more information about this topic.

- Maintenance without outages

  Today it is possible to apply Adabas software updates in the evening and not in the maintenance window during the weekend. Because there is no outage in applications, availability increases and the risk of too many changes at the same time decreases.

- Scalability

  By changing the scheduling environment for batch jobs or changing the policy of WLM for CICS or other applications, it is possible to control the distribution of workload in the sysplex environment. This function also helps to increase application availability.

- Multi-processor usage

  Although Adabas is a single-processor DBMS, with Cluster Services this limitation is removed. Now each Adabas database can use up to 32 processors, one processor per address space.

6.6.3 Experiences

- High availability

  Compared to other system components in the sysplex environment, Adabas Cluster Services at IZB had only *one* outage for *one* hour for *one* Adabas database since its introduction in April 2004.

  In June 2005, IZB experienced a system problem with I/O priority queuing in z/OS (a feature to set a priority for I/Os at different LPARs on one machine). As a result, the PPT of one Adabas Cluster database was destroyed. Therefore IZB had an outage of one hour on one database only. Otherwise, Adabas runs without interruption.

- Starting sequence

  Adabas-related tasks have to be started and stopped in a certain sequence: first Entire Network, then the Adacom address space, and then all Adabas nuclei. Adabas-related tasks in one LPAR should *not* be started or stopped in parallel with the Adabas-related tasks in another LPAR (that is, start one LPAR *after* the other). For additional details about this topic, refer to Chapter 11, “Automation” on page 207”

- PPT and PLOG

  The PPT is a new area in Adabas Version 7.4.2, independent of data sharing. It is located in the ASSO data set and is 32 blocks in size. Information about PLOG data sets, WORK data sets, CLOG data sets, and the current states of active nuclei are located in the PPT.

Therefore, the PPT is very important when operating an Adabas Cluster Environment, as well as in an Adabas noncluster environment. z/OS errors destroyed this table, with no possibility of recovery processing. Now this problem is solved, but IZB recommends taking good care of the PPT and checking the content from time to time.

- Processor dependency (G5 to z990)

  During the time frame when IZB migrated into Adabas Cluster Services, the hardware changed. G5 processors were replaced by z990 servers; see Figure 6-22. In a case like this, it is important to first upgrade the external Coupling Facilities. Otherwise, you will experience an increase in the response times.

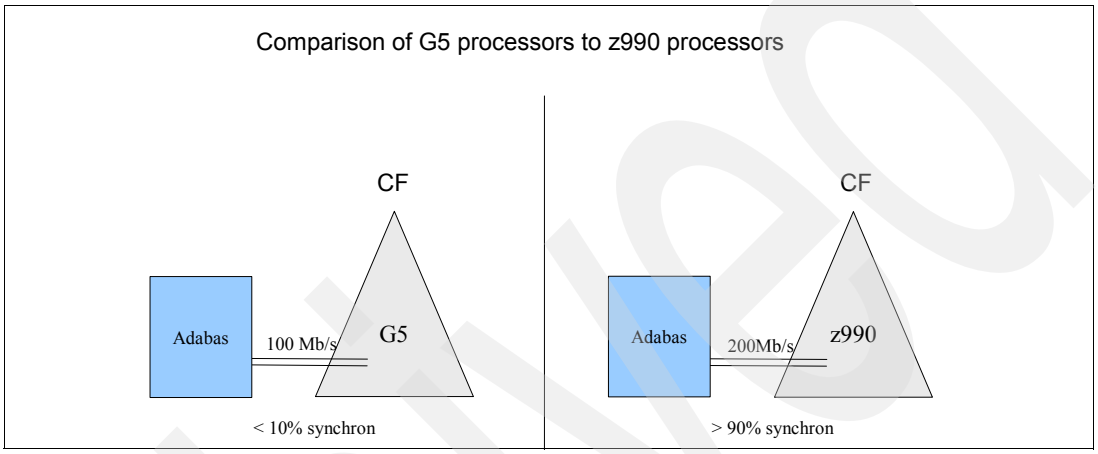


Figure 6-22 Comparison of processors

- Departments involved

  If you want to add Adabas Cluster Services, you have to communicate with four departments:

  – The hardware group, to define the computer center configuration such as Coupling Facilities, processors, main storage, and so on. See Chapter 2, “Developing a multi-site data center” on page 17, for more information.

  – The system group, to define the structures in the Coupling Facility. See Chapter 9, “System” on page 177, for more information.

  – The CICS group, to communicate the corresponding CICSPlex introduction. See Chapter 4, “Migration to CICSPlex” on page 69, for more information.

  – The space group, to request the additional space. See Chapter 8, “Storage management” on page 167, for more information.

6.6.4 Problems

This section describes problems encountered by IZB in the past, problems during the migration phase, and problems encountered today.

- Shared DASD and merging data sets

  It is a very extensive process to create shared DASD and perform the corresponding merge of data sets. For this process, application programmers need to be involved.

- Removal of system affinities in applications

  It was difficult to remove all system affinities from applications. Without removing all system affinities, migration to data sharing is not possible.

- Errors in the system software

  As mentioned in 6.4.2, “Time line and efforts” on page 119, the Adabas versions prior to 7.4.2 had many errors. But today, with Adabas Version 7.4.2, IZB has a very stable DBMS.

  Today the only major challenges remain with IBM Workload Manager (WLM) in z/OS Version 1.5. WLM is unable to balance the IZB workload equally over many LPARs; for example, it does not consider the location of structures. There is no goal to ask WLM to distribute the workload by 50%.

6.6.5 Requirements

IZB has submitted two requirements to Software AG.

First, if only one Adabas nucleus in a cluster environment is active, a synchronization of updates over the Coupling Facility is not necessary. Adabas could recognize this situation and terminate the Coupling Facility communication. Incoming Adabas calls will be routed over Entire Network to the remaining nucleus.

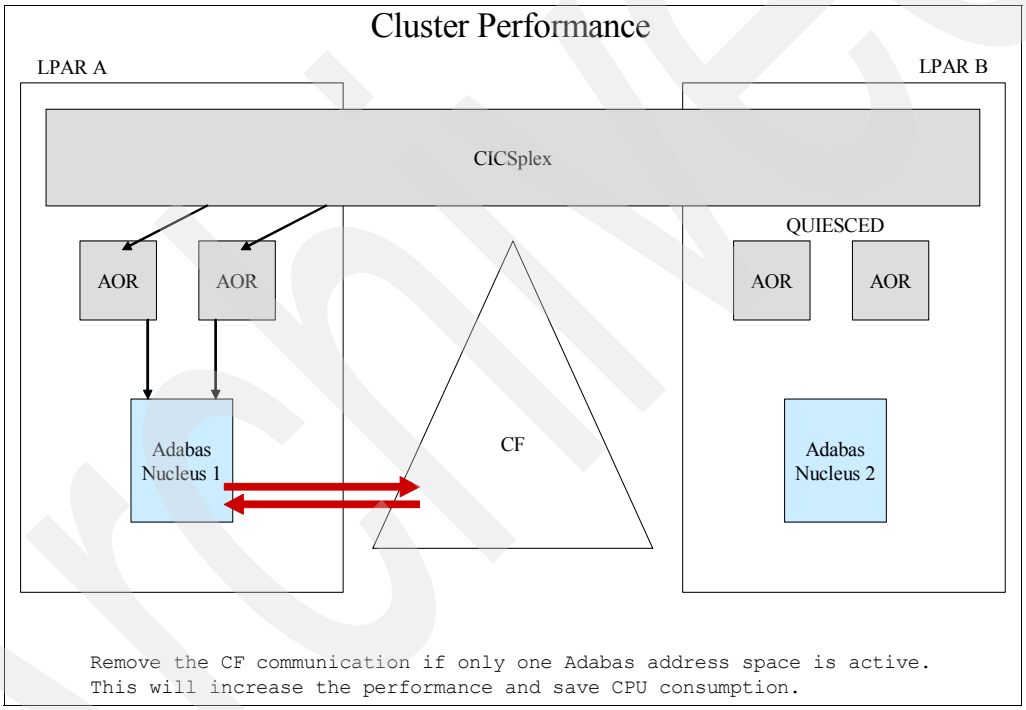


Figure 6-23 Removal of Coupling Facility requests

The second requirement is to remove the requirement that the NUCID (identification of the Adabas address space) must be unique over the entire network, that is, all connected LPARs. For example, if you have a number of Adabas databases and up to 32 address spaces for every database, it is difficult to administer the NUCIDs. IZB’s request is to make the NUCIDs only unique within one physical database.

6.6.6 Performance

Performance considerations are very important when operating Adabas Cluster Services in a production environment. As mentioned in 6.5, “Other Software AG products and tools” on page 123, tools exist that will help a system to achieve good performance.

There are four general points to keep in mind:

- Multisite environment

  In a multi-site workload environment, where workload is distributed across both data centers, 50% of all Coupling Facility requests are remote. See Chapter 2, “Developing a multi-site data center” on page 17 and Chapter 9, “System” on page 177 for more details.

- Synchronous versus asynchronous CF requests

  To achieve good performance, it is very important that most of the CF requests are performed synchronously. This is only possible if the processors in the Coupling Facility have the same or a higher speed as the processors in the z/OS servers. Furthermore, very fast “coupling links” and a local access are needed; see Figure 6-22 on page 133.

- Location of structures

  If you consider the transaction time or runtime of applications, it is very important to know the location of the Coupling Facility structures. The elapsed time of batch jobs can increase ten times if there is a high percentage of updates and therefore many CF requests. In this case it is helpful to route the batch job to the location of the Coupling Facility; see Figure 6-24.

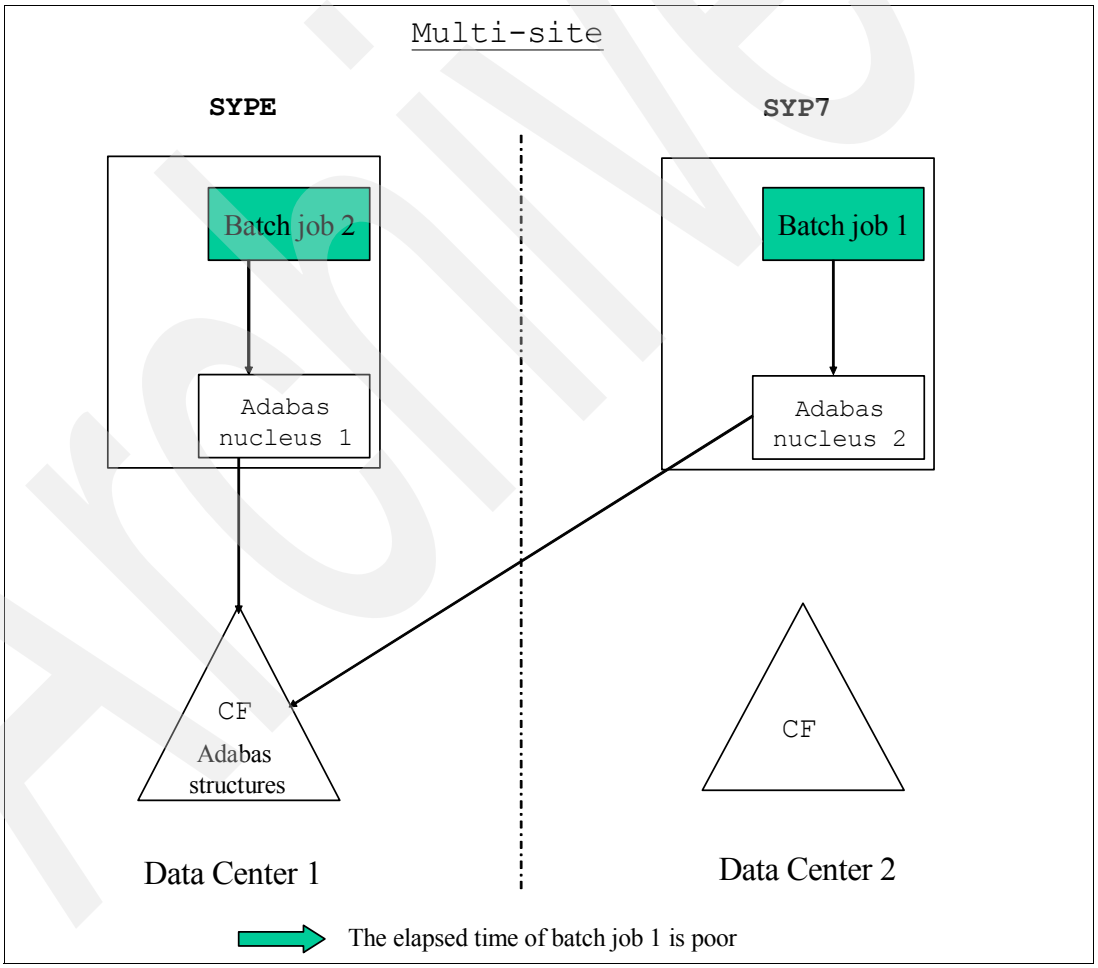


Figure 6-24 Performance of remote CF requests

6.7 Outlook and planned projects

In 2007, IZB will migrate all Adabas databases of Customer B to data sharing. The strategy is to operate two Adabas nuclei against one Adabas database distributed over two LPARs. Figure 6-25 illustrates the distribution over four LPARs. With this configuration, it will be possible to maintain the system software of all Adabas databases without any outage.

IZB wanted to realize this project in 2006, but Customer B will determine when to introduce data sharing for the rest of savings banks, based on cost and manpower considerations.

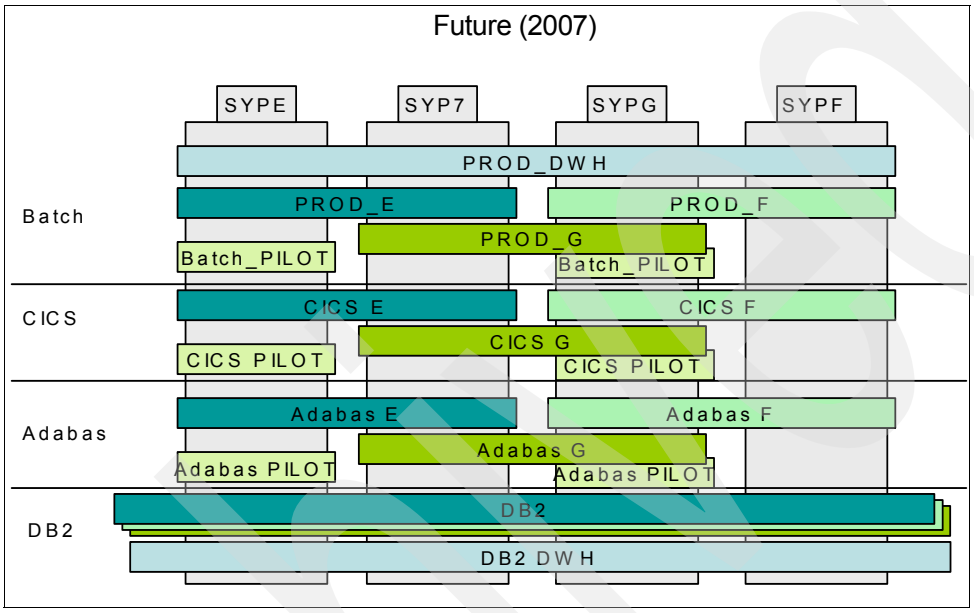


Figure 6-25 Complete Customer B environment in data sharing mode

Another future step will be to introduce data sharing, including Adabas Cluster Services, to other IZB clients.

7

WebSphere

Not long ago, around the world, Web applications were running on workstations and on distributed servers connected by networks. Growing Internet popularity sparked interest in areas such as high availability, workload management, scalability, and security—areas which represent the strengths of the mainframe.

IBM selected Java as a strategic programming language and J2EE™ as a strategic design and development platform on the mainframe. Likewise, IZB and Customer B decided in 2000 to develop their new Java/J2EE e-business applications for the mainframe as well.

This chapter describes IZB's path from a provider of decentralized Web applications to a carrier of a large Webplatform on System z with over 110 Gigabyte real storage and over 1100 MSUs. This Webplatform host is supported across four z/OS LPARs that are part of a large, 10-way sysplex.

Several Internet and intranet Web applications running under WebSphere Application Server for OS/390® V4.0 (in compatibility mode to V3.5) or WebSphere Application Server for z/OS V5.1 connected to DB2, MQSeries, and CICS, currently share this platform and serve thousands of clients who generate more than 100 million Web hits per month. The platform consists of LPARs SYP2, SYP3, SYP8, and SYP9.

Note: A description of WebSphere technical details is beyond the scope of this publication. For WebSphere information, refer to the following IBM documentation.

- For WebSphere Application Server, refer to the WebSphere Information Center:

  http://www-306.ibm.com/software/Webservers/appserv/was/library/index.html

- For details about the IBM HTTP Server (IHS), refer to:

  http://www-306.ibm.com/software/Webservers/httpservers/library/

7.1 Choosing WebSphere on MVS

In the 1990s, IZB provided an Internet home banking application based on applet technology running on several decentralized systems (Sun™ Solaris™) for our clients. This application was developed by an external company, Brokat. A problem with this application was the inflexibility of the applet technology. That is, customers needed to download a bulky applet to their client, and the ability to develop and deploy new application features was limited. Brokat eventually failed and IZB then had to decide how to react to the growing hype of JSP™ and servlet-based Web applications. There were two main decisions to be made: which application server to use, and on which platform.

WebSphere Application Server (WAS) was chosen because of its ability to interface with CICS and DB2; see Chapter 1, “Introduction to IZB” on page 3. As described there, the business logic of the primary applications is based on CICS transactions using DB2 and Adabas. Therefore, it was easy to select an application server that would provide the most compatibility for these components.

Using WebSphere Application Server on OS/390 was part of the motivation to include scalability; see Chapter 3, “Network considerations” on page 43. Virtualizing IZB networks, as described there, had an major impact. As the past showed, communication between mainframe and UNIX people was sometimes challenging, compounded by the inclusion of the networking staff. So reducing communication issues also became an important part of the equation.

Another compelling driver was the chance to optimize PSLC/SYSPLEX pricing. By installing two new LPARs especially for Web applications, IZB was able to provide the necessary LPARs and MIPs, yet could save money on other product licenses, although there would be a larger CPU consumption on OS/390.

Using WebSphere Application Server on OS/390 also gave IZB the chance to use two-phase commit with the option of running with Resource Recovery Services (RRS) in the future.

7.2 Implementation of WebSphere Application Server V3.5

This section describes the IZB implementation of WebSphere Application Server V3.5 in a single HTTP server, as well as in scalable mode, and details the challenges that were encountered.

7.2.1 WebSphere Application Server V3.5 embedded in a single HTTP server

The first application of IZB’s client to be installed on WebSphere Application Server was called Webfiliale. It was a JSP/servlet-based application which aimed to give all of the 80 savings banks a common Web presentation.

IZB decided to use WebSphere Application Server V3.5 instead of Version 4 because V4 was very new and IZB would have been similar to a beta customer. Starting with just a few savings banks, the first step was to install a WebSphere Application Server V3.5 imbedded in an IBM HTTP server on one LPAR, SYP2.

IZB used the IBM HTTP Server for OS/390 (IHS) as the Web server because IHS has features that use special attributes of the System z platform, such as:

- Security with RACF and digital certificates
- Scalability in Parallel Sysplex
- Use of Workload Manager (WLM)

- Fast Response Cache Acceleration (FRCA)

IHS has no Java component, so it is not a J2EE application server. However, it plays a central role in Web applications because it provides browsers with HTML documents. IHS is based on a technique called Common Gateway Interface (CGI), in which different programs can be started in separate address spaces outside the IHS.

CGI in turn is based on a common interface for Web servers, the Go Webserver Application Programming Interface (GWAPI). Another important user of the GWAPI interface is the WebSphere Application Server Plugin. This shows the correlation to WebSphere Application Server, which is, from the view of the IHS, a GWAPI service running with the IHS in a single address space. It is activated through a service directive in the httpd.conf, which is the main definition file of the IHS.

Example 7-1 shows the service directive in the httpd.conf.

Example 7-1 Service directive in the httpd.conf

```
ServerInit /m/izbp/wasp2/WebServerPlugIn/bin/was400plugin.so:init_exit
Service /* /m/izbp/wasp2/WebServerPlugIn/bin/was400plugin.so:service_exit
ServerTerm/m/izbp/wasp2/WebServerPlugIn/bin/was400plugin.so:term_exit/m/izbp/wasp2,/etc/izs
    /was/p2x/Web/was.conf_p2a
```

The WebSphere Application Server Standard edition is closely associated with the IHS; it is virtually plugged in. Therefore it is also known as a WAS plugin. The WAS plugin creates a Java Virtual Machine (JVM™) in the IHS address space, where the servlet engine is started and application programs such as servlets and JSPs are loaded.

Figure 7-1 shows the first configuration, with one WebSphere Application Server V3.5 imbedded in a single IBM HTTP server.

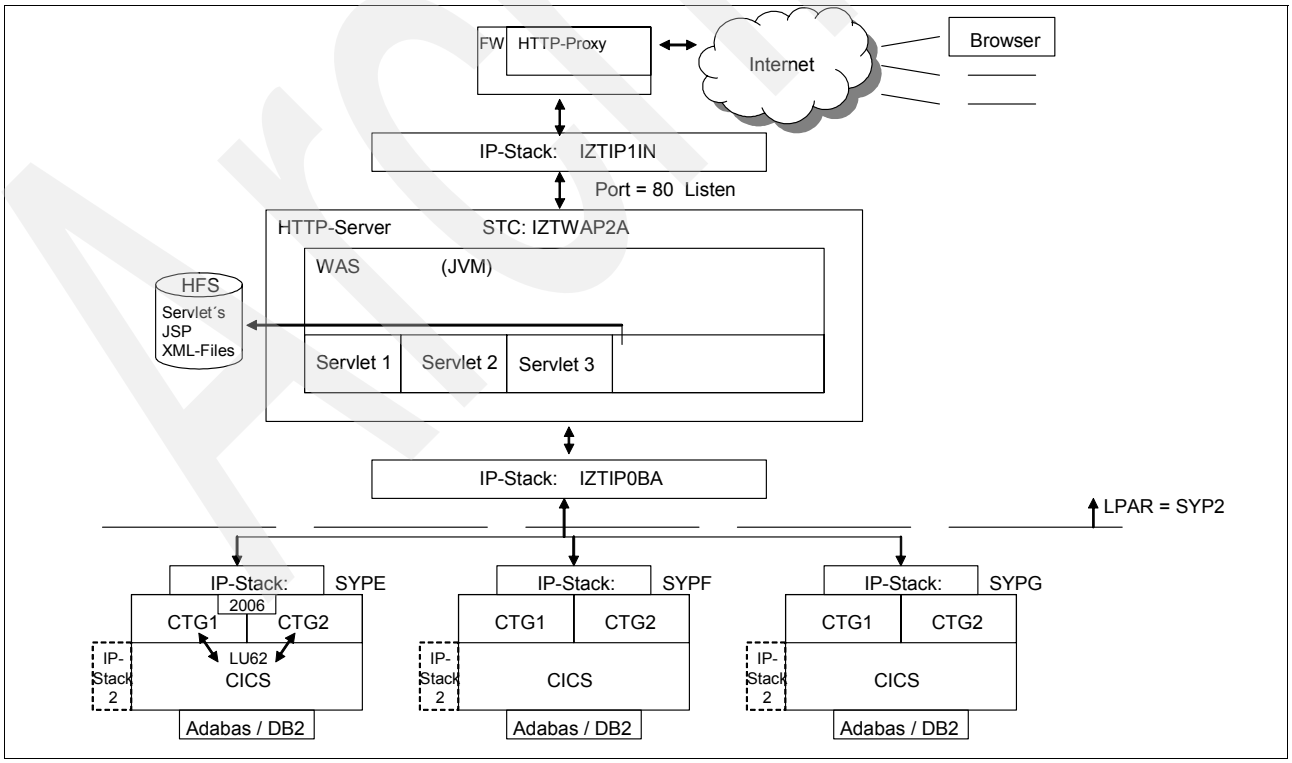


Figure 7-1 Starting point - first configuration

As Figure 7-1 on page 139 shows, IZB started with only one WebSphere Application Server V3.5 imbedded in an IBM HTTP server. Every savings bank had its own application; therefore, servlets existed for every savings bank. The connections to the back-end systems SYPG, SYPF, and SYPE were used by the application for requesting current, daily changing values such as interest rates.

The problem was that if this LPAR had a problem, the application became unavailable. That was a “showstopper” for the planned Internet home banking application as well, which must be available for 24 hours x 7 days a week. The solution was to install the same configuration on the second LPAR, SYP3; see 1.4.4, “Phase 4: 3/2002-10/2003” on page 12. This provided the opportunity to switch incoming traffic to the other LPAR if a problem occurred with SYP2. 7.4.1, “How it works under WebSphere Application Server V3.5 on MVS” on page 159 describes failure and load balancing features.

After a solution for the failover was found, another problem soon appeared. Servlets were crashing in between the JVM due to storage consumption. The heap size was too small, but could not be extended arbitrarily because the size cannot be larger than the virtual storage of the IHS.

This region is limited to 2 GB architecturally. In practice there is 1.5 GB available, because common storage takes up space. Since there would be an application for each of the 80 savings banks, it soon was clear that one JVM would not be enough for the huge number of application programs. The solution for this problem was to use the configuration in scalable mode, as described in the next section.

7.2.2 WebSphere Application Server V3.5 with HTTP server in scalable mode

Scalable mode for IHS is a combination of multiple coherent IHS address spaces controlled by Workload Manager (WLM). One of the IHS (the Queue Manager, or QM) deals with the contact to the clients, the others (the Queue Servers, or QSs) are responsible executing the incoming servlet requests. These work orders are placed on the WLM queue, from which QM and QS derived their names; they handle the same WLM queue.

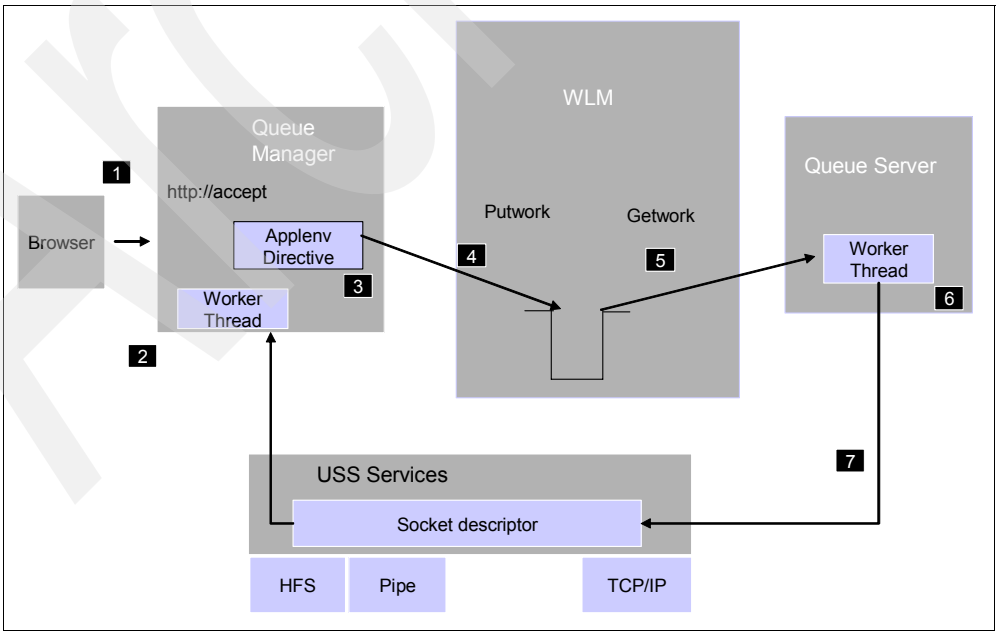


Figure 7-2 Workflow of a request in IHS scalable mode

As illustrated in Figure 7-2 on page 140, a request in IHS scalable mode flows this way:

1. The browser sends a request to the IHS Queue Manager.
2. A worker thread analyzes the request.
3. The requested URI matches to an applenv directive.
4. QM puts the request on the WLM Queue.
5. QS reads the request from the WLM Queue.
6. The JVM in the QS executes the request.
7. QS uses the socket of the QM, through which the request was read, to put the answer directly to the client.

To start the IHS in scalable mode, use the parameter **-SN <subsystemname>** in the startup JCL of the QM. The following examples show the JCL of an IHS in single mode, as QM in scalable mode, and a JCL of a QS.

Example 7-2 shows a sample JCL of an IHS in single mode configuration.

Example 7-2 Sample JCL of an IHS in single mode configuration

```
//IZTWAP2A PROC  ICSPARM='-r /etc/p2.cna',
//  LEPARM='ENVAR("_CEE_ENVFILE=/etc/p2.ena")',
//  LEOPT='RPTOPTS(OFF) RPTSTG(OFF) STORAGE(NONE,NONE,NONE,8K)'
//*******************************************************************
//WEBSRV   EXEC PGM=IMWHTTPD,REGION=0K,TIME=NOLIMIT,
// PARM=('&LEOPT &LEPARM/&ICSPARM')
//STEPLIB  DD DSN=SYS1.CRYPTO.SGSKLOAD,DISP=SHR
//         DD DSN=SYS1.LANGENV.SCEERUN,DISP=SHR
//*******************************************************************
//SYSIN     DD DUMMY
//OUTDSC     OUTPUT DEST=HOLD
//SYSPRINT DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//SYSERR   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//STDOUT   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//STDERR   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//SYSOUT   DD SYSOUT=*,OUTPUT=(*.OUTDSC)
//CEEDUMP  DD SYSOUT=*,OUTPUT=(*.OUTDSC)
```

Example 7-3 shows the sample JCL for the new Queue Manager in scalable mode.

Example 7-3 Sample JCL for the new Queue Manager in scalable mode

```
//IZTWAP2X PROC  ICSPARM='-SN P2X -r /etc/p2.cnx',
//  LEPARM='ENVAR("_CEE_ENVFILE=/etc/p2.enx")',
//  LEOPT='RPTOPTS(OFF)'
//WEBSRV   EXEC PGM=IMWHTTPD,REGION=1500M,TIME=NOLIMIT,
// PARM=('&LEOPT &LEPARM/&ICSPARM')
//STEPLIB  DD DSN=SYS1.CRYPTO.SGSKLOAD,DISP=SHR
//         DD DSN=SYS1.LANGENV.SCEERUN,DISP=SHR
//*******************************************************************
//SYSIN   DD DUMMY
//*OUTDSC    OUTPUT DEST=HOLD
//SYSPRINT   DD SYSOUT=*
//SYSERR      DD SYSOUT=*
//STDOUT      DD SYSOUT=*
//STDERR      DD SYSOUT=*
//SYSOUT      DD SYSOUT=*
//CEEDUMP   DD SYSOUT=*
```

The difference between single server mode and scalable mode is the -*SN <subsystemname>* parameter.

Example 7-4 shows the sample JCL for the Queue Server in scalable mode configuration.

Example 7-4 Sample JCL for the Queue Server in scalable mode configuration

```
//IZTWAP2E PROC IWMSN=,IWMAE=
//  SET SN='-SN '
//  SET AE=' -AE '
//  SET QQ=''''
//WEBSRV   EXEC PROC=IZTWAP2X,TIME=NOLIMIT,
//   ICSPARM=&QQ.&SN.&IWMSN.&AE.&IWMAE.&QQ
```

Every QS starting with the same -*SN <subsystemname>* is associated to the same server complex, which is addressable only through the QM. The server complex could be spanned over different LPARs within a sysplex. So a prerequisite would be shared file systems (HFS/zFS), which were not available in the IZB environment. Therefore, the IZB server complex is located only on one LPAR each. The difference between the QSs is the support of different application environments.

The entire server complex is driven with one common definition file: httpd.conf. An application directive in the httpd.conf file defines a filter for incoming URL requests, the WLM Application Environment, and the transaction class. Through this the server is described.

Example 7-5 Sample application directive statement from the httpd.conf

```
APPLENV /kreissparkasse-augsburg/*              WDFLP2          PWIF
```

In Example 7-5, all requests with the URL `kreissparkasse-augsburg` are routed to the WLM Application Environment WDFLP2 with the transaction class PWIF.

Using secondary directives, each application environment is separately configurable, thus the characteristics of each QS are influenced. One of these secondary directives is called ServerInit. This makes it possible to deal with different was.conf data sets, which are the main definition files of the WebSphere Application Server V3.5; see Example 7-6.

Example 7-6 Secondary directives from the httpd.conf

```
ApplEnvConfig WDFLP2 {
ApplEnvMin 1
ApplEnvMax 1
MaxActiveThreads 40
ServerInit/m/izbp/wasp2/WebServerPlugIn/bin/was400plugin.so:init_exit/m/izbp/wasp2,/etc/izs
    /was/p2x/Web/was.conf_p2a
}
```

That allows you to configure each QS in a different way.

In addition to including startup procedures for each QS in a PROCLIB, the application environment has to be defined in the WLM; refer to Example 7-7 on page 143.

Example 7-7 Two sample workload-managed application environment definitions

```
Appl Environment Name . .      WQS1P2
Description . . . . . . .      WebSphere Queue Server 1
Subsystem type  . . . . .      IWEB
Procedure name  . . . . .      IZTWAP2F
Start parameters  . . . .      IWMSN=&IWMSSNM,IWMAE=WQS1P2


Appl Environment Name . .      WQS2P2
Description . . . . . . .      WebSphere Queue Server 2
Subsystem type  . . . . .      IWEB
Procedure name  . . . . .      IZTWAP2G
Start parameters  . . . .      IWMSN=&IWMSSNM,IWMAE=WQS2P2
```

Scalable mode and the IZB failover concept described in 7.4.1, “How it works under WebSphere Application Server V3.5 on MVS” on page 159 enabled IZB to meet the demands of high availability for the new Internet home banking application, which went into production in March 2003. Therefore, IZB installed eight new queue servers per LPAR (in SYP2 and SYP3) for the Internet home banking application, in addition to the already existing servers for the application Webfiliale. The savings banks had the opportunity to switch from the old UNIX application to the new HTML solution on their own time schedule. By the end of 2003, all 80 savings banks had switched and the old applet application was “history”.

Due to a rollout of the savings banks and new functions in the Internet home banking application, IZB soon reached storage limits again. As a consequence, it was decided to spread the savings banks to 21 QS per LPAR. This allocation continues to this day.

In addition to the Internet server complex for the Internet home banking application, there is also an intranet server complex available on SYP2 and SYP3, serving intranet applications for the savings banks; see Figure 7-3 on page 144. This complex is running with 21 Queue Servers on each side.

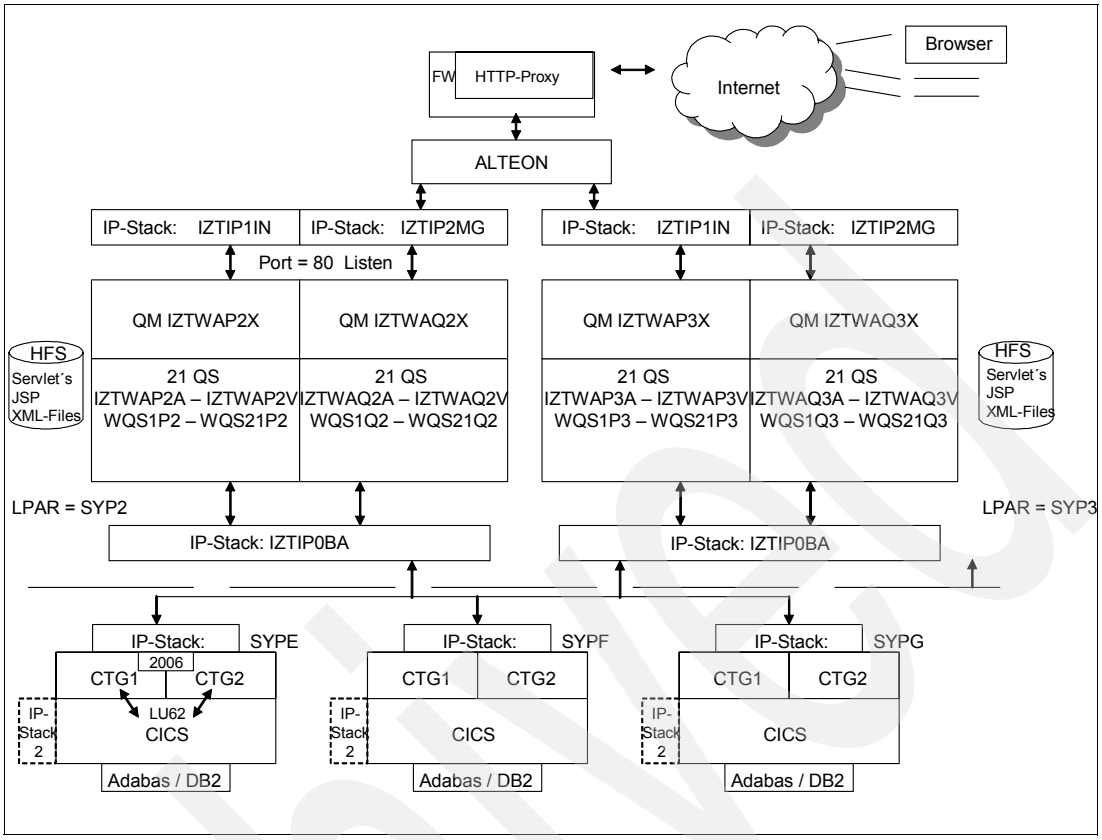


Figure 7-3 WebSphere Application Server V3.5 environment on LPAR SYP2 and SYP3

In 2003, IBM stopped maintenance for WebSphere Application Server V3.5. As mentioned in 7.3, “Implementation of WebSphere Application Server V5” on page 145, IZB and Customer B decided to use Version 5 for the future. Because the existing applications could not be migrated to V5 very quickly, a decision was made to run them with WebSphere Application Server V4.0.

Version 4 was the first one to be ready for the demands of J2EE, and the applications experienced a problem with the existence of elements such as servlet and EJB™ containers. The solution was to disable the J2EE ability of the WebSphere Application Server V4 and run it in compatibility mode like V3.5. This was handled through a parameter in the was.conf data set; see Example 7-8.

Example 7-8 The parameter that disables J2EE ability

```
appserver.java.extraparm=-DDisable.J2EE=true
```

The migration to compatibility mode was completed in July 2003.

7.2.3 Challenges

IZB experienced a problem with WebSphere Application Server V3.5 applications in dealing with JSP compiles. When bringing new software into production, the servers must be stopped. Afterwards the working directories with the compiled JSPs must be deleted.

Restarting the servers normally means making them available for the Internet. This leads to a very large number of JSP compiles in a short time period. The main index pages were called

in first, and were very often compiled twice or more. The JSP compiler seemed to have a problem with parallel work, so the servers were frequently looping.

As a result, Customer B tried to compile the JSPs with the available batch compiler before putting the servers in production again. But the batch compiler experienced problems with Taglibs, which are small Java applications used in the Internet home banking application. The Tablig code is located in various modules that are loaded from the JSPs in complete units. These problems could not be resolved.

As a circumvention, IZB wrote a procedure that runs before the servers are put into production. The procedure initiates a HTTP Get request that compiles using the normal JSP compiler, which does not have a problem with Taglibs. Compiling the 50 most-used JSPs for all 80 customers is a planned activity and takes about 150 minutes. After completion, the servers are connected to the Internet as described in 7.4.1, “How it works under WebSphere Application Server V3.5 on MVS” on page 159.

The Internet home banking application has gained many new features since implementation, so the acceptance and utilization continue to increase. Meanwhile its CPU consumption during a normal workday is too high to support on one LPAR. Thus IZB and Customer B decided to not use the failover scenario described in 7.4.1, “How it works under WebSphere Application Server V3.5 on MVS” on page 159 between 7 a.m and 9 p.m., as this would result in an outage for half of the Internet home banking application users during a problem period.

Therefore, a project was started to migrate the old WebSphere Application Server V3.5 applications to WebSphere Application Server V5 through mid-2006. This is also a prerequisite for a planned change of operating system from z/OS 1.5 to z/OS 1.6. Neither WebSphere Application Server V3.5 nor V4.0.1 (in compatibility mode to V3.5) are operational with z/OS 1.5.

7.3 Implementation of WebSphere Application Server V5

In 2003, IZB’s Customer B decided to start a new project called “recentralizing of decentralized systems (RDS)”. This is a centralized mandatory intranet Web application for employees of the savings banks. This application provides an entrance to the working applications based on Windows® terminal servers. The RDS Portal System is based on a centralized Web application that is coded with J2EE technology (JSP, Servlet, SFA Framework) running on WebSphere Application Server on z/OS and distributed components installed on Windows terminal servers.

For the Web application, IZB and Customer B decided to use WebSphere Application Server on z/OS V5.0. This version is fully ready for Parallel Sysplex, so it can be used across LPAR borders. The decision to use V5 instead of V4 was made because the new version consists of a number of reworked modules. DB2 and LDAP were prerequisites in V4, naming service and interface repository had their own started task, and the administration client was a Java fat client application. Version 5 was designated by IBM to be state of the art in the near future.

IZB ordered WebSphere Application Server V5 in mid-2003. For safety, IZB wanted to separate the old V3.5 and the new V5 worlds initially; see Figure 7-4 on page 146. Therefore, a decision was made to increase the production Webplatform host from two to four LPARs. The first step dealt with the new version only on the new LPARs (SYP8 and SYP9), with the intention of extending the new V5 environment to all four LPARs.

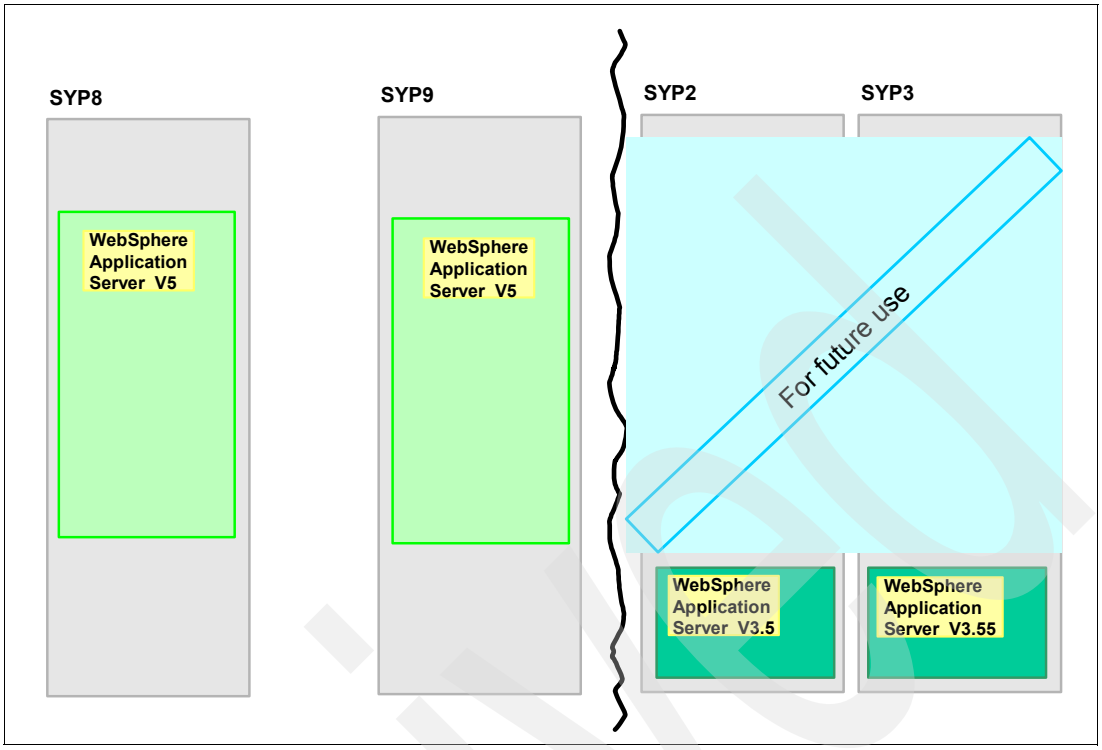


Figure 7-4 Initially planned configuration

7.3.1 Brief description of some WebSphere Application Server V5 terms

This section provides brief explanations of WebSphere V5 terminology, using Figure 7-5 on page 147 as a reference point.

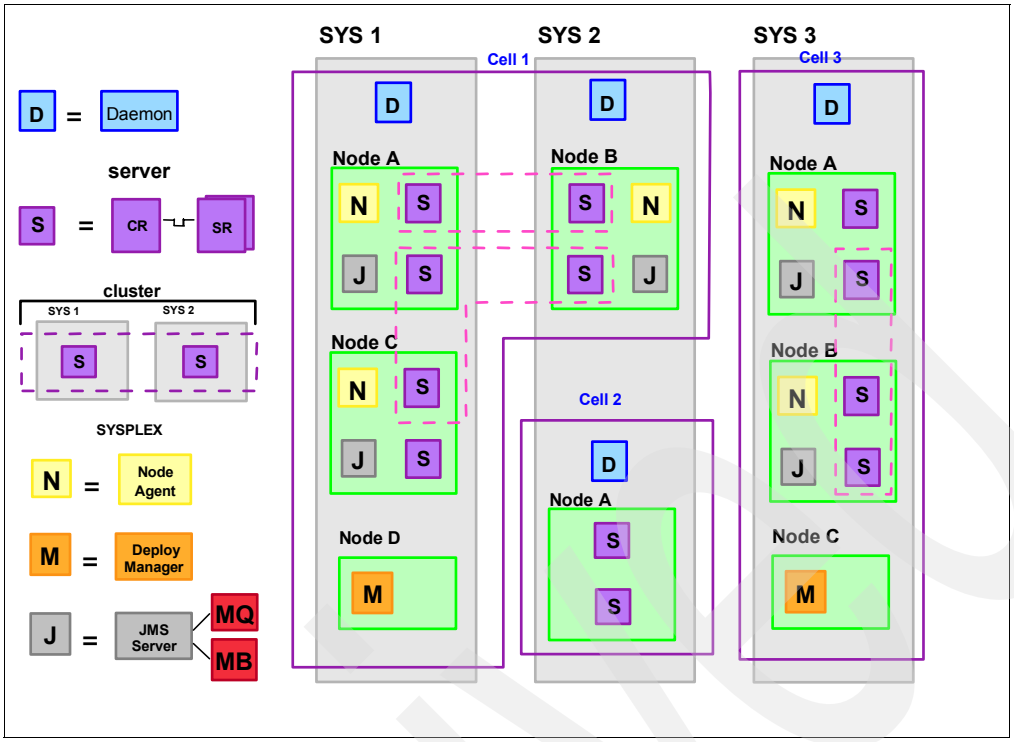


Figure 7-5 Basic WebSphere Application Server V5 configuration

- Cell
  - Logical collection of nodes
  - Boundary of the administrative domain
  - Can be spanned over one or more LPARs in between a Parallel Sysplex, but not to another sysplex
- Node
  - Logical collection of servers on a particular LPAR in a cell
  - Exists mainly for the purposes of administration
  - Normally a node agent is available (except the Deployment Manager node)
- Cluster
  - Logical collection of like-configured servers
  - Can span nodes and systems within the same cell
- Server
  - Consists of a controller and at least one servant
  - Controller
    - Runs system authorized programs
    - Manages tasks such as communication
    - Started by a start command
  - Servant
    - Java Virtual Machine resides here and runs the application programs
    - Started from Workload Manager
    - Dynamic startup can be controlled through the administration console
- Daemon
  - Initial point of client requests
  - One per LPAR and cell respectively

- Node Agent
  - One per node (except the Deployment Manager node)
  - Administers the application server in the node
- Deployment Manager
  - One per cell
  - Hosts the administrative console application
  - Provides cell-level administrative function
- JMS-Server
  - Integrated Java Message Service Provider
  - Not installed at IZB, since full-function WebSphere MQ V5.3.1 is used
- Federation
  - Taking the node structure of a Base Application Server node and merging it into the cell structure of a Deployment Manager cell
- Network Deployment Configuration
  - A group of nodes with a single Deployment Manager node and one or more application server nodes
  - Can span multiple systems in a sysplex
  - Fully utilize clustering

Note: With WebSphere Application Server V5, a choice must be made between two main environments: base, or Network Deployment (ND) Configuration. The ND Configuration was the only one that could support IZB's high availability efforts.

This discussion of IZB's WebSphere Application Server V5 environment refers to the ND Configuration, although a base application server must be installed first.

7.3.2 Installation and running on two LPARs

To install WebSphere Application Server on z/OS V5, IZB followed the instructions given in *Getting Started - IBM WebSphere Application Server for z/OS V5.0.2*, GA22-7957, which can be ordered from here:

http://ehone.ibm.com/public/applications/publications/cgibin/pbi.cgi?SSN=06GRC0015550262588&FNC=PBL&PBL=GA22-7957-04PBCEEB0200001795&TRL=TXTSRH

Because IZB was installing a new product, many problems were encountered. For example, the installation dialog was incorrect, so IZB had to install fixes. New service levels were available at frequent intervals, so receiving an error often meant installing a new service level. Activating security was not possible in the beginning, and later it was difficult to federalize with security switched on. Furthermore there were problems with the Java Service Levels; if applications crashed in a servant, there were no notifications given to the controller, and so on. It was noted that the product seemed to be developed for the decentralized world initially, and adapted to centralized world afterwards. IZB's guideline of not using new software was proven as a result of this initial installation.

As mentioned, the decision was made to use WebSphere Application Server for z/OS V5 in the IZB sysplex environment across LPAR borders in a Network Deployment Configuration. A useful document (document number WP100367), available from Washington Systems Center, described in detail the steps to reach this goal; the document can be found at:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100367

(The Techdocs Web site is the official home of Washington Systems Center documentation. Documentation for newer versions should be located there as well.) As it happened, the situation described in that document fit IZB's plans exactly, so IZB followed the instructions for the most part. The main steps were:

- Planning
- Installing Base Application Server node on SYP8
- Installing Deployment Manager cell on SYP8
- Federate Base Application Server node into Deployment Manager cell
- Installing Base Application Server node on SYP9
- Federate Base Application Server node into Deployment Manager cell
- Building a Cluster across SYP8 and SYP9

The following section focuses on parts of the plan that are worth special attention up front, because they are difficult to change in a production environment.

Naming conventions

Choosing the naming conventions was very difficult, as IZB had to set up names for many parts in the installation, such as cells, nodes, clusters, and servers. Without knowing the amount of future growth, the naming conventions needed to be able to support both horizontal and vertical growth. Table 7-1 lists the IZB naming conventions for cells, nodes, clusters and servers.

Table 7-1 IZB naming conventions for cells, nodes, clusters and servers

Cell	**For example, WpC001**
W	Fix
p	Mark for the WebSphere Installation
	For production systems numeric 0 - 9
	For development systems ascending in alphabetic range a - s
	For integration systems descending in alphabetic range s - a
	For test systems ascending in alphabetic range t- z
C	Fix for "Cell"
001	Continuous number
Note:	A base cell has an additional B in the name, for example, WpCB001

Node	**For example, WpN001**
W	Fix
p	Mark for the WebSphere Installation (same as for cell)
N	Fix for "Node"
001	Continuous number (if character D follows, it is a DM Node)

Cluster	**For example, IZWpA1**
IZW	Fix
p	Mark for the WebSphere Installation (same as for cell)

Cluster	For example, IZWpA1
A1	“Group number” for server with the same functionality

Server	For example, IZWpA11 (Control-Region)
IZW	Fix
p	Mark for the WebSphere Installation (same as for cell)
A1	Application Server A1
1	Continuous number for differentiation of the servers

Started tasks for the controllers and servants should reflect the names for cells, nodes, clusters, and servers, but their names are limited to eight characters. Using the conventions of IZB, the first two characters reflects the client, the third position describes the type of work (batch job, user, started task, and so on).

The eighth character is reserved for the servants of an application server. These servants were marked with an S. Therefore, there are only four characters available to convey a great deal of information, such as the affiliation to a WebSphere cell, and a WebSphere node and the cluster name. For this reason, IZB decided to emulate Example 7-9, which reflects the names for the started tasks for the cells, nodes, clusters, and servers.

Example 7-9 Reflecting the names for the started tasks

STC for Controller	IZWpaan
IZ	Prefix for client
W	For the product WebSphere
p	Mark for the WebSphere Installation (same as for cell)
aa	“Group number” for server with the same functionality
	0a - Daemon
	1a - Deployment Manager
	2a - Node Agent
	Aa - Application Server
n	Consecutive number for particular server

STC for Servant	IZWpaanS
IZWpaan	Same as for Controller
S	Fix for WLM-managed Servant address space

Figure 7-6 on page 151 illustrates the situation when going in for production with RDS and naming conventions.

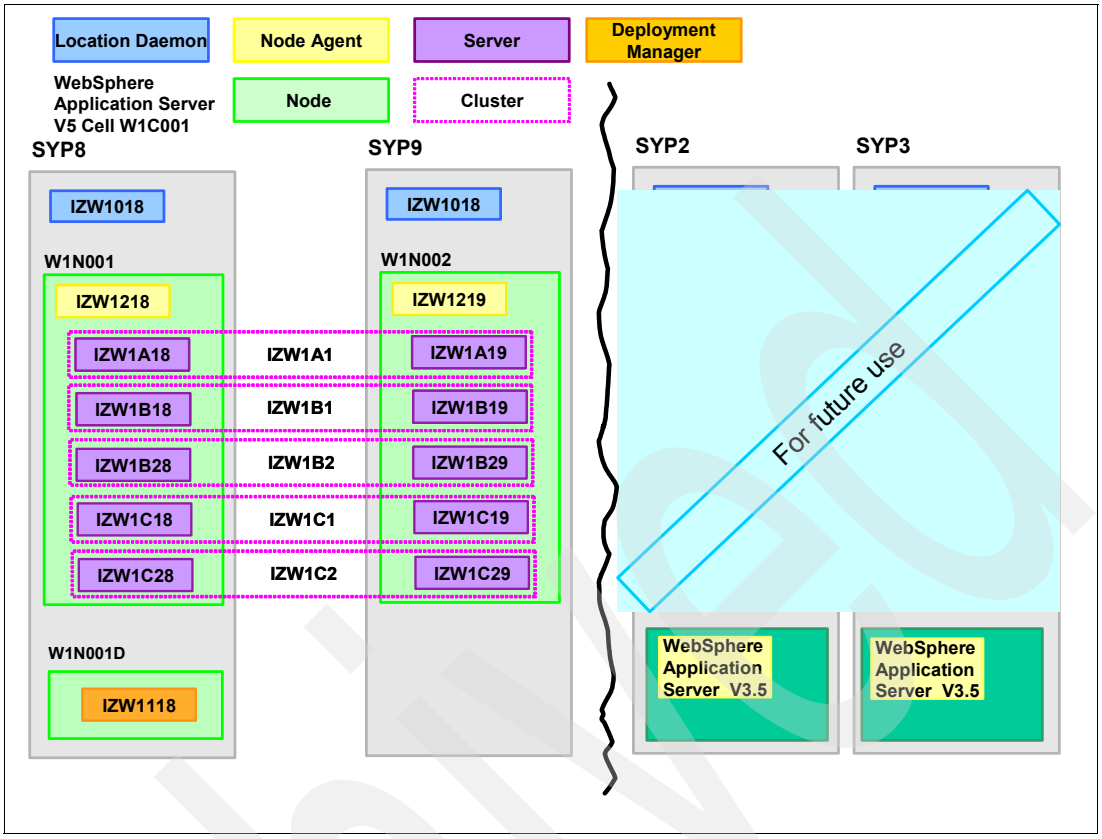


Figure 7-6 Going in to production with RDS and naming conventions

Because IZB has only one server in each cluster per LPAR, it uses the last character as the LPAR identifier. The advantage this offers is that all servers on the same LPAR end with the same character. However, this does not apply for Location Daemon. Every location daemon of a federalized node gets the started task name of the location daemon of the target cell. This is something you cannot change.

The name of the cluster is identical with the first six characters of the application server and is used for the WLM application environment. Different clusters were installed to allow separate applications.

- IZW1A1: Used for self-written monitoring software
- IZW1B1: Used for production version of RDS
- IZW1B2: Used for pilot version of RDS
- IZW1C1 and IZW1C2: Used for future expansion. Not yet active today.

Rule

- Every WebSphere STC starts with IZW*
- 4th character reflects the cell
- If the fifth character is a letter => it is an application server
- If the fifth character is a digit => it is a infrastructure server (Daemon, Deployment Manager, or Node Agent)

Important: When planning to install the first WebSphere Application Server on z/OS, use care when creating the naming conventions. Consider the needs of future expansion in horizontal and vertical directions, because changing names in a running production system is very difficult, and perhaps impossible.

Port concept

For complex Web environments on System z platforms, a very large number of ports are needed:

- Each application server -> at least six ports
- A Deployment Manager -> nine ports
- A Node Agent -> six ports

IZB's target was to find a solution that was as easy as possible, but which also provided a clear assignment from port to product. IZB did not want ports spread over the entire possible range. Therefore it reserved a range of 1000 ports for its Web environment. The ports are located between 11000 and 11999 and are used solely through OMVS services. The flexibility in defining new application servers very quickly should not be lost, in order to allow a reservation of single ports for single jobs. Table 7-2 shows part of IZB's port reservation list.

Table 7-2 Extract of the port reservation list

Server	HTTP	HTTPS	SOAP/ JMX™	ORB/ Boot	ORB SSL	DRS/ Client	Node Discover	Node multi cast	Cell Discover	Protocol IIOP Daemon	Daemon SSL
IZW1018										11018	11019
IZW1118	11016	11017	11011	11014	11015	11013			11012		
IZW1218			11020	11024	11025	11021	11022	11023			
IZW1A18	11006	11007	11001	11008	11009	11000					
IZW1B18	11106	11107	11101	11108	11109	11100					
IZW1B28	11116	11117	11111	11118	11119	11110					
. . .											
IZW2018										11218	11219
IZW2118	11216	11217	11211	11214	11215	11213			11212		
IZW2218			11220	11224	11225	11221	11222	11223			
IZW2A18	11206	11207	11201	11208	11209	11200					

In IZB's environment, 10 ports were reserved for each application server. The list shown in Table 7-2 illustrated the development of the environment in its sequence. The base cell with the application server IZW1A18 gets ports starting from 11000. Subsequently, IZB installed the Deployment Manager (starting with 11010) and the Node Agent (starting with 11020). Initially IZB had no idea how many Node Agents would be required in the future, so it left a gap and defined the second application server starting at 11100. For the later installed cell W2C001, this gap is much smaller.

Inside a cell, all location daemons, all node agents, and all application servers within a cluster listen on the same port number on all of IZB's WebSphere LPARs. This is very important for failure and load balancing with sysplex distribution, as described in 7.4.2, "How it works under WebSphere Application Server V5 on z/OS" on page 161.

Data sets and RACF in a network deployment environment

Starting with the installation of WebSphere Application Server V5, IZB has the following initial environment:

- Different type of data sets: filesystems and MVS data sets

- Shared data sets but no shared filesystems (HFS/zFS)

In choosing names for data sets and RACF profiles, IZB had two main objectives:

- Simple activation of new service in an LPAR
- A single method for the different types of data sets

Using shared data sets and aliases would be easy. During migration time, only two sets of shared data sets need to be stored, and aliases for every data set in the catalog could point to the valid data sets.

But because IZB does not use shared file systems, it has two different solutions for activation; it decided to:

- Store a set of data sets for each LPAR and for each service level
- Use symbolic links for the filesystems

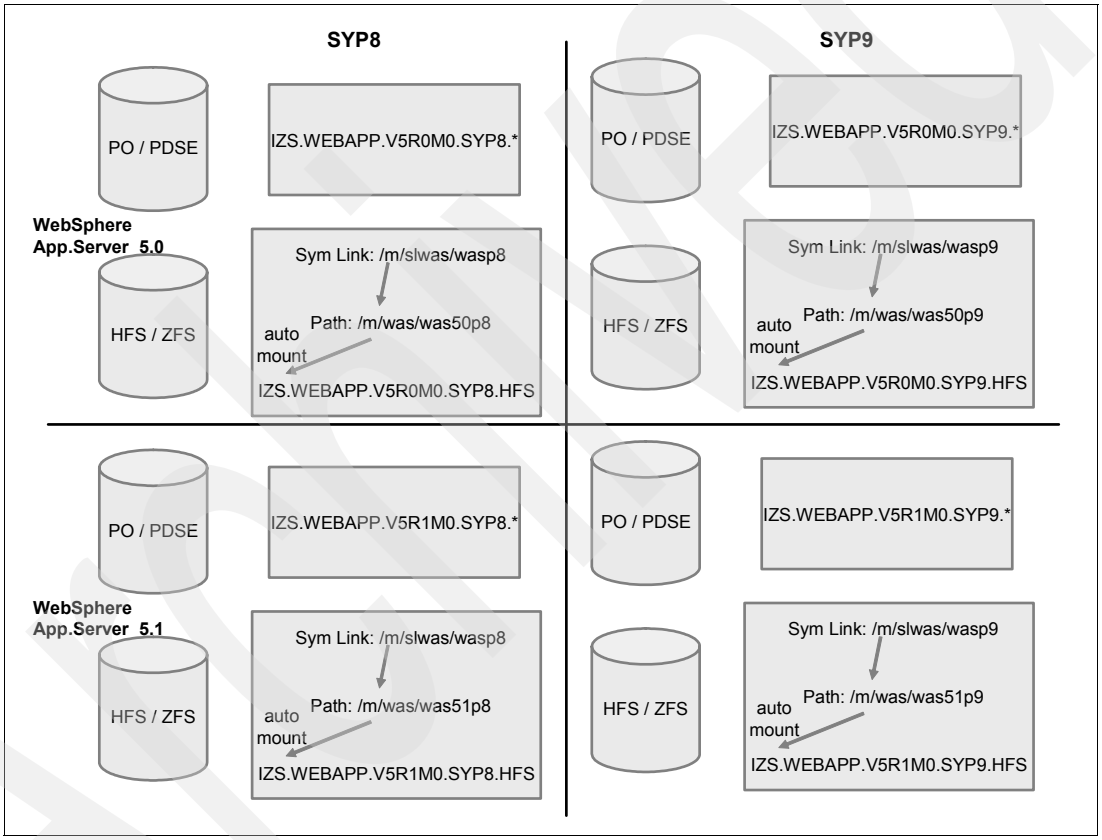


Figure 7-7 The names of data sets at migration time, with two of the LPARs shown

Figure 7-7 shows the names of some IZB data sets at migration time. The first two qualifiers are reserved, in accordance with IZB naming conventions. The third qualifier is used for the service level. This allows IZB, during a migration, to store two sets at different service levels. The fourth qualifier relates the data set to one of the four WebSphere LPARs.

For USS files in an HFS or zFS filesystem, a similar methodology is used. IZB uses symbolic links and the file system is auto-mounted. In Figure 7-7, the 50/51 describes the release and the p8/p9 describes the LPAR. Using symbolic links provides IZB with much more flexibility if the path changes. Here again IZB has to include the release level, due to parallel file systems during a migration.

Theoretically it is possible to store a set of data sets per LPAR and per cell to migrate each node of a cell separately. This requires the use of STEPLIB concatenation in the STC. Actually, IZB uses STEPLIB concatenation because it still uses WebSphere Application Server V3.5, which makes it impossible to use LINKLIST. But in the future IZB plans to switch to LINKLIST for performance reasons. For RACF profiles, the supplied installation is very specific in some parts. IZB attempted to be as generic as possible to allow easy and quick installation of additional nodes, clusters, or application servers, see Example 7-10.

Example 7-10 Generic profiles for our cell W1C001

```
Class: CBIND   Profile: CB.BIND.IZWW1C00.IZW1*
Class: CBIND   Profile: CB.IZWW1C00.IZW1*
Class: SERVER  Profile: CB.*.IZW1*.**
Class: STARTED Profile: IZW1%%%S.*   STDATA(USER(IZW1ZZS) GROUP(IZAW1CF)
Class: STARTED Profile: IZW11%%S.*   STDATA(USER(IZW11ZS) GROUP(IZAW1CF)
```

Note: The first profile in Class: STARTED is for the servants of the application server. The second and more specific profile is for the servant of the Deployment Manager. Remember the naming conventions: “1” in the fifth place => Deployment Manager.

7.3.3 The complete picture

Growing to four LPARs

After installing WebSphere Application Server on z/OS V5.0 on IZB’s two new production systems, SYP8 and SYP9, and running with the application RDS for four savings banks in a pilot service for several months, IZB decided to extend the production cell with two new nodes. IZB completed these tasks:

- Installing Base Application Server node on SYP2
- Federating Base Application Server node into Deployment Manager cell
- Installing Base Application Server node on SYP3
- Federating Base Application Server node into Deployment Manager cell
- Extending the Cluster across SYP2 and SYP3

At this point, IZB reached the installation objective previously described; see Figure 7-8 on page 155.

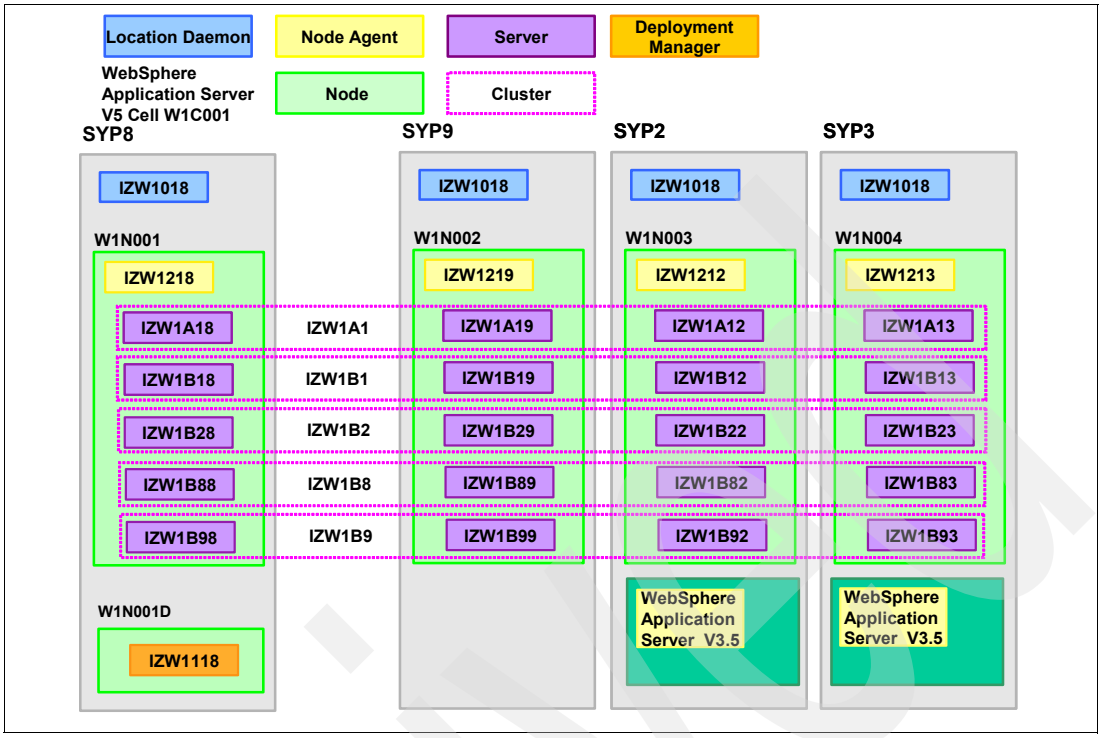


Figure 7-8 Webplatform host after extending WebSphere Application Server V5 to four LPARs

The clusters IZW1C1 and IZW1C2 are not yet active, so they are not shown in Figure 7-8. Instead there are two new clusters, IZW1B8 and IZW1B9.

- IZW1A1: Used for a self-written monitoring utility
- IZW1B1: Used for production version of RDS
- IZW1B2: Used for production version of RDS
- IZW1B8: Used for pilot version of RDS
- IZW1B9: “Sandbox”. New users of RDS are routed to this cluster first.

The connection to the Internet

Figure 7-9 on page 156 gives an overview of the configuration along with the data flow of a client request. For better clarity, Node Agent, Deployment Manager, and Location Service Daemon are not shown. Neither are some other components, such as terminal server farms or LDAP Servers, that are necessary for running the RDS application.

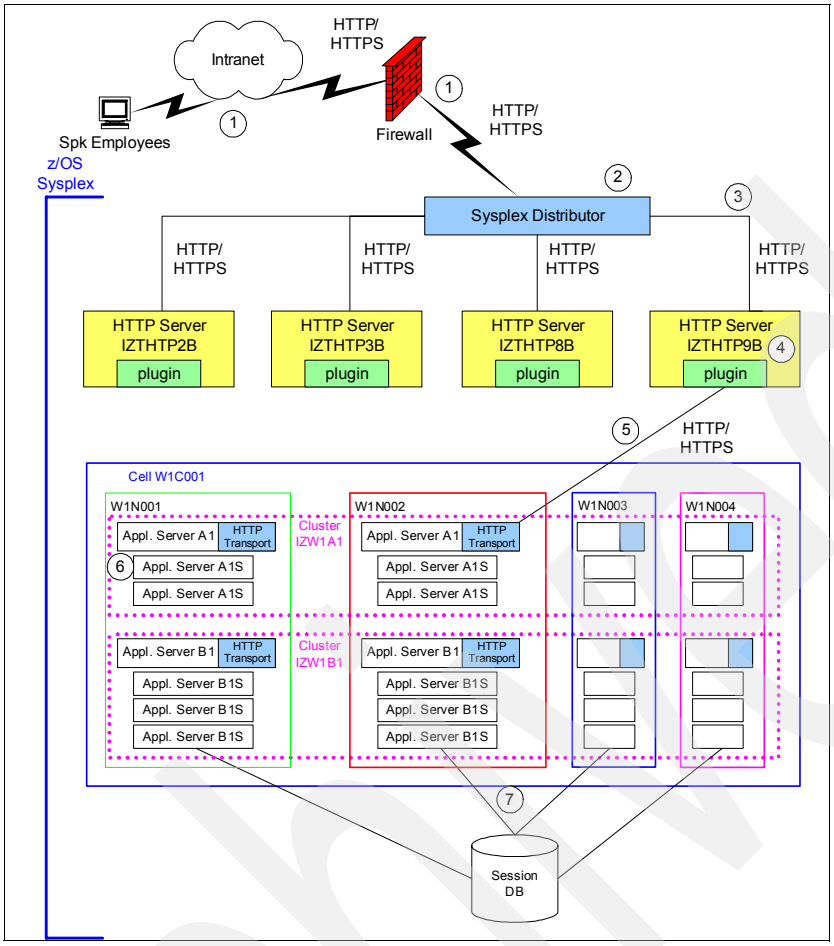


Figure 7-9 Connection to the Internet

The flow is as follows:

1. The request from the employees of the savings bank comes through the intranet and firewall to the Sysplex Distributor.
2. The Sysplex Distributor spreads the request to one of the LPARs, based on WLM criteria.
3. The request is routed to the IHS.
4. The WebSphere Application Server plug-in that is in the IHS determines whether the request is stateful or stateless. The first request is always stateless. This means there is no information such as Session IDs or cookies available for this request.
5. Stateful requests are routed to the correct application server through the information in cookie JSESSIONID (target IP address is the address of the application server). Stateless requests are routed to a server chosen by a round-robin method.
6. The application server inspects the session identifier form the cookie JSESSIONID. The “affinity token” is related to an MVS ENQ, which was created by the server region in which the session object was created. The control region checks to see if the ENQ is still held.

   If the ENQ is still held, then the control region queues the request to the proper server region. If the ENQ is not held, then the control region queues the request to the general WLM queue.
7. Update of the session date in the session database.

The Web application in its special infrastructure

The project RDS has a very complex infrastructure. The Web application is only one part of it. There are many distributed components installed on Windows terminal servers, as well as other decentralized components.

As a result of this design, the Web application has multiple remote connections. Therefore, any problem in the background (for example, lengthy response times or failing components) often leads to problems with the application through accumulation. Referring to Figure 7-9 on page 156, consider the following possible scenario:

- A problem occurs with a component in the background (for example, the session database).
- All threads in the servants are busy.
- Nevertheless, the controller puts more user requests into the WLM queue.
- HTTP server knows nothing of the problem and the WebSphere Application Server plug-in gets no feedback => many threads in HTTP server are busy.
- If all threads in the IHS are busy => requests are rejected for all application servers.
- In the worst case, there is a complete outage of RDS.

Refer to 7.4.2, “How it works under WebSphere Application Server V5 on z/OS” on page 161 for a description of the actions that IZB and Customer B have taken to avoid a complete outage and reduce the impact of this kind of problem.

7.3.4 Migration to WebSphere Application Server for z/OS V5.1

In February 2005, IZB decided to migrate to WebSphere Application Server for z/OS V5.1. Starting with the basic installation using SMP/E and ending with migrating the last production LPAR, it took several months. This was due to errors in early service levels of the new release leading to repeated installation of newer service levels. In addition, IZB had to raise the service level of the old release level 5.0. The migration process is different, depending on which service level the migration starts from, so IZB had to start its migration twice.

Release levels 5.0 and 5.1 can be used parallel in a Network Deployment configuration. By using the data set naming conventions described in “Data sets and RACF in a network deployment environment” on page 152, IZB was able to migrate the servers for each LPAR separately.

Again, a document available from the Washington Systems Center gave a detailed road map for guidance in this complex endeavor. The document number is WP100441 and can be found at:

http://www-03.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100441

IZB followed this document for the most part. Certain basic steps concerning the sequence had to be followed. The Deployment Manager node had to be migrated first. Afterwards the other nodes could be migrated. The application server node on the LPAR where the deployment manager node resided was migrated during the same steps as the DM node. The three nodes on the other LPARs were migrated later.

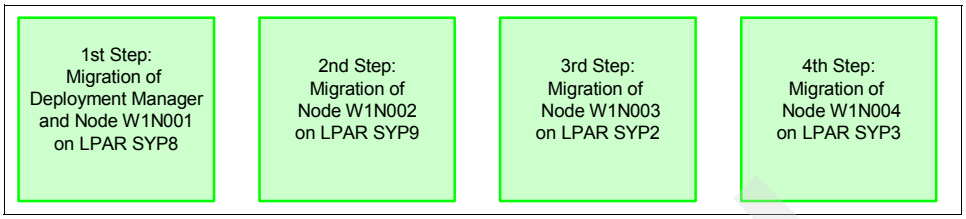


Figure 7-10 Sequence of the migration

On the first production day after the first migration, IZB lost its production server IZW1B28 on the migrated node. The failure was caused by a change of default stack size on the migrated servers, which was not documented. With the increasing load on the systems, IZB incurred storage problems quickly. Because of the Parallel Sysplex and having three production servers in this cluster running on the old release level, IZB was able to avoid an outage.

Circumvention for the problem is a bypass of default stack size by using a special parameter in the JCL. Because the JCL is shared over all four LPARs, IZB had to take care that the change in stack size is activated only for migrated servers. This was done by using the system variable *$SYSNAME* (see note on page 159).

Here are hints and tips regarding migration:

- All servers, even inactive servers, should be restarted before starting the migration.

  Migration job BBOXMIG1 flushes the transaction logs for each server, and you have to restart each server for a successful migration. So ensure that each server is ready to start, to avoid other problems that might accompany a migration.

- Copy the plug-in from the old config data set in the DM config directory *after* migration of DM, but *before* the DM restart with V 5.1.

  After migration of the DM, there is no plug-in in the directory; see Figure 7-11.

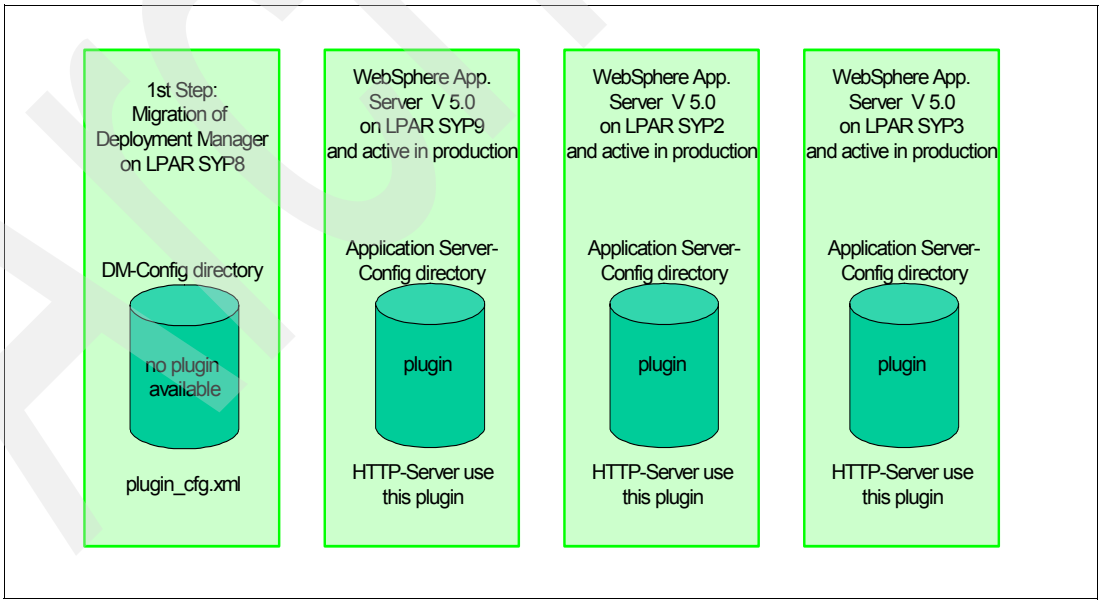


Figure 7-11 The plug-in during migration of the DM

If you restart the DM without a plug-in and the other Node Agents do a synchronization, then the affected HTTP Server has no plug-in available after refresh, and an outage *is* possible.

- Avoid using job BBOXMIG4.

  Job BBOXMIG4 deals with new STC names after migration, which in turn means work for the staff responsible for the automation commands.

> **Note:** After migration, the variable ROOT and the STEPLIBs must point to the new data sets for all affected V5.1 STCs. The old V5.0 STCs must point to the old data sets.
>
> Problem: One shared PROCLIB with only one STC procedure should be started with a different ROOT variable and different STEPLIBs.
>
> Solution: Use system variable *$SYSNAME* in the proclib.

Example 7-11 shows the JCL for the applications servers using system variables.

Example 7-11 JCL for the application servers with the use of system variables

```
//IZW1ZZC  PROC ENV=,PARMS=' ',Z=IZW1ZZCZ
//** set variables specific for each system through INCLUDE IZW1syid **
// INCLUDE MEMBER=IZW1&SYSNAME
//STEPLIB  DD DISP=SHR,DSN=IZS#.WEBAPP.&RELSTAND..&SYSNAME..SBBOLD2
//         DD DISP=SHR,DSN=IZS#.WEBAPP.&RELSTAND..&SYSNAME..SBBOLOAD
```

Example 7-12 shows the Include member for the first migrated LPAR.

Example 7-12 Include member IZW1SYP8 for the first migrated LPAR SYP8

```
//********* set variables specific for each system ********
//* SET ROOT='/m/was/wasconf1'
//* SET RELSTAND='V5R0M0'
// SET ROOT='/m/was/wasconf2'
// SET RELSTAND='V5R1M0'
// SET LE='STACK(128K,128K,ANY,KEEP,128K,128K),NOUSRHDLR' for WAS 5.1
```

- Set *TrustedProxy = true*.

  You have to set *TrustedProxy = true* if the connection through the HTTP plug-in should work after the migration. In the administration console use the path: **Servers** → **Application Servers** → **Web Container** → **Custom Properties** to set the authorized proxy.

- Variable *ras_log_logstreamName*

  If you have not set this variable in Version 5 servers, you must delete it after migration because it is set during the migration process. Errors occurred during startup of the application servers. The path in the administration console is: **Environment** → **Manage WebSphere Variables**. Select a node scope.

7.4 Failure and load balancing

7.4.1 How it works under WebSphere Application Server V3.5 on MVS

As described in 7.2, “Implementation of WebSphere Application Server V3.5” on page 138, there are intranet applications running under WebSphere Application Server V3.5, along with a heavy load from the Internet home banking application. Failure and load balancing are similar for both environments; this section discusses Internet home banking.

Because WebSphere Application Server V3.5 is not ready for Parallel Sysplex, IZB has to maintain complete system and application environments in parallel on both affected LPARs SYP2 and SYP3. For load balancing, half of the savings banks are routed to each server complex, IZTWAP2X and IZTWAP3X. This routing happens through the Uniform Resource Locator (URL).

Clients of the savings banks that are located on SYP2 call the application through portal.izb.de. The other clients call the application through portal1.izb.de. After the request is routed from the Internet through the firewall, it reaches an ALTEON load balancer. This is hardware between the firewall and the mainframe that is responsible for load balancing and routing of requests based on defined rules. The ALTEON recognizes, on the basis of the URL, whether a request is for SYP2 or SYP3, and routes it to the appropriate Queue Manager.

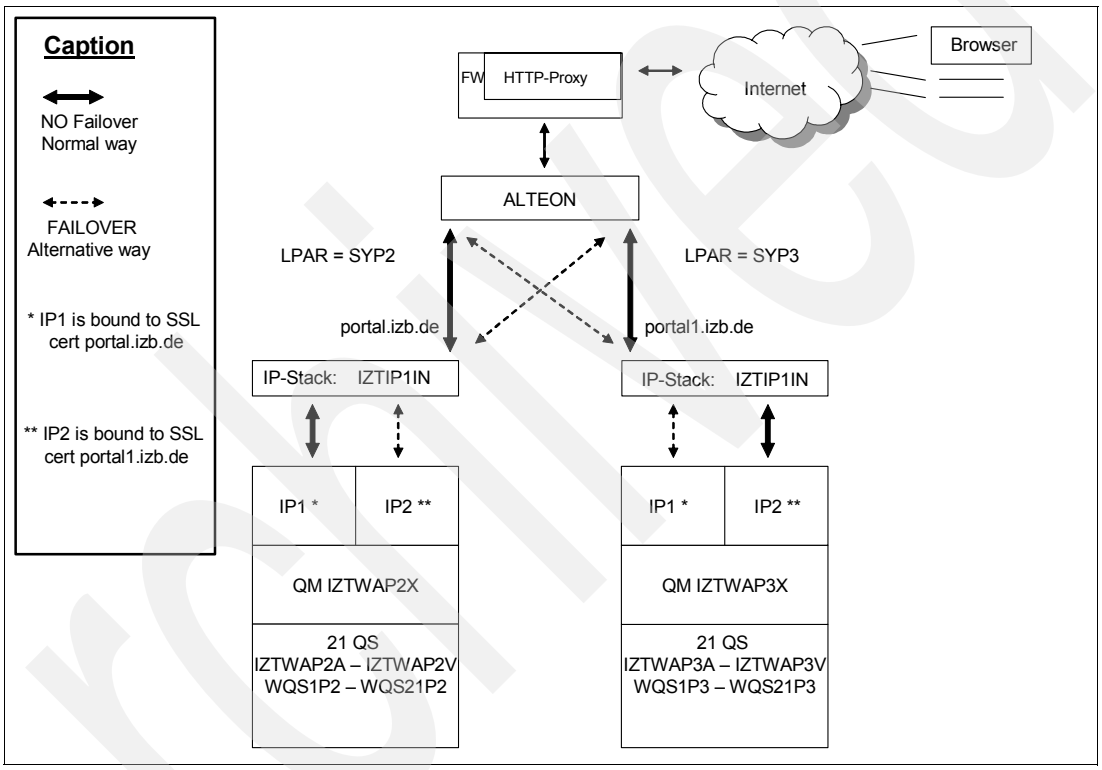


Figure 7-12 Load balancing for the Internet home banking application

Note: The ALTEON load balancer and the firewalls are redundant, as well.

ALTEON continually checks, at short intervals, whether the QM is available. This happens with a simple HTTP Get request on a static HTML page stored in USS. If ALTEON gets an HTTP Statuscode of 200 (request OK), it knows that the real server is available and sends the request to the QM. If the page is not available, ALTEON marks the real server as not available and the incoming requests are routed to the other LPAR. “Not available” pages would result, for example, if the LPAR has an outage or if the QM is not available.

Note: Internet home banking requests are SSL requests. Both SSL authorized certificates for portal.izb.de and portal1.izb.de are located in key databases. For the failover scenario, both are available on SYP2 and SYP3.

For planned maintenance, IZB created REXX execs. These allow IZB to control the ALTEON using some Autooperator commands without human intervention. The REXX ALTCTL (ALTeonConTroL) deletes or creates the static HTML pages in the USS, and allows IZB to control the incoming requests; see Table 7-3.

Table 7-3 The function of REXX ALTCTL

Call	Description
ALTCTL CHECK	Shows the status of all controlled servers
ALTCTL <server> CHECK	Shows the status of a specific <server>
ALTCTL <server> INACT ALL	Deactivates the requests for a specific <server>
ALTCTL <server> INACT <appl>	Deactivates requests for a specific <appl> for a specific <server>
ALTCTL <server> ACT ALL	Activates the requests for a specific <server>
ALTCTL <server> ACT <appl>	Activates requests for a specific <appl> for a specific <server>

7.4.2 How it works under WebSphere Application Server V5 on z/OS

"The Web application in its special infrastructure" on page 157 describes the special problems that IZB faced with the RDS project. IZB undertook many actions to avoid a complete outage and reduce the impact of problems, as described here:

- IHS in scalable mode

  Based on experiences through the implementation of IHS scalable mode for IZB's Internet home banking application, it was decided to use scalable mode for the WebSphere Application Server V5 environment as well. IZB installed five IHS queue servers on each of the four LPARs and separates the applications as follows:

  - QS Q for applications for Cluster IZW1B1
  - QS R for applications for Cluster IZW1B2
  - QS S for applications for Cluster IZW1B8
  - QS T for applications for Cluster IZW1B9
  - QS U for all other applications

  If IZB now experiences a problem with only one application involved, this environment helps to reduce the impact on other applications and clients.

- Sysplex distribution

  The Sysplex Distributor behind the firewall as first load balancer in the mainframe is not new. What is new is the sysplex distribution function between the queue servers and the application servers. As explained in "Port concept" on page 152, inside a cell all Location Daemons, all Node Agents, and all application servers within a cluster listen on the same port number on all of IZB's WebSphere LPARs. This is a requirement for failure and load balancing with sysplex distribution that is only available with z/OS.

  This feature helps IZB to reduce the impact of peak demands on specific LPARs that are caused by a heavy load in the Internet home banking application, on LPARs SYP2 and SYP3.

- The session database

  The session IDs are stored in DB2 running in data sharing mode. The data sharing group is DB40, and one member is located on each LPAR of the Webplatform. Each application has its own session table and tablespace; if one encounters a problem, the others are not affected. Refer to Chapter 5, "DB2 data sharing" on page 93, for more information about this topic.

Figure 3-5 on page 60 illustrates the system configuration, which provides high availability to IZB's clients, even with load fluctuations.

The flow of requests

1. The request from the employees of the savings banks comes through intranet and firewall to the Sysplex Distributor with the port number of the QM.
2. The Sysplex Distributor spreads the request to one of the LPARs based on WLM criteria.
3. The request is routed to the Queue Manager.
4. Based on the URL, the QM knows for which QS the request is meant and sends it to the appropriate QS.
   - Stateless request: The plug-in of the QS sends the request to the IP address of the Sysplex Distributor with the port number of the application server.

     Sysplex Distributor sends the request based on WLM criteria to the appropriate application server.
   - Stateful request: The request is routed directly to the correct application server through the information in cookie JSESSIONID (target IP address is the address of the application server).

Example 7-13 shows that the requests would normally be load balanced by a simple round robin approach. Coding the statement ClusterAddress can influence the transport.

Example 7-13 Extract from the plug-in

```
. . . . . .
<ServerCluster CloneSeparatorChange="false" LoadBalance="Round Robin" Name="IZW1B1"
PostSizeLimit="10000000".........>
 <ClusterAddress name="IZW1B1">
   <Transport hostname="172.16.146.21" port="11106" protocol="http"/>
   <Transport hostname="172.16.146.21" port="11107" protocol="https"/>
 </ClusterAddress>
   <Server CloneID="BAE3F027AC5DC72B00001C9800000026C2FA968E" ...... Name="W1N002_IZW1B19"
.........................>
  <Transport Hostname="SYP9.ESERVER.IZB" Port="11106" Protocol="http"/>
  <Transport Hostname="SYP9.ESERVER.IZB" Port="11107" Protocol="https">
        <Property Name="keyring"
Value="/m/was/wasconf1/DeploymentManager/etc/plugin-key.kdb"/>
        <Property Name="stashfile"
Value="/m/was/wasconf1/DeploymentManager/etc/plugin-key.sth"/>
    <Property Name="certLabel" Value="selfsigned"/>
   </Transport>
  </Server>
. . . . . other server definitions........

</ServerCluster>
 . . . . . .other server cluster definitions.........

<Route ServerCluster="IZW1B1" UriGroup="default_host_IZW1B1_URIs"
VirtualHostGroup="default_host"/>

. . . . . .
```

Concerning a stateless request for cluster IZW1B1 or a stateful request for server IZW1B19, the following happens:

- Based on the Route statement, the request is routed to the ServerCluster IZW1B1 according the UriGroup and the VirtualHostGroup.

- Stateless requests for IZW1B1 take the way through ClusterAddress to the Transport hostname of the Sysplex Distributor with the port number of the servers in this cluster, according to secure or non-secure requests.
- Stateful requests take the way through Server CloneID, which is part of the cookie, directly to the Transport hostname of the server with the port number of the servers in this cluster, according to secure or non-secure requests.

Note: The statements for the use of the sysplex distribution have to be re-coded manually after a plug-in generation.

7.5 The conclusion for IZB

Varied experiences with the Webplatform host in the past years show that it is challenging to handle such a highly complex environment. Both the quality and quantity of the communication between network and database specialists, security administrators and cryptology specialists, software developers and systems programmers, managers and technicians, long-established staff and external consultants, as well as planners in the operating department and hardware capacity planners, changed in many ways

This new communication environment has produced an enormous effect. The cycles for making use of new system software features such as Web services, Message Broker, Portal Server, or WebSphere Server Foundation are getting shorter, along with the cycles for developing and deploying new application software. IZB pays attention to J2EE conformity, especially in application design. In the future, new software needs to match criteria that are specified by IZB and Customer B.

With WebSphere Application Server on z/OS, it is possible to meet current demands on large companies for quick and dynamic growth; it is almost irrelevant whether that growth is in a horizontal or vertical direction. Of course, planning does not end after adding required hardware resources, but continues to look forward. After allocating the necessary hardware resources, the installation of IZB's new Internet cell W2C001 did not take long, because the extensive conceptual work had already been done.

Much effort was invested at IZB with regard to security, especially with a connection to the World Wide Web or if sensitive client data is involved. Numerous aspects needed to be considered, and the security umbrella covers the network security with its firewalls as well as the deepest levels of J2EE security. Hardware and software encryption, Lightweight Directory Access Protocol (LDAP), Resource Access Control Facility (RACF), and digital certificates were included, and as a result, WebSphere Application Server on z/OS and the z/Series platform provides a highly secure environment for IZB's clients.

In the beginning, system monitoring had been done exclusively with small, self-written utilities. The application PIPO, for example, works like an IP ping and performs a simple HTTP Get request against each server in short intervals. If this request doe not receive a positive response twice in a row, an automated message is displayed on the operator console. Because application design gets more and more complex while high availability demands are increasing, IZB decided to use WebSphere Studio Application Monitor (WSAM) to set up substantial system monitoring and automation.

In IZB's Webplatform host, the unequal size of the four LPARs was strongly influential. Because the Internet home banking application with its heavy load is located only on LPARs SYP2 and SYP3, those LPARs are much better supplied with real storage and MSUs. Sometimes this results in suboptimal distribution of work through the Workload Manager, which might give a larger but busier LPAR work that would be better placed in a smaller but

less busy LPAR. As a result, IZB and Customer B will try to spread all applications on all four LPARs with approximately similar loading and resize the LPARs to the same values for improved workload management.

In 2006, IZB plans to migrate to z/OS 1.6. With this release, the new zSeries Application Assist Processors (zAAP) can be used to deliver a specialized z/OS Java execution environment. The Java workload can be placed there and will not be counted on the other LPARs as a result. In turn this will probably reduce the license costs for other software products located on the non-zAAP LPARs, which are balanced on the basis of MIPS usage. The question will be whether the remaining workload on the non-zAAP LPARs is high enough to reach the criterion for PSLC/SYSPLEX pricing again.

Part 3

The system environment

This part describes the system environment and procedures at IZB.

8

Storage management

This chapter describes the changes that IZB made to its storage management environment, practices, and products in order to operate effectively in a sysplex environment. Starting in 1999 with the two data centers in Nuremberg and Munich, the chapter details the challenges that were overcome on the way to achieving a homogenous storage management solution.

8.1 Storage management in IZB in 1999

As described earlier, in 1999 IZB was running two data centers, one located in Munich and one in Nuremberg. PPRC Mirroring was only used for production data at that time, because IZB determined that moving the tapes to the disaster recovery location hundreds of miles away would only be feasible if an actual disaster struck. That is, an outage of just a storage controller or a tape robot would not be sufficient to declare a disaster and experience all of its consequences.

For an overview of IZB's previous environment, read 1.2, "From 1999 until now" on page 5.

8.2 Description of the situation

This section outlines IZB storage management before the relocation project. For a description of the site arrangement, read 1.2.1, "Two data centers, 160 kilometers apart" on page 5.

DASD environment

In 1999, the DASD environment was a collection of different vendors and mirroring concepts. There were Amdahl and EMC storage controllers without mirroring installed. Some Hitachi storage controllers with PPRC mirroring were installed in the data centers in each site. Each DASD controller had different features and performance characteristics.

Additionally, some DASD was shared between development and production systems, which required a great deal of attention from the storage management team; it was somewhat challenging to run two DFHSM systems that did not know each other on the two systems sharing DASD. Figure 8-1 illustrates the DASD configuration at Munich at that time.

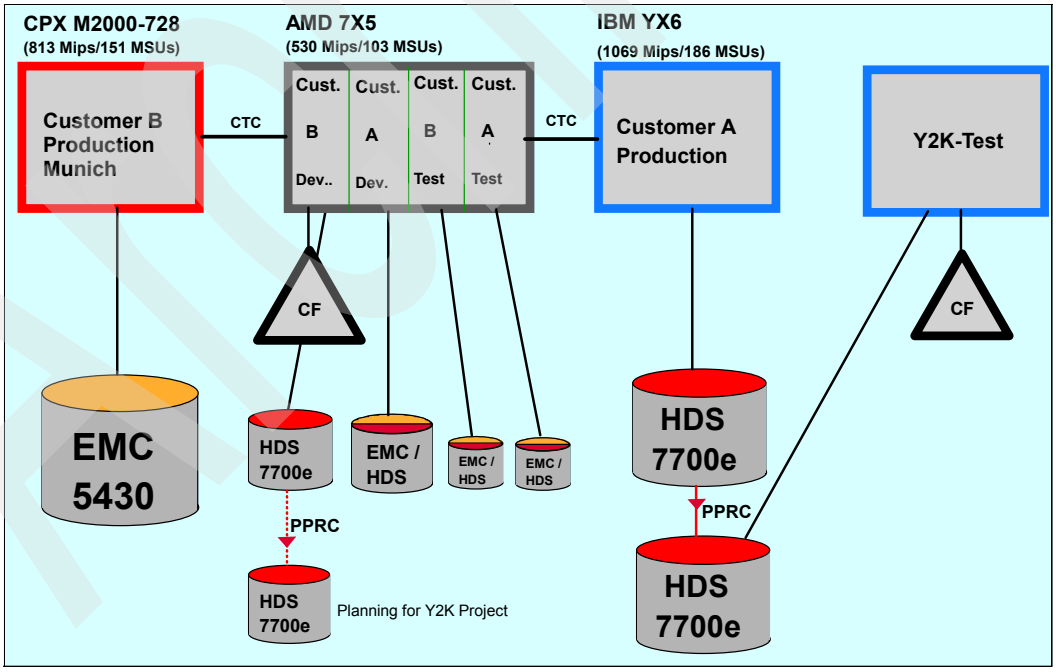


Figure 8-1 IZB's DASD configuration in 1999 at the Munich data center

Figure 8-2 on page 169 illustrates the DASD configuration at Nuremberg at that time.

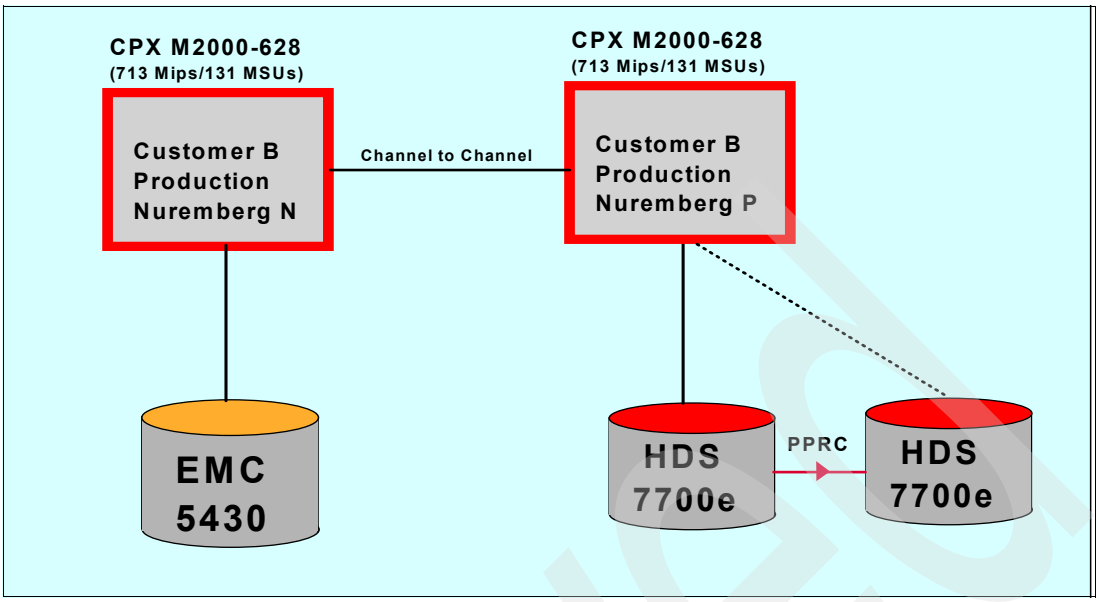


Figure 8-2 IZB's storage configuration in1999 at the Nuremberg data center

Tape environment

The tape environment in 1999 consisted of StorageTek™ silos and 3490E drives. No virtualization or mirroring was established. The tape robots had two different locations in Munich, one in the data center and the other in an interim data room some blocks away (this was required due to the size of the robots). Therefore, the tape environment was not really optimized in those days.

HSM configuration

Most of the systems IZB was running at that point were standalone systems, as was the HSM installation. Actually there were many different HSMs, and each of them had to be maintained and checked every morning. Some of the HSMs were running with DFSMS implemented, while others were not, and this made it even more difficult to maintain the different implementations.

The biggest problem was the shared DASD environment between development and production. If the DASD were accidentally defined in both HSMs, the results were unpredictable. Additionally, the tape resources were dedicated to one system and HSM needed a large number of tapes to be able to end the backup and migration tasks in a timely manner during the night. In some systems, HSM ran nearly 24 hours a day in order to finish the work.

GRS configuration

GRS was not implemented at that time. Every request to DASD that was shared (and IZB did not have a great deal of shared DASD at that time, because Parallel Sysplex was not implemented) was issued with reserve/release technology.

Disaster recovery

The disaster recovery concept in 1999 consisted of a series of tape backups. These backups were regularly sent to a service provider for disaster recovery. Due to the growth of IZB, the costs for disaster recovery became a significant issue. Another problem with this method was that the tapes were not a point-in-time copy. The backups were taken one after another, so catalog entries did not match DASD contents and so on.

Furthermore, disaster recovery tests took a long time because all system tapes had to be recovered first. After that, all image copy tapes had to be restored and the latest log files had to be applied. However, with this solution, data loss was unavoidable because tapes were not written immediately.

Problems

In 1999, IZB was experiencing the following issues:

- The sharing of DASD between production and development systems was often the source of severe problems. A proposal to separate production from development was turned down because the application development group opposed it.
- A large number of tapes was produced. IZB did not use high capacity tapes back then, nor did it use any tape virtualization technology, resulting in a large number of production tapes with very little data on them.
- The infrastructure was somewhat complex, especially in Munich, so installing new hardware was always a challenge.
- Because IZB was growing rapidly in those days, disaster recovery became more and more complex. The interaction between different subsystems made point-in-time copy necessary, but that could not be accomplished with the installed technology.

8.3 The steps between 1999 and today

This section describes the steps undertaken by IZB starting in 1999, covering the most significant changes that occurred during that time frame.

8.3.1 Relocating the Munich data center to Nuremberg

Over approximately 12 months, the Munich data center was relocated to Nuremberg. During this phase, IZB implemented several new technologies and set up a “state of the art” infrastructure. This resulted in a very stable environment and a vastly improved disaster recovery process.

Splitting production and development volumes

At this point, because of the different relocation dates for test, development, and production, the shared DASD between development and production had to be eliminated. Sharing DASD between a system in Nuremberg and one in Munich was not feasible at that time because of the 100-mile distance. Therefore, it became a major task to identify all data sets on the shared volumes and to decide, data set by data set, what to do with each one.

During the planning phase of the relocation, IZB decided it was good practice to define assumptions for all hardware that would be installed (refer to 2.5.3, “DASD” on page 39 for more details about this topic). Those assumptions ultimately paid off, greatly reducing hardware installation and configuration errors.

In addition, the software product XCOM was introduced to the environment to transmit data sets between different systems.

Introducing synchronous mirroring for all DASD

During this time, the old DASD mirroring was removed and synchronous mirroring was introduced for all production data and development LPARs. This was necessary because, in case of a disaster, IZB wanted to give its application development department the ability to continue to develop software programs.

Introducing tape virtualization

Prior to the data center move to Nuremberg, IZB did not use any tape virtualization. Therefore, migration to larger tapes was unfeasible, because only software such as DFHSM would have written these large tapes to full capacity. But none of the homegrown programs could fill a tape with gigabytes of capacity. So in order to reduce the number of tapes to be relocated, the virtualization solution had to be implemented first.

IZB analyzed three different virtualization solutions: IBM VTS (which was very new to the market); STK's virtualization solution; and the Computer Associates software VTAPE. IZB chose to use VTAPE because it met all IZB requirements and was very easy to handle. VTAPE emulates tapes but actually writes on classic z/OS DASD. If the z/OS DASD areas assigned to VTAPE are occupied beyond a defined threshold, VTAPE starts to migrate the DASD data to tape.

In a subproject of the relocation, all tape activity (except DFHSM and Output Management) was migrated from classic tapes to virtual tapes. So IZB had only three tape users: DFHSM; the output management products from Beta Systems; and VTAPE. These products were able to use large capacity tapes. Through introducing tape virtualization and the use of larger tapes, IZB was able to reduce the number of tapes from 41,000 to 13,500.

Introducing tape duplexing

Because some of IZB's batch production work involved tape as a direct output medium, it was essential to be able to duplicate these tapes. VTAPE from Computer Associates was able to meet this requirement.

VTAPE does not simply write two output tapes when migrating data from DASD to tape. Instead, it uses different high level qualifiers for the original data set on tape A versus the backup data set on tape B. Additionally, it maintains a table to relate user data set name to VTAPE data set name. In case of a disaster, VTAPE has been configured to switch the high level qualifiers so that all allocations will be directed to tape B instead of tape A.

As a result of the reduction in the total amount of tapes (from 41.000 to 13.500), it became feasible to install a backup robot. Without virtualization, the space to install the new robots would not have been available.

The introduction of tape duplexing was a critical milestone for IZB because it was then able to have all vital data in both data centers, thereby providing the highest level of readiness for disaster recovery. Figure 8-3 on page 172 illustrates the tape environment after the relocation to Nuremberg.

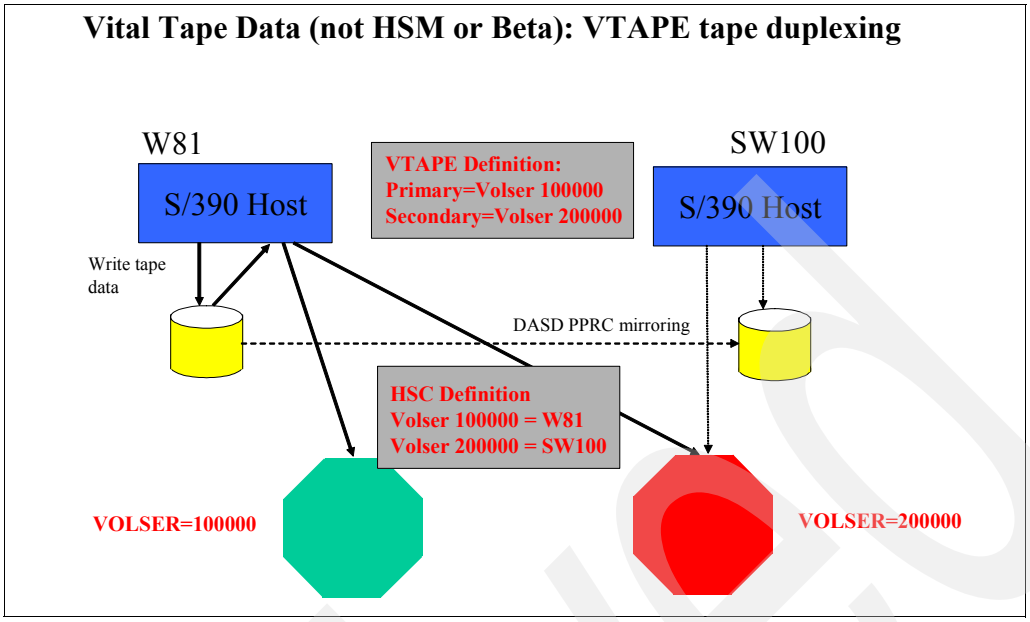


Figure 8-3 The IZB tape environment after relocation to Nuremberg

Change of cabling technology

During this period, the infrastructure in the Nuremberg data center was also revamped and enhanced. Connectivity to remote DASD and tape devices had previously been accomplished using traditional ESCON connections. Now, IZB became one of the first clients in the world to use FICON bridge mode channels to connect CPCs with remote devices.

Cross-site connections were one of the most cost-intensive elements of the design the new data center architecture. As a result of the cabling technology change, however, the number of connections needed between the two sites was dramatically decreased.

During the project, the FICON bridge mode links were tested with the newly installed G6 servers.

8.3.2 Change of DASD hardware

During the relocation of the data center in 1999 and 2000, the largest DASD storage machine installed had a capacity of 4 TB. About 18 machines on the primary site and 17 on the secondary site were installed at this time. Because of the development of larger storage controllers and larger DASD (model 27 instead of model 3), it became possible to consolidate the storage controllers.

Migration to large volumes (model 27) and PAV

Because IZB was close to the upper limit of devices (63998), it was necessary to reduce the number of addresses. This was achieved through two coordinated actions:

- IZB migrated most data from model 3 devices to model 27 devices.

  A model 27 device offers nine times the capacity of a model 3 device. A model 27 only needs two or three Parallel Access Volumes (PAVs) per unit in order to perform well (rather than eight PAVs). In this way, IZB was able to save a large number of addresses.

- IZB developed a firm guideline for its clients which stated that only model 27 devices would be allocated PAV devices; they would not be defined for model 3 devices.

  IZB also ran performance tests with model 3 and model 27 addresses, in which the model 27 addresses were defined with PAV. The results of the testing convinced many IZB clients to migrate to model 27 devices.

Consolidation of storage controllers

Consolidating storage controllers made storage management easier for IZB. Prior to consolidation, storage controllers were dedicated to LPARs or sysplexes rather than being shared among sysplexes. This strategy was adequate for the relocation of the data center, because each syspslex was moved individually from Munich to Nuremberg. So an outage on one storage controller would only impact one sysplex, while all other sysplexes could continue. However, such an approach was inadequate for storage controllers able to run tens and hundreds of terabytes. Therefore, IZB revised this configuration.

Every device address in IZB's data centers is only present once; there are absolutely no duplicate devices. That concept was essential to keeping an accurate view of the whole installation. IZB maintained this approach even when consolidating the storage controllers from 35 (including the secondary controllers) to 8.

As a result of the consolidation, all ESCON cabling could be replaced with FICON, which reduced the complexity tremendously.

Introduction of FICON cascading and fibre channel PPRC links

When consolidating the storage controllers from 35 to 8, IZB also introduced FICON cascading and fibre channel PPRC. With the help of FICON cascading, IZB was able to dramatically reduce the number of cross-site links, because the ISL links between the two FICON switches could be used in both directions (from Site 1 to Site 2 and vice versa). Because of this technology, IZB was able to switch from a single site data center to a multi-site datacenter.

IZB undertook an ESP on FICON cascading, and performed intense testing in this area. The testing verified that the performance impact on the remote DASD was extremely low.

8.3.3 Exploiting sysplex technology

After completing the consolidation, and introducing FICON cascading technology into its data centers, IZB began considering the exploitation of Parallel Sysplex and how the Parallel Sysplex concept could help with storage issues. IZB analyzed and then implemented two features: DFHSM shared recall queue, and enhanced catalog sharing (ECS).

DFHSM shared recall queue

IZB installed two additional systems to handle support activities, and called them SUSY systems. Only the HSMs on these SUSY systems are used to write tapes, but all systems are able to read from the tapes (for Recovery or ML2 recall actions).

Prior to implementing the DFHSM Common Recall Queue (CRQ), recall requests could only be done by the DFHSM host where the request was initiated. This led to problems, because tapes were in use by other HSM systems. By using the common recall queue, every system can recall data sets, even if the request was not initiated in that system; see Figure 8-4 on page 174.

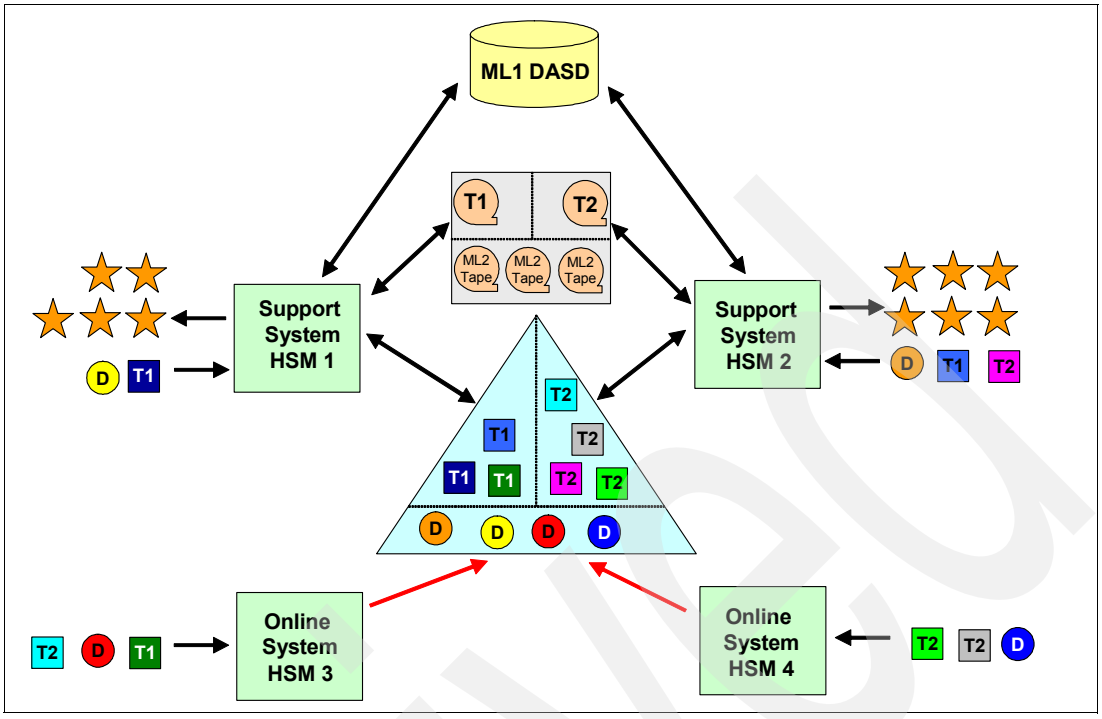


Figure 8-4 Concept of DFHSM Common Recall Queue

Enhanced catalog sharing

As the sysplex became larger and larger, IZB looked for technology to improve the overall response time of its applications. To address this problem, IZB implemented enhanced catalog sharing (ECS), as illustrated in Figure 8-5 on page 175.

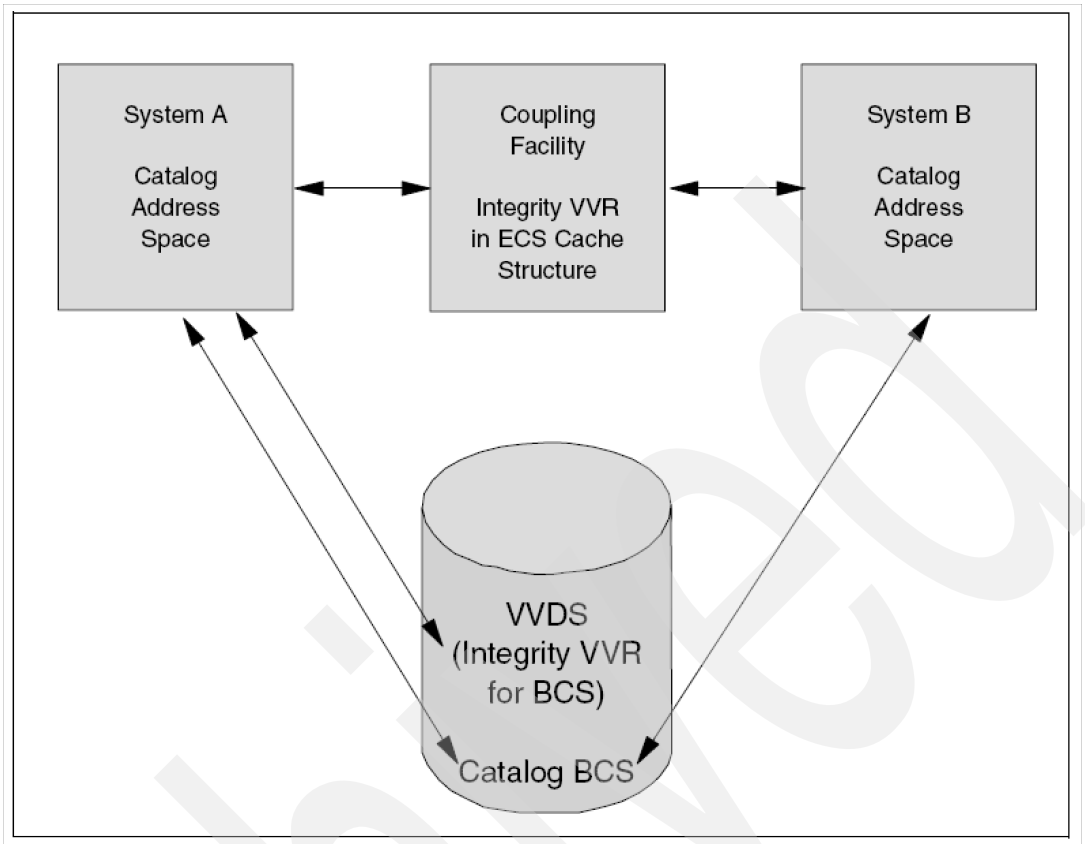


Figure 8-5 VVR Integrity using Enhanced Catalog sharing

After extensive testing and the application of the latest PTF levels available at that time, IZB implemented enhanced catalog sharing in all its sysplexes. For a comprehensive description of ECS, refer to *Enhanced Catalog Sharing and Management*, SC24-5594, which can be accessed at:

http://www.redbooks.ibm.com/abstracts/sg245594.html

8.4 Disaster recovery aspects

Improving IZB's disaster recovery capability involved two major steps in this long-term project:

1. Moving the Munich data center to Nuremberg and setting up a second data center within PPRC distance.
2. Changing from a single site workload mode to a multi-site workload mode. As a result, there are two active sites instead of an active site and a passive site.

8.4.1 Changes in disaster recovery when moving from Munich to Nuremberg

Major changes in disaster recovery were introduced with the full implementation of PPRC and the cross-site cabling infrastructure. The change from primary DASD to secondary DASD was not difficult and could be accomplished with user-written REXX procedures. Therefore, as the first test, all DASD was switched from primaries to secondaries, and an IPL was performed from the secondary DASD.

Also, because tapes were mirrored, switching the tape environment was easy too and could be performed with simple commands. However, in contrast to the PPRC DASD mirroring technology, going back to normal processing mode was not easy with tapes. All tape usage was stopped and the test was to read a data set from a test tape drive that was duplexed. After the tape resources were switched, the tape could be read without any problems.

The problem with this active/passive configuration was that the passive data center was only used during testing. This meant that the infrastructure was only tested during a backup test. And it could be some time between an infrastructure change and the next disaster recovery test. During such a period, no one could be sure that the changes had been done correctly.

Nonetheless, compared to the situation before, the improvements were dramatic. During the very first test of the disaster recovery concept, IZB could verify that the secondary data center was working as expected and could take over the workload in case of a disaster.

8.4.2 Introduction of a multi-site architecture

After FICON cascading was implemented, IZB switched from a single site workload to a multi-site workload. Now the secondary data center was not only a backup site, but a also full production site. This offers the benefit that changes to the infrastructure are directly tested, because production workload is running in this site permanently. This reduces the probability that systems cannot be started because of an incorrect infrastructure setup to near zero.

8.5 Conclusion

In the course of this project IZB gained the following insights regarding storage management:

- Implementing strict configuration techniques can help to avoid errors. For example, at IZB, addresses are used only once; there are no duplicates.
- Implementing new technology has to be done carefully, with enough time for testing. However, new technology with useful features can offer a great opportunity to optimize an environment.
- Concepts that are helpful today may become obsolete in a year or two. For example, the concept of dedicating storage controllers to sysplexes became impractical after large storage controllers became available.
- Naming conventions are essential, as discussed in Chapter 2, “Developing a multi-site data center” on page 17.

9

System

This chapter discusses operating system layout and maintenance, which is the basis for all other work when migrating from single systems to Parallel Sysplex.

9.1 Background

As mentioned, when IZB was founded in 1994, it started with eight LPARs from two customers (see Chapter 1, “Introduction to IZB” on page 3). On each LPAR, a single MVS system was run. A systems programmer team was created, and its first job was to consolidate the MVS systems, so that all system data sets would have the same names. The target was a single MVS layout for both customers to avoid double work, such as maintenance.

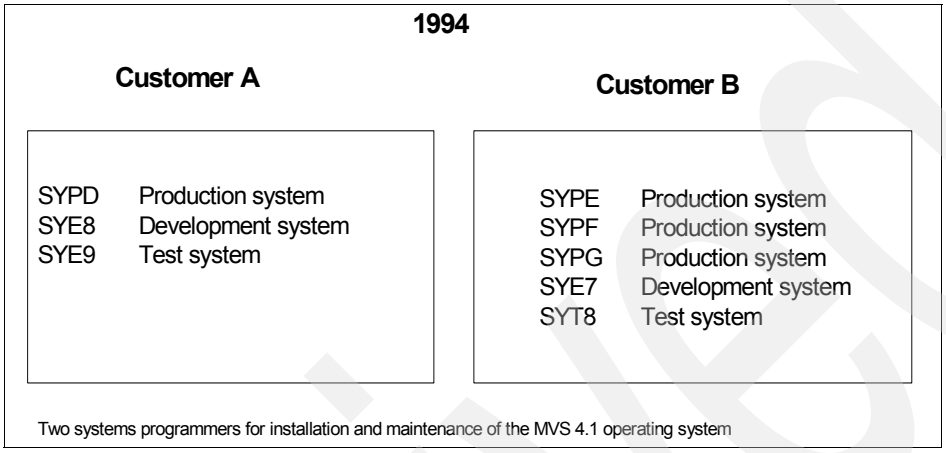


Figure 9-1 IZB’s operating systems in 1994

The team built a skeletal structure of the MVS systems, which was a compromise between two source systems. Two test systems reflected the special features of each customer. Into these test systems, the systems programmers applied the maintenance and built the maintenance basis for the production systems.

This layout was used from 1995 to 2001. With the decision to move to a Parallel Sysplex, the systems needed a new layout (see 9.2.3, “New operating system layout” on page 182) that standardized more then just data set names. The following sections cover the previous system layout, the problems, and IZB’s migration to a new system layout that is the basis for all future systems.

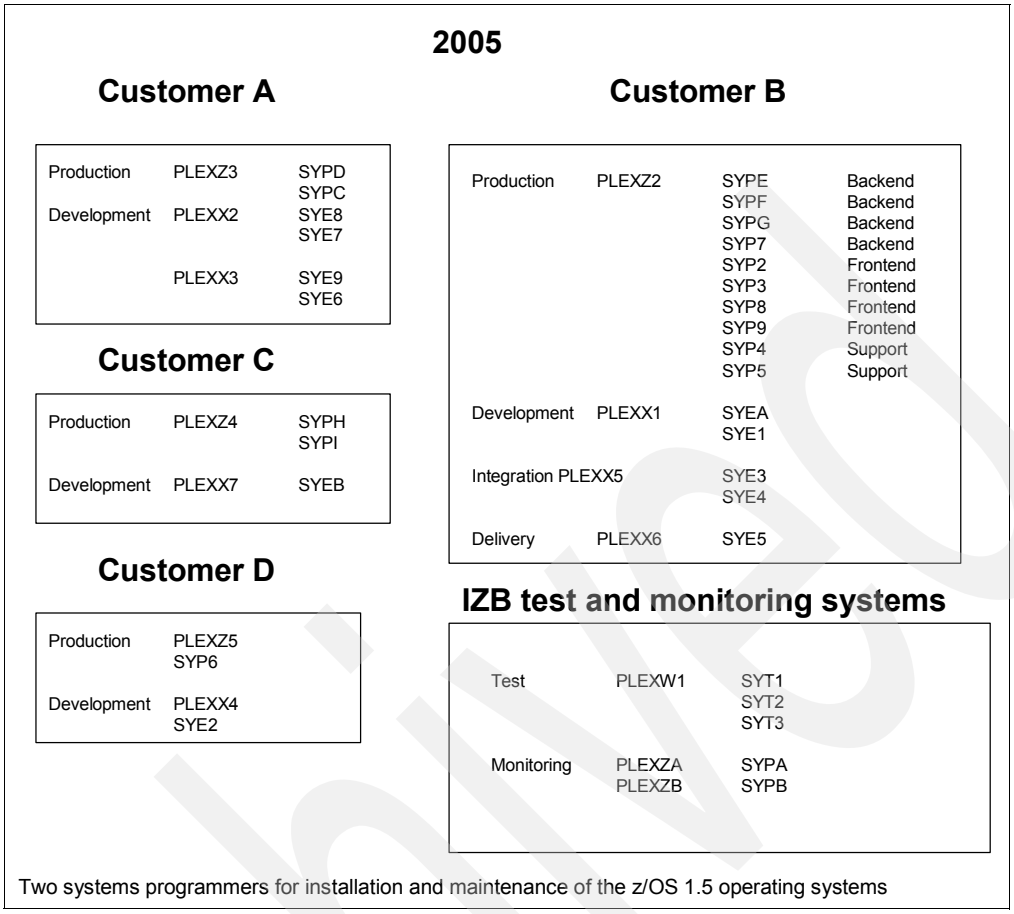


Figure 9-2 IZB's operating systems in 2005

9.2 System layout

Both the old and new system layouts are described in this section.

9.2.1 Old operating system layout

The old system resided on two 3390 Model 3 volumes that contained all the target data sets. These target data sets had the names delivered by IBM. All system volume names began with SY followed by a two-character system suffix and two characters that describe the contents of the volume. Table 9-1 on page 180 lists the system volumes layout from 1994 to 2000.

Table 9-1 System volumes layout from 1994 to 2000

Volume name	Content
SYxxR1 / RA	System residence, first/second Maintenance set Target data sets
SYxxR2 / RB	System residence spill volume, first/second Maintenance set Target data sets
SYxxMC	Master catalog volume Two master catalogs, one for each Maintenance set IODF SYS1.DUMPnn SYS1.PARMLIB SYS1.PROCLIB Secondary RACF database SYS1.MANnn SYS1.TRACE
SYxxC1	Volume for control data sets Primary RACF database SYS1.BRODCAST User LINKLIB User LPALIB
SYxxPn	Page volumes Page data sets

For each system, two sets of system residence volumes existed, one active set and one set for maintenance. Because there were different requirements from IZB's clients, and thus different user modifications, the systems programmer team had to find a way to separate these modifications. In a simple case, the modification only changed a load module, so they built this load module and brought it into a customer-specific load library. Some other modifications such as user exits had to be recoded with requesting the system ID to address the client requests.

The increasing size of the IBM target data sets was a problem. Because one 3390 Model 3 volume could no longer contain all the target data sets, a spill volume was needed. The data sets on the first sysres could be cataloged with six stars (******), but all data sets on the spill volume had to be cataloged with its volser. So two master catalogs were needed in case of a sysres switch. All catalog changes had to be defined on both master catalogs.

To install new maintenance on a production system, several steps were necessary:

- Bring target volumes online
- Initialize target volumes
- Run a full volume copy batch job
- Copy customer-specific load libraries
- Check new master catalog to see if all changes are included

Much more work was necessary to deliver a new operating system:

- Fill target volumes with new SYS1 data sets
- Change SYS1.PROCLIB members
- Change SYS1.PARMLIB members
- Change all IZB and customer proclibs to reflect new data set names
- Change REXX/CLIST procedures to reflect new data set names

- Change master catalog entries
- Change TSO logon procedures if ISPF data set names were changed

To prepare a new sysres for delivery, a systems programmer had to invest 5 to 8 hours of effort. Some of the changes had to be done a few minutes before system shutdown. The active parmlib and the active proclib had to be changed, so ISB had the risk to kill its running system. ISV software load libraries were put on volumes that were shared over all systems without GRS. If changes were necessary, they impacted all systems.

9.2.2 Reasons for a new layout

The decision by IZB and its clients to install Parallel Sysplex with new LPARs was only one argument for redesigning the operating system. With the old operating system layout, the systems programmer team had the following problems.

- Too much manual work for maintenance:
  - Copy load modules for different systems
  - System parameters are different from system to system
  - Some parameters had to be changed before IPL
  - Changes in master catalog had to be done twice because each sysres owned its own master catalog
- Extensive effort was needed to prepare a new operating system release:
  - Last minute changes of PARMLIB and PROCLIB
  - Changes in client proclibs
  - Changes in client batch jobs and tasks
  - Changes in master catalog
  - Wasted time to check readjusted copy jobs
  - Change of the same parameters in every parmlib/proclib
- Operational risks:
  - Reserve/enqueue situations when copying system data sets
  - IPL problems because error in parmlib/proclib
  - Long downtime of production system if there was a parmlib error
  - Standby system sometimes not executable
- Conclusions:
  - Some modifications were without SMP/E information
  - Only one SMP/E target zone for eight systems
  - No documentation about the maintenance level of a single system

Requests to the new operating system layout

- Ready for Parallel Sysplex
- All systems must be equal
- No changes on running systems
- SMP/E must reflect all system modifications for each system
- All changes must be documented
- Easy to deliver maintenance without manual work
- No further changes in client libraries
- Minimize outage time
- Easy build of new LPARs
- One master catalog for one sysplex
- No write access from a test system to a production system; separation between test, development and production

The systems programmer team's "wish list" comprised the following tasks:

- Apply PTF
- Press a button
- IPL
- Work done

9.2.3 New operating system layout

The new operating system layout was drafted in 2000. After initial tryouts on IZB's test systems, the layout was rolled out to all other systems, parallel with the new operating system OS/390 V 2.8. For this implementation, the systems programmer team needed nine months.

One fundamental requirement was strict naming conventions for data sets and system volumes. The systems programmers renamed nearly all system data sets, in order to become independent of changes by IBM. There would be no more version numbers in data set names; all data sets would have the high level qualifier SYS1.

The volume assignment was designed for performance and not for utilization, so one 3390 Model 3 volume was needed just for the primary JES2 checkpoint data set. All volumes in a Parallel Sysplex are shared. All systems of a Parallel Sysplex are running from the same sysres volumes (with the exception of rolling out a new maintenance level) with a shared master catalog. Table 9-2 lists the volume assignments.

Naming conventions

Table 9-2 Types and naming conventions since 2000

Type	Convention

System names for new systems
SYxn
	SY= SY for all new systems
	x = P for production systems
	x = E for Entwicklung ("Development")
	T = Test
	n = Counter, 1-9, A-Z
	Example:
	`SYP7 = production system number 7`
	`SYE3 = developments system number 3`

Sysplex names
PLEXxn
	PLEX = identifier for a sysplex
	x = Z for production
	x = X for development
	x = W for Test
	Example:
	`PLEXZ2 = production sysplex number 2`
	`PLEXX1 = development sysplex number 1`

Sysplex volumes SSxxyn	SS = system volume xx = sysplex suffix y = content n = counter Example: `SSZ2C1 = catalog volume PLEXZ2`

Contents of system volumes

Table 9-2 lists the contents of the system volumes.

Table 9-3 Volume naming conventions since 2000

Volume name	Content
SSxxR1, SSxxRA, SSxxRX	Sysres volume, three sets all target data sets need for IPL and SYS1.PARMLIB SYS1.PROCLIB
SSxxR2, SSxxRB, SSxxRY	Sysres Spill volume target data sets
SSxxR3, SSxxRC, SSxxRZ	Sysres Spill volume target data sets
SSxxH1, SSxxHA, SSxxHX	Sysres HFS Volume USS Root HFS
SSxxC1	Master catalog volume Master catalog IODF BRODCAST Page data sets for standby system, Spool data sets and checkpoint data sets for standby JES2
SSxxP1 - SSxxPn	Page volumes numbers depend on system size
SSxxO1, SSxxO2, SSxxO3	Volumes for operational system data sets SYS1.MANxx RACF database TSO/E procedures Procedure libraries
SSxxU2	USS work volume USS work files like /temp /var /etc
SSxxX1, SSxxX2, SSxxX3	Couple data set volumes contains sysplex couple data sets each volume on a different DASD controller
SSxxK1	JES2 checkpoint volume
SSxxK2	JES2 secondary checkpoint volume
SSxxSA - SSxxSn	Stand alone dump volume numbers depend on system size

What is new in this operating system layout

In this layout, all SYS1 target data sets that are maintained by SMP/E are on sysres volumes, as is the USS root HFS. All system data sets on these volumes were cataloged with system variables:

- ******: IPL volume
- &sysr2: sysres spill volume 2
- &sysr3: sysres spill volume 3

These variables were set in an IEASYMxx member in SYS1.PARMLIB; see Example 9-1 on page 184.

Example 9-1 IEASYMxx member

```
SYSDEF
SYMDEF(&SYSR2='&SYSR1(1:5).B')
SYMDEF(&SYSR3=&SYSR1(1:5).C')
```

The systems programmer team needed three IEASYMxx members to reflect the three sysres sets. Because the master catalog entries are variable, no second master catalog is needed and so the load address is identical on every IPL.

Each sysres set has its own SMP/E target zone to reflect all changes. For each modification, there is an SMP/E usermod; modules from subsystems such as CICS, which must be linked or copied in LPALIB or LINKLIB, were placed in a separate load library. With an SMP/E usermod, they were picked up and linked/copied into the right system library.

Standby operating system

All SYS1 data sets that are necessary to IPL a basic operating system are on the first sysres volume. The master catalog volume contains all data sets to IPL a standby operating system:

- Standby page data sets
- Standby RACF database
- Standby logrec data set
- Standby JES2 spool data set
- Standby JES2 checkpoint data set

Because SYS1.PARMLIB and SYS1.PROCLIB are on the first sysres volume and contain procedures to start a standby JES2 and a standby VTAM, IZB was able to start a standby operating system with only two volumes in every defined LPAR. These standby operating systems need no further maintenance.

In case of a damaged master catalog, the systems programmer is able to start a standby operating system from every available master catalog on another system that could be varied online.

Another important step IZB undertook in 2001 was to standardize its mainframe operating systems. A team of specialists from several areas determined which software components IZB needed in a new mainframe image. The team created a list of IBM and ISV products for a base system, and determined standard parameters for the operating system and naming conventions for started tasks, batch jobs, and so on. The result was a firm guideline for building new system images. The existing operating systems were adjusted.

9.2.4 SYS1.PARMLIB

SYS1.PARMLIB was built by the systems programmer team with the traditional rules:

- Suffix = 00: Used by all operating systems
- Suffix = Plexclone: Used by all systems of one Parallel Sysplex
- Suffix = Systemclone: Used by a single system

An innovation was the placement of SYS1.PARMLIB on the first sysres volume, so each SYS1.PARMLIB on a sysres set can reflect changes that come with a new operating system release or maintenance update. If a new release of z/OS brings new LPA data sets, new linklist data sets, and a few other parmlib changes, the systems programmer can update the new parmlib without changing the running systems.

The starting point is one source SYS1.PARMLIB (for each z/OS release) that resides on the maintenance operating system (see 9.2.5, “System cloning” on page 186). This source

parmlib contains all members of all Parallel Sysplexes and thus all members of each operating system. By cloning an operating system, SYS1.PARMLIB was copied to a target LPAR. SYS1.PARMLIB contains only members that are changed by IZB. Unchanged members remain in SYS1.IBM.PARMLIB, which is concatenated ahead of SYS1.PARMLIB.

All system data sets that must dedicated to one system, such as page data sets, SYS1.MANxx, and so on, have a sysname qualifier in their data set name. In SYS1.PARMLIB members, the system symbolics are used for consistency, for example:

```
PAGE=SYS1.PAGE.COMMON.&SYSNAME,
```

Data sets that are used in SYS1.PARMLIB members and in product procedures are referenced by system variables. So only system variables need to be updated if a data set name is changed, rather than several parmlib members and procedures.

IZB uses automation and monitoring tools from ISV Boole & Babbage. Some data sets must be defined in SYS1.PARMLIB, startup procedures for automation tasks, CICs tasks, DB/2 tasks, and IMS tasks:

- SYMDEF(&BBPUTLV='PT0501B1')

  Definition in IEASYM member; the symbol &BBPUTLV contains a BOOLE Putlevel for system automation and monitoring software.
- IZS#.BOOLE.&BBPUTLV..BBLINK

  Definition in LNKLST member.
- //BBACTDEF DD DISP=SHR,DSN=IZS.BOOLE.&BBPUTLV..BBACTDEF

  Definition in the procedure JCL of a started task.

With these definitions, IZB staff who supervise ISV software have a easy way to apply maintenance on the systems with an IPL, as only one change in the IEASYM member is necessary.

Normally all operating systems in a Parallel Sysplex run from the same sysres set. If a new maintenance level is rolled out, it is possible that one or more operating systems would run from different sysres sets and from different SYS1.PARMLIBs. As a result, the systems programmer must update three SYS1.PARMLIBs: the two that are in use, and the source parmlib in the maintenance operating system. If the third sysres set is also in use, the programmer must make any updates four times.

Because of this complexity, the systems programmer team wrote a REXX/ISPF application for all SYS1.PARMLIB updates called PARMLIB TOOL. This parmlib tool runs on every z/OS LPAR and performs the following functions:

- Show only the parmlib members from the actual Parallel Sysplex.
- Browse or edit parmlib members.
- Require a short documentation when a member is updated.
- Update a member of the active parmlib.
- Copy this update to all SYS1.PARMLIBs in the Parallel Sysplex, if they run on the same operating system release.
- Copy the update to source SYS1.PARMLIB on the maintenance z/OS LPAR.
- Show documentation of all parmlib changes.
- Create a backup member of each updated member.

Some checks are also implemented. So the systems programmer can only update a member that has not been updated in source parmlib before. At times, especially in a new operating system release, a systems programmer updates some members in source parmlib. This is no problem because one source parmlib exists for every z/OS release. The parmlib tool only

updates members in SYS1.PARMLIBs of the same release level. Figure 9-3 illustrates the parmlib management flow in a Parallel Sysplex.

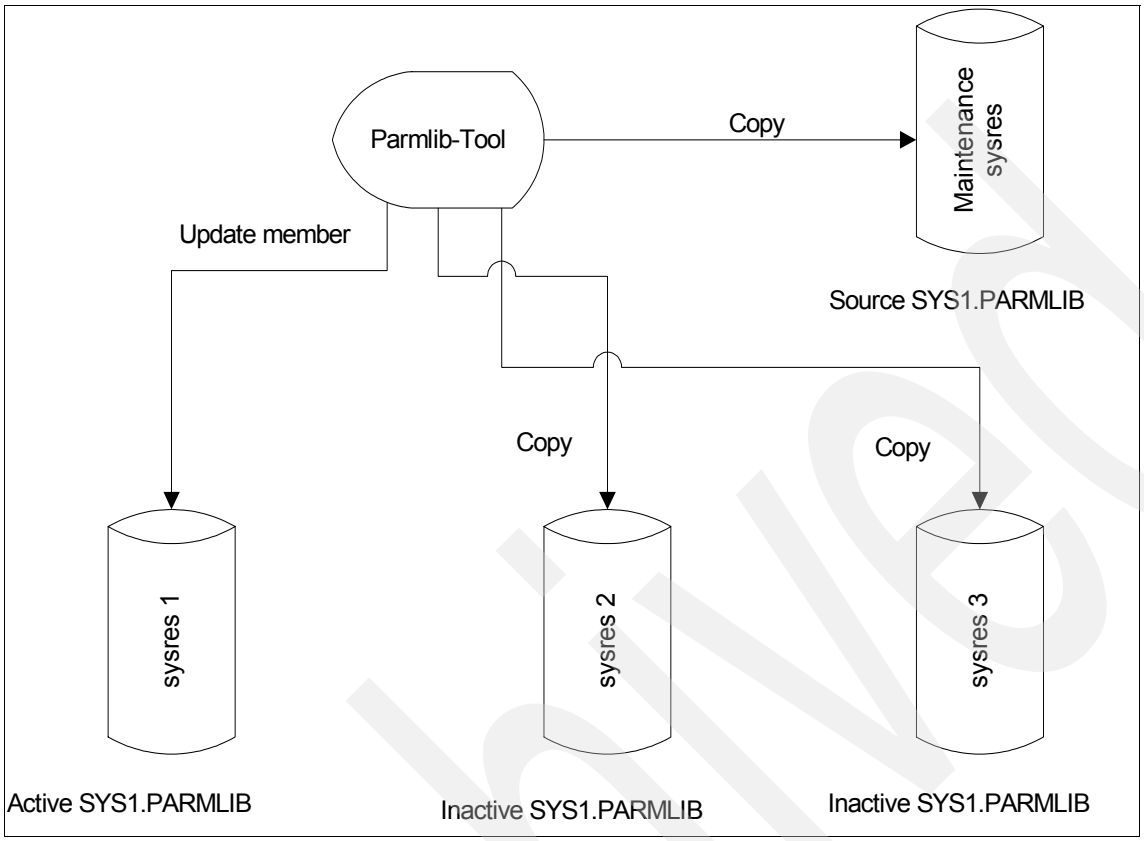


Figure 9-3 Parmlib management in a Parallel Sysplex

Because systems programmers can only update the SYS1.PARMLIB with the parmlib tool, IZB implemented a security routine so that only few persons in the systems programmer team are able to change members with a 00 Suffix and the PROGxx members which contain the APF authorized libraries.

9.2.5 System cloning

After the systems programer team established the new operating system layout, it worked for a short time with old-fashioned batch jobs to bring a new maintenance level from the maintenance system to the production systems. To meet all demands, they wrote a REXX/ISPF application to clone the z/OS systems. The following section addresses maintenance strategy; refer to Figure 9-4 on page 187.

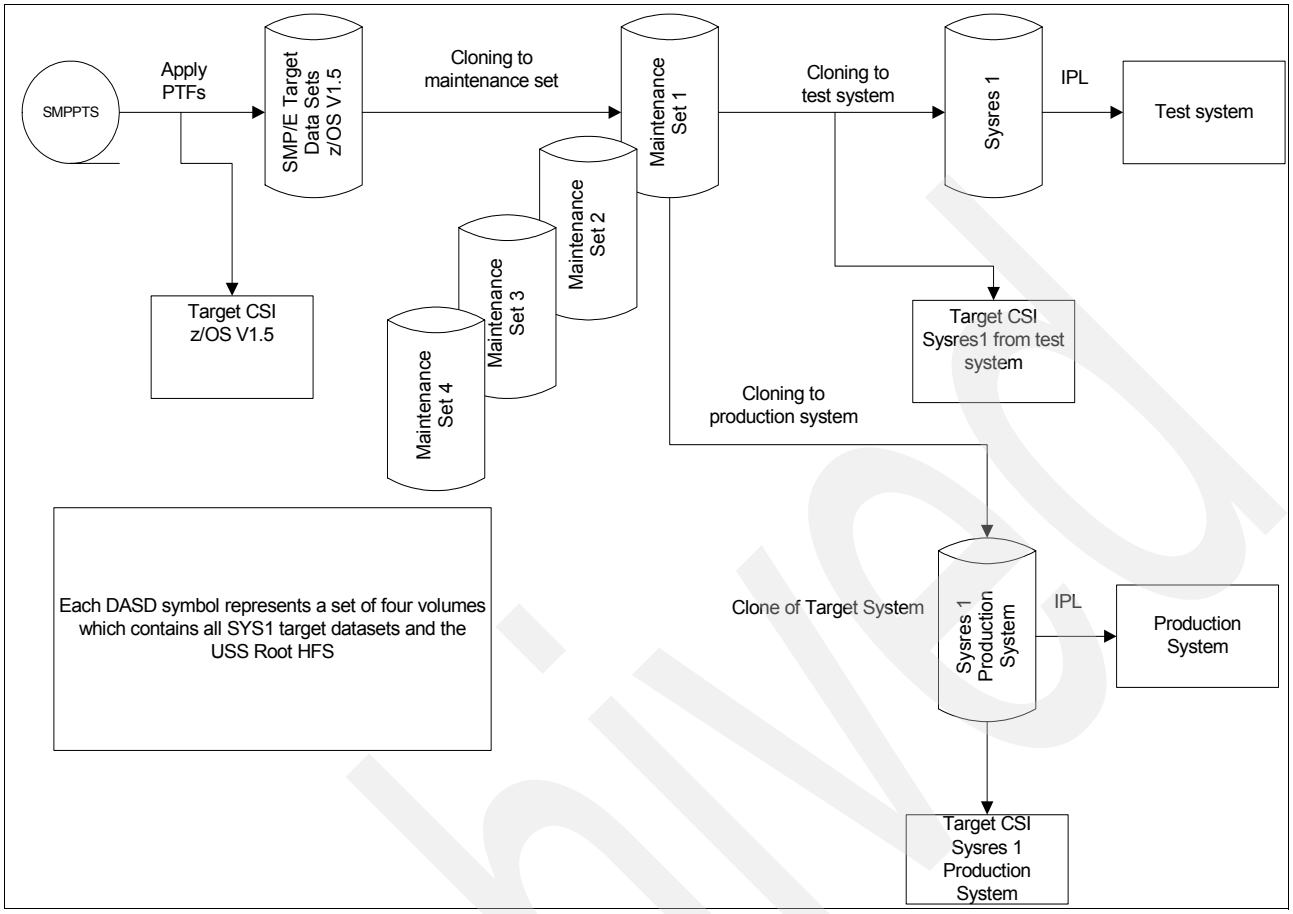


Figure 9-4 Operating system maintenance

All z/OS maintenance is applied using SMP/E into a set of target data sets that IZB calls the maintenance system. After SMP/E applies one or more PTFs, the first cloning (copy) goes to one of four maintenance sets, so that maintenance level is in a parked position. To test this new maintenance level, it is cloned to one of three sysres sets of IZB's Test sysplex.

In the meantime, new PTFs can be applied into maintenance system and a new maintenance set can created. If the test are successful, the tested maintenance level can be cloned to a development/production system. Each clone creates a target CSI to reflect all PTFs on the sysres. Modifications such as userexits are only received in the maintenance system, not applied. The apply occurs only on the target system.

But the clone tool does much more. Remember the dream of the system programmer team (Apply PTF, push a button, IPL, work done). So the systems programmers built the following functions into the clone tool, which only runs in IZB's test sysplex:

- Send a job to the target system to determine which sysres set is not in use.
- Bring all SMP/E volumes and maintenance volumes online.
- Create a job stream for copy jobs.
- Perform a full volume copy of maintenance volumes.
- Copy USS Root HFS and convert to zFS.
- Copy SYS1.PARMLIB (source parmlib) and SYS1.PROCLIB from the maintenance system.
- Apply all usermods for the target system.
- Create SMP/E target CSI.
- Create an IEASYMxx member to set the correct &sysr2 and &sysr3 variables.

- Take the job output out of JES2 and archive it on disk.
- If the clone runs on a maintenance set, document which new PTFs were applied

The clone tool internally has ISPF tables with information about system names, user modifications, NJE nodes, sysplex names, and so on. The clone tool is now the information center for systems programmers. It contains all the information about system maintenance, IPL time, which usermods are in which sysplex, and which system is running from which sysres. The next few graphics show screen shots from the clone tool, with translations following.

```
_____________________________________________________________________
Achtung: die Anwendung steht nur in der Wartungsumgebung zur Verfügung!
_____________________________________________________________________

                    Tool zum Clonen von Systemen
               A d m i n i s t r a t i o n m e n u e
               ____________________________________

                        MAIN OPTION MENUE

                          System=IZT1

                    01 Clone-Prozesse initieren
                    02 Clone-Prozesse anzeigen
                    03 Profile definieren
                    04 USERMOD-Administration
                    05 USERMOD/System-Beziehungen
                    06 IPL-Information (alle Systeme)
                    07 Volumes für Targetzones
                    08 Systeminformationen
                    09 PARMLIB/SYSCLONE-VARIABLE

                    bitte auswählen ==>
```

Figure 9-5 Clone tool's main menu

Clone tool main menu:

01	Initiate a new clone
02	Browse clone jobs
03	Define profiles (a set of job skeletons)
04	Definition file for all user modifications
05	Assignment user modifications to sysplex
06	IPL information about all systems
07	Definition volumes for SMP/E target zones
08	Global system information
09	Definition of variables

Figure 9-6 on page 189 shows different maintenance sets and information about the date and time when the set was built. Sets with an open state can used for cloning.

```
                    Tool zum Clonen von Systemen        Row 1 to 11 of 73
              C o n t e x t - A d m i n i s t r a t o r
                       M a s t e r - M o d e

 ____________________________________________________________________________
| Ein Context dient der verbalen Beschreibung eines definierten Schnitt- |
| punktes innerhalb der OS/390-ZOS-Systemwartung. Er sollte kurz und     |
| verständlich sein und ist Ankerpunkt zu zugeordneten Cloneprozessen.   |
 ____________________________________________________________________________

Auswahl: I/W = ins D = del S = sel Z = zoom IPL-Info C = Close B = Browse
         U = update Context
A  Rel  Context                            Userid   Datum   Zeit Set State
  16-01 zOS 1.6 Basisinstallation          xxxxxxx 10.10.05 16:23 3  open
  15-19 SMSVSAM,LE PTFs,IMS-LNKLST         xxxxxxx 27.09.05 11:33 1  open
  15-16 VIPA PTFs restored                 yyyyyyy 09.08.05 14:41 2  open
  14-22 z/OS 1.4 UCB Overlay PTFs          yyyyyyy 17.03.05 10:15 4  open
  15-15 Websphere, SMSVSAM                 xxxxxxx 21.07.05 15:20 1 closed
  15-14 1.5 USS-Websphere PTF              xxxxxxx 30.06.05 10:11 3 closed
  15-13 Node. Desc SMSVSAM CICS            xxxxxxx 24.06.05 12:58 2 closed
  15-12 VSAM RLS, RMF + LE PTFs            xxxxxxx 09.05.05 18:19 1 closed
  15-11 VSAM RLS und RMF PTF's             xxxxxxx 03.05.05 12:16 1 closed
  15-10 z/OS 1.5 SMP/E PTF                 xxxxxxx 12.04.05 14:05 2 closed
  15-09 z/OS 1.5 UCB Overlay PTFs          xxxxxxx 17.03.05 13:34 3 closed
```

Figure 9-6 Maintenance sets showing date and time of build

The columns show:

Rel	z/OS Release and IZB internal maintenance number
	16-01 = z/OS V1.6 first maintenance (basic installation)
	15-19 = z/OS V1.5 maintenance number 19
Context	Brief information about the main changes
User ID	User ID of submitter
Datum	Date of cloning
Zeit	Time of cloning
Set	Maintenance set that was used
State	State of the set
	Open = Can be used for distribution
	Closed = Cannot be used for distribution, old maintenance
	Only used for documentation
	Busy = Clone job is actually running

Figure 9-7 on page 190 shows the information after cloning a maintenance set to the Parallel Sysplex.

```
. . . . . .
                    Tool zum Clonen von Systemen         Row 1 to 15 of 15
                C l o n e p r o z e s s e / S t a t u s
                        B r o w s e - M o d e
 ____________________________________________________________________
| Mit dieser Funktion können Sie Clone-Prozesse initieren und die Ergebnisse  |
| anzeigen lassen. Bitte beachten Sie die zulässigen Eingabemöglichkeiten.    |
 ____________________________________________________________________

Context: SMSVSAM,LE PTFs,IMS-LNKLST         Rel: 15  Set: 01  Status: open
Auswahl: B = Ergebnis anzeigen Z = zoom IPL-Info X exp.IPL
Enter=REFRESH           Wartungsumgebung von SS15R1

                   ________________________________
___________________|  Initiation- |  neue  Load  Load |_____________________
A  System Node  JC |  Date/Time   | SYSRES addr  parm | RC  Status Cloneprozess
____________________________________________________________________________

   PLEXX1 JESE   P 07.10.2005 08:42 SSX1RA 5F8D 5F72X1 0004 ended normal
   SYE5   JEE5   P 26.09.2005 09:03 SSE5RA 75D1 770FE5 0004 ended normal
   PLEXX5 JEX5   P 22.09.2005 14:11 SSX5RX 7554 7547X5 0004 ended normal
   PLEXZ2 JESP   O
   PLEXX3 JEBE   P 19.09.2005 15:54 SSX3RA 7953 7933X3 0004 ended normal
   PLEXX2 JEBB   P 28.09.2005 14:45 SSX2RX 9187 9162X2 0004 ended normal
   PLEXZ3 JEBA   P 21.09.2005 08:42 SSZ3R1 3BA2 3BDDZ3 0004 ended normal
   PLEXW1 JEW1   A 28.09.2005 13:56 SSW1R1 3E96 3E8BW1 0004 ended normal
   SYEB   JEEB   G 15.09.2005 13:29 SSHERA 6831 68DAHE 0004 ended normal
   SYPI   JEPI   G 20.09.2005 06:58 SSHVR1 2900 2816HV 0004 ended normal
   SYPH   JEPH   G
   SYP6   JEP6   P
   SYE2   JEE2   P 12.09.2005 10:33 SSE2RA 7E2E 7E27E2 0004 ended normal
   SYPA   JEPA   P 07.10.2005 10:41 SSPARA 3EE6 3ED8PA 0004 ended normal
```

Figure 9-7 Information after cloning maintenance set to the Parallel Sysplex

Information in the rows includes:

System	Sysplex or system name
Node	JES2 node name
Date/Time	Date and time of cloning
Sysres	New Sysres name
Loadadr	New Load address of target system
Loadparm	Loadparm, information only
RC	Highest return code of clone job
Status	State of clone job, ended normal, flushed or running

Figure 9-8 on page 191 shows the usermod administration.

```
COMMAND ===>              B r o w s e - M o d e            Row 1 to 32 of 132
(update,normal)       Tool zum definieren von Usermods
                U S E R M O D - A D M I N I S T R A T I O N

    ____________________________________________________________________
   | Durch Eingabe von UPDATE im Command-Field werden die Zuordungsfelder |
   | freigegeben. Mit Eingabe von NORMAL werden diese wieder verriegelt.  |
    ____________________________________________________________________

  Usermods+-----------------+SYSTEMS--->            weitere Systeme mit PF11
     |                       ____________________________________________
Name |  Description         | X1 |SYE5| X5 | Z2 | X3 | X2 | Z3 | W1 |SYEB|SYPI|
_____V_______________________|____|____|____|____|____|____|____|____|____|____|
#00A01A ASSEMBLY mit IEV90  | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00A02A ASSEMBLER DEFAULT O | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#E2A02A ASSEMBLER DEFAULT O |    |    |    |    |    |    |    |    |    |    |
#P6A02A ASSEMBLER DEFAULT O |    |    |    |    |    |    |    |    |    |    |
#00D01A bypass RACF-Checkin | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00D02A bypass enqueuing on | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00D03A ZAP in Modul ADRPAT | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00D04A DADSM PREALLOCATION | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00D05A ZAP for ISAM LOAD w | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00D06A 3480 Message Displa | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00D07A DFDSS OPTIONS INSTA | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
#00F01A APL2 Inst-Options f | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00F02A Nickname FMT        | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00F03A Nickname File ISI   | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00F05A Bookmanager Options | X  | X  | X  | X  | X  | X  | X  | X  |    |    |
#00F07A CICS SVC (IZB: SVC  | X  | X  | X  | X  | X  | X  | X  | X  | X  | X  |
```

Figure 9-8 Usermod administration

The columns on the left side list the name of the usermod and a short description. An `X` under the sysplex or system suffix means that this usermod was applied to this system. To use this tool, the systems programmer simply defines a new sysmod and types an `X` in the appropriate column. The clone job picks up this information and applies the usermod.

Because IZB has different clients with different requirements, the systems programmers built different user exits, such as IEFUSI, for each customer. It is much easier to maintain a few exits than one big complex that contains all requirements for all clients.

9.2.6 Conclusion

During the period 2000 to 2001, IZB mainframe specialists had many challenges to deal with, including:

- Standardizing the operating system.
- Building a new operating system layout.
- Developing clone and parmlib tools (initially, the tools did not have all the functions described).
- Making adjustments in other areas (job scheduling, networking, output management, operations, and so on).

Implementing all of these innovations was sometimes difficult, for both IZB and its clients. Many changed windows and some unexpected interruptions resulted from these changes in IZB's mainframe environment.

On the other hand, the benefits associated with the changes were realized starting in 2002, when IZB's mainframe specialists built 17 new z/OS images. After all hardware requirements became available (DASD space, IODF, and power on reset (POR)), in order to activate a new LPAR, the systems programmer team needed two days to establish a new z/OS standard image. After three to four weeks, the ISV software was installed and the system were tested and ready for client production.

If a client with a single operating system decides to implement Parallel Sysplex, it does not represent a problem since all single images are designed for Parallel Sysplex. Because all systems now run with nearly identical parameters, the rate of problems has decreased.

With standardized operating systems, no additional staff is needed to install and maintain a higher number of LPARs. For instance, in 1994 IZB needed two systems programmers to maintain eight operating systems. In 2005, two systems programmer were able to install and maintain 31 LPARs.

Benefits

- Very cost effective installation and maintenance of the operating system.
- Higher stability of the operating system.
- Easier management of the environment.
- Transparent documentation for all system changes.
- Minimized outage times.
- Eliminate an IPL as a result of no last-minute changes.
- Better usage of staff and hardware and software resources.

Lessons learned

- Standardizing the operating systems must be done as soon as possible.
- Clear naming conventions are essential.
- Implementation of Parallel Sysplex is easier than expected, but a good test environment is required.
- Invest time and money to develop your own tools for repetitive jobs.
- Finally, remember the systems programmer dream: Apply PTF, push a button, IPL, work done!

At IZB, the dream has been realized. Systems programmers must push two or three buttons, but the boring work like cloning systems has been greatly simplified, creating time for more challenging tasks.

9.3 Managing Parallel Sysplex

This section describes how IZB developed its Parallel Sysplex environment over time.

Note: A description of how to implement a Parallel Sysplex is beyond the scope of this document; for complete information about that topic refer to *z/OS V1R7.0 MVS Setting Up a Sysplex*, SA22-7625, which is also orderable on the Web:

http://ehone.ibm.com/public/applications/publications/cgibin/pbi.cgi?SSN=06GRC0015550262588&FNC=PBL&PBL=SA22-7625-12PBCEEBO200002265&TRL=TXTSRH

The following lists milestones of IZB's Parallel Sysplex environments:

- 1996: First experiences with basic sysplex on test systems
- 1997: All systems in monoplex mode because of WLM
- 2000: Parallel sysplex PLEXZ1 of three productions systems because of IBM pricing (systems SYPF,SYPG and SYPE)
- 2000: Building of sysplex PLEXX1 (development)
- 2001: Split PLEXZ1 in PLEXZ1 and PLEXZ2
- 2001: Rebuilding of PLEXZ2, SYPE and new system SYP2
- 2002: New system SYP3 in PLEXZ2
- 2002: New system SYP7 in PLEXZ2 for ADABAS clustering
- 2002: New systems SYP4 and SYP5 in PLEXZ2 (supporter systems)
- 2002: Build integration sysplex PLEXX5 (systems SYE3 and SYE4)
- 2002: Build system SYE5 (quality assurance)
- 2003: Build systems SYPA and SYPB for PPRC management
- 2003: Implement development sysplex PLEXX3 (SYE9, SYE6)
- 2003: Implement development sysplex PLEXX4 (SYE8, SYE7)
- 2004: Build PLEXZ3 with production systems SYPD and SYPC
- 2004: New systems SYP7 and SYP8 in PLEXZ2
- 2004: Merging systems SYPF and SYPG in PLEXZ2

The first sysplex was built on an IZB test system in order to gain experience with the new technology; the basic sysplex was installed in 1996. In 1997, the test system was migrated to a Parallel Sysplex. At first, only the operations team used functions such as console sharing or operlog. In this environment the systems programmers gained experience with GRS and the Coupling Facilities, and tested failure scenarios. With the introduction of WLM and the withdrawal of IPS and ICS, all systems were migrated into a basic sysplex in monoplex mode, because the WLM definitions reside in a coupling data set.

The first production sysplex (PLEXZ1) was built in 2000 and run as a Parallel Sysplex. At this time, GRS was running in ring mode. In this mode one system was the master of GRS and had to be the first one IPLed. The other systems joined the GRS ring at IPL time.

While migrating to OS/390 V2.8, the systems programmers established GRS in star mode, which is much easier to handle than GRS ring mode. More and more subsystems began to make use of Parallel Sysplex technology. RACF was the next user of the Coupling Facility (CF). When building PLEXZ1, the systemS programmers consolidated the three RACF databases from SYPF, SYPG, and SYPE into one. This RACF DB was shared across the three systems and for better performance (avoiding much DASD access), RACF used a Coupling Facility. From this point, only one RACF DB was needed for all this client's production systems.

In 2001, IZB began to establish data sharing. System SYPE was removed from PLEXZ1 and became the first member of the sysplex PLEXZ2. In the following years this sysplex grew to ten systems. Because of IZB's standardized system layout, it was relatively easy to migrate new systems into this sysplex. With the first CICS structures and the use of VTAM generic resources, the requirements to the sysplex increased. The XCF connections over the CF links proved to be too slow, so additional XCF paths were needed. Normally, these paths are defined in the CF. With IZB's new operating system layout, only one coupling member exists for the whole sysplex, so the path definitions in this parmlib member turned out to be very

complex. (Today IZB has eight XCF paths from every system to every system in PLEXZ2.) This problem was solved with a REXX exec that starts the necessary XCF paths at IPL time. For each system, there is a table that contains the CTC path addresses. This results in better clarity when a user displays the active/inactive connections. If there is an error, it is easy to restart the failing connections.

With the increase of sysplex-compliant subsystems and the implementation of data sharing, the number of CF structures increased (over 200 in PLEXZ2 today). For better clarity and easier administration, the systems programmers wrote a REXX/ISPF application that uses standard XCF commands and prepares the results in ISPF panels. With a few simple actions, a user is able to repeat complex commands (such as display, rebuild, or force connection) on one or more structures, as shown in the following examples. Figure 9-9 shows the installed Coupling Facilities of sysplex PLEXZ2.

```
Cmd=>               S Y S P L E X - I n f o r m a t i o n      Row 1 to 2 of 2
                        Übersicht Coupling-Facilities            scroll=> CSR
                                MVS/XCF-Overview
Systeminformation:
PLEXZ2: SYPE,SYPF,SYPG,SYP2,SYP3,SYP4,SYP5,SYP7,SYP8,SYP9

Auswahl => Z Zoom S Structuren
A Name     Status      Couple-Facility Information
  CFNG1    in use      002084.IBM.83.00000007C26F PARTITION: 01   CPCID: 00
  CFNH1    in use      002084.IBM.83.00000007C27F PARTITION: 01   CPCID: 00
****************************** Bottom of data ******************************
```

Figure 9-9 Sysplex information panel

Figure 9-10 shows the structure names, the Coupling Facility name, which job use the structure, and the associated connection name. Also shown is which LPAR (system) uses the structure and its status (active, inactive, allocated).

```
Cmd=>               S Y S P L E X - I N F O R M A T I O  Row 288 to 321 of 509
 (L strnm,SHCDS)       Übersicht Structuren/Connections      CF STR on false CF
                          alle Coupling-Facilities           CP Change pending
System: SYP5        <-PF10/PF11-><Enter=Refresh,PF1=Help>    DL pending-delete
PLEXZ2: SYPE,SYPF,SYPG,SYP2,SYP3,SYP4,SYP5,SYP7,SYP8,SYP9    SD Stop Duplexing
Auswahl => D Detail RN RB normal  RO RB other
           SD Stop duplex  FC Force Conn FA Force Conn,ALL FS Force Structure


A  Structure-Name   CFname   Jobname   Conname          Lpar    Status

   IRRXCF00_P001    CFNG1    RACFDS    IRRP001§SYP2     SYP2    ACTIVE
   IRRXCF00_P001    CFNG1    RACFDS    IRRP001§SYPG     SYPG    ACTIVE
   IRRXCF00_P001    CFNG1    RACFDS    IRRP001§SYPE     SYPE    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYPF     SYPF    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP9     SYP9    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP8     SYP8    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP7     SYP7    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP5     SYP5    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP4     SYP4    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP3     SYP3    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYP2     SYP2    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYPG     SYPG    ACTIVE
   ISGLOCK          CFNG1    GRS       ISGLOCK#SYPE     SYPE    ACTIVE
   ISTGENERIC       CFNH1    NET       DEBBSM0N_SYP2    SYP2 CF ACTIVE
   ISTGENERIC       CFNH1    NET       DEBBSM0NDEBBSP03 SYPG CF ACTIVE
```

Figure 9-10 CF structure administration panel

In front of the state field, more information triggered by actions can be shown:

- CF: Structure on false Coupling Facility
- CP: Change pending
- DL: Structure pending delete
- SD: Stop duplexing

Against this structure the following actions are possible:

- D: Detail view, shows all details for the structure
- RN: Rebuild structure to normal location
- RO: Rebuild structure to other location
- SD: Stop duplexing
- FC: Force connection
- FA: Force all connections
- FS: Force structure

This tool simplifies Coupling Facility operation.

To facilitate policy definition, the systems programmers have developed a set of prepared batch jobs. These jobs are used if new structures or logstreams definitions are needed. For new structure definitions, it is sometimes a challenge to determine the required space for the structure. The IBM CF Sizer may help in those situations; this wizard can be found at:

http://www.ibm.com

In addition to this administration work, the systems programmer team is responsible for disaster recovery planning. Until 2004, IZB had a production data center and a standby data center (see Chapter 2, “Developing a multi-site data center” on page 17). Along with the active Coupling Facilities in the production data center, IZB had a standby Coupling Facility for each Parallel Sysplex in the backup data center.

This backup Coupling Facility was defined in the corresponding coupling member, but not physically connected. Backup tests showed that IZB would run into problems during a disaster, when the production data center was down, because the information held in the mirrored couple data sets were incompatible with the empty Coupling Facility.

In order to maintain the agreed restart time limit, the systems programmers developed procedures to restart several Parallel Sysplex environments as quickly as possible. The procedures started one system of a sysplex in monoplex mode, created new definitions of all sysplex structures, and IPLed in Parallel Sysplex mode. Experience showed that this plan was faster than analyzing problems under time constraints. After establishing multi-site production, every production sysplex now has a Coupling Facility at each site, so this problem no longer exists.

With increasing numbers of Parallel Sysplex environments for development systems, there was a hardware cost increase for Coupling Facilities. Part of this increase IZB could eliminate by using of internal Coupling Facilities (ICFs). The competitive configurations are all LPARs (operating system and Coupling Facilities) from one sysplex in one CPC. The costly coupling links could be replaced by internal links; however, redundancy would be lost. Therefore, IZB uses this less costly method only for test and development sysplexes.

By using internal Coupling Facilities, IZB found that sharing Coupling Facility processors resulted in an enormous increase in CPU consumption by the GRS address space. By using dynamic processor sharing, this CPU consumption increases even more.

In the current stage of expansion of sysplex PLEXZ2, the following functions are in use:

- RACF database sharing
- GRS star mode
- CICS DFHLOG and DFHSHUNT
- VSAM record level sharing
- DB2 data sharing
- ADABAS data sharing
- MQSeries
- VTAM generic resources
- TPX generic
- Automatic tape switching
- Enhanced catalog sharing
- JES2 checkpoint
- System logger
- HSM common recall queue

9.3.1 System layout conclusion

The path that IZB followed from single systems to Parallel Sysplex required several years. In the first years (1996 to 1999), the systems programmers experienced some challenges with the new technology. But since 2000 the problems have decreased and the environment became much more stable. At times IZB's strategy was driven by IBM pricing policy, but overall the benefits realized are significant enough for the time and money invested.

Benefits for IZB

- Cost-effective mainframe environment due to sysplex pricing
- More flexibility and scalability
- Less outage time due to redundancy of applications
- Workload balancing effects better use of available capacity

Benefits to systems programmer team

- Managing a 10-way sysplex is only slightly more work than managing a single system, but much less labor-intensive than managing 10 single systems
- Being involved in a technically interesting environment

9.4 Managing workload

The WLM Policy was copied from Cheryl Watson's Starter Policy and modified according to the naming conventions of IZB's started task and transaction workload. For the batch workload, IZB assigned two separate service classes via the Initiator Classification rules. One was for regular batch and had a velocity goal of 30, the other was assigned a discretionary goal so that WLM could let work run in this service class if the other workload left CPU cycles.

After IZB adapted the original Cheryl Watson Policy, it was decided that the naming conventions need more revision as there were 12 different sysplex environments in the installation. Because most of these systems had very different characteristics, it became quite confusing to assign different goals to service classes with the same name and identify a probable cause of trouble without reading the WLM Policy Definition first.

So IZB decided to create service class names that make it easier to identify goals assigned to service classes. IZB divided the service classes into three categories. One was for transaction-related goals, and prefix T was assigned to this category. The next was for

started tasks, which was assigned prefix S. Finally, the batch workloads category was assigned prefix B.The second qualifier identifies the Importance of the workload.

Because IZB needed to be prepared for Resource Capping and WLM Managed Initiator, it took the third qualifier for identifying this type of work; therefore, the third qualifier could be U for Unlimited Resources, R for Resource Group, or W for WLM Managed Initiator. The fourth qualifier was used to identify Periods in transaction workloads, so this got a numeric value to identify how many Periods a transaction workload has. To make this more readable, an underscore was used for the fifth qualifier.

The sixth Qualifier identifies the type of goal: Velocity, Response Time or Discretionary. Using this Service Class Naming Convention simplified identifying the underlying goal and missed goals. Therefore, a transaction with an Importance of 1, with two Periods and a Response Time goal of 0,4 seconds would have a service class named T1U2_0:4. A batch Discretionary work would get a service class named B6U1_00D. A started task with Importance 3 and Velocity 40 would get a service class named S3U1_040. Table 9-4 lists the Workload Manager (WLM) policy naming conventions.

Table 9-4 WLM naming conventions - classification rules

Column	WLM meaning	Type
1-3	Workload	BAT = Batch ADA = Adabas CIC = CICS DB2 = DB2 IMS = IMS MON=Monitor MQS = MQSeries NET = Network Services OUT = Output Management Services SMS = Storage Management Services STC = Started Task Services Basis System WEB = WebSphere
4-8	Identification code	FTP = File Transfer Product STC Subsystem HTT = HTTP Server IWEB Subsystem LAR = LARS /LARA STC Subsystem Three qualifiers for identification of the application Or Transaction name SB20 for CICS Subsystem (CICSB20) PRAP for CB und WEB Subsystem (WEBPRAP)

Table 9-5 WLM naming conventions for service classes

Column	WLM meaning	Type
1	Workload Type	B = Batch S = Started Task Order Process T = Transaction
2	Importance	1 - 6
3	Resource Type	R = Resource Group U = Unlimited W = WLM managed
4	Number of periods	1 – 3 for Workloads with periods
5	Placeholder	_
6-8	Goal	001-085 for Velocity Goals 0:02 - 9:99 for Response Time Goals in seconds 01M - 99M for Response Time Goals in minutes 01S - 24S for Response Time Goals in hours

Table 9-6 illustrates the relationship of workload to service class.

Table 9-6 Relationship of workload to service class

I	Service Classes Started Tasks	Service Classes Transactions	Service Classes Batch	Defined Workloads in seperate Service Classes
0	SYSSTC			Standard operating system, Network Started Tasks
1	S1U1_070 S1U1_055 S1U1_040 S1U1_025	T1U1_0:4 T1U1_0:7 T1U1_1:0		Automation and Monitoring with Velocity Goal Database Systems with Velocity Goal Transactions out of Subsystems
2	S2U1_055 S2U1_040 S2U1_025	T2U1_0:5 T2U1_1:0 T2U2_0:5 T2U2_1:0 T2U3_0:5		Near System Software products like HSM, VTAPE, HSC, CA1 Transactions out of Subsystems TSO
3	S3U1_040 S3U1_025	T3U2_1:0 T3U2_2:0 T2U2_0:5 P2 T2U2_1:0 P2		Output Management Online applications like BETA, LARA UNIX System Service Transactions Transactions out of Subsystems TSO
4	S4U1_030 S4U1_020	T4U2_1:0 T4U2_5:0 T2U3_0:5 P3 T3U2_2:0 P2	B4U1_050 B4U1_035 B4U1_020	Archive and Print systems Test Transactions Time critical Batch
5	S5U1_010	T4U2_5:0 P2	B5U1_030	Unclassified Work Basket Standard Batch
6			B6W1_00D	Default for WLM managed Batch Test Transactions

When IZB merged the two sysplex systems, the names of the service classes made it easy to merge the two definitions into one. Next, IZB had to classify all started task, batch, and transaction workloads that had been classified in the original system, in the policy of the target sysplex and make some minor adjustments in the Velocity goals.

For the batch workload, it was decided after a short review to keep the Initiator assignment of the target sysplex and later reclassify batch if necessary. The started tasks and transaction goals were quite similar, and IZB also kept the definition of the target sysplex and added two service classes for goals which it did not cover in the target system's policy.

9.4.1 Parallel Sysplex - conclusions

Workload management in a Parallel Sysplex is not a simple task. IZB had 10 systems with completely different workloads and an asymmetric configuration. Assigning the appropriate goals to the workloads was difficult. In order to achieve optimal results and assign the right priorities between the front-end systems, back-end systems, supporter systems, and WebSphere systems, IZB invested a great deal of time and effort in analyzing the workloads on all systems.

The RMF reports provided useful information. The predicted velocity was implemented in the sysplex-wide policy. IZB did not adjust these velocity goals to single systems because that is difficult to manage and would have produced overhead in WLM. IZB would have needed to readjust these velocities with every mayor software change. Therefore, it was decided to concentrate on the most important task, which was assigning the right priorities to the workloads and (wherever possible) to assigning response time percentage goals, as these are independent (in contrast to velocity goals).

Note: Keep in mind that you see the effectiveness of your service policy only at peak times, when systems are running at more than 95% utilization. This is when WLM begins to manage the resources.

Lessons learned

- Analyze the relationships between your applications. Understand which applications need service from others. This knowledge is fundamental and necessary.
- Do not accept every recommendation from every ISV, because their products will always "need the highest performance". Instead, rely on your RMF reports and your knowledge of the interaction between the applications such as database systems for guidance.
- Invest sufficient time in measurement and analysis.
- Adjust only few parameters at a time and analyze the result.s

These simple tips represent the only method that IZB has found to achieve usable results.

10

Output management

This chapter documents IZB's work to enable their output management processes to run in a sysplex environment.

10.1 Output management in IZB

At IZB, the entire output management task is divided into these parts:

- JES spool
- Local printing
- Remote printing
- Output browser
- List archive

10.1.1 JES spool

Most IZB systems and application programmers simply store their output from batch and TSO into a Hold class in JES2. This enables them to browse and delete it easily with SDSF, which is installed on all systems. The system deletes held output automatically after 10 days.

10.1.2 Local printing

Local printing is handled by specifying a local destination and the designated output class. Output is printed on channel-attached printers in the data center.

10.1.3 Remote printing

Most printing is done on remote printers. IZB uses only IP-attached printers; these are normal PC laser printers using PCL5 or higher as their print control language. IZB uses an ISV software package from Levi Ray Shoup (LRS) to convert all output from JES2 line and page (AFP™) formats into PCL and send it over TCP/IP to the printer. The allocation of output to these printers is done by class and destination.

10.1.4 Output browser

For browsing and archiving output, IZB uses BETA92 and BETA93. BETA92 collects batch joblog and syslog data, then stores it on DASD and tape. It offers several features to browse the spool data sets online. Output can be stored for up to 11 years. BETA93 is mainly used for list output produced by batch. Output can be printed, stored on DASD and tape, and distributed to several different addressees.

10.1.5 List archive

List archiving and retrieving is done by BETA93FR. This specialized BETA93 handles indexing of lists, stores data on WORM media, and offers an search interface for the indexed data. It is a long-term archive that holds data up to 11 years.

10.2 The problem

In the past, these output management tasks were done on each system in isolation. So all tools needed for output management were installed on each system, and most of the ISV products were licensed on each CPC. The output classes in JES2 used by the products were different on each system. Because of the large amount of production output, a failure of one of the output management products would result in a spool full condition.

In order to reduce the number of licenses and to make maintenance of the software easier and more reliable, IZB decided to concentrate the output management software onto only a

few systems, and to make these highly available. This consolidation would affect not only the sysplex members, but also the development and test systems, as well as part of the production and development environment of the clients.

The consolidation of output management software had to be completed prior to the sysplex merge. Otherwise, at the moment the sysplex was merged, all printing software would try to get hold of the output, which could have resulted in duplicate printing, or no printing at all. The output management products had their own databases, so merging the sysplex without merging the databases would result in several inaccessible or duplicate archived lists.

When IZB began these efforts, it was clear that it would have to supply output management services to more than 30 systems. And it would have to maintain several instances of output management software on several systems to serve only a few people.

Although the number of license fees of the ISVs was a real issue, the growth of CPU power contributed to this expense, because all CPCs had to be licensed. By concentrating the products on only two systems, the amount of licensed software would be reduced dramatically.

10.3 Implementation

IZB decided to focus on the output management products of the two new supporter systems, SYP4 and SYP5. The plan was to reduce the number of MSUs that would be charged by ISVs. So IZB installed all output management products on SYP4 and SYP5, and all libraries are shared among these systems. Most of these products are not sysplex-enabled, so they do not use data sharing. The started tasks or jobs representing the software were started only on one system; on the other system it was installed, but not started. During a system outage, it is easy to tell the automation tool to restart all these products on the remaining sysplex member.

After installing the products, IZB defined new and different output naming conventions, so that each product could read output from the JES multi-access spool (MAS) without interfering with other products. However, the output characteristics of the batch and online applications on all the systems did not match the new naming conventions.

In addition to the normal JES sysout allocations via JCL, IZB also has a large number of batch and online applications that allocate sysout data sets dynamically. Therefore, the attributes used for this output are not easy to change. After estimating the amount of effort needed to change all these programs, IZB decided to develop its own output router task to reroute the JES output.

This application was installed on each system, and it works this way: rule by rule, the output is dynamically changed after the original job places it into the MAS output queue. In this way, IZB was able to adapt all output from all clients—even those with different naming conventions—to a global IZB naming convention, and then route the output from all installed JES2 spools through NJE to the big sysplex JES2-MAS. All this took place without changes to JCL or programs. Then, after the output arrives in the MAS, it is processed by the installed output management software.

10.4 Dynamic output routing via IZTOPR

IZTOPR is the name of the started task that IZB created in 1996 to route output between different JES2 systems. The program scans the JES2 spool and registers output with

traditional output characteristics, and then alters the attributes of the output according to certain rules; see Figure 10-1. The input to the program is a large Rule File.

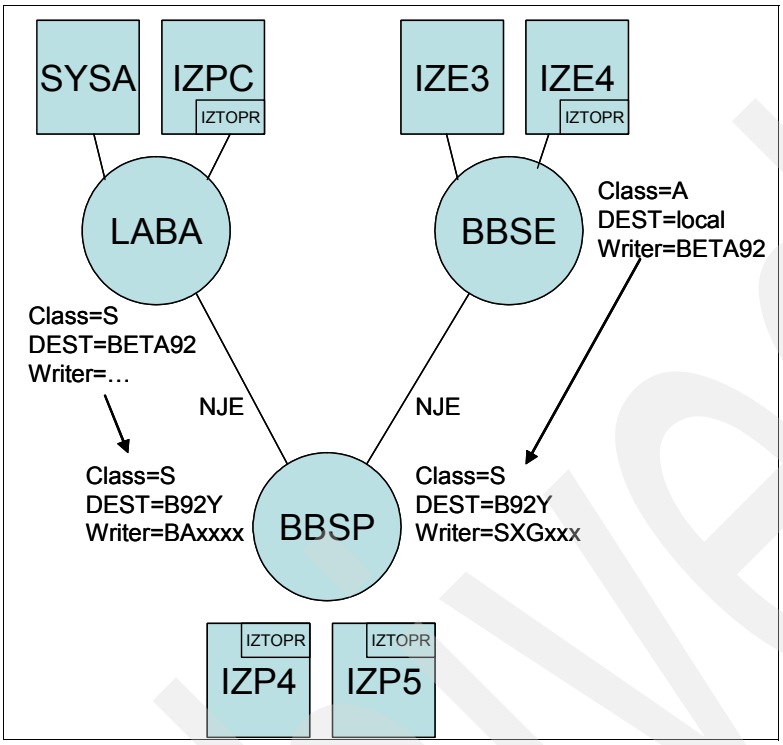


Figure 10-1 Dynamic output routing via IZTOPR

10.5 Lessons learned

As a result of enabling its output management processes to run in a sysplex environment, IZB gained the following insights:

- After all output was migrated to JES2, processing became very slow; it took significantly longer to process output through the JES2 SAPI than on the original system. IZB discovered that this occurred because the hold and dormancy parameter of the JES2 MAS needed to be changed for the system reading most of the output (see Chapter 9, “System” on page 177 for details).
- Because output could arrive from different systems, the JESSPOOL protection in RACF needed to be revised to avoid any security gaps between the old solution and the new one.
- The naming convention for output in JES2 is very important, and is heavily dependent on the client’s application. It is challenging to define a new convention and convert all definitions to use it. But you do not want to do it twice, so it is important to find and adopt a suitable new naming convention.

10.6 Conclusion

At IZB, most of the output management software (archiving, printing, remote-printing) was not originally sysplex-ready, so these products were installed only on one system. Therefore, they needed to be set up so they could be started on a different system in case of system failure. And this required good automation tools in order to ensure that, in each sysplex, only

one instance of the output management software is started. Even though a special tool is used for rerouting output to special systems, this will be needed for dynamically allocated output. By contrast, if JCL allocates the JES output, then changing the JCL will be sufficient.

Points to note:

- Shifting output is a month-long task, because it has to be carefully planned, and you will be shifting job by job—so start early.
- Check all software contracts; you may be able to save license fees by performing a consolidation.

11

Automation

Since IZB was founded in 1994, automation has become an important component in the mainframe arena. A classic MVS operating environment, still found in many companies and consisting of an operator located in front of a bank of consoles and several monitors, has not existed at IZB since 1996. At IZB, personnel have been distributed since 2001 across three locations: Munich, Nuremberg, and Offenbach. The objective from the very beginning was to run the control and monitoring as much as possible in a fully automated manner.

For example, should an error or event occur, such as the threshold value being exceeded because of too high CPU consumption or a looping transaction, then an alert is sent using e-mail. During non-staffed hours, an SMS is sent. In addition, in the majority of cases a problem ticket is opened by the AutoOperator.

An important component of the automation and alert management is the ZIS system. Automation is taken over by the ZIS system, whenever MVS automation tools are not working anymore, such as automatic SA Dump. For more details about the ZIS system, refer to11.6, “A la carte” on page 219 and 11.3, “Automatic IPL” on page 211.

11.1 Standard automation

Shortly after the installation of the first sysplex in 1996, IZG started with monitoring and automation. The first sysplex consisted of two LPARs, one external Coupling Facility, and one Sysplex Timer. The first steps towards automation started with the fact that messages such as `IXL 158I PATH chpid IS NOW NOT-OPERATIONAL TO CUID` and `IXC518I SYSTEM SYS1` were monitored. Those messages were caught by the AutoOperator and an alert was sent. During non-staffed hours at that time, a simple alert was sent to a pager. Today, IZB uses SMS with an exact error message.

The first significant automation success was the automatic move of a system out of the sysplex with a shutdown and an adjacent IPL of the system. The move, using available processor technology, was quite labor-intensive. With today's connection through the HMC consoles, however, this task has become quite simple. The communication to the HMC console is done using SNMP (formerly coax). Functions such as "v xcf,sys1,off", resetnormal, load, and so on, are executed by the ZIS system.

Over the next two years the sysplex automation and the monitoring were expanded, and monitoring with the BMC MainView AlarmManager was set up. For more details about this topic, refer to 6.5.4, "BMC's monitor product "Mainview"" on page 128 and 11.5, "Automating and monitoring with AlarmManager" on page 214.

Figure 11-1 illustrates why automation is a necessity at IZB.

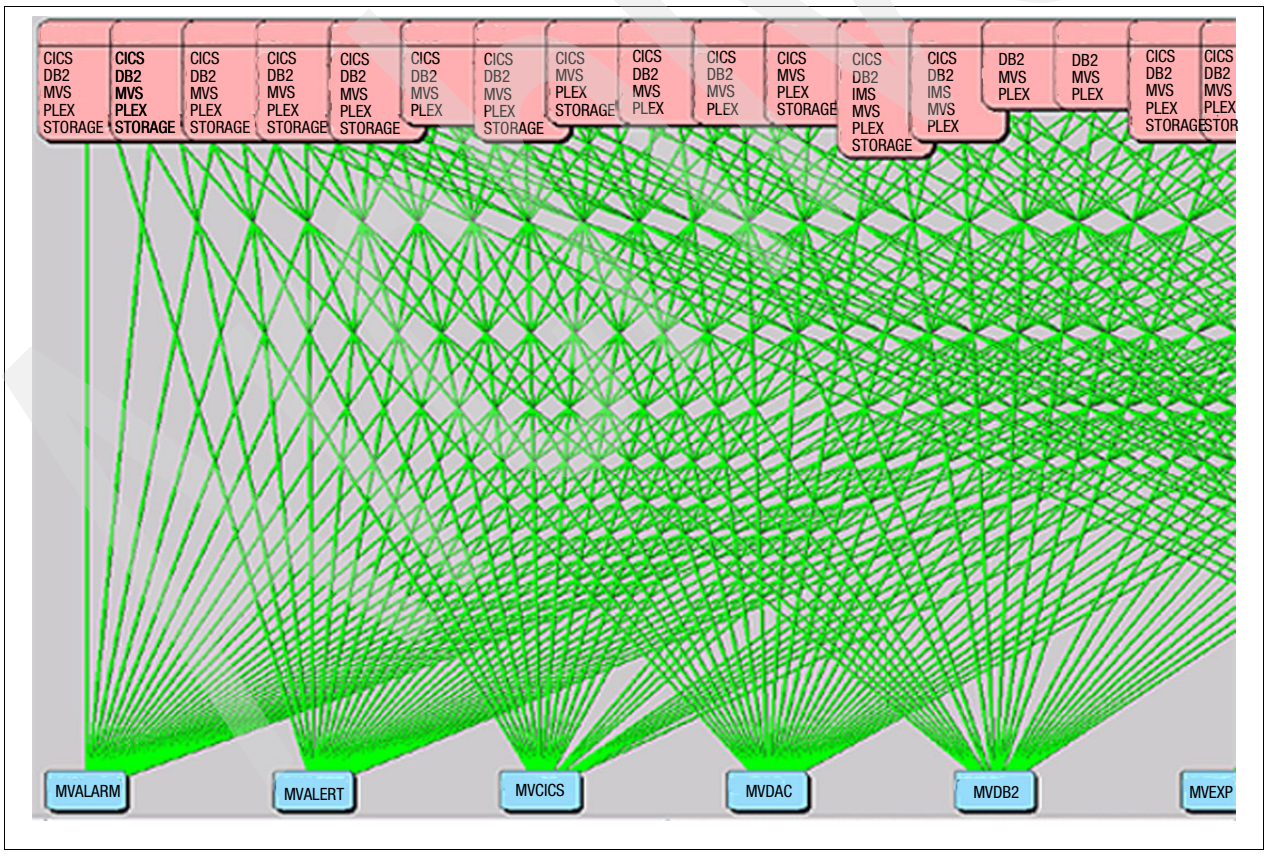


Figure 11-1 Excerpt of automation and its distribution

In 1999, when the first production sysplex went live, all preparations were made and barely any changes had to be made, as the first steps towards an automation standard were taken.

11.2 Automating in three steps

Approximately 2,900 Started Tasks (STC) are defined within the IZB automation environment. However, with that many STCs it is not always easy to keep an overview and control them. So IZB began to consider how to set up an automation standard. The objectives were:

- To keep a unified system structure
- To simplify and accelerate the addition of new systems

This led to the idea of setting up the automation in three steps, as described in this section; see 11.2, “Automating in three steps” on page 209.

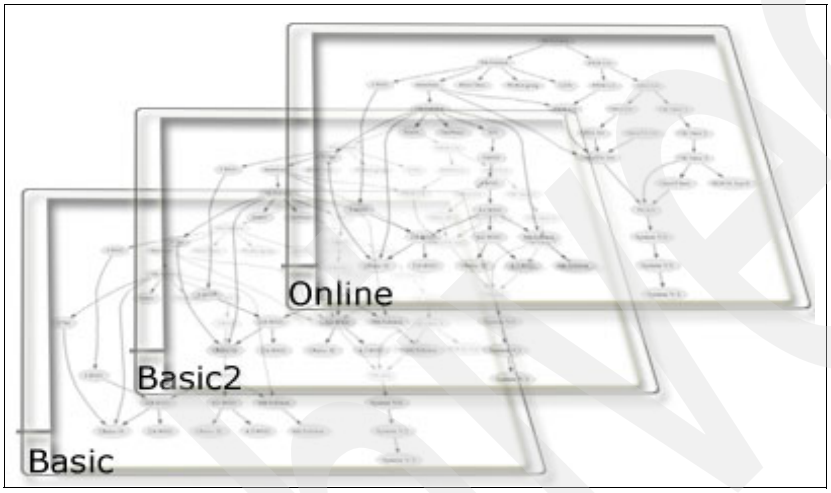


Figure 11-2 Overview of the three steps

Step one: BASIC

BASIC includes all important started tasks such as JES2, VTAM, LLA, TSO, and MVS Monitor to start a functional operating system and the ability to log in. This step is identical on all systems and very helpful for conversion or maintenance work, as IZB quickly built a functional system. Figure 11-3 on page 210 illustrates the BASIC structure.

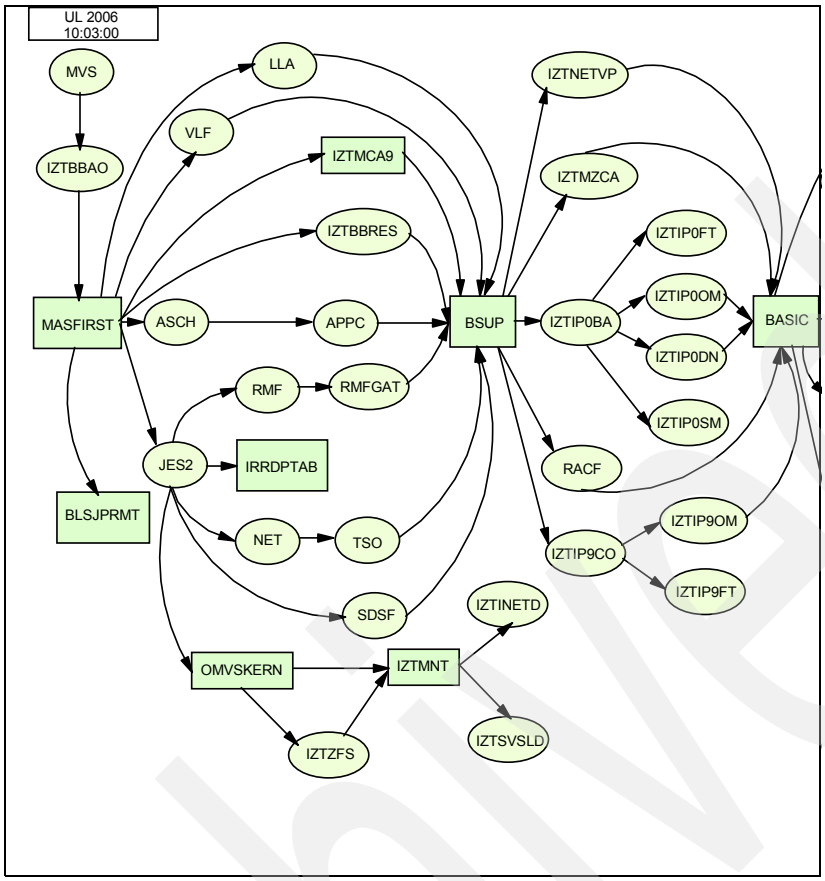


Figure 11-3 Basic structure

Step two: BASIC2

Step two required that all started tasks of BASIC are activated. In Step two, all started tasks like tape processing, other TCP/IP connections, and online monitors are started. BASIC2 is the preparation for the remaining started tasks. This step was separated out in order to accelerate the process of BASIC; also, BASIC2 was not identical on all systems.

Step three: Online systems and tools

Step three requires that all Started Tasks of BASIC and BASIC2 are operational. Here, all online systems (such as Adabas, DB2, CICS, IMS) and other started tasks (such as Tivoli® Workload Scheduler), are started. This step is different from system to system, but includes a structure and dependencies, such as CICSPLEX; see Figure 11-4 on page 211.

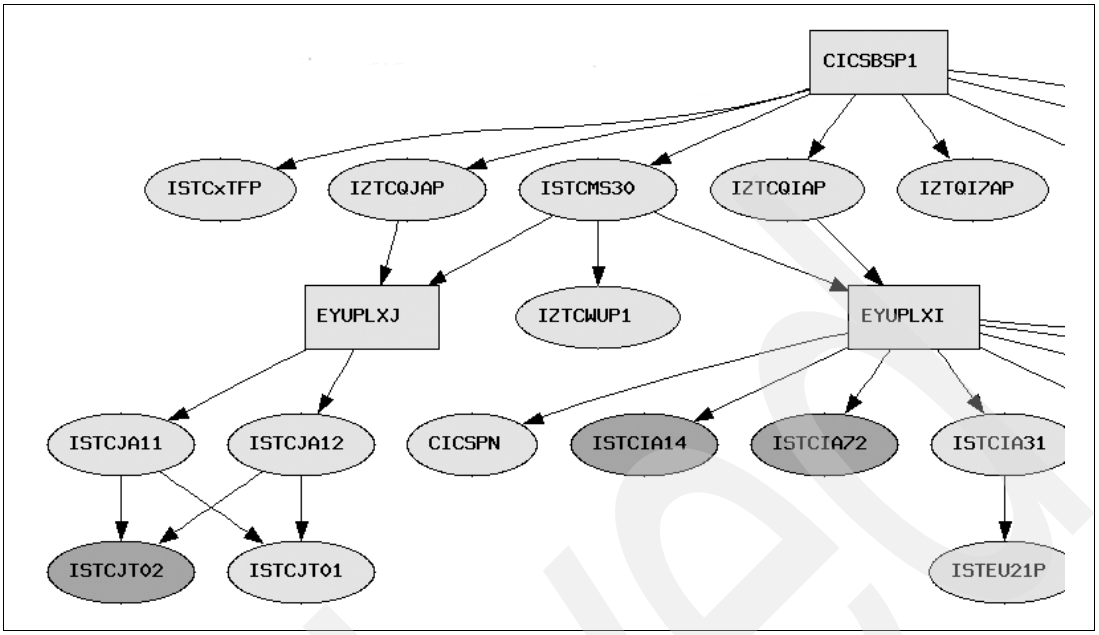


Figure 11-4 CICS structure

With the standardization, IZB achieved the following objectives:

- The starting of the started tasks and their mutual dependencies is nearly identical on all systems.
- For the installation of the BASIC automation (step 1 and 2) on a new LPAR, just one day is required. Formerly, two to three days were required, because a pre-build frame had to be adopted with as little effort as possible and then had to be implemented.
- Each step can be started and stopped individually.
- The IPL for a BASIC system was accelerated.; only four minutes elapse until TSO logon is available.
- A unified system overview was achieved, which is a simplification for everybody.
- A much simpler operating environment results.

11.3 Automatic IPL

Because of the three-step automation setup, it is possible to stop and start the systems at any time, up to a certain level. For example, if only the BASIC system is supposed to remain active, this can be achieved with one single command. The same applies to an IPL. For an IPL, two variables always exist:

- Variable 1: *System shutdown*
- Variable 2: *Re-IPL*

Variable 2 is always set. There is always a *re-ipl*, unless Variable 1 is on, meaning that the system is to remained stopped for maintenance work. The timing control is done with the AO scheduler. This enables the systems to be IPLed without manual intervention during the night or at any point in time. In case of unexpected problems, alerting occurs from the ZIS system.

To IPL, the AutoOperator terminates the started tasks in the order of the defined dependencies. When all started tasks are stopped, then the AutoOperator shuts down. Prior to this, the two IPL variables are released as WTOs to the ZIS console. The messages are

caught by the ZIS, and the last shutdown phase is initiated. First a `z eod` is transmitted. After this is completed, the system is moved with `v xcf,xxx,off` out of the sysplex. The subsequent message is caught again by the ZIS system and `xxx,sysname=sys1` is transmitted.

Then the ZIS system transmits an SNMP on the HCM console containing a “Reset Normal”. At the end a check is run to see if a system is still active within the sysplex. If it is, then the message `xy` is answered with DOWN. After the IPL default value is set to re-IPL, an automatic IPL is initiated.

The procedure is similar to a shutdown. First the deposited IPL variables are read out of a ZIS table and transmitted over SNMP to the HMC console, and then the IPL is initiated. As the first started task, the AutoOperator is started with the parameter **`submstr`**. After the AutoOperator has started, it reports with a message to the console and the ZIS procedure terminates. Afterwards the AutoOperator initiates all defined Started Tasks in the correct order or, if required, until a certain step.

If the IPL variable *System shutdown* been set, there will be no re-IPL. An IPL can be initiated from the ZIS Menu at any time (see 11.6, “A la carte” on page 219). The procedure is the same as described here.

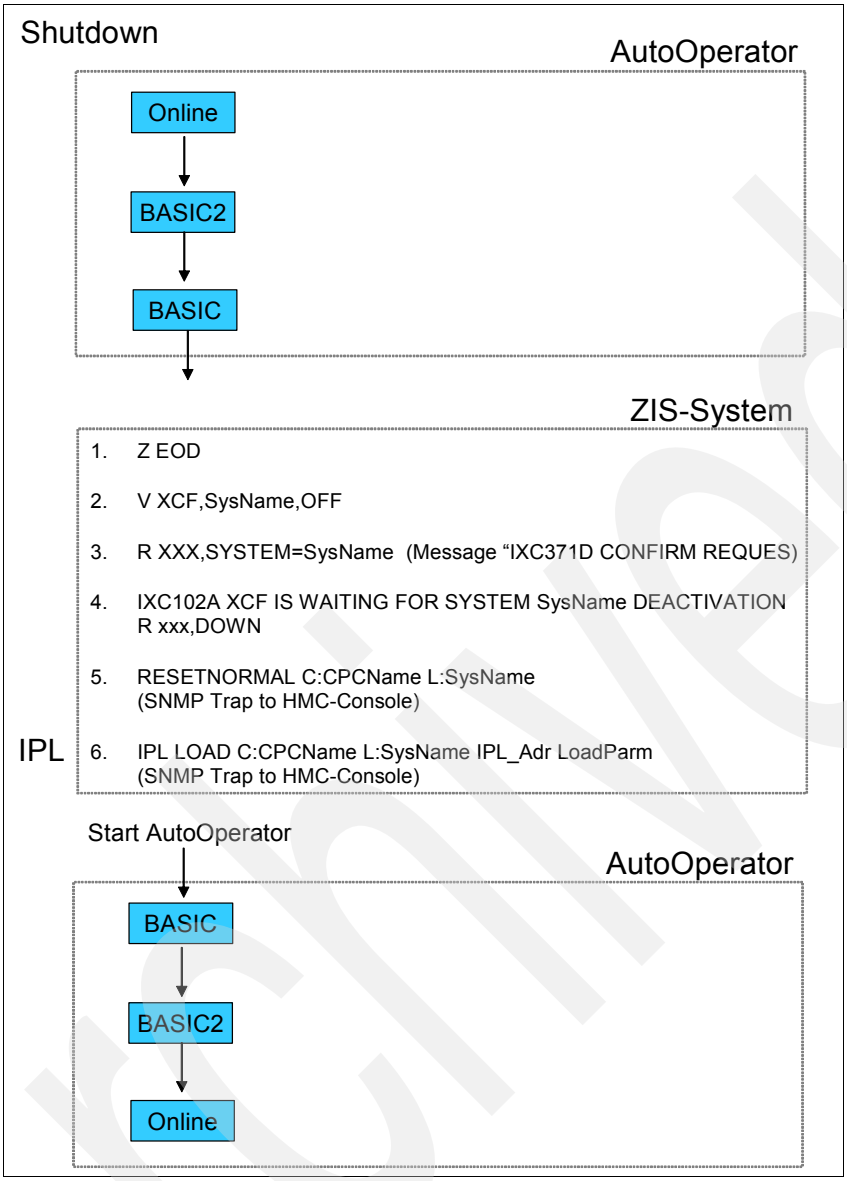


Figure 11-5 Shutdown and IPL process

The ZIS procedures are in-house REXX execs that can be changed any time, and can be adjusted to IZB needs and requirements.

11.4 Starting, stopping, and monitoring started tasks

The start and stop commands of STC are stored in tables within the AutoOperator, and are afterwards set as variables. This feature allows for one started task to have several start and stop variants that can connect to recovery routines.

Starts and stops are timed. Should the start of a STC be unsuccessful after a defined period of time and therefore no success message is sent, an alert message is provided.

As an example, for CICS, three start variables are set:

- Start 1: *AUTO*
- Start 2: *INITIAL*
- Start 3: *COLD*

The CICS regions are generally started with the parameter *AUTO*. If, for example, a CICS cold start is required (start 3), then prior to the start, the AO variables are set using a procedure in *COLD*. After that, all CICS regions will start with *COLD*. After the start, the default value is automatically reset. The change can be performed as well for just one or two CICS regions. This feature enables the user to easily change the start command at any time.

The procedure of the stop commands is similar. During the stop of an STC, the standard STOPP command is executed. If not successful, then the next command is executed.

As an example, for CICS, three stop variables are set:

- Stop 1: *F CICS, cemt p shut*
- Stop 2: *F CICS, cemt p shut immed*
- Stop 3: *C CICS*

Started tasks may be active on only one system within the sysplex but, if required, might need to be activated within a backup on another system. These are defined and marked within the AutoOperator and therefore can be activated at any time.

11.5 Automating and monitoring with AlarmManager

The BMC AlarmManager is core function for IZB's automation and monitoring solution. If set up correctly, the AlarmManager will work like a pre-warn system.

MAINVIEW AlarmManager generates alarms when thresholds from specific MAINVIEW product views are exceeded. Data from multiple systems and subsystems are summarized and monitored. When the threshold values are exceeded, alerts will be sent.

Using MAINVIEW AlarmManager, alarm definitions can be created and modified to display meaningful messages for a site's specific requirements. Alarms can be set for any (or all) of five severity levels, ranging from informational to critical.

AlarmManager connects to every MAINVIEW Monitor, such as z/OS, CICS, DB2, and IMS; see Figure 11-6 on page 215.

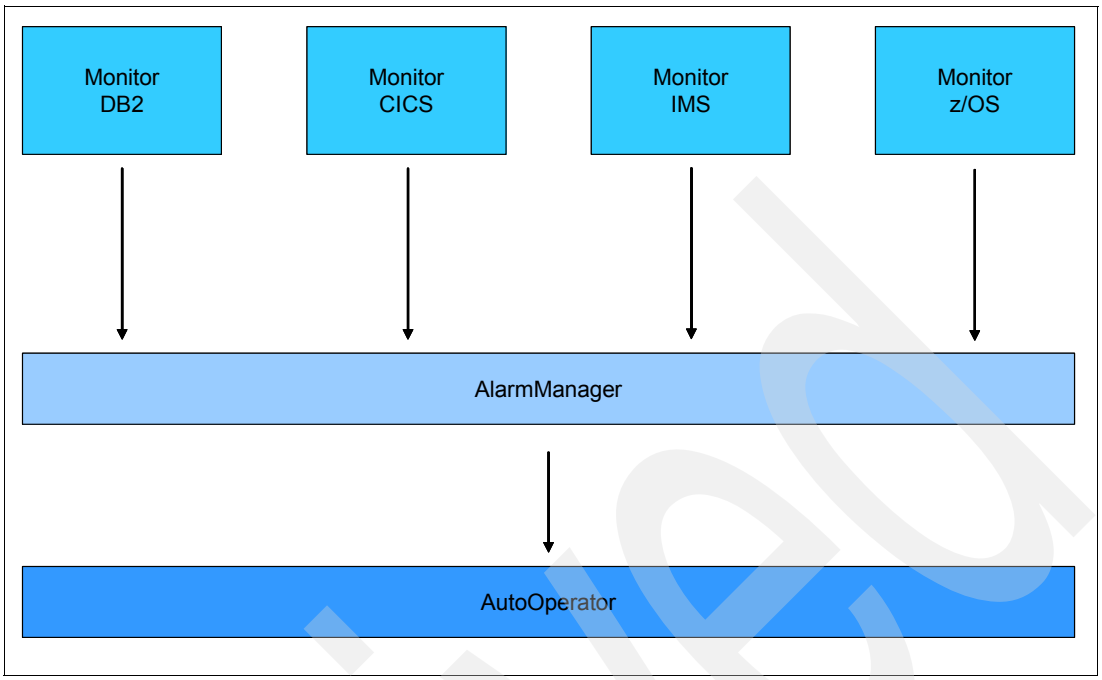


Figure 11-6 AlarmManager

You can trigger every field (by setting thresholds) in every view of these monitors. When the AlarmManager recognizes that a threshold is reached, it will produce a predefined message for this condition. You can create a rule about these messages that starts an action (exec, e-mail). AutoOperator also is able to check every additional field in the views of the Monitors by using an exec that works with the values of these fields (compare, and so on).

MAINVIEW AlarmManager provides the following features and benefits:

- Alarm conditions are monitored based on default thresholds or thresholds set up by the user.
- Thresholds determine whether an alarm is recorded and displayed as critical, major, minor, or a warning or informational message.
- Messages are color-coded to indicate alert severity.
- A set of views is available to display alerts according to severity and chronology.
- Hyperlinking on an alert message displays the view where the exception occurred.
- Alarm definitions can be customized to display messages meaningful to a specific site.
- Time and days for monitoring can be specified in each alarm definition, as well as monitoring frequency.
- User-created help panels can provide specific instructions when a certain alarm occurs.
- Alarms can be forwarded to MAINVIEW AutoOperator for automatic actions.

In MAINVIEW for z/OS, the View CPUSTAT for the value of CPU BUSY is triggered; see Example 11-1.

Example 11-1 CPU Busy

```
24NOV2005  16:12:23 ------ MAINVIEW WINDOW INTERFACE (V4.2.05) ----------------
COMMAND  ===>                                                SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =CPUSTAT===========SYPB=====*========24NOV2005==16:12:23====MVMVS====D====1
```

```
C No TYPE      CPU Busy(I)    TSO  BAT  STC       CPU Busy(R)    TSO  BAT  STC
- -- --        0.....50...100 Busy Busy Busy      0.....50...100 Busy Busy Busy
  00 CP   30.3                          30.3 64.8                 0.0      64.8
```

A threshold was set at 99%. If the threshold is reached, add 1 to a counter and set the time of the first appearance. If the message occurs six times in an interval of fifteen minutes, information in an e-mail message will be sent.

In case of an AUXILIARY STORAGE SHORTAGE, AutoOperator will connect to the View JSLOT of MVS Monitor to detect the job that is using the most auxiliary storage SLOTS; see Example 11-2.

Example 11-2 JSLOT

```
24NOV2005  15:57:07 ------ MAINVIEW WINDOW INTERFACE (V4.2.04) ----------------
COMMAND  ===>                                                 SCROLL ===> PAGE
CURR WIN ===> 1        ALT WIN ===>
 W1 =JSLOTX===========SYPC=====*========24NOV2005==15:53:21====MVMVS====U====7
C Jobname  Jobnumber Typ JSLOTS Job
- -------- --------- --- ------ Status
  IM00MQ02 JOB13286  BAT    448 ACTIVE
  IM02AUGE JOB13596  BAT    436 ACTIVE
  IM00MP22 JOB09133  BAT    272 ACTIVE
  IM00MP24 JOB09143  BAT    270 ACTIVE
  IM00MP23 JOB09135  BAT    267 ACTIVE
  IM00MP21 JOB10578  BAT    123 ACTIVE
  IM00MP20 JOB10572  BAT    119 ACTIVE
```

With this information, different actions can be started. IZB uses two options:

- A CANCEL of the job will take place in development and test systems.
- For production systems, an e-mail will be sent to the automation group.

11.5.1 Console definition

The linkage of the multi-console support (MCS) consoles is done from the ZIS system and is maintained by the automation team; see Figure 11-7. IZB decided on the ZIS linkage because only this linkage ensures that:

- The remote access is a secured one.
- Only certain people have access to the MCS consoles.

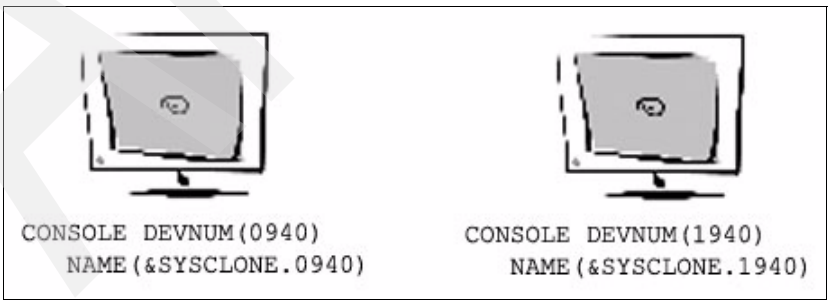


Figure 11-7 Primary console and secondary console

Another reason for the ZIS linkage was because it contains a console monitoring function. This enables messages of any kind to be defined and captured, such as messages that generate an SMS alert. To keep the maintenance effort low, only one console member exists for all IZB systems

Prior to the changeover, IZB had several console names with different addresses. For standardization purposes, all primary consoles were defined with address 0940 and all secondary ones (also used for backup) were defined with address 1940; see the following examples.

Definitions for MCS console

Example 11-3 Primary MCS console definitions

```
CONSOLE DEVNUM(0940)
   NAME(&SYSCLONE.0940)
   ALTGRP(BACKUP)
   MSCOPE(*)
   DEL(RD)
   MFORM(J,T,S)
   MONITOR(JOBNAMES-T,SESS-T)
```

Example 11-4 Secondary MCS console definitions

```
CONSOLE DEVNUM(1940)
   NAME(&SYSCLONE.1940)
   ALTGRP(BACKUP)
   MSCOPE(*)
   DEL(RD)
   MFORM(J,T,S)
   MONITOR(JOBNAMES-T,SESS-T)
```

Definitions for SMCS console

Example 11-5 SMCS console definitions

```
CONSOLE DEVNUM(SMCS)
   NAME(SMCS&SYSCLONE.00)
   LOGON(REQUIRED)
   ALTGRP(BACKUP)
   MSCOPE(*)
   DEL(R)
   MFORM(J,T,S)
   MONITOR(JOBNAMES-T,SESS-T)
```

The consoles for print and tape have been set up as SNA Multiple Console Support (SMCS) consoles. The advantage offered by SMCS consoles is that they do not occupy a port on a 2074. The SMCS consoles have been defined with parameter and Sysclone to avoid having each system create their own SMCS consoles within the sysplex. Therefore only one console member exists for all systems inside and outside of the ‘plex system. Modifications need only be made on the test and development ‘plex system, and then distributed to the other ‘plex systems.

11.5.2 The distribution system

In order to keep all systems on the same level of automation and monitoring, IZB has developed a distribution system; see Figure 11-8 on page 218.

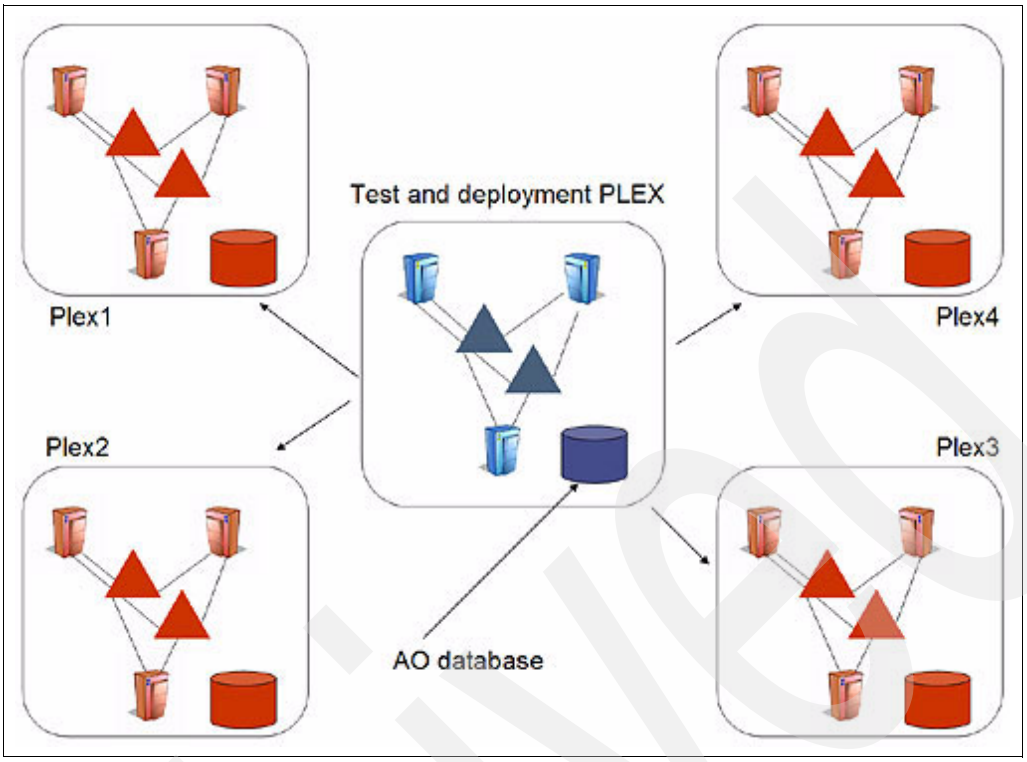


Figure 11-8 IZB distribution system

Development and testing of new automation rules and monitoring threshold values is done on a test and development 'plex. This 'plex is also used for:

- Testing by simulating errors, such as breakdown of a Coupling Facility in operating mode
- Testing and development of new products or new software releases

When developing new automation rules and guidelines, extensive test scenarios are executed on all test systems. When the executions are successful, distribution occurs to all production systems. The distribution to all production systems is done with a user-written utility, shown in Figure 11-9.

```
>    < TRANSFER UBBPROC     (REXX)
>    < TRANSFER TESTPROC    (REXX)
>    < TRANSFER UBBPLIB     (PANELS)
>    < TRANSFER UBBVTAM     (VTAMNODES)
>    < TRANSFER UBBPARM     (RULES)
>    < TRANSFER CNTL        (JOBS)
>    < TRANSFER TABLE       (TABLES)
>    < TRANSFER DOC         (TICKETS)
>    < TRANSFER SNCC.REXX   (EXECS)
>    < TRANSFER SNCC.PLIB   (PANELS)
>    < TRANSFER SBBSDEF     (SCREENS)
>    < TRANSFER SBBVDEF     (VIEWS)
>    < TRANSFER TABLE2      (TABLE-LONG)
>    < TRANSFER ALARME      (GROUPS)

>            < MEMBER NAME
```

Figure 11-9 Distribution tool

Among other things, AutoOperator procedures and Alert Manager definitions are sent with the distribution tool. If a REXX procedure is to be sent, then the field Transfer UPPROC is

chosen and the procedure name is entered in the lowest line. Afterwards, the distribution is carried out.

11.6 A la carte

Another automation tool is the ZIS system. It is used at IZB primarily for the following tasks:

- Monitoring of system Heartbeat-Check
- Automatic IPL (reset, load, and so on over the HMC console)
- Automatic SA dump
- Notification of problems (using SMS, e-mail, pager, and so on)
- Automatic adjustment of CPU (power/speed) weight
- Automatic on and off switching of CBU (Capacity Backup Upgrade)
- Remote access to the systems (over telephone or LAN)

All 2074 and HMC consoles are connected to ZIS systems. Two HMC consoles and one MCS console per location are installed for each system being monitored by the ZIS system; see Figure 11-10.

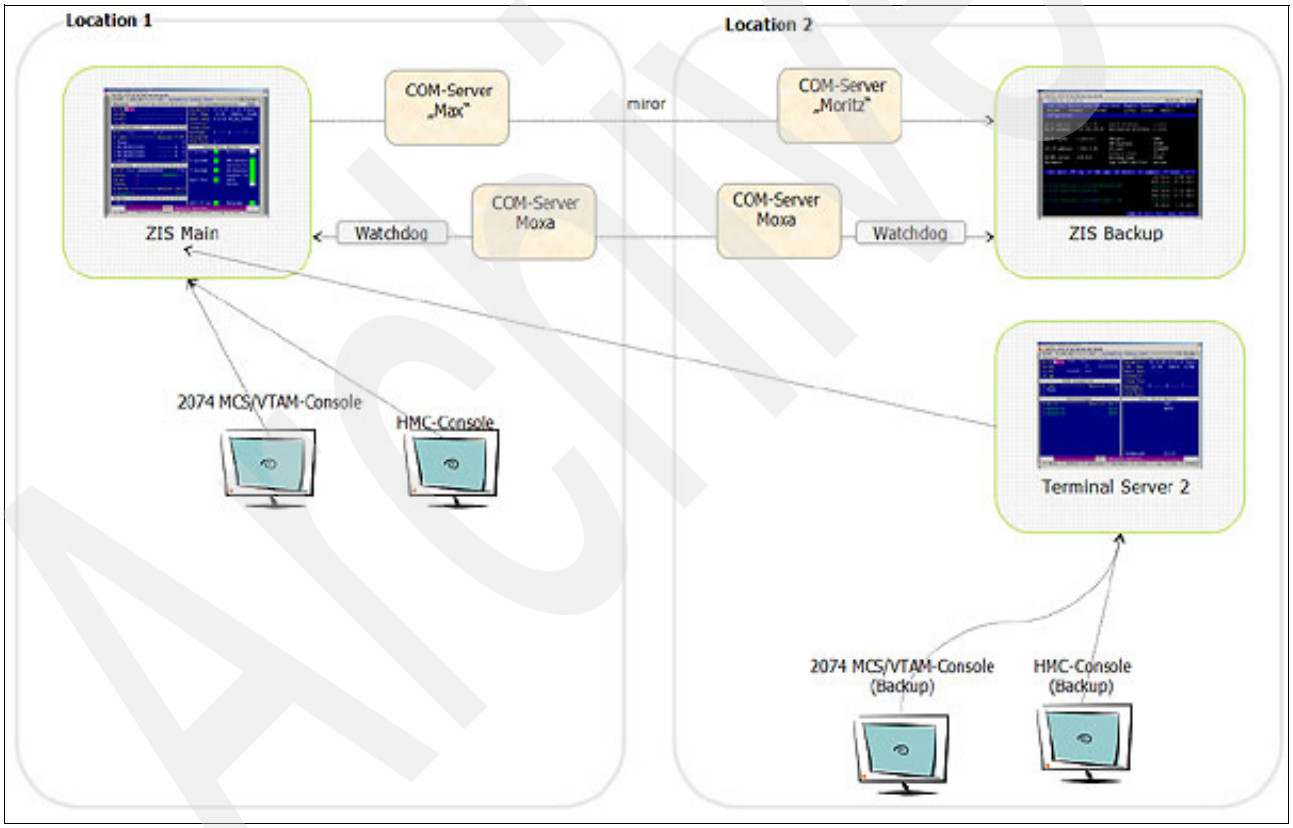


Figure 11-10 ZIS configuration

The ZIS system provides process management of cross-platform systems with a large number of interfaces to virtually all systems relevant for computer center operation:

- Mainframes
- Heterogeneous systems
- Servers
- Network components
- Infrastructure and security features

The ZIS system is composed of a series of console emulations and a large number of TCP/IP interfaces, as well as various hardware interfaces including KVM and video interfaces. They allow a complete integration of the computer center systems into the management concept.

To minimize downtime, remote interfaces are available for specialists to correct errors around the clock. This is a unique combination on the market providing concise monitors (process, systems and network management with full alarm functions) including a large number of operating interfaces (power management, consoles) and links to documentation files and video images on an integrated user interface.

ZIS system operation includes a primary system as well as a second system that is a reflection of the first. Other consoles are connected over a terminal server. In the event of a failure of the primary system, the secondary system automatically boots up and takes over all functions.

With the move and construction of a new datacenter in Nuremberg, as described in Chapter 2, “Developing a multi-site data center” on page 17, the ZIS system underwent an overhaul as well. From two separate systems, one system was created in two locations. If necessary, they can be operated separately, as tested during IZB backup checks. Since the configuration of the complex is divided between two locations, when backing up the components and each location, the system might need to be started on a different CPC with different parameters. These are listed in a separate menu.

Depending on the situation, the parameters can be changed by a switch button, after which the automatic IPL can be started. The reset and IPL load for the respective systems is performed through the ZIS system using an SNMP trap at the HMC console. As a result, monitoring is taken over by the MCS console. After the Auto-Operator has been started and is active, the ZIS system hands over control. All other system starts are taken over by the Auto-Operator as well. For more details, refer to 11.2, “Automating in three steps” on page 209.

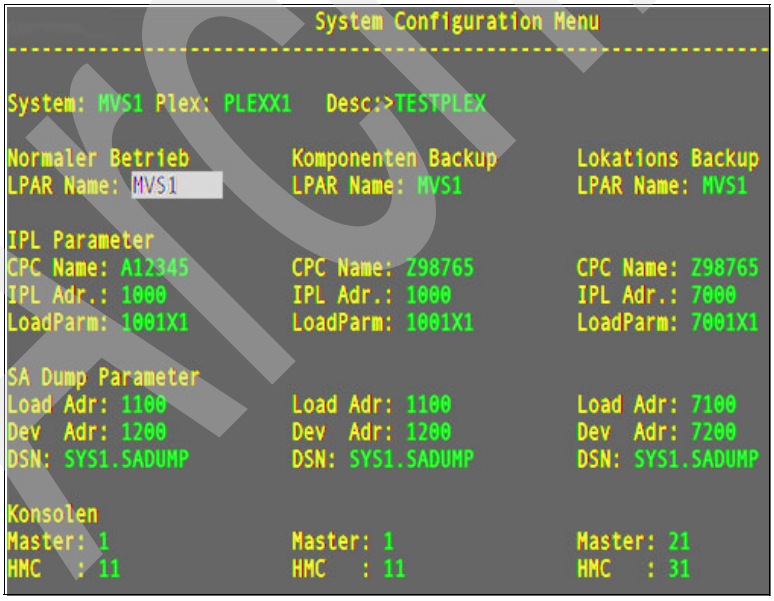


Figure 11-11 IPL and SA dump menu

11.7 The tools that were used

IZB used the following tools for this automation project:

- BMC tools
  - MAINVIEW AutoOperator for z/OS Version 6.4
  - MAINVIEW for DB2 Version 8.1
  - MAINVIEW for CICS Version 5.8
  - MAINVIEW for IMS Version 4.1
  - MAINVIEW for OS/390 Version 2.7
- LeuTek Company tool
  - ZIS Version 4.20

11.8 Conclusions

How long did the conversion to automation take at IZB? Automation is a dynamic process that must be worked on continually. This is why a realistic indication of how long it will take until completed cannot be given.

Successful automation has these characteristics:

- Automation has to be developed and defined by people who work with it on a daily basis.
- The needs of the client must be taken into consideration and worked on with the other teams.
- Regular processes and solutions have to be discussed with the automation team (many discussions are required).
- The needs of automation have to be adjusted to the system environment (every company is different).
- New tools have to be included.
- Ensure the automation is easy, understandable, clear, and well documented.

11.9 Outlook

What are the next steps for IZB? It is about to expand automation in the direction of “event management” and visualization of the business processes, using the concepts that IZB have already found to be successful.

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 224. Note that some of the documents referenced here may be available in softcopy only.

- *Merging Systems into a Sysplex*, SG24-6818
- *FICON Native Implementation and Reference Guide*, SG24-6266
- *Networking with z/OS and Cisco Routers: An Interoperability Guide*, SG24-6297
- *TCP/IP in a Sysplex*, SG24-5235
- *SNA in a Parallel Sysplex Environment*, SG24-2113
- *J.D. Edwards OneWorld XE Implementation on IBM @server iSeries Servers*, SG24-6529
- *OS/390 Parallel Sysplex Configuration, Volume 1: Overview*, SG24-5637
- *CICS and VSAM Record Level Sharing: Planning Guide*, SG24-4765
- *OS/390 Parallel Sysplex Configuration, Volume 2: Cookbook*, SG24-5638
- *z/OS Systems Programmers Guide to: Sysplex Aggregation*, REDP-3967

Other publications

These publications are also relevant as further information sources:

- *zSeries Capacity Backup (CBU) User's Guide (Level -02h)*, SC28-6823
- *z/OS V1R4.0-V1R5.0 MVS Programming Sysplex Services Reference,* SA22-7618
- *System z9 and zSeries Capacity on Demand User's Guide*, SC28-6846
- *z/OS V1R1.0 Parallel Sysplex Application Migration*, SA22-7662
- *Enhanced Catalog Sharing and Management*, SC24-5594
- *CS IP Configuration Guide*, SC31-8725
- *CICS System Definition Guide*, SC34-6428
- *CICSPlex SM Concepts and Planning,* GC33-0786
- *CICSPlex SM for CICS TS z/OS, Administration,* SC34-6256
- *CICSPlex SM for CICS TS z/OS, Managing Workloads,* SC34-6259
- *CICS Transactions Affinities Utility Guide,* SC34-6013
- *OS/390 V2R5.0 Parallel Sysplex Application Migration,* GC28-1863
- *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935

- *Getting Started - IBM WebSphere Application Server for z/OS V5.0.2*, GA22-7957
- *Adabas Implementation Guide for Cluster Services*, available from Software AG:

  http://documentation.softwareag.com

Online resources

These Web sites and URLs are also relevant as further information sources:

- IBM Web pages for Parallel Sysplex

  http://www-03.ibm.com/servers/eserver/zseries/pso/

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

R

S

T

U

V

W

Y

Z

Exploiting Parallel Sysplex: A Real Customer Perspective

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

Exploiting Parallel Sysplex: A Real Customer Perspective

Quantifiable benefits

Actual implementation efforts

Lessons learned in the real world

IBM System z is well known for its reliability, availability, and serviceability (RAS). So how can your enterprise obtain all this benefit, along with low total cost of ownership (TCO) and excellent centralized administration? And what other benefits can you realize by implementing a Parallel Sysplex configuration? This IBM Redbook answers these questions by documenting the experiences of a real life client that undertook this process. Informatik Zentrum Frankfurt und München (IZB) completed a Parallel Sysplex implementation, and it shares its perspective with you in this detailed analysis.

IZB is a large banking service provider in Germany. Five years ago, it had two data centers with differing standards, no sysplexes, underpowered hardware, and an expensive, tape-based disaster recovery strategy. It lacked DASD or tape mirroring; it needed to customize systems for each client; and it could not aggregate CPUs, resulting in elevated software licensing costs. Today, by exploiting Parallel Sysplex features, IZB is achieving maximum service levels and financial value from its System z environment, and looking forward to growing its client base.

This publication provides step-by-step information about how IZB set up its System z environment. Covering the areas of processors, disk, networking, middleware, databases, peripheral output, and backup and recovery, the book was written to demonstrate how other installations can derive similar value from their System z environments.

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

SG24-7108-00 ISBN 0738494976