z/OS Systems
Programmers Guide to: Sysplex Aggregation

Understand the requirements for aggregation
Considerations for different sysplex types
Using workload balancing to help maintain aggregation

Frank Kyne
Matthias Bangert
Marcy Zimmer

ibm.com/redbooks
Note: Before using this information and the product it supports, read the information in “Notices” on page vii.
## Contents

Notices .................................................................................. vii  
Trademarks .......................................................................... viii  

Preface ................................................................................. ix  
The team that wrote this Redpaper ................................................ ix  
Become a published author ........................................................ xi  
Comments welcome ............................................................ xi  

Chapter 1. Introduction ............................................................. 1  
1.1 Objective ........................................................................ 2  
1.2 Sysplex aggregation ......................................................... 2  
1.3 Parallel Sysplex options and considerations ......................... 3  
1.4 Sample scenarios ............................................................ 3  
1.5 Workload routers and work managers ................................. 4  

Chapter 2. Sysplex aggregation .................................................. 5  
2.1 Sysplex aggregation overview ........................................... 6  
2.1.1 Background ................................................................ 7  
2.1.2 Terminology ............................................................. 9  
2.2 Sample configurations ..................................................... 10  
2.2.1 Sample configuration 1 ................................................ 10  
2.2.2 Sample configuration 2 ............................................... 12  
2.2.3 Sample configuration 3 ............................................... 14  
2.3 Aggregation rules and criteria .......................................... 17  
2.3.1 Aggregation requirements .......................................... 17  
2.3.2 Comparing aggregation to subcapacity charging requirements ............................................... 20  
2.4 Other considerations ....................................................... 20  
2.4.1 What is “normal mode of operation” ............................ 20  
2.4.2 Validation requirements ............................................. 21  
2.4.3 Sysplex members with different local times .................. 24  
2.5 Getting started ............................................................. 24  
2.6 Challenges to sysplex aggregation implementation ............. 25  

Chapter 3. Parallel Sysplex options and considerations ................. 27  
3.1 Different sysplex configuration types .................................. 28  
3.1.1 BronzePlex ............................................................ 28  
3.1.2 PlatinumPlex ........................................................ 29  
3.1.3 Multi-site sysplex .................................................... 30  
3.2 Advantages of a PlatinumPlex ......................................... 32  
3.2.1 Availability .......................................................... 32  
3.2.2 Systems management .............................................. 36  
3.2.3 Maintaining sysplex aggregation qualification .............. 37  
3.3 Sysplex aggregation qualification ..................................... 39  
3.4 To merge or not - a sample scenario ................................. 40  
3.5 Summary ....................................................................... 43  

Chapter 4. Sample scenarios ..................................................... 45  
4.1 Introduction .................................................................... 46  
4.2 Scenario 1 - Checking all the alternatives ......................... 46
Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing, IBM Corporation, North Castle Drive Armonk, NY 10504-1785 U.S.A.

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

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

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

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

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

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

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

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:
- @server®
- Redbooks (logo) ™
- z/OS®
- z/VM®
- zSeries®
- z9™
- AIX®
- CICS®
- CICSPlex®
- Domino®
- DB2 Connect™
- DB2 Universal Database™
- DB2®
- DFSMSdss™
- DFSMSHsm™
- DFSORT™
- DRDA®
- GDPS®
- HyperSwap™
- IBM®
- IMS™
- IMS/ESA®
- Lotus Notes®
- Lotus®
- MVS™
- MVS/ESA™
- NetView®
- Notes®
- OS/390®
- Parallel Sysplex®
- PR/SM™
- Redbooks™
- RACF®
- RMF™
- Sysplex Timer®
- System z9™
- Tivoli®
- VTAM®
- WebSphere®

The following terms are trademarks of other companies:
- Java, and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
- Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.
Preface

This Redpaper is aimed at technical management and the Systems Programmers that will have the task of implementing any changes that may arise in the course of adjusting your environment to meet the sysplex aggregation criteria. It provides information that will help you understand and quantify the impact that sysplex aggregation qualification may have on your availability and your IT costs.

Should you decide that changes are required to meet the qualification criteria, it will help you design/create a solution to create a sysplex environment that delivers both the availability and systems management benefits of Parallel Sysplex® and the cost savings that can be achieved through the exploitation of sysplex aggregation.

The team that wrote this Redpaper

This Redpaper was produced by a team of specialists from around the world working at the International Technical Support Organization, Poughkeepsie Center.

Frank Kyne is a 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. Frank is also responsible for the GDPS® product documentation. Before joining the ITSO seven 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® Field Technical Sales Support organization in Munich, Germany. Matthias' areas of speciality are disaster recovery and data center merges. He is a project leader for different customer projects as well as a regular speaker at customer briefings and events. Before joining IBM in 1999, Matthias worked for seven years as an MVS Systems Programmer and six years as a System Engineer specializing in storage subsystems.

Marcy Zimmer is a member of the IBM System z9™ and zSeries Pricing and Investment Analysis team in IBM Poughkeepsie, New York. She has 6 years of experience in large systems and software pricing and offerings. She holds a bachelor's degree in Information and Decision Sciences and a master's degree in Information Systems Management from Carnegie Mellon University. Marcy's last 5 years at IBM have been focused on the area of zSeries software pricing including: Sub-Capacity pricing, Sub-Capacity Planning Tool and Sub-Capacity Reporting Tool design/support, defined capacity, software asset management, customer outreach, and Parallel Sysplex aggregation.

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

Rich Conway
International Technical Support Organization, Poughkeepsie Center

Peter Baueuerle
IBM Germany

Bill Bitner
IBM USA
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 customers.

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 papers to be as helpful as possible. Send us your comments about this Redpaper 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 email to:
  redbook@us.ibm.com
- Mail your comments to:
  IBM Corporation, International Technical Support Organization
  Dept. HYJ  Mail Station P099
Introduction

This chapter introduces the objective of this RedPaper and describes the topics that will be presented in the remainder of the document.
1.1 Objective

The objective of this document is to help you understand the precise requirements for sysplex aggregation, and to suggest ways that you can achieve and maintain qualification for sysplex aggregation pricing while balancing technical and financial objectives. It is aimed not only at those installations that would like to implement further aggregation, but also at those that have already arrived at their target number of “PricingPlexes”\(^1\), to help you ensure that you continue to conform to the aggregation qualification criteria with the minimum manual intervention.

This document has two target audiences:

- If you already have two or more CPCs aggregated, we suggest ways to configure your systems so as to ensure that you continue to meet the aggregation criteria with minimal ongoing effort.
- If you have not yet achieved your target number of PricingPlexes, it will help you evaluate the technical considerations and options related to moving to a smaller number of PricingPlexes. We want you to obtain the best balance of the availability and systems management benefits of Parallel Sysplex, and the potential financial benefits of a smaller number of PricingPlexes.

1.2 Sysplex aggregation

Sysplex aggregation is a set of rules that IBM have defined that determine the basis that will be used when deciding how some z/OS®-related software should be charged. The concept of sysplex aggregation was introduced when IBM announced Parallel Sysplex and the 9672 range of CPCs\(^2\) in 1994. At its most simplistic, sysplex aggregation allows you to pay for software on two CPCs as if those CPCs were in fact a single CPC. The reason this is attractive is that the software price per additional unit of processing power decreases as the CPC size increases, providing a curve similar to that shown in Figure 1-1 on page 3. You can see that as the number of MSUs increases, the incremental cost of each additional MSU gets smaller and smaller. From a purely financial perspective, everyone wants to be on the right side of the curve, to minimize the software cost of capacity upgrades.

---

\(^1\) We use the term “PricingPlex” to describe the group of CPCs that are treated as an aggregate for software pricing purposes.

\(^2\) We use the term CPC to describe a Central Processing Complex. Other commonly used names are processor, CEC, box, or footprint, however we will use CPC throughout for consistency.
2.3.1, “Aggregation requirements” on page 17 provides a detailed description of the precise criteria for being able to avail of the financial benefits of sysplex aggregation.

At this stage in this RedPaper, we want to make it clear that we are in favor of sysplex aggregation. Sysplex aggregation improves the cost-competitiveness of the System z9 and zSeries platforms and makes z/OS a very cost-effective operating system for new applications such as WebSphere®, ERP applications, Lotus® Domino®, and so on. However, we do have concerns about the technical implications of placing completely unrelated systems in the same sysplex. The objective of this document is to help you identify the correct balance between meeting the sysplex aggregation criteria, without negatively impacting the systems management and application availability leadership that System z9 and zSeries is famous for.

1.3 Parallel Sysplex options and considerations

There are many ways a Parallel Sysplex can be configured, from a shared-nothing model, where the intent is that all the members behave as if they were not in the same sysplex, to a shared-everything model, where all work can run on any system in the sysplex. If you are coming from a non-sysplex configuration, achieving either type of Parallel Sysplex will require some amount of work.

In general, we believe that the shared everything model provides far more value than the shared-nothing model—apart from anything else, creating shared-nothing sysplexes often entails far more work than people realize when they embark on such a project, while delivering reduced technical benefits compared to the shared-everything model. Chapter 3, “Parallel Sysplex options and considerations” on page 27 provides a valuable discussion about the benefits, drawbacks, and considerations for each of the different types of Parallel Sysplex, not just in the short term, but also on an ongoing basis.

1.4 Sample scenarios

There are as many different ways to configure your systems and your work as there are z/OS customers, so it is difficult to provide answers or guidelines that are the definitive correct
answer in every case. However, there are some configurations and challenges that are common across many customers. Therefore, we have provided a number of sample configurations, based on actual customer scenarios, that show a “before” and “after” configuration. In some cases, the solution resulted in more CPCs being included in the aggregation group, and in other cases such a change did not make sense - we discuss both situations in Chapter 4, “Sample scenarios” on page 45.

1.5 Workload routers and work managers

Regardless of the type of Parallel Sysplex you implement, in order to get both the financial benefits of sysplex aggregation and the availability benefits of Parallel Sysplex, you are going to have to give consideration to what work runs in each system in your sysplex. From a software pricing perspective, balancing workload becomes critical to meeting IBM’s aggregation requirements. Chapter 5, “Workload balancing and sysplex aggregation” on page 65 discusses each of the IBM-provided workload routers (like TCP/IP Sysplex Distributor) and work managers (like IMS™ Transaction Manager), and how they can be used to help achieve a more even balance of work across all the systems in your sysplex.
Chapter 2. Sysplex aggregation

This chapter describes the background to the sysplex aggregation rules and benefits, provides a detailed list and clarification of the precise aggregation criteria, and addresses some of the most commonly-asked questions.
2.1 Sysplex aggregation overview

So, what exactly is sysplex aggregation? Sysplex aggregation provides the ability to pay for software on more than one CPC as if the software was executing on a single large CPC of the same capacity as the aggregate capacity of the individual CPCs. The customer benefit is that the price per additional MSU for software typically decreases as the number of MSUs increases. This is shown graphically in Figure 2-1. While you can have a sysplex across multiple LPARs in a single CPC, the concept of aggregation only applies to summing the MSUs for two or more CPCs when all the qualification criteria (which we will discuss in 2.3.1, “Aggregation requirements” on page 17) are met.

Figure 2-1  Effect of decreasing unit price as capacity increases

Aggregation is best explained with an example. Let’s say you have two CPCs - a 300 MSU one and a 1500 MSU one. If the CPCs meet the aggregation criteria, you would pay for your software as if that were a single 1800 MSU CPC, rather than two separate CPCs. If you look at the chart in Figure 2-2 on page 7, you will see that the price for the first 300 MSUs (highlighted by “Y”) is significantly more than the price of an additional 300 MSU if you are already at the 1500 MSU mark, as shown by “X”.

1 Note that sysplex aggregation only applies to PSLC and Variable WLC products. This document does not address the Fixed WLC or One Time Charge products as they are unaffected by sysplex aggregation.

2 There are two types of sysplex - a “base” sysplex, and a Parallel Sysplex. A Parallel Sysplex is a base sysplex that also has one or more Coupling Facilities. Because you must have a Parallel Sysplex in order to qualify for sysplex aggregation, when we use the term “sysplex” in this document we are referring to a Parallel Sysplex unless specifically stated otherwise.
You can see from Figure 2-2 that the ability to aggregate these two CPCs could result in a significant reduction in how much you are paying for the software—to put it in perspective, if you had to pay for the 300 MSU CPC separately, z/OS on that CPC would cost you 82,500 RedBits (our mythical currency) per month, and 174,500 RedBits per month on the 1500 MSU CPC.

However, by being able to aggregate these two CPCs, you would instead pay for 1800 MSUs of z/OS, costing 188,000 RedBits per month. These theoretical prices, along with the average prices per MSU, are shown in Table 2-1 - you can see that the aggregated price (188,000 RedBits) is significantly less than the price if the CPCs cannot be aggregated (257,000 RedBits).

### Table 2-1 Pricing comparisons

<table>
<thead>
<tr>
<th>CPC</th>
<th>MSUs</th>
<th>Average price for each MSU</th>
<th>Monthly price (in RedBits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>300</td>
<td>275</td>
<td>82500</td>
</tr>
<tr>
<td>B</td>
<td>1500</td>
<td>116</td>
<td>174500</td>
</tr>
<tr>
<td>A and B separately</td>
<td></td>
<td></td>
<td>257000</td>
</tr>
<tr>
<td>A and B aggregated</td>
<td>1800</td>
<td>104</td>
<td>188000</td>
</tr>
</tbody>
</table>

### 2.1.1 Background

In this section we discuss the history of sysplex aggregation, and the most recent (at the time of writing) announcement dated November 2, 2004, together with the Sysplex Verification Package released in September 2005. In order to fully comprehend sysplex aggregation and its associated rules, and the reasons why we produced this document, it is important that you understand the information in this section. We will discuss the specific rules as we progress through this chapter, but we urge all readers, even those not interested in the technical details, to read this section.
Sysplex aggregation pricing was introduced when IBM announced Parallel Sysplex and the 9672 range of CPCs in 1994. The intent was to encourage customers to embrace and benefit from the new sysplex technology. Since then, this capability has proven to be the single most effective way for customers to reduce their IBM mainframe software costs.

One of the questions we sometimes get is why did IBM set a threshold that more than 50% of the workload on a CPC must be in a given sysplex in order to qualify for sysplex aggregation. There are two reasons for this:

- IBM wanted customers to genuinely benefit from Parallel Sysplex technology. If no threshold, or a very low one, had been set, people could set up a small “dummy” sysplex purely to qualify a CPC for aggregation. However doing this would not impart any of the benefits of sysplex on the production systems. You must remember that this criteria was set back in 1994, when sysplex was still quite new, and its advantages were not as widely acknowledged as they are today.

- The other reason is that it seemed a reasonable assumption that at least 50% of most installation’s mainframe utilization is accounted for by their production systems. As IBM was trying to encourage the use of sysplex with production systems, the threshold was set at 50%.

Setting the utilization at just 50% allows you to have multiple sysplexes, and even non-sysplexed LPARs, on the same CPCs as your production systems, while still potentially being able to avail of sysplex aggregation.

At the time of writing, it is eleven years since the introduction of sysplex aggregation. The use of sysplex is now common across many customers, and most customers with multiple CPCs could benefit from sysplex aggregation to some extent. As the uptake of sysplex has accelerated, and new products and exploiters have been announced, there have been a number of adjustments and clarifications of the criteria, with the most recent (at this time) being in November 2004. The most important points about this clarification (which is available on the Web at: http://www-306.ibm.com/common/ssi/ca/8/897/ENUS204-268/ENUS204-268.PDF) are:

- The list of qualifying enablement functions was updated to include more recent exploiters, such as Enhanced Catalog Sharing.

- IBM stated that “Additional or upgraded machines in existing sysplexes must satisfy these criteria before the associated machine MSUs will qualify for aggregated PSLC or aggregated WLC pricing”.

- IBM stated that it may periodically request verification that the aggregation criteria are still being met.

Communication with IBM regarding your aggregation qualification is discussed further in “Sysplex Verification Package” on page 23.

Because it may be some time since you originally qualified, and your configuration may have gone through many changes and upgrades in the interim, it is possible that one or more of your CPCs may no longer qualify for sysplex aggregation pricing. One of the objectives of this document is to help you identify this situation, and hopefully address it in a manner that not only lets that CPC re-qualify, but also removes the need for tweaking, adjusting, and re-balancing your workload and LPARs every time a CPC is upgraded or replaced.
2.1.2 Terminology

Unfortunately, the topic of software pricing is a complex one. You start off with an environment that is extremely flexible in how it can be configured (System z9/zSeries and Parallel Sysplex), and add in pricing options that are intended to encourage:

- The use of System z9 and zSeries as platforms for new applications.
- The use of sysplex to allow customers to benefit from the availability and systems management advantages it can provide.

The result is a topic that is complex and laden with terminology.

To start with, it is important to understand the difference between **licensing** and **pricing**.

- If you execute a piece of IBM System z9 or zSeries software on a CPC, you must have a license to do so. A license for an IBM monthly license charge product is specific to the product and a CPC with a particular serial number. The license for a Parallel Sysplex License Charging (PSLC) or a Variable Workload License Charging (VWLC) product is sold in terms of Millions of Service Units (MSUs). If you are executing a product on a 1500 MSU CPC, you must have a 1500 MSU license, specifying the serial number of that CPC.3
- The price for a product, that is, how much you pay IBM each month, depends on the pricing metric that is used for that product. Two well-known examples are PSLC and VWLC. Both of these metrics support sysplex aggregation.

To make this topic a little easier to discuss and comprehend, we have provided a number of sample configurations and some new terms. As we move from the more straightforward to some of the more complex configurations, we will use some terms to refer to groups of systems and CPCs. Some of these are familiar and others are completely new terms, introduced in this document specifically for brevity:

### Parallel Sysplex
This is the group of MVS-based systems that share a sysplex CDS, have a common time source, and are connected to one or more common Coupling Facilities.

### PrimaryPlex
To determine if a CPC qualifies for aggregation, you must determine the utilization of each MVS-based LPAR on that CPC. If the summed average utilization over the 40 hours of the prime shift of all the LPARs belonging to any given sysplex exceeds 50% of the MSUs used by all MVS-based LPARs on that CPC, that sysplex is determined to be the “PrimaryPlex” for that CPC.

While the PrimaryPlex does not need to be your production sysplex, it normally would be because your production sysplex is usually the largest consumer of capacity.

### PricingPlex
A PricingPlex is the group of CPCs that have the same PrimaryPlex. In Figure 2-4 on page 12, the PricingPlex consists of CPCs A and B because the LPARs in the Prod sysplex are consuming more than 50% of the MVS-based MSUs on each of those CPCs. Even though there is a member of the Prod sysplex running on CPC C, that CPC is not in the same PricingPlex as CPCs A and B. In fact, in this example, CPC C is not in any PricingPlex because no sysplex is consuming more than 50% of the used MVS-based MSUs on that CPC.

Note that it is your responsibility to notify IBM of which of your CPCs are in the same PricingPlex and therefore you wish to have billed as an aggregate. It is possible that you could have two CPCs that actually...

---

3 For products that support sub-capacity charging, the number of MSUs in the license is adjusted automatically every month based on the information provided in your SubCapacity Reporting Tool (SCRT) report.
qualify to be in the same PricingPlex, however you are not obtaining the pricing advantage - this could happen if you have not informed IBM of this relationship, or if you have not previously signed the required contracts and agreements.

Qualifying enablement function

In order to qualify for sysplex aggregation, there must be at least one “qualifying enablement function” that is used by every system in the sysplex that is running on any of the PricingPlex CPCs. A qualifying enablement function is one that uses one of a list of CF structures that have been specified by IBM. The reason for this list is that IBM is encouraging customers to use functions that are unique to a Parallel Sysplex environment. The list of qualifying enablement functions is contained in 2.3.1, “Aggregation requirements” on page 17.

2.2 Sample configurations

When discussing multi-dimension, complex topics like this, the use of examples can assist in understanding the principles being presented. Therefore, we have developed three configurations to illustrate the considerations. We start with the most simple - a single CPC - and move on through 2 stages, ending with a more complex configuration that would be similar to that seen in many of our larger customers today.

2.2.1 Sample configuration 1

So, let’s start with a nice simple example. Figure 2-3 shows a configuration with a single 1500 MSU CPC (CPC A). LPAR PRDA1 is executing z/OS, CICS®, and DB2® and is in a monoplex called Prod. LPAR DEVA1 is also executing z/OS, CICS, and DB2 and is in a monoplex called Dev. And LPAR TSTA1 is also executing z/OS, CICS, and DB2, and is in a monoplex called Test.

Figure 2-3   Single CPC with three monoplexes
In this example, the pricing metric is either PSLC or full-capacity VWLC. So, in this configuration, you would require a 1500 MSU license for each of z/OS, CICS, and DB2, and your monthly software bill would be for 1500 MSUs of capacity for each of those products. Because there is only a single CPC, the question of sysplex aggregation does not arise.

Table 2-2 shows the LPAR utilization information in tabular format, and Table 2-3 shows the information about product licenses.

### Table 2-2  CPC, LPAR, and sysplex utilization

<table>
<thead>
<tr>
<th>CPC</th>
<th>LPAR/System</th>
<th>Sysplex</th>
<th>Avg MSU consumption</th>
<th>% of MVS-based MSUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TSTA1</td>
<td>Test</td>
<td>135</td>
<td>11</td>
</tr>
<tr>
<td>A</td>
<td>DEVA1</td>
<td>Dev</td>
<td>287</td>
<td>21</td>
</tr>
<tr>
<td>A</td>
<td>PRDA1</td>
<td>Prod</td>
<td>928</td>
<td>68</td>
</tr>
</tbody>
</table>

### Table 2-3  Products licensed to CPC

<table>
<thead>
<tr>
<th>Product</th>
<th>CPC A</th>
</tr>
</thead>
<tbody>
<tr>
<td>CICS</td>
<td>Yes (1500 MSU)</td>
</tr>
<tr>
<td>DB2</td>
<td>Yes (1500 MSU)</td>
</tr>
<tr>
<td>z/OS</td>
<td>Yes (1500 MSU)</td>
</tr>
</tbody>
</table>

**Important:** In Figure 2-3 on page 10 and all subsequent examples, the notes in the picture have the following meanings:

- The percent number in the top box shows the percent of white space (that is, unused capacity) on each CPC.
- The number in the top right of each LPAR is the number of MSUs used by that LPAR, averaged over the prime 40 hours in the week.
- The number under the CPC name is the full capacity in MSUs of that CPC.
- The C, D, and Z in each LPAR indicate which software is executing in that LPAR as indicated by the key at the bottom of the figure.
- The name in each LPAR, PRDA1 for example, is the name of the system running in that LPAR. For simplicity, we assume that the name of each LPAR matches the name of the system running in the LPAR.

In this example, the pricing metric is either PSLC or full-capacity VWLC. So, in this configuration, you would require a 1500 MSU license for each of z/OS, CICS, and DB2, and your monthly software bill would be for 1500 MSUs of capacity for each of those products. Because there is only a single CPC, the question of sysplex aggregation does not arise.

Table 2-2 shows the LPAR utilization information in tabular format, and Table 2-3 shows the information about product licenses.

**Note:** In reality, we expect that most customers will take advantage of sub-capacity pricing to lower their software costs. However, even though sysplex aggregation and sub-capacity pricing are both concerned with CPC utilization and software pricing, they measure utilization in different ways - this is discussed in more detail in 2.3.2, “Comparing aggregation to subcapacity charging requirements” on page 20. For more information about sub-capacity pricing, refer to z/OS Planning for Workload License Charges, SA22-7506.

Therefore, in order to avoid confusion, all the examples in this chapter will be provided on the basis that software is being paid on a full capacity basis. Regardless, however, of whether software is being paid on the basis of full capacity or subcapacity, the utilizations used to determine sysplex aggregation qualification are the percent of used MVS-based MSUs that each LPAR is responsible for, not the percent of total CPC capacity.
2.2.2 Sample configuration 2

Due to business growth and increasing availability requirements, you decide to purchase two additional CPCs (CPCs B and C) at 1000 MSUs each, and to implement a Parallel Sysplex. This results in the configuration shown in Figure 2-4. We now have three multi-system sysplexes (Prod, Dev, and Test), each of which contains at least one member on each of the CPCs.

![Figure 2-4 Three CPCs with multi-system sysplexes](image)

So, in this case, you would need three licenses for each of z/OS, CICS, and DB2. Each product would require one 1500 MSU license and two 1000 MSU licenses. How much you actually pay each month depends on which sysplex members reside on each CPC, and the percent of the total used MVS-based MSUs that are consumed by each member - in other words, whether the CPC can be aggregated. Again, for simplicity, we will assume that you are paying on a full capacity basis.

And this is where the terminology starts to get a bit confusing. You will see references to “a processor being in a given Parallel Sysplex”, even though we all know that it is z/OS systems, rather than whole CPCs, that are in a Parallel Sysplex. In fact, the entity that is being referred to here is what we call a PricingPlex (see 2.1.2, “Terminology” on page 9 for a description of PricingPlex).

However, because software is aggregated at the CPC level, for software pricing discussions, we consider that a whole CPC belongs to a given PricingPlex. Which PricingPlex depends on the utilization of the LPARs on that CPC, and which sysplex each LPAR is a member of.

If we look at Figure 2-4, CPC A would be considered, from a software pricing perspective, to be in PricingPlex Prod, because the LPARs that are in the PROD sysplex (PRDA1 and PRDA2) are consuming more than 50% of the MSUs used by MVS-based LPARs on that CPC. Similarly, CPC B would also be considered to be in the Prod PricingPlex because LPAR PRDB1 is consuming more than 50% of the MSUs used by MVS-based LPARs on that CPC. However, even though CPC C contains members from all three sysplexes, from a software
pricing perspective, it is not considered to be in any PricingPlex because no sysplex is using more than 50% of the MSUs used by MVS-based LPARs on that CPC. This is shown in tabular format in Table 2-4.

**Table 2-4  CPC, LPAR, and sysplex utilization**

<table>
<thead>
<tr>
<th>CPC</th>
<th>LPAR/System</th>
<th>Sysplex</th>
<th>Avg MSU consumption</th>
<th>% of MVS-based MSUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TSTA1</td>
<td>Test</td>
<td>109</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>DEVA1</td>
<td>Dev</td>
<td>164</td>
<td>12</td>
</tr>
<tr>
<td>A</td>
<td>PRDA1</td>
<td>Prod</td>
<td>800</td>
<td>58</td>
</tr>
<tr>
<td>A</td>
<td>PRDA2</td>
<td>Prod</td>
<td>300</td>
<td>22</td>
</tr>
<tr>
<td>B</td>
<td>TSTB1</td>
<td>Test</td>
<td>164</td>
<td>18</td>
</tr>
<tr>
<td>B</td>
<td>DEVB1</td>
<td>Dev</td>
<td>200</td>
<td>22</td>
</tr>
<tr>
<td>B</td>
<td>PRDB1</td>
<td>Prod</td>
<td>546</td>
<td>60</td>
</tr>
<tr>
<td>PricingPlex Prod Total</td>
<td>-</td>
<td>-</td>
<td>2500 (based on full capacity charging)</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>TSTC1</td>
<td>Test</td>
<td>216</td>
<td>25</td>
</tr>
<tr>
<td>C</td>
<td>DEVC1</td>
<td>Dev</td>
<td>312</td>
<td>36</td>
</tr>
<tr>
<td>C</td>
<td>PRDC1</td>
<td>Prod</td>
<td>337</td>
<td>39</td>
</tr>
<tr>
<td>CPC C Total</td>
<td>-</td>
<td>-</td>
<td>1000 (based on full capacity charging)</td>
<td>-</td>
</tr>
</tbody>
</table>

Based on this configuration and these utilizations, CPCs A and B are considered to be in the same PricingPlex and therefore can be aggregated (we have shaded the rows pertaining to the Prod sysplex in green in the table above). CPC C is not considered to be in any PricingPlex (from a software pricing perspective), so its MSUs cannot be aggregated with any other CPC. The net of all this is that you would pay for 2500 MSU of capacity for the Prod PricingPlex (CPCs A and B) and for 1000 MSU for CPC C, assuming that all products (z/OS, CICS, and DB2) are executed on every CPC and that you are paying on a full capacity basis. This is shown in Table 2-4. Table 2-5 shows the software licences that would be required for each product, by CPC, and the number of MSUs that would be used for pricing.

**Table 2-5  Licences and MSUs for pricing by product and CPC**

<table>
<thead>
<tr>
<th>Licence</th>
<th>CICS</th>
<th>DB2</th>
<th>z/OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Licence</td>
<td>No. of MSUs for pricing</td>
<td>Licence</td>
<td>No. of MSUs for pricing</td>
</tr>
<tr>
<td>CPC A</td>
<td>1500</td>
<td>2500</td>
<td>1500</td>
</tr>
<tr>
<td>CPC B</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
<tr>
<td>CPC C</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
</tr>
</tbody>
</table>

Chapter 2. Sysplex aggregation 13
Hopefully these examples have helped you understand the basic concept of sysplex aggregation. You can see that as the number of LPARs, sysplexes, and CPCs increases, discussing what is or is not included in the aggregation can get quite complex. But don’t worry - the PLEXCALC tool that we will discuss in “PLEXCALC” on page 21 makes the process significantly easier by reporting the utilization of each LPAR and identifying which PricingPlex a CPC is considered to be a member of.

2.2.3 Sample configuration 3

Before moving on to discuss the precise aggregation qualification criteria, we will go through one final, more complex, example.

As shown in Figure 2-5 on page 15, you now have added two more 1250 MSU CPCs (business is going really well!). This configuration also reflects the following changes:

- While the Prod sysplex has members on all five CPCs, it only accounts for more than 50% of the MSUs used by MVS-based LPARs on two of the CPCs.
- A new sysplex for quality assurance has been set up. Because you are close to rolling a significant new application into production, the quality assurance LPARs are very busy, consuming more than 50% of the MSUs used by MVS-based LPARs on CPCs D and E. And because of the rate of growth of your business, your expansion plans, and your stringent availability requirements, it is expected that the number of MSUs used by these systems will be fairly constant into the foreseeable future.
- Some of the software products are only executed in a subset of the LPARs. You have decided to start using WebSphere MQ, but it is not in production yet. You will notice that it is being used in two of the Test LPARs, and four of the Dev LPARs.
- Two new LPARs (called VMLxx) have been set up for z/VM® and Linux® use. Note that in this example these LPARs are using general purpose CPs. If you were using IFLs for these LPARs, the utilization of those LPARs would be ignored in all discussions about sysplex aggregation.

All these changes are shown in the configuration in Figure 2-5 on page 15.
Figure 2-5  More complex aggregation example

The configuration is described in tabular format in Table 2-6.

Table 2-6  CPC, LPAR, and sysplex utilization

<table>
<thead>
<tr>
<th>CPC</th>
<th>LPAR/System</th>
<th>Sysplex</th>
<th>Avg MSU consumption</th>
<th>% of MVS-based MSUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TSTA1</td>
<td>Test</td>
<td>109</td>
<td>8</td>
</tr>
<tr>
<td>A</td>
<td>DEVA1</td>
<td>Dev</td>
<td>164</td>
<td>12</td>
</tr>
<tr>
<td>A</td>
<td>PRDA1</td>
<td>Prod</td>
<td>800</td>
<td>58</td>
</tr>
<tr>
<td>A</td>
<td>PRDA2</td>
<td>Prod</td>
<td>300</td>
<td>22</td>
</tr>
<tr>
<td>B</td>
<td>TSTB1</td>
<td>Test</td>
<td>164</td>
<td>18</td>
</tr>
<tr>
<td>B</td>
<td>DEVB1</td>
<td>Dev</td>
<td>200</td>
<td>22</td>
</tr>
<tr>
<td>B</td>
<td>PRDB1</td>
<td>Prod</td>
<td>546</td>
<td>60</td>
</tr>
<tr>
<td>PricingPlex Prod Total</td>
<td>-</td>
<td>-</td>
<td>2500 (based on full capacity pricing)</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>TSTC1</td>
<td>Test</td>
<td>216</td>
<td>25</td>
</tr>
<tr>
<td>C</td>
<td>DEVC1</td>
<td>Dev</td>
<td>312</td>
<td>36</td>
</tr>
<tr>
<td>C</td>
<td>PRDC1</td>
<td>Prod</td>
<td>337</td>
<td>39</td>
</tr>
</tbody>
</table>
What interesting observations can we make based on this configuration?

- This configuration consists of two PricingPlexes: CPCs A and B and CPCs D and E (because the QAPLEX systems are using more than 50% of the used MVS-based MSUs on those CPCs), and one “standalone” CPC - CPC C.

- The utilization of the two QAPLEX systems are just over the 50% required to be able to aggregate those two CPCs. It is important that effective workload balancing is implemented between those two systems to ensure they continue to meet the criteria. If the workload balance were to change so that more work ran in QAD1, the utilization of QAE1 could drop below 50% of that CPC, meaning that you would no longer be able to aggregate those two CPCs.

- Considering that the PRDD1 LPAR which is in the Prod sysplex only accounts for 17% of CPC D, it is unlikely that CPC D will ever be in the same PricingPlex as CPC A. However, if you were to move the PRDD1 LPAR, or its workload, onto CPC C, and move some or all of the TSTC1 workload onto CPC D, it might be possible to aggregate CPC C into the same PricingPlex as CPCs A and B.

If you could achieve this, you would be left with two PricingPlexes rather than three, and in general, the fewer PricingPlexes you have, the lower your software costs should be.

- Remember that the ability to aggregate for software pricing happens at the CPC level. So, if we look at WebSphere MQ, we see that it only executes in the Test and Dev LPARs. However, even though it is the Prod and QA LPARs that qualify the CPCs for aggregation, we are still able to aggregate the MQ licences on CPCs D and E. Because CPC C is not in any PricingPlex, you cannot aggregate the MQ licences on that CPC. And in the Prod
Chapter 2. Sysplex aggregation

PricingPlex, MQ is again only executing on one CPC, so there is no question of aggregation in that case. All this information is shown in Table 2-7.

Table 2-7 Licences and MSUs for pricing by product and CPC

<table>
<thead>
<tr>
<th>Licences and MSUs for pricing</th>
<th>CPC A</th>
<th>CPC B</th>
<th>CPC C</th>
<th>CPC D</th>
<th>CPC E</th>
</tr>
</thead>
<tbody>
<tr>
<td>LICENCE</td>
<td>NO. OF MSUs FOR PRICING</td>
<td>LICENCE</td>
<td>NO. OF MSUs FOR PRICING</td>
<td>LICENCE</td>
<td>NO. OF MSUs FOR PRICING</td>
</tr>
<tr>
<td>CICS</td>
<td>WebSphere MQ</td>
<td>DB2</td>
<td>z/OS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1500</td>
<td>0</td>
<td>1000</td>
<td>1500</td>
<td>1500</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td>1000</td>
<td></td>
</tr>
<tr>
<td>1250</td>
<td>1250</td>
<td>1250</td>
<td>1250</td>
<td>1250</td>
<td></td>
</tr>
</tbody>
</table>

a. All the above numbers are based on paying on the basis of full capacity. Given that VM/Linux is using a significant portion of CPCs D and E, and that WebSphere MQ is only executing in a subset of LPARs, subcapacity charging should definitely be used in this case to reduce software costs.

Now that you understand the sysplex aggregation concepts and benefits, we move on to describe the current aggregation rules in detail.

2.3 Aggregation rules and criteria

There are a number of rules and criteria that IBM have specified that determine whether a given CPC is eligible for sysplex aggregation. A very important point is that your systems must always conform to these rules—it is not sufficient that they just conform when the CPC is originally installed. The restatement of the sysplex aggregation criteria specifically states “You must also notify IBM when you believe that your systems no longer qualify for pricing under this criteria”. The complete restatement announcement is available on the Web at:


2.3.1 Aggregation requirements

The sysplex aggregation criteria (as we understand them at the time of writing) break out into hardware, software, and operational requirements, and are as follows:

► Sysplex aggregation hardware requirements:
  - All CPCs in the Parallel Sysplex must have a common time source. This means they must all be connected to the same Sysplex Timer® or STP network.
  - Because the Sysplex Timer and STP support connection to up to 24 CPCs, it is very likely that all your CPCs are already connected to the same common time source.
  - All systems in the Parallel Sysplex must be connected to at least one common Coupling Facility (CF).
  - However from a technical perspective, we always recommend that you have at least two CFs, and that all the CFs in a given sysplex are accessible to all members of that sysplex.
  - You can use either shared or dedicated CPs in your z/OS LPARs.
Sysplex aggregation software requirements:
- MVS/ESA™ 5.2.2, OS/390®, or z/OS must be executing in all LPARs that are to be considered for aggregation.
- All z/OS, z/OS.e, and OS/390 images that comprise the Parallel Sysplex environment must have at least one common systems enablement function activated to use the CF across all images in the PricingPlex. Eligible systems enablement functions are:
  - Application data sharing, including:
    - IMS TM: with IMS DB or DB2,
    - CICS: with IMS DB, DB2, or VSAM/RLS
    - TSO and DB2 data sharing
    - An eligible Independent Software Vendor’s Data Base from Group C of the PSLC Exhibit.
  - WebSphere MQ shared message queues
  - HSM common recall queue
  - Enhanced Catalog Sharing
  - GRS Star
  - JES2 Checkpoint in the Coupling Facility
  - RACF® database caching
  - SmartBatch multisystem processing
  - VTAM® Generic Resources
  - VTAM MultINode Persistent Sessions
  - Automated tape sharing and switching (prior to z/OS 1.2)
  - System Logger SYSLOG (OPERLOG)
  - System Logger LOGREC
  - System Logger Resource Recovery Services

Even though there are other CF exploiters, like XCF for example, it is only one or more items in the list above that will meet the requirement for use of a systems enablement function.

Sysplex aggregation operational requirements:
- The z/OS, z/OS.e, and OS/390 images participating in the above systems enablement function(s), in the same sysplex, must account for at least 50% of the total MSUs consumed by MVS-based systems on each eligible CPC.

Note: This is not 50% of the total capacity the CPC, but 50% of the used capacity. Further, it is 50% of the capacity used by MVS-based systems (including those running under VM), so it excludes capacity used by VM itself and its other guests, Coupling Facility, or Linux LPARs, as well as any capacity used on any special purpose engines (zAAPs, for example). This is a very important point that is often misunderstood.

- To determine eligibility for the 50% rule, the following calculation is applied for each CPC:
  - Sum the utilization of all the systems in each sysplex on the CPC for the 8-hour prime shift, for the 5 working days in the week (a total of 40 hours).
  - Sum the utilization for all MVS-based LPARs on the CPC for the same time period.
  - Divide the utilization of the largest sysplex on this CPC by the total MVS-based utilization on this CPC.
  - In order for this CPC to be eligible for aggregation, the PrimaryPlex utilization must be using more than 50% of the total for every week in the month.
- The PLEXCALC tool provided by IBM (and discussed further in “PLEXCALC” on page 21) does these calculations for you, using SMF Type 70 records from every MVS-based LPAR (or VM guest) in the CPC.
- If it happens that exactly 50% of the used capacity on a CPC is related to one sysplex, and the other 50% is related to a different sysplex, you can choose which sysplex you want the CPC to belong to from a pricing perspective.
- The prime shift is defined as the 8-consecutive-hour period covering your normal working day. It defaults to 9:00 to 17:00, however if your business differs from this standard model, you can make a case to IBM explaining why some alternate consecutive period actually better reflects your prime shift. You also have the ability to exclude one or two hours for the lunch period - see “Running the PLEXCALC tool” on page 22 for more information.

If you have LPARs with different GMT offsets (regardless of which sysplex they are a member of), refer to Appendix A, “Handling systems with different time offsets” on page 97.

- As we said previously, sysplex aggregation is based on the average utilization over the 8 hour prime shift—this is different to sub-capacity charging which is based on the highest rolling 4-hour average at any time over every day in the month. This is discussed further in 2.3.2, “Comparing aggregation to subcapacity charging requirements” on page 20, but we just want to mention it here to remind you that there is a difference between these two measurements.

- If you run any MVS-based guests under VM, the CPU consumption of the MVS-based guests must be taken into account. There can be situations when the CPU time reported in the SMF Type 70 records for z/OS systems running under VM can be over-stated. When CPU utilization is very high, it is possible that the physical CP can be stolen from VM while the z/OS guest is dispatched on the logical CP. In this case, the CPU time in the Type 70 records includes the time that the logical CPs are not dispatched on a physical CP.

While the use of the Type 70 records may slightly overestimate the CPU time for the z/OS guests in some cases, given the information accessible to RMF™ and the Sysplex Calculator tool (described in “PLEXCALC” on page 21), this is the best solution currently available. The alternate would be to take the utilization of the whole VM LPAR and treat that as if it was an MVS-based LPAR as is currently done with SCRT. However, that would more than likely result in an even larger over-estimate.

WLM in z/OS 1.7 has been enhanced to provide more accurate utilization information for z/OS guests running under z/VM.

- If you are using zAAPs, any work dispatched on the zAAP is not included in any of the calculations.

One thing to be aware of if you are planning to start exploiting zAAPs, is the impact they could have on your sysplex aggregation qualification.

zAAPs are nice for two reasons:

- They deliver cheaper “MIPS” than standard CPs. So, in hardware terms, it costs less to process the same amount of Java™ work using zAAPs than if you were to use CPs.
- When calculating the capacity used by your z/OS images for charging purposes, IBM does not take into account the capacity used on the zAAPs. Especially if you are using sub-capacity pricing, this can result in a sizeable reduction in your software bill if you can move workload from the CPs over to the zAAPs.
However, you need to consider what impact the zAAPs will have on your sysplex aggregation qualification. You should run the zAAP projection tool in every LPAR to size how much work you will be moving from the CPs to the zAAPs. Then apply that to the utilization of each LPAR to estimate the “after” utilization of each LPAR. Because it is possible that moving the work to the zAAPs could potentially change the workload profile so much that the CPC no longer is in the same PricingPlex, you should also work with your IBM representative to project the impact of the new utilizations on your software bills.

The use of zAAPs and the mechanisms to control how much work is routed to them is discussed in more detail in 4.3, “Scenario 2 - Spending in order to save” on page 50, and 5.4.5, “zAAPs” on page 92.

There are numerous pricing options for z/OS and selected subsystems - z/OS.e and New Application Licence Charging (NALC) are two examples. While these options have an impact on your monthly software bill, they do not have any impact on your calculations for sysplex aggregation. For example, if you have a system executing z/OS.e that is part of a sysplex, the utilization of that system would be included in all the sysplex aggregation calculations just as any z/OS or OS/390 system would. Equally, a CPC on which the pricing metric is NALC, can also be included in the sysplex aggregation criteria.

These criteria are subject to amendments or clarifications from time to time, so you should always check the Parallel Sysplex Aggregation Web site for the latest information. The URL for the Web site is:


2.3.2 Comparing aggregation to subcapacity charging requirements

Because sysplex aggregation and sub-capacity charging (and the associated SCRT tool) are both associated with how much you pay for your software, it is easy to confuse the requirements and criteria for these two mechanisms.

Even though these two things are related, they have different objectives and measurements. The objective of sysplex aggregation qualification is to encourage the use of sysplex capabilities in production environments. The measurements are based on the actual utilization during your prime business hours.

Sub-capacity pricing, on the other hand, is intended to provide a mechanism for you to pay for software based more on how the software is used (as measured by an averaged utilization of the LPARs the software executes in), rather than on the total capacity of the CPC it is executing on. Sub-capacity pricing doesn’t care what sysplex a product is executing in. And to protect you from short term workload spikes, the sub-capacity is based on the highest rolling 4-hour average of each LPAR over a month. Every day of the week is considered, as is every hour of the day.

2.4 Other considerations

2.4.1 What is “normal mode of operation”

The wording in the sysplex aggregation criteria specifies that the environment that is used to validate qualification must be the “normal mode of operation”. That means that you are not meant to set up a sysplex just for the validation measurement, and then revert to a different configuration for normal operations.
“Normal operation” also means that you are not expected to measure during a period when you are suffering from a disaster or conducting a disaster recovery test. IBM understands that the configuration you may be running during a disaster or DR test may differ from that used in normal operations, so it is not expected that you have to provide a validation measurement for this period.

Similarly, if you have a temporary hardware failure, such as a CPC outage, IBM understands that moving LPARs during the outage period may result in some CPCs not meeting the qualification criteria during this window, so once again you are not expected to provide a measurement for this time.

Normal operation would also not apply to any period when you are using Capacity Backup Upgrade (CBU) or On Off Capacity on Demand (OOCoD) engines. CBU would normally only be used for a short time during a Disaster Recovery test, or else in case of a disaster. OOCoD would similarly only be used during times of peak processing. For both these situations, you would not be expected to provide a report for these times.

Finally, normal mode of operation also covers your workloads. If there are times when your production system usage spikes very much higher than its normal profile, such as during year-end processing for example, this would not be a time that should be used to verify qualification for sysplex aggregation.

### 2.4.2 Validation requirements

**PLEXCALC**

The Sysplex Calculator tool (referred to in this document as PLEXCALC) is a no-charge tool provided on an as-is basis that can be downloaded from the following Web site:


This tool calculates the average utilization of each MVS-based system that it has SMF records (Type 70s) for. In order to get an accurate result, you must provide it with the Type 70 records for every MVS-based system on every CPC in the installation, including test systems, development systems, system programmer sandbox systems, and those running under VM. The reason PLEXCALC needs the records from every system is that the field it uses to calculate the LPAR utilization is only provided for the system that created the record - the SMF records from any other system will not contain the required information. If PLEXCALC discovers LPARs that are using CPs (as opposed to ICFs or IFLs) for which is has not been provided with any Type 70 records, it will issue a warning message.

The tool is provided on an as-is basis, and is not intended to conclusively determine if your CPCs qualify for aggregation—that determination can only be made by IBM. However, the PLEXCALC report helps IBM make that determination, and helps you ensure that your systems continue to meet the 50% threshold qualification criteria.

A sample output from the PLEXCALC tool is shown in Figure 2-6 on page 22.
The PLEXCALC report will help you quickly identify the PrimaryPlex on each CPC. It also helps you see how much of a buffer you have above the 50% threshold. For example, if the utilization of the PrimaryPlex is just 51.3%, you may want to consider adding more work to those systems to decrease the likelihood of that CPC dropping out of this PricingPlex. If the CPC does not qualify for aggregation, or if it is a member of a PricingPlex other than the one you intended, you can easily identify how large the shortfall is (in terms of MSUs).

Even though you do not currently have to send the output from PLEXCALC to IBM on a regular basis, you should still run it periodically to identify if you have CPCs that are in danger of falling outside the aggregation qualification criteria so you can make adjustments as necessary.

**Running the PLEXCALC tool**

**Tip:** The provided PLEXCALC job includes the object deck for the PLEXCALC program. A number of users have encountered problems with this when they upload the file using their 3270 emulation product's file transfer function, even when they specified that the file should be sent in Binary format. To avoid such problems, we recommend using FTP with the BIN option to send the PLEXCALC file to your z/OS system.
The PLEXCALC tool takes as input SMF Type 70 records from every MVS-based system in your installation. If data is missing from any system, an alert message will be included in the report created by PLEXCALC. Although you only need to give IBM a PLEXCALC report for one week, we recommend that you pass a full month’s worth of SMF records to PLEXCALC; this will allow you to check that the PrimaryPlex on each CPC is consistent from week to week. The records do not need to be sorted prior to passing to PLEXCALC.

If you just provide the SMF records, and don’t override any of the PLEXCALC parameters, it will assume that the prime hours for all systems is 09:00 to 17:00 local time. If you have systems with different local time offsets, refer to “Appendix A, “Handling systems with different time offsets” on page 97”.

By default, PLEXCALC will assume that you want to use the 8-hour period starting at 9:00 as your peak period. If you have a sound business reason, you can use a start time other than 09:00.

Similarly, PLEXCALC assumes that you want to use the lunch period as part of your prime period. However, if you wish, you can specify one or two consecutive hours to be your lunch period and therefore excluded from the calculations. For example, if you specify `PARM=(/9,,CUSTOMER NAME)`, PLEXCALC will determine the prime shift to be 09:00 to 16:59. If you specify `PARM=(/9(12-13),,CUSTOMER NAME)`, you are telling PLEXCALC that you have a one hour lunch period starting at 12:00. In this case, PLEXCALC will determine your prime shift to be 09:00 to 11:59, and 13:00 to 17:59.

If you are already availing of sub-capacity charging, you must collect the SMF Type 70 and 89 records from all your systems for the SCRT tool anyway. Therefore, if you want to run PLEXCALC looking back over the last few months, you can use the data that you are already collecting and archiving for SCRT.

**PLEXCALC report**

The output from PLEXCALC consists of a data set, in CSV format, that you can then download and view using your favorite spreadsheet program. The output may consist of a number of reports, one for each week of data that you submit.

The report starts with a list of all the CPCs and the LPARs on each CPC. The next section (the lines in the PLEXCALC report starting with “Based on IBM’s Parallel Sysplex Aggregation criteria...”) reports the PrimaryPlex (if any) for each CPC. The next section contains one column per LPAR that Type 70 records were provided for. This section contains the LPAR name, and the system name and SMF ID, the percent of MVS-based MSUs on this CPC that this LPAR consumed over the prime shift for the week, the average number of MSUs consumed by this LPAR over the prime shift for the week, followed by an hour-by-hour breakdown of the number of MSUs used by each LPAR. Finally, there is a list of any footnotes in the report - for example, in Figure 2-6 on page 22, footnote 1 is used to identify the LPARs that were used as the base time that all other LPARs on that CPC were set to.

**Sysplex Verification Package**

To help you confirm to IBM that your configuration conforms to the sysplex aggregation requirements, IBM has provided a package known as the Sysplex Verification Package. The package can be downloaded from the following URL:

The package should be completed and submitted to IBM in all of the following circumstances:

- You add an additional machine to an existing PricingPlex.
  
  When you install an additional CPC that you wish to include in an existing PricingPlex, it is your responsibility to inform IBM if you believe that CPC will qualify for sysplex aggregation. Within 60 days of the first of the month following when you submit that notification, you must send IBM a completed Sysplex Verification Package to verify that the CPC does in fact qualify for sysplex aggregation. This allows time for you to move LPARs from the old CPCs over to the new one and to reach its normal operating mode.

- You create a new PricingPlex, or

- You have an anniversary checkpoint for an overlay agreement, such as an Enterprise License Agreement (known as Enterprise Software and Services Offering in some regions) or an Open Infrastructure Offering.

In all these cases, you must send the following:

- A completed questionnaire (the questionnaire is included in the package that you download).
- A PLEXCALC report for one week—you can select which week you submit.
- An RMF PostProcessor CF Interval report for the same period as the PLEXCALC report. Note that in order to get the RMF CF report you must be running RMF Monitor III. If you do not run Monitor III today, you can find information in the Package about how to run Monitor III with minimal overhead.

2.4.3 **Sysplex members with different local times**

It is possible to have systems in a sysplex or on the same CPC that do not have the same local time. In such situations, it can be quite complex to determine which time range should be used in the PLEXCALC tool. Rather than taking up space here with a discussion that only applies to a small minority of installations, we discuss this situation in Appendix A, “Handling systems with different time offsets” on page 97.

2.5 **Getting started**

Now that you understand the benefits and requirements of sysplex aggregation, where do you go from here? Regardless of whether you are already benefitting from sysplex aggregation (and want to continue doing so) or you want to expand the number of CPCs in your PricingPlex, the first step is to confirm how compliant all your CPCs are today.

To get started, gather at least one week’s worth of SMF Type 70 records from every MVS-based system in your installation. It is very important that you don’t omit any systems from this analysis as the reports you will be reviewing express the utilization of each system as a percent of the total utilization. If you omit some systems, the resulting report will not provide a representative picture of where you stand.

Having collected all the SMF records, you should input all that data to a single run of the PLEXCALC tool, ensuring that the values used for working days (default is Monday to Friday) and prime hours (default is 09:00 to 17:00) are appropriate for your environment.

The resulting PLEXCALC report will tell you which sysplex, if any, is the PrimaryPlex for each CPC for each full week covered by the input data. Review this report to ensure that the contents conform to your expectations.
There is an additional step that we recommend that must be carried out manually. We believe that it is valuable information to know what percentage of your total used MSUs (across all CPCs) are attributable to each sysplex. For example, if you found that the PrimaryPlex for some of your CPCs was not the sysplex that you expected it to be, your next step should be to determine if that sysplex is consuming enough capacity to be able to account for greater than 50% of those CPCs if you were to do some workload rebalancing.

PLEXCALC reports the average MSUs for each system, however it doesn’t report the total MSUs for each sysplex. Therefore we recommend that you get the total MSUs across all MVS-based systems, get the total MSUs used by each sysplex, and determine the percentage of total MSUs that each sysplex accounts for.

This exercise will help you find out:
- Do you have a sysplex that consumes enough MSUs to potentially use more than 50% of each CPC (assuming you had some way to evenly distribute the work across all systems)?
- If you have multiple PricingPlexes, do the PrimaryPlexes for each one consume sufficient capacity on the CPCs in the PricingPlex that you are comfortably above the 50% threshold?
- Do you have any sysplex that is large enough to allow you to achieve the level of aggregation you desire? If not, would the utilization, if rebalanced, allow you to at least decrease the number of PricingPlexes from the current number?

### 2.6 Challenges to sysplex aggregation implementation

Some of the most common challenges customers face when trying to maintain sysplex aggregation qualification, or trying to get a CPC qualified for the first time are:

- If you have not implemented data sharing, the granularity of units of work that can be moved between LPARs tends to be quite large. For example, you cannot move DB2 batch jobs to another system if that system is not a member of the same data sharing group as where the jobs run now. Rather, you would need to move the whole DB2 subsystem, together with its attendant CICS regions, batch jobs, TSO users, and so on. This might increase the load on the target system so that it now qualifies, but decrease the load so much on the previous CPC that it no longer qualifies!

To use an analogy, work that does not support data sharing is like large jagged rocks. If you have to place these in a number of glass containers, there will be a lot of unused space, and it can take some time to get the right mix of rocks in the containers so that each one is equally full. On the other hand, work that supports data sharing is more like sand - instead of moving a whole DB2 subsystem and all the associated work, you can move work at the transaction level. It is much easier to evenly fill your glass containers with sand than with the rocks. And when you upgrade your glass container to a 1500 MSU one, rebalancing the contents is a trivial task.

The whole topic of the role and benefits of data sharing is covered in Chapter 3, “Parallel Sysplex options and considerations” on page 27.

- You may have implemented data sharing, but have not done anything to dynamically balance the incoming work across your systems. So, even though two systems are in the same DB2 data sharing group, all the DB2 batch jobs still run in just one of the systems.

An example of this type of configuration is shown in Figure 2-7 on page 26. You can see that the production sysplex is consuming the bulk of CPC A, just under half of CPC B, and under a third of CPC C. If you could rebalance the work across those three CPCs, two, and maybe even all three, CPCs would qualify to be in the same PricingPlex.
This is something that we see quite frequently, and thankfully there are many functions available from IBM to help you address this. The whole topic of workload balancing and workload routing is discussed in Chapter 5, “Workload balancing and sysplex aggregation” on page 65.

- You may have data centers that are some distance apart. You would like to place them in the same sysplex, to get the benefits of sysplex aggregation, however the distance between the sites would significantly affect the response time for any I/Os and CF requests that must be sent between sites.

  The topic of multi-site sysplexes, together with the advantages and challenges they provide, is discussed in 3.1.3, “Multi-site sysplex” on page 30.

- You do not have any single sysplex that accounts for more than 50% of your used MVS-based MSUs.

  In this case, there are two things you can do. Whether you select one or the other, or both, depends on how close to the 50% you are. If you are very close to the 50%, you probably qualified in the past, but workload changes have caused a CPC to no longer qualify. In this case, you may be able to move back above the threshold by moving workloads from other systems into the PrimaryPlex systems. This work may come from other systems in the sysplex, or it could even come from a different sysplex. This is discussed in 5.5, “Redistributing workloads and processing” on page 95.

  If you are so far away from the 50% threshold that workload changes will not help you achieve qualification, then you must consider the benefits, drawbacks, and costs of merging systems or sysplexes into the largest sysplex. For more information on this, refer to 3.4, “To merge or not - a sample scenario” on page 40.

- You have different clients and for any number of reasons cannot place them in the same sysplex, and none of your sysplexes are large enough to qualify for aggregation.

  In this case, there is very little you can do from a technical perspective. If you cannot address the reasons why the systems cannot reside in the same sysplex, you may find it difficult to identify a configuration that will qualify for aggregation.

While this list does not cover every possible reason why you may have difficulty meeting the aggregation criteria, it does cover the most common ones. The remainder of this document contains information to help you address these challenges in the most effective manner, bearing in mind that different installations have different business requirements.
Parallel Sysplex options and considerations

The most basic requirement for sysplex aggregation is that you must have a sysplex that spans multiple CPCs. However there are many ways a sysplex can be configured.

This chapter discusses two different ways that a Parallel Sysplex can be implemented, and describes the benefits and drawbacks of each type. It also discusses how the attributes of each type of sysplex impact your continuing qualification for sysplex aggregation in the light of future workload, hardware, and configuration changes.
3.1 Different sysplex configuration types

Customers have implemented their Parallel Sysplexes in many different ways, driven by their availability, capacity, systems management, and budgetary requirements. Some exploit IBM's Parallel Sysplex technology to the maximum, including multi-site sysplexes, data sharing with dynamic workload balancing, and extensive resource sharing. Others use it only for limited resource sharing. However, to get the most benefits and the optimum value from this unique technology, you should be fully exploiting the Parallel Sysplex capabilities that apply to you. Although there is definitely an investment, in terms of people’s time to fully exploit the features, the payback can be unmatched levels of service and availability.

In this chapter we discuss and contrast two completely different, but both quite common, examples of a Parallel Sysplex implementation. In the IBM RedBook *Merging Systems into a Sysplex*, SG24-6818, three different types of sysplexes are described. The terms “BronzePlex” and “PlatinumPlex” were introduced in that document, to describe two of the different types of sysplexes, and we will be using them again in this RedPaper. We do not discuss a “GoldPlex” - in the context of the discussion here (sysplex aggregation), a GoldPlex is not significantly different to a BronzePlex.

We start the discussion about different sysplex types by describing what we mean by a BronzePlex and a PlatinumPlex. We also touch briefly on the topic of multi-site sysplex (which could be either a PlatinumPlex or a BronzePlex) and the considerations relating to sysplex aggregation.

3.1.1 BronzePlex

If we look at Parallel Sysplex exploitation, at one end of the scale, you can take completely unrelated systems and place them in the same sysplex, attempting to share as little as possible. In fact, you may even try to manage the systems as if they are not even in the same sysplex. This would typically be done solely to get the financial benefits of sysplex aggregation charging. In this case, there will be practically no sharing between the systems in the sysplex, and therefore limited technical benefit from being in the same sysplex. We use the term “BronzePlex” to describe this type of sysplex.

However, the original intent of sysplex was to provide the ability to join a number of systems and have them appear and behave as one logical entity - single system image in other words. Specifically, sysplex was not designed to support subplexes—that is, subsets of the sysplex that have nothing in common with the other members of the sysplex except that they share the same sysplex Couple Data Sets and common time source. For example, the design point was not to have development and production systems in the same sysplex. While some products do support the idea of subplexes (DFSMShsm™, for example), others do not (GRS, for example). For these reasons, we strongly advocate that installations minimize the use of BronzePlexes, or use them as an intermediate step on the path to realizing the full benefits of Parallel Sysplexes.

More importantly, from a sysplex aggregation perspective, while a BronzePlex may help you qualify for sysplex aggregation with your current configuration, it does nothing to help you maintain qualification. As the workload profile changes and the hardware configuration is modified (CPC upgrades and processor consolidation) you will have to:

- Constantly monitor the utilization of the systems against the sysplex aggregation qualification criteria.
- If the workload on a CPC no longer meets the qualification criteria, you need to identify a workload elsewhere in the sysplex that is large enough to bring the failing system back
into qualification, while at the same time ensuring that moving that workload does not cause the donor system to now fail to qualify.

- Manually move that work from one system to another, making whatever JCL, process, and operational changes may be required to cater for the new location.

What you really want is a configuration where the work will automatically flow around the systems in the sysplex based on available capacity, allowing sysplex aggregation qualification to be maintained transparently by fine tuning system controls rather than disruptively moving workloads or whole systems around the CPCs. This brings us to the other type of sysplex, the PlatinumPlex.

### 3.1.2 PlatinumPlex

The second configuration we have considered is where the objective is to maximize the benefits of sysplex and sysplex aggregation pricing. In this case, after the merge is complete, the systems in the sysplex will be sharing *everything* with all the other members of the sysplex. For example, there will be:

- A single logical pool of user data, so work can access its data from any system in the sysplex.
- A single system image to the user, so they just log on to “CICS” or “TSO”, without having to know how many regions or systems there are that could potentially serve that request. And if one of more of the servers are down, there is no change in the user’s procedures as long as at least one server remains available in the sysplex.
- A single security environment to minimize administration tasks, and to ensure that the same security rules are enforced regardless of which system is being used to access the data.
- A single JES environment, so batch jobs can be submitted anywhere, and are potentially able to run anywhere in the sysplex.
- A single automation focal point and set of automation rules, to enable a sysplex-wide view of events, and to allow the automation processes to be designed once and run everywhere.
- A single shared sysres between all the systems, to minimize software management effort and ensure consistency across all systems. You may decide to actually have a number of physical sysres to avoid a single point of failure, but they would all be replicas of each other.
- Basically just one of everything in the sysplex.

This configuration provides the maximum in systems management benefits and efficiency, so we have called it the “PlatinumPlex”. It is shown diagramatically in Figure 3-1 on page 30.
Although this is really the optimum for a sysplex, and we recommend that you have a plan in place to get there, the main differentiator between a PlatinumPlex and a BronzePlex is that in a PlatinumPlex all user data can be accessed from all members of a sysplex, and any work can run in any system, meaning that moving work from one system to another can be done at the transaction- rather than the database manager-level. You can have a PlatinumPlex that has not yet implemented all the available options—it is possible, for example, to have a PlatinumPlex with two JES2 MASes. However, in a PlatinumPlex, it is not feasible to have more than one RACFplex, more than one SMSplex, more than one HSMplex, or more than one tape management systems, for example. Remember that one of the objectives of sysplex is optimized systems management; having multiple security and DASD environments to maintain does not deliver that level of optimization. It also does not provide the flexibility to run your work on any system in the sysplex.

Many people think that implementing a BronzePlex is a trivial exercise, compared to creating a PlatinumPlex, because little is shared between the systems. However, this can turn out not to be the case. From a technical, and even a financial, perspective, if you are going to put time and effort into creating a BronzePlex, we would prefer to see you invest some additional effort up front to achieve a PlatinumPlex, resulting in reduced cost and effort over the medium and longer terms.

### 3.1.3 Multi-site sysplex

The last few years have seen a dramatic increase in the number of installations running multiple data centers. The predominant reasons for this are for improved availability, for disaster recovery purposes, or to meet government regulations. The technology supporting
the ability to span a sysplex across sites has also improved, with the maximum supported
distance at the time of writing being up to 100 km., as shown in Figure 3-2.

Given the existence of multiple sites in relatively close proximity, there are a number of
reasons why you might want to have a single sysplex that spans those sites:

- You want to spread your critical applications across both sites, to allow for speedy
  recovery of applications following a site failure.
- You plan to implement GDPS/PPRC, and a multi-site sysplex is a requisite for this offering.
- You would like to be able to have CPCs in both sites in a single PricingPlex.

If you are considering a multi-site sysplex for any of these reasons, you really need to
consider:

- Will you run production in one or both sites? Remember that from a sysplex aggregation
  perspective, simply having a sysplex that spans two sites is not sufficient to ensure that
  you can aggregate all CPCs. The qualification rules that apply to the CPCs in the primary
  site apply equally to those in the secondary site (must be using a qualifying exploiter, must
  have the same PrimaryPlex, and so on).
- Do you plan on doing data sharing across the two sites?
- What is the distance between the sites? What impact will the distance have on your
  workloads?
- What is the target sysplex type? Will you have a PlatinumPlex spanning both sites or a
  BronzePlex? The benefits and drawbacks of each sysplex type apply equally regardless of
  whether the sysplex is contained in a single site or spans two sites.

A frequent question we receive is “what is the maximum distance you can do data sharing
over?” Unfortunately, the answer is “it depends”. With some effort, you can estimate the
impact the distance between the sites will have on your DASD and CF response times. You
must then project that impact to your applications to see if they can withstand the increased
response times.

A very important point is that the distance will impact you even if you are not doing data
sharing across the sites. If you have one sysplex in Site1 and another sysplex in Site2, and

<table>
<thead>
<tr>
<th>Site 1</th>
<th>Intermediate Site w/amplifiers</th>
<th>Site 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPC</td>
<td>40 km max</td>
<td>CPC</td>
</tr>
<tr>
<td>ETR</td>
<td>2 ETR link / CPC</td>
<td>ETR</td>
</tr>
<tr>
<td></td>
<td>2 CLO Links</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1 ETR link / CPC</td>
<td>1 ETR link / CPC</td>
</tr>
<tr>
<td></td>
<td>Building with amplifiers</td>
<td>Building with amplifiers</td>
</tr>
<tr>
<td></td>
<td>100 km max</td>
<td></td>
</tr>
</tbody>
</table>

*Figure 3-2  Multi-site sysplex distances*
you would like to merge them into a single cross-site sysplex to avail of sysplex aggregation, there will still be an impact. For example, if you are using current technology (z990) and GRS Star in each sysplex and the CPCs are within ICB distance of each other, you could expect lock response times in the region of 10 microseconds. If you merge these into a single sysplex, the GRS structure will be remote to the systems in one of the sites. If the centers are just 20km apart, you would expect the GRS response time for one sysplex to increase from 10 to maybe 300 to 350 microseconds. So, even though the technology supports up to 100km between the sites, you must project what that will do to your DASD and CF response times, and in turn, how the increased response times will impact your applications and the utilization of the attached systems.

To get the real availability and resiliency benefits from the two sites, you really need to have multi-site data sharing, together with a mechanism to transparently switch between primary and secondary DASD volumes—a capability that is provided by the HyperSwap™ function in GDPS/PPRC and GDPS/HyperSwap Manager. For more information on GDPS, refer to the IBM Redbook GDPS Family: An introduction to Concepts and Facilities, SG24-6374.

### 3.2 Advantages of a PlatinumPlex

Looking at sysplex in terms of how it can help reduce your software costs and provide real ongoing technical benefits, there are a number of sound reasons for deciding to implement a PlatinumPlex. This section reviews those benefits.

#### 3.2.1 Availability

There are few IT departments today that are not under pressure to deliver better application availability, preferably at a lower cost. No matter how good your availability may be, there is always pressure to improve it further. Looking at the costs of an outage, as shown in Table 3-1, you can see why businesses are willing to invest in order to avoid outages.

<table>
<thead>
<tr>
<th>Financial Impact of Downtime Per Hour (by various industries)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Brokerage Retail</strong></td>
</tr>
<tr>
<td><strong>Credit Card Sales Authorization</strong></td>
</tr>
<tr>
<td><strong>Airline Reservation Centers</strong></td>
</tr>
<tr>
<td><strong>Package Shipping Services</strong></td>
</tr>
<tr>
<td><strong>Manufacturing Industry</strong></td>
</tr>
<tr>
<td><strong>Banking Industry</strong></td>
</tr>
<tr>
<td><strong>Transportation Industry</strong></td>
</tr>
</tbody>
</table>

Source: Contingency Planning Research & Strategic Research Corp.

In this section we will briefly touch on three aspects of availability:

- Planned system outages
- Unplanned system outages
- Data outages
Planned system outages

As shown in the Gartner Group survey presented in Figure 3-3, the vast majority of outages in a System z9 or zSeries environment are planned outages. Despite the reliability and flexibility of modern System z9 and zSeries systems, there will always be a time when a planned outage is required, for example to replace a CPC or to upgrade to a new release of the operating system. Therefore, our objective is to find a mechanism to cater for planned system outages that will not have an impact on the users of applications running on that system.

***Figure 3-3  System outage reasons***

It would probably be good to start with a formal definition of exactly what is a planned outage:

“An unavailability of service that is known in advance to all users and maintainers of that service.”

Back in the “good old days”, planned outages were not a major problem. Systems had hundreds rather than thousands or tens of thousands of users, and nearly all the users were inside the company. As a result, it was easy to let all the users know of outages in advance.

Now, however, many of your users might be outside your company, either customers on the Internet or staff in business partners, both of whom have no idea of your “planned” outage windows. So, while an outage might be planned as far as you are concerned, for your users, it is just an unacceptable lack of service. As a result, the old attitudes of not being as concerned about planned outages are quickly disappearing.

Before we proceed, we need to discuss precisely what it is that users are concerned about. It isn’t systems - most of your users neither know nor care what system, or even what platform, their applications run on. Applications should be like the electricity service - you flick a light switch and you expect the light to come on - you don’t know or care how the power was generated. Equally, users simply expect to be able to use their application at all times, preferably without having to worry about an alternative process if the normal logon process doesn’t work. Now, this is good news for us as systems programmers, because while we do
not have the capability to eliminate all IPLs, we do have the capability to avoid planned system outages causing outages to applications.

In an appropriately planned and configured Parallel Sysplex, it should rarely be necessary to have a planned outage that will impact your application availability. The only event that could result in a complete lack of application availability should be a sysplex IPL, and they should be rare or non-existent. The goal should be to have at least one sysplex member up and running at all times—this potentially enables you to maintain constant application availability and also provides a platform that can be used to resolve any problems that may be preventing another system from successfully IPLing.

If you are able to run copies of your critical applications on all members of the sysplex, as you would in a PlatinumPlex, it should be possible to maintain the availability of those applications across planned outages. Parallel Sysplex technology can help you dynamically balance your workload across the systems, automatically re-routing incoming work requests to whichever server instances are available, steering them away from the system that is about to be shut down—this obviously requires that every system has the ability to read and update all data; data sharing in other words.

In a BronzePlex, on the other hand, a planned outage requires, at a minimum, that you will need to manually move work managers (DB2 and IMS subsystems, for example) from one system to another, impacting all users of those services during the move. In the worst case, where nothing is shared between the systems, there will not be a viable alternative system to move the work to, because no other system has access to the data, and even if it did, the security environment would be different - as a result, the application will be unavailable for the entire duration of the outage.

### Unplanned system outages

Unplanned outages hit you unexpectedly, usually at the worst possible time. And in most cases, your applications are off the air before you have an opportunity to do an orderly shutdown and move them off the failing system. The potential causes of an unplanned outage fall into one of two categories.

- You can suffer unplanned outages as a result of hardware failures.
  
  In times where storage devices can store tens of terabytes of data and CPCs support ever-increasing numbers of users, the number of physical boxes in the IT-department continually decreases. While this means there are fewer components to fail, the impact of each failure, if it does happen, is greater.
  
  If you have a PlatinumPlex, your critical applications would be spread across the CPCs, meaning that they would continue to be available after a CPC failure, even if it meant some of the users have to log on again. A BronzePlex will not buy you anything as there will only be one instance of each application, meaning that the application will be unavailable if the CPC it is running on fails.

  For planned or unplanned DASD outages, the HyperSwap capability in GDPS/PPRC and GDPS HyperSwap Manager transparently switches to the secondary volumes without requiring a restart of the connected systems. All the systems in a given GDPS must be in the same sysplex, however both BronzePlexes and PlatinumPlexes are supported (see “Role of GDPS” on page 36).

- You can suffer unplanned outages because of software failures or human error (systems management problems).
  
  Once again, having multiple instances of an application, as in a PlatinumPlex, will protect you from the outage of a subsystem or even a whole system. Just as for a CPC failure, a BronzePlex does not provide any protection against such a failure.
From a systems management perspective, you are far more likely to have an integrated view, and single point of control over the whole sysplex in a PlatinumPlex. This provides the ability for humans and automation to make better informed decisions, thereby reducing the risk of procedural errors causing an outage.

The largest problem we see in a BronzePlex is that you are running a sysplex configuration in a way that it was not designed for. You are neither treating the systems as a single image, nor are they completely independent of each other.

**Data outages**

Some installations build Parallel Sysplexes that include test, development, and production workloads in the same sysplex. This may appear to be a simpler solution, because there are fewer sysplexes to manage. However running production and non-production jobs in the same Parallel Sysplex, with access to the same DASD pool, introduces the potential for the wrong data to be used somewhere.

For example, what would happen if somebody moves a test batch job for a new application into production without updating all the JCL? If you have a clearly-separated environment (PlatinumPlex), that is not a problem: when the job runs in production it would get an abend saying “data set not found” (because it tried to access the test data). But what would happen if the sysplex consists of test and production systems and all the data is available to each system? Now, instead of failing, the job runs “successfully”, ending with an RC=0. No one notices, because this is one of thousands of jobs ending with RC=0, so it may be some time before someone notices that they have the wrong data.

To avoid this type of problem, the safest way is to isolate test, development, and production sysplexes. Depending on your current configuration, providing this level of isolation may result in higher hardware costs because you may need additional CF LPARs, additional CF links, and so on. However, the added security and auditability provided by segregation more than outweighs the one-off costs of splitting the sysplexes.

**Protecting production systems**

There are a number of other things that must be considered should you be contemplating including production and non-production systems in the same sysplex.

In general, IBM have very good, robust isolation capabilities on zSeries (LPAR, for example) and databases and data can be effectively isolated. However, some things in a sysplex are meant to be shared:

- XCF needs to communicate with all the images in the same plex and it can’t tell the difference between important (production) images and not so important (test) images.
- Some things are sysplex-wide in scope (CFRM policy, Couple Data Sets, SFM policy, and so on) and therefore can’t be isolated.

Certain functions cannot be used or enabled until all images are at the required software level—examples would be CF structure duplexing, GRS star, and so on. In a sysplex that consists of both test and production systems, such new features can’t be fully tested before the prerequisite software level is introduced to the production images, delaying the introduction of the new features and potentially increasing the risks to the production systems.

Similarly, new functions often require toleration support to be installed in all members of the sysplex prior to being enabled. Having production and test or development systems in the same sysplex elongates the time to roll out such new functions because of the number and sensitivity of the systems involved.
Some functions, such as GRS star, must be enabled on all members of the sysplex or disabled on all members. If you wish to enable these features on the test systems for evaluation, you must by definition also enable them on the production systems at the same time, which is probably not your preferred approach.

You also need to think about how you handle hardware resources that are shared between the test and production systems. For example, you might share a CF link between test and production LPARs. In normal operation, this may provide perfectly acceptable service times, but what happens if there is a change or a bug in the test workload, resulting in a flood of requests to the CF from the test LPAR? Even if the CF has enough capacity to handle the increased load, the additional traffic on the CF link could cause an unacceptable degradation in CF response times for the production system. Having an integrated test/production configuration makes it more difficult to avoid sharing such resources than if production and non-production systems were in separate sysplexes.

Role of GDPS

Given the context of this book, sysplex aggregation, we will limit our GDPS discussion to GDPS/PPRC, which supports a multi-site sysplex\(^1\).

GDPS/PPRC complements a multi-site Parallel Sysplex implementation by providing a single, automated solution to dynamically manage storage subsystem mirroring (disk and tape), CPCs, and systems. It is designed to help a business attain near-continuous availability and a disaster recovery capability with no or minimal data loss. GDPS/PPRC is designed to minimize and potentially eliminate the impact of any hardware or site failure, including disasters, or a planned outage.

The availability of HyperSwap greatly enhances the attractiveness of running data sharing across both sites (multi-site workload, in GDPS terminology). With HyperSwap, you can execute a planned site shutdown with no application outage. HyperSwap also allows you to continue processing across a DASD subsystem failure, again with no application outage.

If you already have two sites that are within a distance that would make cross-site data sharing viable, and you are considering merging the sysplexes for aggregation purposes, you may find that the resulting software savings could fund the GDPS project, giving you a complete near-continuous availability and disaster recovery solution while still saving you money.

For more information on the various GDPS offerings, refer to the IBM Redbook GDPS Family: An introduction to Concepts and Facilities, SG24-6374 or the GDPS Web site: http://www.ibm.com/servers/eserver/zseries/gdps/

3.2.2 Systems management

Direct software costs (the money you send the vendor every month) are only one aspect of the cost of running an IT service. Costs are typically broken into hardware, software, and people. Being able to exploit sysplex aggregation should reduce the software component. However, for most installations, the fastest growing cost is people. The type of sysplex you implement (PlatinumPlex or BronzePlex) can impact your people costs.

There are many aspects to systems management, such as operations, automation, batch management, storage management, security management, and performance management,\(^1\)

\(^1\) GDPS/PPRC HyperSwap Manager also supports multi-site sysplex and HyperSwap, however it does not provide all the system management capabilities that are delivered by GDPS/PPRC. For more information on GDPS/PPRC HyperSwap Manager, and a comparison of the two offerings, refer to the IBM Redbook GDPS Family: An introduction to Concepts and Facilities, SG24-6374.
to name just a few. One way to reduce your costs is minimize the effort you need to put into managing and maintaining each of these. PlatinumPlex plays an important role in this effort by providing you with a single system image for the sysplex.

There are technical reasons for wanting a single system image - for example, if you have a single volume that is protected by two or more RACF databases, there is the possibility that a user will have different levels of access to a resource depending on which system he is running on. Your security will be more reliable if there is just a single RACF database controlling access to all sysplex resources.

However, there are also financial reasons for having a single RACF database for the entire sysplex. Rather than having to define your users and resources multiple times, in each RACF database, you only need to define them once, saving a considerable portion of your security administrator's time. You also save the time that is spent dealing with the confusion that arises as a result of having multiple databases.

The same argument applies to nearly all the systems management principles - apart from the technical benefits, there are nearly always financial benefits as a result of only having to do things once per sysplex. It is important to consider this when deciding what type of sysplex you want to implement.

### 3.2.3 Maintaining sysplex aggregation qualification

For many installations, the most challenging aspect of sysplex aggregation qualification is trying to maintain a configuration where the PrimaryPlex uses more than 50% of the used MVS MSUs on each of the CPCs in the PricingPlex. Not only do the workloads in each system change on an ongoing basis, but every time you upgrade or replace a CPC, the capacity profile of the installation changes so much that you may have to move systems between CPCs, and workloads between systems, in order to maintain qualification.

However, this is much more of an issue in a BronzePlex than in a PlatinumPlex. In a BronzePlex, the moveable units of work are things like DB2 or IMS subsystems, their attendant CICS regions, plus any batch jobs that use the data they manage. If a DB2 subsystem doesn’t exploit data sharing, there is only one place you can run the jobs that use the data owned by that DB2.

On the other hand, if you are fully exploiting data sharing, the moveable unit of work is a single transaction or a single batch job. And if you are exploiting dynamic workload balancing, the system can automatically move the work to where the capacity exists. This in effect allows you to more fully exploit all your installed capacity, and may allow you to postpone capacity upgrades.

---

**Figure 3-4** PlatinumPlex versus BronzePlex
Another advantage of a PlatinumPlex shows itself when you upgrade or replace a CPC. As an analogy, we use two glass containers (as shown in Figure 3-4 on page 37). Consider the two glasses on the left as being a BronzePlex, with each of the moveable units of work being a large irregular-shaped stone, and the glasses on the right representing a PlatinumPlex, where the moveable units of work are small pebbles.

Now consider what happens when you upgrade one of your CPCs, as shown in Figure 3-5. The figure shows how in a BronzePlex, you may have to manually move selected rocks (DB2 subsystems) between glasses (CPCs) to maintain aggregation qualification, especially if the upgraded CPC has latent demand which will change the workload balance after the upgrade. The right half of the figure shows the same upgrade when you are using a PlatinumPlex. In this case, the work has automatically flowed to where the additional capacity is, and little if any manual adjustment should be required.

An additional benefit of the workload balancing capability of a PlatinumPlex is that you can grow your capacity in three ways:

- You can increase the number of logical CPs available to the LPAR, thereby increasing the capacity of each system. You can have up to 32 logical CPs in a z/OS 1.6 or later LPAR.
- You can increase the number of systems in the sysplex.
- You can increase the speed of your systems by moving to newer, faster, CPCs.

Some workloads, like WebSphere, scale extremely well with large numbers of logical CPs, while others scale better by increasing the speed of the CPCs or the number of members in the sysplex. By having the ability to easily and automatically move work anywhere in the sysplex, you can avail of the full variety of upgrade options, selecting the one that is the most cost-effective for you at a given time.

In general, adding a new member to an existing data sharing sysplex increases overhead by about 0.5%. To understand the impact of changing CPC type or model, refer to the Large Systems Performance Reference (LSPR) numbers available at:


When looking at upgrade options, remember that as well as MSUs, consideration must be given to the workload type, the impact of the change on virtual storage constraints, single points of failure, possible additional I/O and CF connectivity, and systems management considerations. There is no question, however, that it is far easier to add a new system image to a PlatinumPlex than to a BronzePlex.
3.3 Sysplex aggregation qualification

We have now discussed the different types of sysplex, and elaborated on some of the benefits of a PlatinumPlex, especially in relation to sysplex aggregation. The next step is to analyze your current configuration to confirm that it still qualifies for sysplex aggregation, and make any changes that may be required. Having confirmed that you are still in compliance, you should take the opportunity to review your sysplex configuration to ensure that you will remain in compliance in the future, preferably with little or no manual intervention.

If you do not currently qualify, or if you would like to increase the number of CPCs in your PricingPlex, or if you just barely qualify and would like a larger cushion, there are a number of options open to you. However, before we get into that discussion, let’s discuss verifying your current qualification.

Situations in which you may want to re-validate your qualification are:

- Some time has passed since you informed IBM about which CPCs qualify for sysplex aggregation, and workloads have come and gone in that time, so you just want to be sure that the qualification criteria are still being met.
- A CPC was upgraded, but the workload distribution across the CPCs in the sysplex was not adjusted to reflect the new distribution of capacity.
- A larger CPC was installed to consolidate the workloads from a number of smaller CPCs.
- Additional CPCs have been added to your configuration, perhaps because of a merger.
- More systems have been created because, for example, your company decided to build a quality assurance environment to check all software and applications in a production-like environment.
- zAAPs have been installed and will be used to offload a significant amount of production Java workload from the general purpose CPs (remember that cycles consumed on zAAP engines are not included in the aggregation qualification calculation).
- Workload has been migrated from your PrimaryPlex z/OS systems to other platforms, reducing the number of MSUs used by the PrimaryPlex systems.
- Work was consolidated from other platforms to z/OS, but not into your PrimaryPlex systems. Or maybe it was all consolidated into a single PrimaryPlex system, rather than balancing it across all systems in that sysplex.

Any of these events can affect the balance of utilization between your LPARs, potentially resulting in a CPC falling out of qualification. In order to understand your current situation, you should run the PLEXCALC tool, inputting SMF records from every z/OS system in your installation. As discussed in “PLEXCALC” on page 21, PLEXCALC will help you identify which CPCs meet the qualification criteria, and if so, by how much.

If you find that work needs to be done to increase the relative share of CPC consumption by the PrimaryPlex, your initial reaction may be that you need to merge more systems into the PrimaryPlex in order to push up its utilization. However merging a system into a sysplex is not a trivial effort, so before you embark on that path, you should investigate if there are other, easier, ways of achieving your objective. The following steps will help you identify the scale of the adjustment that is required:

- For each CPC in your configuration, check how much of the capacity is used by each of your sysplexes. You can use PLEXCALC to get this information. Are there CPCs that no longer meet the qualification criteria? If so, how far short are they (in terms of MSUs)?

If you find that you are only narrowly missing the target 50% on just one CPC, and the other CPCs in the PricingPlex are also close to the 50% target, the solution may be to move some work from outside the sysplex into that system. An example might be the
processing of SMF records for billing or management reports that is currently run on a
development system - it might be easy to move those jobs into the affected system.
Moving workloads in this way is discussed in 5.5, “Redistributing workloads and
processing” on page 95.

► Check how many MSUs your largest\(^2\) sysplex is consuming across all installed CPCs, and
what percent of the total consumed MVS MSUs this represents. If that sysplex is using
more than, say, 60% of all MVS MSUs across all CPCs, the solution may just be improved
workload balancing.

On the other hand, if you do not have any sysplex that even comes close to 50% of the
MVS MSUs across all the CPCs that you would like to have in the PricingPlex, you
probably have a more challenging task ahead of you.

Chapter 5, “Workload balancing and sysplex aggregation” on page 65 contains information
about the various components and products that play a role in dynamic workload balancing,
and also contains some other potential ways to affect system utilization and help you achieve
the qualification criteria. We urge you to read that chapter before making a final
merge/no-merge decision.

If your investigations show that the only way to meet the qualification criteria is to merge more
systems into your sysplex, the next section shows how these considerations were applied to
our example configuration.

Tip: Before opting for a BronzePlex on the basis of it being a “quick fix”, we urge you to
consider the medium and long term views as well. You may find that having a BronzePlex
as your final objective is a bit like buying an old car: it appears cheap at the time you buy it,
but longer term you find that this was actually the more expensive option because of the
ongoing support and maintenance costs.

3.4 To merge or not - a sample scenario

This configuration is based on the example we introduced in 2.2.3, “Sample configuration 3”
on page 14. Since the final configuration we discussed, a data center consolidation has taken
place in the company. You installed a new 1500 MSU CPC in the same computer room as
your existing five machines. The new CPC is to run the workload from a remote data center,
which was previously located 2000 miles away. This results in the configuration shown in
Figure 3-6 on page 41.

\(^2\) “largest” sysplex means the sysplex that consume the largest number of MSUs of all your sysplexes.
The workload running in CPC F is very similar to the PRD workload in the other CPCs, however it serves subsidiaries in other parts of the country.

As a result of these changes, you now have the following situation:

- CPCs A and B qualify for sysplex aggregation, based on systems PRDA1/PRDA2 and PRDB1 consuming more than 50% of the MVS-based MSUs on their respective CPCs. This gives you PricingPlex A+B.
- CPCs D and E also qualify for sysplex aggregation, based on systems QAD1 and QAE1 consuming more than 50% of the MVS-based MSUs on those CPCs. This gives you PricingPlex D+E.
- Neither CPCs C nor F qualify for sysplex aggregation. On CPC F, system NEWF1 is not part of any multi-system sysplex and on CPC C, no sysplex accounts for more than 50% of the used MVS MSUs, so neither of these CPCs are in a PricingPlex.

Having completed the move of system NEWF1 to CPC F, you wish to address the fact that that system is a single point of failure and that having all that workload in a single z/OS system cannot deliver the levels of service and availability your business requires. For example, planned IPLs of this system have always implied an availability impact for all the users of that system. The fact that NEWF1 now resides in a machine room with other CPCs introduces the possibility of cloning the NEWF1 workload across multiple CPCs.
To increase availability by exploiting the available hardware and sysplex technology, you make the following decisions:

- Merge system NEWF1 into the existing PRD sysplex. Because the PRD sysplex is exploiting data sharing, this would remove the NEWF1 system as a single point of failure. Because of the objective to remove the system as a single point of failure, the decision was made to go with a PlatinumPlex—a BronzePlex would not have delivered this capability.

- Use dynamic workload balancing wherever possible to spread all workloads across all CPCs. We will focus on this in Chapter 5, “Workload balancing and sysplex aggregation” on page 65.

We have to be honest and admit that, in addition to the prospect of improved availability and improved system management, the software savings that could be achieved by including CPC F in the PricingPlex was also a major consideration in the decision to merge system NEWF1 into the PROD PricingPlex. Adding CPC F to the table we were using earlier, results in the new numbers shown in Table 3-2. It is obvious that if you add system NEWF1 to the Prod sysplex then you will be able to include CPC F in PricingPlex A+B, along with CPCs A and B.

<table>
<thead>
<tr>
<th>CPC</th>
<th>LPAR/System</th>
<th>Sysplex</th>
<th>Avg MSU consumption</th>
<th>% of MVS-based MSUs</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>TSTA1</td>
<td>Test</td>
<td>109</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>DEVA1</td>
<td>Dev</td>
<td>164</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>PRDA1</td>
<td>Prod</td>
<td>800</td>
<td>58</td>
</tr>
<tr>
<td></td>
<td>PRDA2</td>
<td>Prod</td>
<td>300</td>
<td>22</td>
</tr>
<tr>
<td>B</td>
<td>TSTB1</td>
<td>Test</td>
<td>164</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>DEVB1</td>
<td>Dev</td>
<td>200</td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>PRDB1</td>
<td>Prod</td>
<td>546</td>
<td>60</td>
</tr>
<tr>
<td>F</td>
<td>NEWF1</td>
<td>None (Target is Prod)</td>
<td>1230</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>PricingPlex A + B + F Total</td>
<td>-</td>
<td>3513</td>
<td>-</td>
</tr>
<tr>
<td>C</td>
<td>TSTC1</td>
<td>Test</td>
<td>216</td>
<td>25</td>
</tr>
<tr>
<td></td>
<td>DEVC1</td>
<td>Dev</td>
<td>312</td>
<td>36</td>
</tr>
<tr>
<td></td>
<td>PRDC1</td>
<td>Prod</td>
<td>337</td>
<td>39</td>
</tr>
<tr>
<td></td>
<td>CPC C Total</td>
<td>-</td>
<td>865</td>
<td>-</td>
</tr>
<tr>
<td>D</td>
<td>VMLD1</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>QAD1</td>
<td>QA</td>
<td>566</td>
<td>53</td>
</tr>
</tbody>
</table>
Interestingly, if you now add up the MSUs used by all the MVS-based systems, and the MSUs used by members of the Prod sysplex, you will find that the Prod sysplex systems are using more than 56% of all MVS-based MSUs consumed across all six CPCs. This means that after completing the merge and the workload balancing exercise, you have the potential of being able to have all six CPCs in the same PricingPlex should you decide to pursue that objective.

### 3.5 Summary

It would be unusual to find an IT department today that is not being given the following objectives by management:

- Reduce hardware and software costs
- Improve application availability
- Make more effective use of the existing workforce

Achieving sysplex aggregation will help you achieve part of the first objective (software cost savings). Using a PlatinumPlex to help you achieve sysplex aggregation will also:

- Help you address hardware costs by letting you more effectively use all the available capacity across all installed CPCs
- Help you reduce personnel costs by eliminating duplicate systems management and administration tasks
- Improve application availability by letting you reduce single points of failure for your applications
- Allow the existing staff to handling increasing workloads and increasingly complex applications by reducing the number of instances (RACF databases, sysres', storage pools and policies) that need to be maintained

We understand that IT departments are under pressure to reduce costs now, but don’t forget the longer term savings, both in software costs and people costs, that can be achieved by implementing a PlatinumPlex. When embarking on a sysplex aggregation project, show management that you can deliver even greater savings and better service by expanding the project to include the eventual implementation of a PlatinumPlex.
Chapter 4. Sample scenarios

This chapter provides a description of a number of real life customer configurations. In each case, it discusses the original configuration and the considerations involved when investigating the applicability and benefits of sysplex aggregation to their particular case.

**Important:** In this chapter we present a variety of typical customer situations. In order to illustrate the relative software costs of each option, we once again use our fictional currency, the RedBit.

Please note, however, that for simplicity we have assumed that all configurations are using full-capacity rather than subcapacity pricing. In practice, we believe that many people using Workload Licence Charging (WLC) would actually be exploiting subcapacity pricing to further reduce their software costs. However, to avoid the added complexity of getting into discussions about rolling 4-hour averages, we have priced all examples on the basis of full capacity pricing.
4.1 Introduction

In Chapter 2, “Sysplex aggregation” on page 5 we introduced the concept of sysplex aggregation and discussed the qualification criteria that must be met in order to be able to avail of this facility. We also introduced a sample customer configuration that was carried forward into Chapter 3, “Parallel Sysplex options and considerations” on page 27 where we discussed the considerations relating to the type of sysplex that you can implement.

In this chapter we provide a few more sample configurations, representing customers with different configurations and different requirements. None of these examples exactly reflect a real customer, however each scenario was inspired by a real customer situation.

In the examples we used in 2.1, “Sysplex aggregation overview” on page 6 we introduced a fictitious currency called “RedBits” to help illustrate the differing costs of our scenarios. In the examples in this chapter, we provide prices (in RedBits) for the following representative software stack:

- 5625-DB2 DB2 V8 UDB for z/OS
- 5655-M15 CICS TS for z/OS 3.1
- 5694-A01 z/OS V1 BASE
- 5694-A01 z/OS V1 DFSMSdss™+hsm
- 5694-A01 z/OS V1 DFSORT™
- 5694-A01 z/OS V1 RMF
- 5694-A01 z/OS V1 SDSF
- 5694-A01 z/OS V1 Security Server
- 5697-ENV Tivoli® NetView® for z/OS V5

4.2 Scenario 1 - Checking all the alternatives

The first example customer had two data centers, one in San Francisco, and the other in Las Vegas. The company headquarters had been in Las Vegas, but moved some years ago to Chicago.

Initial configuration

The data centers had the following configuration:

- San Francisco
  - There was a single processor, a z990 model 316.
  - There were three production systems (PRDA1, PRDA2, PRDA3), all in the same sysplex. The systems shared data sets and exploited resource sharing to some extent.
  - There was one development system and one test system. Both of these were monoplexes.
- Las Vegas
  - There was a single processor, a z990 model 304.
  - There were two production systems (PRDB1, PRDB2) in a sysplex.

The production systems were similar to those in San Francisco in that they exploited resource sharing but didn’t yet exploit data sharing. The lack of data sharing and workload balancing was the primary reason why the CPU consumption of the two
production systems was so different (50% for PRDB2 and 10% for PRDB1) as seen in Figure 4-1.

- There was one development system and one test system, both in their own monoplexes.

- Note that the production systems in San Francisco and Las Vegas ran totally different applications.

![Figure 4-1 Original situation for scenario 1](image.png)

In Figure 4-1, the percentage numbers above the CPCs indicate the z/OS average utilization over the 40-hour prime shift (one week). Information about the configuration is shown in Table 4-1 and Table 4-2.

**Table 4-1  CPC A: z990-316, 761 MSUs, prime shift 40 hour average: 92% (700 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODSF</td>
<td>PRDA1</td>
<td>30</td>
<td>210</td>
</tr>
<tr>
<td>PRODSF</td>
<td>PRDA2</td>
<td>25</td>
<td>175</td>
</tr>
<tr>
<td>PRODSF</td>
<td>PRDA3</td>
<td>20</td>
<td>140</td>
</tr>
<tr>
<td>no sysplex</td>
<td>DEVA1</td>
<td>15</td>
<td>105</td>
</tr>
<tr>
<td>no sysplex</td>
<td>TSTA1</td>
<td>10</td>
<td>70</td>
</tr>
</tbody>
</table>

**Table 4-2  CPC B: z990-304, 248 MSUs, prime shift 40 hour average: 84% (209 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODLV</td>
<td>PRDB1</td>
<td>10</td>
<td>21</td>
</tr>
</tbody>
</table>
From a software pricing perspective it was not possible to do any aggregation: there was only one CPC in each data center, meaning there was nothing to aggregate, and they couldn’t aggregate across sites because of the distance between the sites. The overall software price for this configuration (using the software stack we listed in 4.1, “Introduction” on page 46) was 499,500 RedBits per month.

Considerations
Taking this as the starting configuration, a number of things were happening in the company:

- Because of increasing competitive pressure, together with a move to provide Internet access to applications running on z/OS, the IT department was under pressure to provide a more resilient configuration, delivering better availability now, together with a disaster recovery capability in the medium term.
- Company management wanted to investigate the possibility of decreasing software costs across the two CPCs.

Because of these two requirements, the systems were transferred from San Francisco and Las Vegas to the corporate headquarters in Chicago. This introduces the possibility of being able to exploit the presence of a second CPC for production work in case of a hardware failure. It also moves the two CPCs within sysplex distance of each other, meaning that the use of sysplex aggregation can be investigated as a means of lowering software costs.

The customer’s strategic direction is to utilize GDPS/PPRC to deliver the near-continuous availability and disaster recovery capability required by the business units. However, before starting that project, the customer first wishes to improve availability by exploiting data sharing and resource sharing within a single site.

Implemented solution
In order to quickly deliver the savings possible with sysplex aggregation, the customer’s first reaction was to investigate implementing a BronzePlex, merging the former San Francisco and Las Vegas production sysplexes in to a single sysplex. However, because the software, applications, and workloads running in the two sysplexes are completely different, there was not an obvious technical benefit from having everything in the same sysplex.

As an alternative way of achieving sysplex aggregation, the customer looked at the impact of moving systems around between the CPCs. It transpired that moving one of the San Francisco production systems to the Las Vegas CPC, and one of the Las Vegas production systems and a test system to the San Francisco CPC would result in roughly the same overall CPC utilization as before, however the San Francisco sysplex (PRODSF) would now be consuming more than 50% of the MVS-based MSUs on both CPCs, as shown in Figure 4-2 on page 49. This would require some connectivity changes, however these would have been required anyway in order for the Las Vegas CPC to be able to act as a backup CPC to the San Francisco one and vice versa. Overall, however, the elapsed time and amount of effort would be significantly less than merging all the production systems into a single BronzePlex or PlatinumPlex, the same software cost savings are achieved, and the CPC is removed as a single point of failure in each production sysplex.

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODLV</td>
<td>PRDB2</td>
<td>50</td>
<td>104</td>
</tr>
<tr>
<td>no sysplex</td>
<td>DEVB1</td>
<td>20</td>
<td>42</td>
</tr>
<tr>
<td>no sysplex</td>
<td>TSTB1</td>
<td>20</td>
<td>42</td>
</tr>
</tbody>
</table>
While the customer could have continued to use Internal Coupling Facilities for resource sharing, because they are embarking on a data sharing implementation project, they also installed two standalone Coupling Facilities at the same time. Once the infrastructure is in place, they will commence the data sharing/workload balancing project in the PRODSF sysplex systems. While the PRODLV systems are not major players from a sysplex aggregation perspective, the data sharing project will then move on to those systems to help them achieve similar levels of availability.

![Figure 4-2  Situation after the configuration changes](image)

The projected utilizations of the various systems are shown in Table 4-3, “CPC A (Chicago): z990-316, 761 MSUs, prime shift 40 hour average: 93% (706 MSUs)” on page 49 and Table 4-4 on page 50. We highlighted the rows pertaining to the PrimaryPlex systems - you can see that they will account for more than 50% of the total MVS-based MSUs on each CPC. This means that both CPCs qualify for aggregation. The number of MSUs consumed by each system is extracted from the PLEXCALC report.

Based on this configuration and the number of MSUs, the customer’s software cost would reduce by 124,000 RedBits a month, decreasing from 499,500 RedBits to 375,500 RedBits.

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODSF</td>
<td>PRDA1</td>
<td>30</td>
<td>210</td>
</tr>
<tr>
<td>PRODSF</td>
<td>PRDA2</td>
<td>25</td>
<td>175</td>
</tr>
<tr>
<td>PRODLV</td>
<td>PRDB2</td>
<td>15</td>
<td>104</td>
</tr>
<tr>
<td>no sysplex</td>
<td>DEVA1</td>
<td>15</td>
<td>105</td>
</tr>
<tr>
<td>no sysplex</td>
<td>TSTA1</td>
<td>6</td>
<td>42</td>
</tr>
</tbody>
</table>
In addition to the software savings, the new configuration brings the following benefits:

- The customer now has the infrastructure in place to start implementing data sharing (which requires failure-isolated Coupling Facilities).
- Both production sysplexes have members on both CPCs, meaning that a CPC outage should not bring down all members of either sysplex.
- The configuration is strategic in terms of a later implementation of multi-site data sharing with GDPS/PPRC.

4.3 Scenario 2 - Spending in order to save

The customer in the next scenario had three data centers in Germany, two on a campus in Munich, and an older one in Berlin. In addition to z/OS, they also use Linux and z/VM. One of the z/VM systems is used for a 3-way test sysplex, and the other is used for Linux guests. They are running WebSphere on the production systems, with this being the fastest growing part of the production workload.

Initial configuration

The following sections describe the configurations at the start of the project.

**Munich**

In the Munich campus there were three sysplexes on two CPCs, one in each data center in the campus:

- The production sysplex, PRODM, consisting of systems PRDA1 and PRDB1. This sysplex exploits resource sharing today, and will exploit data sharing in the near future.
- The development sysplex, DEVM, consisting of systems DEVA1 and DEVB1. These systems exploit both resource sharing and data sharing (they are running the data sharing applications that will soon move into production).
- The test sysplex, TESTM, consisting of systems TSTB1, TSTB2, and TSTB3. However, this sysplex is a little different because it runs under z/VM. That gives their systems programmers the ideal test environment to test CF outages and application recovery. However all the members of the sysplex must reside on a single CPC (because they must all run under the same z/VM image) and therefore cannot be moved around to address workload balancing issues.
**Berlin**

In Berlin there is only a single machine, running one production system and one quality assurance system. These systems are completely separated from each other, and are each in their own monoplex, with no shared resources.

**Considerations**

This customer is like most customers around the world - the IT department is being asked to do more for less. Due to an increasingly competitive business environment, the company is trying to reduce costs where possible. However, there is also a requirement for more capacity (the capacity planning specialists have recommended a 20% increase of MSUs on CPC A), and the users of the production system in Berlin are demanding a decrease in planned and unplanned outage times.

The availability enhancements for the Berlin system can be accomplished by using data sharing and workload balancing, however a second CPC is really necessary in order to fully exploit data sharing and workload balancing.

The configuration of the Munich and Berlin data centers is shown in Figure 4-3.

![Figure 4-3 Original situation for scenario 2](image)

Table 4-5, Table 4-6 on page 52, and Table 4-7 on page 52 describe the original situation. The green rows in these tables highlight the systems that are members of the PrimaryPlex on each CPC - note that there are no green rows for CPC C, because it is a single CPC with no other CPC that it can aggregate with.

**Table 4-5  CPC A (Munich): z990-308 (+ 4 IFLs), 448 MSUs, prime shift 40 hour average: 82% (368 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODM</td>
<td>PRDA1</td>
<td>75</td>
<td>276</td>
</tr>
</tbody>
</table>
Because the z/VM system called LINUX runs only Linux guests, the CPU consumption of that LPAR is not taken into account for sysplex aggregation qualification.

**Table 4-6  CPC B (Munich): z990-310, 538 MSUs, prime shift 40 hour average: 76% (410 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEVM</td>
<td>DEVA1</td>
<td>25</td>
<td>92</td>
</tr>
<tr>
<td>none</td>
<td>LINUX</td>
<td>4 IFL engines</td>
<td>0</td>
</tr>
</tbody>
</table>

In Table 4-6 you will see that the utilization of the z/VM LPAR has been taken into account in the sysplex aggregation calculations. To be precise, the portion of the z/VM utilization attributable to the z/OS guests has been taken into account - the CPU consumption of the CF virtual machines (CFB1 and CFB2), for example, is ignored.

**Table 4-7  CPC C (Berlin): z990-308, MSUs: 448, prime shift 40 hour average: 85% (381 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODM</td>
<td>PRDB1</td>
<td>52</td>
<td>213</td>
</tr>
<tr>
<td>DEVM</td>
<td>DEVB1</td>
<td>23</td>
<td>94</td>
</tr>
<tr>
<td>TESTM</td>
<td>TSTB1 (VM)</td>
<td>8</td>
<td>33</td>
</tr>
<tr>
<td>TESTM</td>
<td>TSTB2 (VM)</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>TESTM</td>
<td>TSTB3 (VM)</td>
<td>10</td>
<td>41</td>
</tr>
<tr>
<td>none</td>
<td>CFB1 (VM)</td>
<td>does not run z/OS</td>
<td>0</td>
</tr>
<tr>
<td>none</td>
<td>CFB2 (VM)</td>
<td>does not run z/OS</td>
<td>0</td>
</tr>
</tbody>
</table>

From a software point of view, the Munich configuration costs 371,000 RedBits per month and the Berlin configuration currently costs 245,000 RedBits per month. That sums up to a total software bill (using the software stack described in 4.1, “Introduction” on page 46) of 616,000 RedBits per month.

**Implemented solution**

To find a strategy to address the diverging requirements of reduced costs and greater availability and capacity, the customer calls a meeting consisting of their technical and management staff and their IBM representatives. The objective of the meeting is to investigate all the technical options to identify the one that best addresses their business and technical objectives. After much discussion, the following plan was agreed:

- The highest priority is to consolidate the Berlin data center into the Munich campus.
  Placing all the CPCs in just two locations within sysplex distance of each other is a prerequisite to the other action items that were agreed. Additionally, the Berlin data center is outdated and would require significant capital investment to bring it to the same state-of-the-art level as the centers in the Munich campus.
It is acknowledged that the only way to address the demands for improved availability for the Berlin production system is to implement data sharing and workload balancing, to remove the single points of failure for those applications. This was one of the reasons for moving the Berlin CPC to Munich - so the Berlin applications could be spread across multiple CPCs.

However, due to application development resource constraints, this work cannot start until the current data sharing project for the Munich systems has completed. Despite the fact that moving to a BronzePlex followed by a move to a PlatinumPlex will require more effort than going directly to a PlatinumPlex, the business cannot wait for the PlatinumPlex project to obtain the cost savings from sysplex aggregation. As a result, the customer decided to start a BronzePlex project immediately following the move of the Berlin CPC to the Munich data center.

Additional capacity is required to handle the growing production workload on the PRDA system. WebSphere already accounts for a sizeable portion of the CPU consumption of that system. Because of the size of the Java workload, it is determined that judicious use of zAAPs can help address the capacity requirement.

The projected configuration reflecting these changes is shown in Figure 4-4.

![Figure 4-4 Implemented solution for scenario 2 including two zAAPs for Java workload](image)

Coming back to the use of zAAPs; a recent study using the zAAP Projection Tool indicated that the Java workload is consuming more than two CP’s worth of capacity. By purchasing two zAAPs on CPC A, they can move that workload to the zAAPs, freeing up capacity on the CPs. The capacity planning group identified a need for 20% more capacity - moving the work to the zAAPs will free up sufficient capacity on the existing CPs to be able to deliver that capacity.

It would have been possible to install a third zAAP and move yet more work from the CPs to the zAAPs (thereby lowering the z/OS software bill still further assuming subcapacity charging is being used), however the customer decides to wait for more growth in the Java workload before pursuing this option.
Using the CPC consumption in Table 4-5 on page 51 as a base, installing two zAAPs on CPC A would move roughly 112 MSUs of work from the CPs to the zAAPs. Because there is little Java work running in the DEVA1 system, the installation of the zAAPs would have little impact on that system. This would result in the utilizations shown in Table 4-8 - the PrimaryPlex is now using 64% of the MVS-based MSUs, compared to 75% before the installation of the zAAPs, still comfortably above the 50% threshold.

Table 4-8  
CPC A: z990-308 + 4 IFLs + 2 ZAAPs, 448 MSUs, prime shift 40 hour average: 82% (367 MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRDM</td>
<td>PRDA1</td>
<td>64</td>
<td>164</td>
</tr>
<tr>
<td>DEVVM</td>
<td>DEVA1</td>
<td>36</td>
<td>92</td>
</tr>
<tr>
<td>none</td>
<td>LINUX</td>
<td>4 IFL engines</td>
<td>0</td>
</tr>
</tbody>
</table>

The installation of a zAAP in CPC B would decrease the utilization of the PrimaryPlex to roughly 157 MSUs - less than 50% of the total used MVS-based MSUs, resulting in that CPC no longer qualifying for aggregation. The decision was made to postpone the installation of a zAAP on this CPC until the data sharing project is further progressed (this is expected to result in a move of some z/OS workload from CPC A to CPC B). As a result, the configuration of CPC B, as shown in Table 4-9, is unchanged.

Table 4-9  
CPC B: z990-310, 538 MSUs, prime shift 40 hour average: 76% (538 MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODM</td>
<td>PRDB1</td>
<td>52</td>
<td>213</td>
</tr>
<tr>
<td>DEVVM</td>
<td>DEVB1</td>
<td>23</td>
<td>94</td>
</tr>
<tr>
<td>TESTM</td>
<td>TSTB1 (VM)</td>
<td>8</td>
<td>33</td>
</tr>
<tr>
<td>TESTM</td>
<td>TSTB2 (VM)</td>
<td>7</td>
<td>29</td>
</tr>
<tr>
<td>TESTM</td>
<td>TSTB3 (VM)</td>
<td>10</td>
<td>41</td>
</tr>
<tr>
<td>none</td>
<td>CFB1 (VM)</td>
<td>does not run zOS</td>
<td>0</td>
</tr>
<tr>
<td>none</td>
<td>CFB2 (VM)</td>
<td>does not run zOS</td>
<td>0</td>
</tr>
</tbody>
</table>

Finally, system PRDC1 on CPC C, which moved from Berlin to Munich, will be added to the PRODM sysplex. Because system PRDC1 is using more than 50% of the MVS-based MSUs on CPC C, sysplex PRODM will be the PrimaryPlex on CPC C, as shown by the green-shaded row in Table 4-10. And because sysplex PRODM is the PrimaryPlex on all three CPCs, all three boxes can be aggregated together.

Table 4-10  
CPC C (Berlin): z990-308, 448 MSUs, prime shift 40 hour average:85% (381 MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODM</td>
<td>PRDC1</td>
<td>83</td>
<td>316</td>
</tr>
<tr>
<td>no sysplex</td>
<td>QAC1</td>
<td>17</td>
<td>65</td>
</tr>
</tbody>
</table>
Being able to aggregate CPC C with CPCs A and B has the effect of reducing the total software bill from 616,000 RedBits per month to 453,000 RedBits. In reality, assuming the customer is availing of subcapacity charging, he would see a further saving as a result of moving some of the Java work from the CPs to the zAAPs.

**Summary**

To summarize, then, the benefits this customer obtained by making the configuration changes discussed above are:

- In the near term, they will have a second CPC available to take over the workload from the Berlin system should the CPC it is running on require a planned outage or suffer a failure.
- As soon as they are able to implement the BronzePlex between system PRDC1 and the other PRODM sysplex systems, they will be able to aggregate CPC C with CPCs A and B, delivering an immediate software savings.
- In the medium term, the Berlin system will start benefitting from the availability advantages resulting from implementing data sharing and workload balancing for those applications.
- By installing zAAPs on CPC A, they have added the additional capacity required for the growth of their production systems, but without any increase in their software costs.

In fact, immediately following the installation of the two zAAPs, they should see a decrease in their software bill for CPC A because work that was previously running on a CP will be moved to the zAAPs. As the production workload grows to take up the capacity freed up by the zAAPs, the software bill will gradually rise back to current levels as the CPs get busier.

- By having all the production systems located within sysplex distance of each other, they have the infrastructure in place to eventually implement data sharing and dynamic workload balancing for all their production applications.

**4.4 Scenario 3 - aggregation isn’t always the right answer**

In this scenario we would like to show you a customer situation where the decision was made not to merge sysplexes, because the technical drawbacks were larger than the benefits of a potential BronzePlex.

**Initial configuration**

This customer had two data centers within sysplex distance of each other (7 kms). Each center contained one CPC and one standalone Coupling Facility. The existing CPCs use all available engines in today’s largest z990, with 16 CPs for z/OS workload, and 16 CPs reserved for disaster recovery using Capacity Backup Upgrade (CBU). Due to a strong business climate, the systems are growing and more capacity is needed.

The customer’s configuration is shown in Figure 4-5 on page 56. As before, LPARs with the same color are members of the same sysplex. The systems in each CPC are as follows:

- TST sysplex: LPARs TSTA1 and TSTB1
- DEV sysplex: LPARs DEVA1 and DEVB1
- PRD sysplex: LPARs PRDA1 and PRDA2 on CPC A, and LPARs PRDB1 and PRDB2 on CPC B
The numbers above the CPCs in the figure (95 and 85 in this case) indicate the average CPC utilization over the 40-hour prime shift. The CPCs are running at 100% busy for prolonged periods during the peak hours.

The customer has recently implemented data sharing and is experiencing additional overhead on the z/OS systems because of the relatively slow CFs (9672) being used.

More details about the configuration are contained in Table 4-11 and Table 4-12.

Table 4-11  CPC A: z990-316, 761 MSUs, prime shift 40 hour average: 95% (722MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRD</td>
<td>PRDA1</td>
<td>45</td>
<td>325</td>
</tr>
<tr>
<td>PRD</td>
<td>PRDA2</td>
<td>32</td>
<td>231</td>
</tr>
<tr>
<td>DEV</td>
<td>DEVA1</td>
<td>17</td>
<td>123</td>
</tr>
<tr>
<td>TST</td>
<td>TSTA1</td>
<td>6</td>
<td>43</td>
</tr>
</tbody>
</table>

Table 4-12  CPC B: z990-316, 761 MSUs, prime shift 40 hour average: 85% (647 MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRD</td>
<td>PRDB1</td>
<td>36</td>
<td>233</td>
</tr>
<tr>
<td>PRD</td>
<td>PRDB2</td>
<td>34</td>
<td>220</td>
</tr>
<tr>
<td>DEV</td>
<td>DEVB1</td>
<td>25</td>
<td>162</td>
</tr>
<tr>
<td>TST</td>
<td>TSTB1</td>
<td>5</td>
<td>32</td>
</tr>
</tbody>
</table>
Due to the fact that the production sysplex is the PrimaryPlex on both CPCs, the customer is able to aggregate across both CPCs. The software price for this configuration (using the software stack described at the beginning of this chapter) is 471,000 RedBits per month.

**Considerations**

Both CPCs are already running constantly at 100% busy for 2 to 3 hours every day, resulting in poor response times and elapsed times for the less important workloads. In addition, there are plans in place to roll out more applications on the production systems. And as the customer continues to expand their use of DB2 data sharing to more applications, that will also generate a demand for more capacity. Because the customer is already at the maximum size of a z990 (16 CPs plus 16 CBU PUs), they have no choice but to install one or more additional processors (remember that this happened before the announcement of System z9, when the z990 was the largest available zSeries CPC).

At the same time, they realize that something needs to be done about the Coupling Facilities. They already need to add more storage to the CFs in support of the existing data sharing workload. And as the load on the CFs increases, they see that the 9672s will no longer be capable of delivering the required performance. Therefore, the 9672s are going to have to be replaced with more powerful Coupling Facilities.

**Possible Options**

There are a number of ways the customer can address these requirements. However, there are some restrictions that they must live within:

- For disaster recovery purposes, they prefer to have an equal or similar number of MSUs in each site. This precludes the option of buying just one more CPC - whatever option they go for will have to include two similarly sized CPCs, one for each site.
- Their preference is to avoid creating any more production images. Due to a number of promotions and departures, their operators are not very experienced, so they would not like to ask them to manage any more production systems.
- They don’t want to mix production and non-production images in the same sysplex. They only recently completed a project to separate their production environment and don’t want to have to undo all that work.
- They are unhappy that the technology used for the CFs has fallen so far behind that used for the production systems, and would prefer that this not happen again in the future.
- They have investigated the use of System Managed Duplexing for the structures that require failure independence and have determined that the impact on response times caused by the distance between the sites (about 7km) would be more than their applications could handle. Therefore, they either have to buy two more standalone CFs, or they can use ICFs only if the CPCs containing the CFs do not contain any production systems.

Given these criteria, the customer looked at the following options.

**Option 1**

They could add two more CPCs, one in each data center, and move two of the existing production systems to the new boxes.

They would also have to purchase two new standalone Coupling Facilities to replace the two CFs.

This option had the advantage that they could aggregate all four CPCs. On the other hand, replacing the 9672 CFs with two more standalone CFs left them exposed once again to the CF technology in the future falling behind that used by the production systems.
**Option 2**

They could install two new CPCs with ICFs and migrate two of their production systems to the new CPCs.

This has the double advantage of being able to aggregate all four CPCs and also reducing the chances of the CF technology falling behind that of the production systems.

On the other hand, this configuration would require the use of System Managed Duplexing across the sites, and they have already determined that the response time achievable from that configuration would make it difficult to complete their batch work within the small (and shrinking) batch window.

**Option 3**

The third option was to install two new CPCs with ICFs and migrate the test and development systems to these CPCs.

This has the advantage that their is no need for System Managed Duplexing because the CFs would be failure isolated from the production systems. Also, it reduces the likelihood of the CF technology drifting behind again.

The disadvantage is a financial one - because there would be no production systems on the new CPCs, they could not be aggregated with the existing CPCs. They could, however, be aggregated with each other because the development sysplex would be the PrimaryPlex on each of the new CPCs.

**Implemented solution**

After prolonged discussions about the different options, the customer decided to go with option 3. The final configuration is shown in Figure 4-6.

![Figure 4-6 Implemented solution](image)

From a sysplex aggregation perspective, the customer now had two PricingPlexes instead of one (the old CPCs were not changed). Due to the production sysplex being the PrimaryPlex
on CPCs A and B, those CPCs form one PricingPlex. And because the development sysplex is the PrimaryPlex on CPCs C and D, they form another PricingPlex. The details about the new CPCs are listed in Table 4-13 and Table 4-14.

Table 4-13  CPC C: z990-304 (+ 2 ICFs), 248 MSUs, prime shift 40 hour average: 85% (211 MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>ICFC1</td>
<td>2 dedicated ICFs</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>DEVC1</td>
<td>82</td>
<td>173</td>
</tr>
<tr>
<td>TST</td>
<td>TSTC1</td>
<td>18</td>
<td>38</td>
</tr>
</tbody>
</table>

Table 4-14  CPC D: z990-304 (+2 ICFs), 248 MSUs, prime shift 40 hour average: 82% (203 MSUs)

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>-</td>
<td>ICFD1</td>
<td>2 dedicated ICFs</td>
<td>0</td>
</tr>
<tr>
<td>DEV</td>
<td>DEVD1</td>
<td>76</td>
<td>155</td>
</tr>
<tr>
<td>TST</td>
<td>TSTD1</td>
<td>24</td>
<td>49</td>
</tr>
</tbody>
</table>

Obviously these changes will result in an increased software bill. Even if CPCs C and D were in the same PricingPlex as CPCs A and B, there would have been an increase because more capacity is being used, however the increase would have been smaller. The actual new costs for software are 730,000 RedBits per month, compared to 557,000 RedBits per month if all CPCs had been in the same PricingPlex. But remember that, in practice, they would be using subcapacity charging, so the actual bill for the CPC A+B PricingPlex would drop to some degree when they move the test and development systems off those CPCs.

4.5 Scenario 4 - More for less?

In this final scenario, the customer has two CPCs in the same data center. They have four sysplexes spread over the two CPCs, but unfortunately none of the sysplexes account for more than 50% of the MSUs used by MVS-based LPARs on that CPC, meaning that they cannot aggregate the two CPCs. Based on this configuration, the current software cost for the software stack defined in 4.1, “Introduction” on page 46 is 399,000 RedBits per month.

As part of an ongoing cost saving effort, two things are happening:

- Management have heard about the concept of sysplex aggregation and want the technicians to review the possibility of aggregating the two CPCs.
- The company is a large user of Lotus Notes®, with 15,000 Notes users supported by a growing population of PC servers (currently sized at just over 300 servers). While the unit cost per PC is low, the cost of replacing all 300 servers is significant. More importantly, the cost of managing and maintaining them has reached the point where it is a serious management concern. Therefore, management have requested that consolidation options be investigated and costed.

Initial configuration
The sysplex configuration at the beginning of this project is as follows:

- Test sysplex: Systems TSTA1 (on CPC A) and TSTB1 (on CPC B)
- Development sysplex: Systems DEVA1 (on CPC A) and DEVB1 (on CPC B)
- Quality assurance sysplex: Systems QAA1 (on CPC A) and QAB1 (on CPC B)
- Production sysplex: Systems PRDA1 (on CPC A) and PRDB1 (on CPC B)

The configuration is shown diagramatically in Figure 4-7. It is also shown in more detail in Table 4-15 and Table 4-16.

![Figure 4-7 Configuration at start of project](image)

**Table 4-15 CPC A: z990-305, 302 MSUs, prime shift 40 hour average: 81%, (244 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODN</td>
<td>PRDA1</td>
<td>45</td>
<td>110</td>
</tr>
<tr>
<td>QAN</td>
<td>QAC1</td>
<td>28</td>
<td>68</td>
</tr>
<tr>
<td>DEVN</td>
<td>DEVA1</td>
<td>20</td>
<td>49</td>
</tr>
<tr>
<td>TSTN</td>
<td>TSTA1</td>
<td>7</td>
<td>17</td>
</tr>
</tbody>
</table>

While the PRODN sysplex is the largest sysplex on CPC A, it isn’t large enough to consistently account for over 50% of the used MVS-based MSUs.

**Table 4-16 CPC B: z990-306, 352 MSUs, prime shift 40 hour average: 85% (300 MSUs)**

<table>
<thead>
<tr>
<th>Sysplex Name</th>
<th>System Name</th>
<th>% of total MVS-based MSUs used</th>
<th>Number of MVS-based MSUs consumed</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRODN</td>
<td>PRDB1</td>
<td>46</td>
<td>138</td>
</tr>
<tr>
<td>QAN</td>
<td>QAB1</td>
<td>24</td>
<td>72</td>
</tr>
<tr>
<td>DEVN</td>
<td>DEVB1</td>
<td>25</td>
<td>75</td>
</tr>
<tr>
<td>TSTN</td>
<td>TSTB1</td>
<td>5</td>
<td>15</td>
</tr>
</tbody>
</table>
Similarly, the PRODN sysplex is the largest sysplex on CPC B, but again it isn’t large enough to consistently account for over 50% of the used MVS-based MSUs.

**Implemented solution**

The customer is running DB2 data sharing in the production sysplex in order to meet their availability targets. Due to past unpleasant experiences, compounded by concerns by the companies auditors about mixing production and non-production data in the same environment, the system programming department is not in favor of merging the production sysplex with any of the other sysplexes. While a BronzePlex might provide the ability to bring more systems into the same sysplex while maintaining a degree of separation, it would provide no benefits other than a reduction in software costs. So, it appeared that there was no solution to satisfy both the technicians and the finance people.

However, remember that we said that there was also a request to investigate some way to reduce costs by consolidating the Notes servers onto another platform. A Total Cost of Ownership (TCO) study was done, which showed that the current mainframe environment could be cost effectively upgraded in a way that would reduce the cost of running the Notes service and help them achieve the goal of consolidating the two CPCs by pushing the Production sysplex utilization above 50% of the used MVS-based MSUs on both CPCs.

Specifically, the decision was made to make the following changes to the mainframe environment:

- Upgrade the two CPCs by adding two CPs to each CPC.
- Install Lotus Notes under z/OS in two new systems (LNA1 and LNB1), one on CPC A, one in CPC B (for availability purposes). These new systems would be part of the Production sysplex.
- Migrate all Lotus Notes users from the 300 old PCs to the Lotus Notes z/OS systems.
- The target configuration, together with projected utilizations is shown in Figure 4-8.

```
<table>
<thead>
<tr>
<th></th>
<th>CPC A z990-307</th>
<th>CPC B z990-308</th>
</tr>
</thead>
<tbody>
<tr>
<td>TSTA1</td>
<td>15%</td>
<td>TSTB1</td>
</tr>
<tr>
<td>DEVA1</td>
<td>21%</td>
<td>20%</td>
</tr>
<tr>
<td>QAA1</td>
<td>34%</td>
<td>19%</td>
</tr>
<tr>
<td>PRDA1</td>
<td>25%</td>
<td>PRDB1</td>
</tr>
<tr>
<td>LNA1</td>
<td>81%</td>
<td>LNB1</td>
</tr>
<tr>
<td>LNB1</td>
<td>85%</td>
<td></td>
</tr>
</tbody>
</table>
```

*Figure 4-8  Installation after migration of all Lotus Notes Users to z/OS*

After the new CPs have been installed and all users have been migrated, the situation from a sysplex aggregation point of view is shown in Table 4-17 on page 62 and Table 4-18 on page 62.
The green shaded rows show the LPARs that constitute the PrimaryPlex on each of the CPCs. Together the LPARs PRDA1 and LNA1 use more than 50% of the consumed MVS MSUs on CPC A, and LPARs PRDB1 and LNB1 do the same on CPC B.

Based on these projections, the following is the result of this project:

- The customer was able to avoid having to replace 300 PC servers. The staff that were previously supporting those servers have been re-assigned to other projects that were understaffed and actively recruiting.
- The mainframes have additional capacity that can potentially benefit any of the z/OS workloads that may require it at a given time.
- Despite the fact that each CPC has more than 100 more MSUs additional capacity, the monthly software costs for the two CPCs have been reduced from 399,000 to 359,000 RedBits per month. In fact, in a real situation, the customer would exploit sub-capacity charging to only pay for Notes in the two LPARs it is running in, and by not paying for CICS or DB2 in either of those LPARs. So, the software savings would probably work out at even more than 40,000 RedBits. And that is excluding the savings associated with being able to eliminate over 300 PC servers.

### 4.6 Summary

As you can see from the four customer examples, there is no single answer that is right in every case. Sometimes the answer is as simple as re-balancing your existing systems and workloads. Other times, you need to be a bit more inventive. In some cases, creating a BronzePlex is the correct course of action, and in other cases, the correct thing to do is to leave things as they are and accept higher software costs in return for better availability or performance.
In order to determine the right action for your situation, you must carefully analyze your current and projected configuration, along with various requirements such as reduced costs, improved security, better availability, delivering a disaster recovery capability, and so on. The following suggestions may help you arrive at the right answer:

- We recommend creating a color-coded spreadsheet showing each of your CPCs and the LPARs within each CPC. Use the colors to identify LPARs that are in the same sysplex. The spreadsheet should contain the capacity of each CPC, along with the number of MSUs used by each LPAR, and the percent of MVS-based MSUs that each LPAR accounts for. You can use the PLEXCALC tool to help you determine these numbers.

Having such a spreadsheet is invaluable if you want to project the impact of moving systems from one CPC to another. We have included an example in Figure 4-9.

![Figure 4-9 Example of a spreadsheet to check the aggregation](image)

- If you cannot qualify through workload balancing or moving systems between CPCs, then determine the risk associated with merging un-related systems into one sysplex. But please don’t consider a BronzePlex to be the final result of a merge as you will more than likely end up going through this exercise all over again when you have the next round of CPC upgrades.

- Before adding ZAAPs, project the impact this will have on the utilization of all MVS-based LPARs on that CPC, especially if the PrimaryPlex is already close to the 50% threshold.

- Use workload balancing mechanisms (as described in Chapter 5, “Workload balancing and sysplex aggregation” on page 65) wherever possible to use the system to help you automatically balance the work, and reduce or eliminate the re-balancing effort for future CPC upgrades or changes.

Review the list of workload balancing features and workload managers in Chapter 5, “Workload balancing and sysplex aggregation” on page 65 to determine if you are exploiting all the capabilities that are potentially available to you.
Before going to the effort of merging systems into a BronzePlex, investigate whether consolidating work from other platforms can help you achieve the aggregation criteria.
Workload balancing and sysplex aggregation

This chapter describes the various IBM products and functions that control or influence where each incoming piece of work gets executed. In each case, we discuss the criteria that are used to determine where the work should be routed, and if it is possible to influence those decisions. We also discuss related system attributes and functions (like Intelligent Resource Director and Defined Capacities), and how they relate to the objective of balanced workloads and CPU consumption.
5.1 Introduction

The main reason for striving to achieve sysplex aggregation is the cost savings that can potentially be obtained. The single most effective way to reduce your IBM mainframe software costs is through sysplex aggregation if your configuration and workload profile support this capability.

However, the “cost of ownership” of any system consists of more than just hardware or software costs—people costs are typically the largest and fastest growing part of most IT budgets. Therefore, it is important not only that you are able to configure your systems so that they meet the sysplex aggregation criteria; you also want to be able to maintain that qualification with as little manual intervention as possible. This is where dynamic workload balancing comes into the equation. There are obviously other benefits from dynamic workload balancing—improved availability, potentially better performance, better return from your hardware investment by letting you use that resource wherever it may be—however in this document our primary concern is helping you obtain and retain sysplex aggregation qualification with minimal manual intervention.

We want to stress at this point that workload balancing is not a magic panacea that will ensure that you will continue to meet the qualification criteria with no manual intervention. Unfortunately, sysplex aggregation is a complex topic, and none of the workload balancing components are designed specifically to help you meet the criteria. Broadly speaking, there are three models for workload balancing:

**Capacity based**  
This model attempts to direct work to whichever system has the most available capacity (in terms of MSUs).

From a sysplex aggregation perspective, the restriction with this model is that it will not on its own result in balanced utilization if the CPCs are of significantly different capacities.

**Round robin**  
This model attempts to route an equal number of work requests to each server.

Once again, from a sysplex aggregation perspective, the limitation with this model is that it will not on its own result in balanced utilization if the CPCs are of significantly different capacities because a similar amount of work will result in lower utilization on the larger CPC.

**Goal achievement**  
This model attempts to route work to the server that is currently delivering performance that meets or exceeds the installation’s objectives.

From a sysplex aggregation perspective, the problem with this model is that faster CPCs will deliver better response times at higher utilizations than slower CPCs, once again resulting in uneven utilization across the CPCs.

Despite these limitations, it is still an excellent investment to use workload balancing to distribute work around the members of the sysplex. If you remember, in 3.2.3, “Maintaining sysplex aggregation qualification” on page 37, we discussed how it is easier to achieve balanced utilization when work can be moved in units of transactions rather than complete database managers and all the associated batch work and transaction managers. It is very common in customers that are having difficulty achieving sysplex aggregation to see very unbalanced utilizations across the CPCs; these typically result from large monolithic workloads that cannot be easily broken up (usually because of affinities to a particular piece of data or service).
By distributing work across all the CPCs in the PricingPlex you will probably have a starting point that is much closer to your objective, and this sound foundation will be delivered by the system rather than manually by humans.

In this chapter we discuss all the IBM products and components that have a role in workload balancing. Your objective should be to exploit the functions that are appropriate for your environment and workload so that work is automatically distributed around the sysplex, and then use the tuning “knobs” we will discuss (like LPAR weights) to fine tune the balance of work across the members of the sysplex. This delivers many benefits, including:

- You can maintain sysplex aggregation qualification with far less ongoing manual intervention.
- You will have better protection from unplanned outages as your critical workloads will be spread across multiple, failure-isolated, servers.
- Both planned and unplanned outages will be more transparent to users - when users logon or submit an item of work, the system will automatically route it to an available server. As long as at least one server is available within the sysplex, users should not even be aware that there is an outage.
- Because work can execute anywhere in the sysplex, you can more fully utilize the available capacity across all CPCs. It is not unusual when customers are not using workload balancing to see them upgrading one CPC while there is still unused capacity on another CPC in the sysplex.

Before we get into discussing each of the components, we start with a brief discussion about the role of the MVS Workload Manager in workload balancing. Many of the products we discuss work with WLM when deciding where work should be processed. And even among the products that don’t use WLM, many provide a user exit where you can influence the routing decisions; those exits could potentially call a WLM service to gather information that could then be used in the routing decision.

### 5.2 MVS Workload Manager

When discussing the MVS Workload Manager (WLM) component in relation to workload balancing, many people think that WLM has an objective to provide balanced utilization across the various members of the sysplex. In fact, this is a falsehood. WLM’s objectives actually are:

- To manage the resources of the sysplex so that workloads achieve their performance and throughput objectives, based on their importance as specified by the installation,
- To attempt to ensure that resources are not over-committed to the extent that performance is impacted.
- To optimize and maximize the use of sysplex resources in order to get the work through the system/sysplex as quickly as possible.

You will notice that there is no mention of *balanced CPU utilization* in this list. Imagine that you have a 9672-R23 and a System z9 in the same sysplex - all other things being equal, the z9 will provide far better performance than the 9672, so balancing utilization across both CPCs would certainly not help achieve the first or third of the three objectives above.

Another common misconception is that WLM *pushes* work to the various work managers in the sysplex. Again, this is generally not true. WLM does provide capacity information to some of the workload routers (VTAM Generic Resources and TCP/IP Sysplex Distributor, for example). However, it is up to those routers, and how you customize them, whether they even use that information or not.
An important thing to understand in the relationship between WLM and the workload routers in the concept of available capacity. You must remember that WLM's view of the world (or at least, his world), extends beyond the LPAR that it is executing in. WLM also has access to information, provided by PR/SM™, about the weights and utilization of all the other LPARs on the same CPC. And in addition to that, it also has information about the utilization and goal attainment of all the other systems in the same sysplex that it gets from the WLMs on the other systems.

So, when WLM is calculating where there is available capacity to run future work, it looks at the following:

- The share of the CPC that this LPAR is entitled to, based on its weight and the number of online logical CPs in the LPAR.
- The actual utilization of this LPAR over the last few minutes.
- The total amount of unused capacity on this CPC, based on the total CPC capacity and the actual utilization of all LPARs on this CPC.

All this information is input into a complex calculation. However, to grossly over-simplify things, the available capacity is determined to be:

- The difference between the actual utilization of this LPAR and the capacity it is guaranteed as a result of its weight, bearing in mind its number of online logical CPs.
- Plus some portion of the total unused capacity on the CPC, apportioned based on the relative weight of this LPAR compared to the other LPARs on the CPC.
- In some cases, WLM also takes into account the importance of the new work request, and includes capacity used by lower importance workloads in the available capacity value (because that lower importance work could be displaced to cater for the new, higher importance, work).

The resulting available capacity number (calculated in terms of MSUs) is then used in various ways to provide information to the workload routers.

One very important thing that you must understand is that sysplex aggregation is a financial entity. There is no technical reason for its existence. It does not (directly) help you get better performance, or cause more work to be processed. Therefore, it is a concept that WLM knows nothing about. As a result, there is no function in WLM to say “distribute work so that this sysplex uses more than 50% of the used MVS MSUs on each CPC”.

Also, while WLM knows where all the systems in the sysplex are running, it makes no differentiation between two systems running on the same CPC, and two running on different CPCs. This means that WLM does not do any calculations along the lines of “I see that two systems in my sysplex are using more than 50% of the used MSUs on CPC X, so I will favor systems on other CPCs for future work”.

Finally, before we move on to talk about the individual workload routers and workload managers, we want to make it clear that WLM on a given system does not make changes to other systems. All WLMs in the sysplex are peers - there is no concept of a “master” WLM. WLM on every system in the sysplex has information not only about the work on its own system, it also knows what is going on in every other system in the sysplex. However, WLM on one system will never, for example, start initiators on another system. What it may do sometimes is say “I am full, and there is capacity over on system B, so I will stop some initiators, and let my peer over on system B start some initiators over there”. WLM on system B, when it sees that WLM on system A is not starting any more initiators, may then react by starting his own initiators.
More information about WLM is available on the WLM Web site at:

5.3 Workload balancing components

There are two aspects to fully exploiting Parallel Sysplex:

- Data sharing, which provides the ability to run any work on any system in the sysplex.
- Dynamic workload balancing, which takes incoming work requests (such as CICS or IMS transactions or batch jobs) and routes them to an appropriate system for execution.

In order to be able to fully exploit dynamic workload balancing, you want every piece of work to potentially be able to run on any system in the sysplex. This effectively means that a prerequisite for effective workload balancing is data sharing. However, data sharing implementation is comprehensively covered in product manuals and other Redbooks (a number of which are listed in the bibliography) so we will not discuss this further in this document.

In this section, we list the workload balancing components ordered by subsystem type (CICS, DB2, IMS, and so on), starting off with VTAM Generic Resources and Sysplex Distributor as these interact with most of the subsystems and play a vital role in determining which server a user will be dispatched on when they first logon.

5.3.1 VTAM Generic Resources

The VTAM Generic Resources function improves availability and potentially balances the SNA session load across the systems in a sysplex. It does this by treating a group of application servers as a single entity from the user’s point of view and dynamically determining where each new session request should be routed.

For example, you could have three CICS regions, with VTAM APPLIDs of CICSPRD1, CICSPRD2 and CICSPRD3, and define a generic name of CICSPROD for this group of regions. When a user is logging on, they would simply specify CICSPROD as the target application in the logon request. As long as at least one of the three regions is available, the logon request will be successful. And if more than one region is up, VTAM will use one of a number of balancing algorithms to distribute the loads across the available regions.

Figure 5-1 on page 70 shows the elements of VTAM Generic Resources.
WLM interactions

Whenever an application starts and indicates that it wishes to have a generic resource name, VTAM registers that instance with WLM. As a result, when VTAM subsequently asks WLM for a recommendation about which application instance in a generic resource group to send a request to, WLM understands which address spaces are in the generic resource group and will send back a recommended instance based on the goal achievement of each of the members of the generic resource group.

Every time VTAM attempts to establish a new session in response to a logon request using a generic resource group name, it calls WLM for a recommendation about where to route a session request to. In response, WLM returns the name of just one instance. The criteria WLM uses to select which instance are:

- The goal achievement of each of the members of the group
- If all instances have the same goal achievement, WLM will prefer the instance with the smaller number of sessions

WLM refreshes this information once per policy adjustment interval (currently once every ten seconds).

User control

The actions VTAM takes when setting up a new session to a generic resource are as follows:

1. If there is an existing affinity between the LU and an instance of the target application, VTAM will select that instance.

2. VTAM’s default mode is to prefer local instances of the target application (that is, an instance on the same z/OS system as VTAM). This default can be changed by turning off bits GRRFNPLA and GRRFNPLL in the VTAM Generic Resource user exit (ISTEXCGR). However, because the exit is only called in step 5 below, changing the setting of the bits at that point will only affect how VTAM behaves for the next request, not the current one. As a result, the most common way to use this would be to set the bits once to determine how VTAM behaves for all applications for all future requests.

3. VTAM next determines which instance of the GR application has the fewest sessions. If VTAM has been told not to use WLM (by turning off bit GRRFWL MX in ISTEXCGR) it will route the session to that instance.
4. VTAM then calls WLM, passing the generic resource name and other information.

If you let VTAM prefer a local instance, it will check how many sessions there are for each of the local instances, and then send a list of all the local instances to WLM, requesting it to recommend one from that list.

If you override VTAM’s default so that it treats a local instance the same as remote ones, VTAM will send a list of all the instances across the sysplex. In response WLM returns the APPLID of the instance that is best achieving its objective.

5. Finally, VTAM calls the ISTEXCGR exit. In this exit you can specify that VTAM should ignore the WLM recommendation and instead use its own recommendation based on the number of sessions in each instance. If you wish, you could also add code to make your own decision about where to route the request, ignoring both WLM’s and VTAM’s recommendations.

For most customers, the decision is whether to balance based on the goal achievement of the available instances (the default) or based on the number of sessions with each instance (by telling VTAM to ignore the WLM recommendation). If you are interested in exploiting VTAM GR’s workload balancing capability, you will probably want to override its preference for local instances by coding the ISTEXCGR exit - a sample is provided in SYS1.SAMPLIB.

If you feel that more control is required, or that a different algorithm would be more appropriate in your environment, you have the option of using the ISTEXCGR exit. In this case, you might wish to look at the sample ISTEXCUV exit provided and described in the IBM Redbook *Managing Software for Availability in a Parallel Sysplex*, SG24-5451, to get some ideas of the type of function you might implement.

If you do not use VTAM Generic Resources, there is a VTAM exit you can use, together with VTAM USERVAR support, to distribute session requests across a number of application instances. The exit is ISTEXCUV, and it is possible to dynamically alter the exit to control the percent of requests sent to each instance. Note that this exit only applies to applications that are not using the Generic Resources function.

**Further information**

For more information on setting up and using VTAM GR and/or the VTAM USERVAR support, refer to:

- IBM Redbook *Managing Software for Availability in a Parallel Sysplex*, SG24-5451
- IBM Redbook *SNA in a Parallel Sysplex*, SG24-2113
- IBM Redbook *Using VTAM Generic Resources with IMS*, SG24-5487
- z/OS Communications Server: *SNA Network Implementation Guide*, SC31-8777

**5.3.2 Sysplex Distributor**

Sysplex Distributor is a function introduced in IBM Communications Server for OS/390 V2R10. It builds on the XCF dynamics and Dynamic Virtual IP Address (VIPA) support that was already available in OS/390 to deliver improved availability and workload balancing for TCP/IP applications in a sysplex. Using Sysplex Distributor, a TCP/IP workload can be distributed to multiple server instances in the sysplex without requiring changes to clients or networking hardware and without delays in connection setup. Comms Server for z/OS provides the ability to implement a dynamic VIPA as a single network-visible IP address for a set of hosts that belong to the same sysplex. Any client located anywhere in the IP network sees the sysplex cluster as one IP address, regardless of the number of hosts that it includes.
With Sysplex Distributor, clients receive the benefits of workload distribution provided by both WLM and the Quality of Service (QoS) Policy Agent. In addition, Sysplex Distributor ensures high availability of the IP applications running in the sysplex, even if a physical network interface fails or an entire IP stack or z/OS is lost.

**WLM interactions**

When an IP connection request arrives for an application that is exploiting the Sysplex Distributor support, the Sysplex Distributor routing stack does not complete the connection itself as it usually would. Instead, it consults WLM to obtain capacity and goal achievement information about each application instance. The routing node also consults the Service Policy Agent for network performance and defined policies that might affect the distribution decision. How this information is used depends on the release of z/OS you are using, and the options you have specified on the IP definitions.

Prior to z/OS 1.5, WLM provided Comms Server with available capacity information about each system containing an instance of the target application. Sysplex Distributor would then distribute incoming requests based on the available capacity in each system. So, for example, if there are just two systems in the sysplex, and system A has 100 spare MSUs and system B has 50 spare MSUs, Sysplex Distributor would send roughly twice as many requests to system A as to system B. This behavior is the default as long as all systems are in WLM Goal Mode and SYSPLEXROUTING has been specified in every stack.

In z/OS 1.5, it was recognized that you may not always want to distribute sessions in this manner. For example, in some cases you might prefer to have an even distribution of sessions across all available servers, regardless of the available capacity on each. To enable this capability, a new keyword, DISTMETHOD ROUNDROBIN, was introduced on the VIPADISTRIBUTE statement.

Then in z/OS 1.7, the interface to WLM was further enhanced with the option to get information about the goal achievement of the individual servers, rather than available capacity information about each system as was the case previously. To use this distribution mechanism, you must indicate that you wish to use DISTMETHOD SERVERWLM rather than BASEWLM or ROUNDROBIN.

Note that regardless of which distribution method is used, Sysplex Distributor only calls WLM once a minute to obtain an updated list of routing recommendations.

In addition to balancing the requests across the systems in the sysplex, Comms Server also provides the ability to have multiple servers on a single system listen on the same port, thereby balancing the requests to a given system across the servers in that system. To enable this in the TCP/IP stack, you must add the SHAREPORT keyword to the PORT profile statement that reserves the TCP port for the server instances. In z/OS 1.7, this capability has been enhanced to allow Sysplex Distributor to target a specific server instance when using the SERVERWLM option. To enable this capability when multiple servers are sharing a port, replace the SHAREPORT keyword with SHAREPORTWLM on the PORT statement.

With all available relevant information, including precisely which target stacks have server applications ready for work and bearing in mind the distribution method you specified, the routing stack selects a target stack and forwards the request to that target stack for processing. The routing stack also remembers the full connection information (source and destination IP addresses and ports), so that future TCP segments for that connection are sent.

---

1 Prior to z/OS 1.7, WLM only provided information about the available capacity on each system containing an application instance. The information about goal achievement of the instances is new in z/OS 1.7.

2 The DISTMETHOD option does not have any effect on incoming connection requests that have an active affinity established to a specific server instance (by way of the TIMEDAFFINITY parameter). When an affinity exists, it has priority over the DISTMETHOD setting.
on to the same target stack for processing. When the connection ends, the target stack notifies the routing stack that the connection has ended, so the routing stack can discard the connection routing entry for the ended connection. The next connection from the same client will be distributed independently of the earlier one.

**User control**

There are no Comms Server user exits to let you influence the selection of target stack.

However, if you define a Sysplex Distributor performance policy on the target stacks, it is possible to take network performance into account when making routing decisions. You can also define a Sysplex Distributor routing policy on the distributing host to make adjustments to the distribution decisions.

If your objective is to use Sysplex Distributor to effect a more even distribution of CPC utilization across the sysplex, you will probably want to use the BASEWLM distribution method. While this mechanism does not ignore server performance, it does place more emphasis on sending work to whichever system has the most spare capacity.

**Further information**

For more information about Sysplex Distributor, how it works, and how to implement it, refer to:

- IBM Redbook *TCP/IP in a Sysplex*, SG24-5235
  - At the time of writing, a new Redbooks about TCP/IP on z/OS is planned to be available in 2006 - the title is not available at this time, but the book number will be SG24-7171.
- The chapter entitled “TCP/IP in a Sysplex” in *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communication Server: New function summary*, GC31-8771

### 5.3.3 Load Balancing Advisor

The z/OS Load Balancing Advisor, introduced in Communications Server for z/OS 1.6, communicates with external load balancers and one or more Load Balancing Agents. The main function of the Load Balancing Advisor is to provide external TCP/IP load balancing solutions, such as the Cisco Content Switching Module (CSM), with recommendations on which TCP/IP applications and target z/OS systems within a sysplex are best equipped to handle new TCP/IP workload requests. These recommendations can then be used by the load balancer to determine how to route new requests to the target applications and systems (that is, how many requests should be routed to each target). The recommendations provided by the Advisor are dynamic, and can change as the conditions of the target systems and applications change.

The Load Balancing Advisor provides information to the external load balancers that is similar to the information that is available to Sysplex Distributor. Specifically, it provides the following information:

- Status information about each target system and application, allowing the load balancer to bypass systems or application instances that are currently not available.
- Information from WLM about the available capacity on each system in the sysplex.
- Information from WLM about the goal achievement of each server address space.
- TCP/IP network performance within the sysplex.
WLM interactions
Just as Sysplex Distributor can get information from WLM about various aspects of performance and capacity, equally Load Balancer can obtain the same information and pass it on to the external load balancer. Specifically, it is obtains the following information from WLM:

WLM system-level recommendations
WLM provides Load Balancing Advisor with weighted recommendations for each appropriate system in the sysplex. The WLM recommendations are derived using several measures, including each system's available CPU capacity, or capacity that can be displaced by higher importance workloads. The latter is important for scenarios where systems might be 100% utilized, but some might be running work that has a lower importance (as defined in the WLM policy) and can therefore be displaced by higher importance workload.

WLM server-level recommendations
WLM provides information about goal achievement for every potential target server address space in the sysplex. If a system contains multiple potential target servers, information about each one will be returned.

The default interval for Load Balancing Advisor to update its information is once a minute, however this can be overridden in the Load Balancing Advisor customization.

User control
The only level of control you have over the information provided to the external load balancers by the Load Balancing Advisor is whether the recommendations will be based on available capacity or goal achievement. It is not possible in the Load Balancing Advisor to override or adjust the WLM recommendation.

Use the optional \texttt{wlm} Load Balancing Advisor customization statement to specify the default type of WLM recommendation to be used for all groups. There are two choices for this, the \texttt{BASEWLM} and \texttt{SERVERWLM} values. If you do not specify this statement, the default is \texttt{BASEWLM}. If you want a specific group of applications to use a type of WLM recommendation other than the default, you can override the default WLM recommendation type for that group on a port number basis using the \texttt{port\_list} statement. The WLM recommendation is a key component used in determining the net weight assigned to a member.

Further information
For further information on setting up and using the Load Balancing Advisor, refer to:

- The chapter entitled “z/OS Load Balancing Advisor” in \textit{z/OS Communications Server: IP Configuration Guide}, SC31-8775

5.3.4 WLM/DNS
WLM/DNS provided intelligent sysplex distribution of requests through cooperation between the WLM and the DNS server. For customers who elect to place a DNS in a z/OS sysplex, DNS could invoke WLM sysplex routing services to determine the best system to service a given client request.

However, z/OS Communications Server removed support for the BIND DNS 4.9.3 function in z/OS 1.7, thereby removing the WLM/DNS support at the same time. BIND DNS 9.2.0 has been available since z/OS V1.4, and customers should implement this version as a
replacement for BIND DNS 4.9.3. Customers that use the WLM/DNS feature of BIND DNS 4.9.3 should investigate Sysplex Distributor or alternative solutions.

5.3.5 WLM-Managed initiators

WLM provides a mechanism for classification of work and controls the allocation of system resources (for example, CPU and memory) to units of work during the execution phase. However, prior to the introduction of WLM-Managed initiators, WLM had no control over when or where batch jobs would be started, and had to work in reaction mode as JES placed jobs into execution independent of current conditions.

Beginning with OS/390 V2R4, you have the option of using a set of initiators where WLM, rather than JES, controls how many are active, and the rate at which queued jobs are initiated.

WLM interactions

JES has been extended to classify batch jobs following converter processing and to provide job backlog information to WLM, by service class, on each image in the JES MAS. WLM acquires the job backlog information and uses it in determining whether or not installation goals are being met. WLM starts, stops, or changes job selection criteria for WLM-Managed initiators based upon availability of work, goal attainment, and processor resource availability within the sysplex. For example, if WLM determines that a more important workload, like CICS, is missing its goals on a system because it is being delayed by a resource that batch is using a lot of, WLM will reduce the number of available WLM-Managed initiators on that system, thereby reducing the number of batch jobs being run.

In keeping with its goal to maximize system throughput, WLM will attempt to keep the CPU as busy as possible, and may start new initiators (as appropriate, based on goal achievement) until the available capacity in that system is less than 5%. At that point, WLM will check to see if other systems in the JES MAS have more spare capacity. If they do, it will stop WLM-Managed initiators on this system. The result is that the goal achievement of the service class used for batch jobs will decrease. This in turn will prompt WLM on one of the other systems in the JES MAS to start more initiators. Note, however, that WLM on one system will not direct WLM on a different system to start more initiators. Each WLM operates semi-independently, reacting to a drop in the goal achievement of the service class at the sysplex level. Also, it is important to understand that in relation to WLM-Managed initiators, WLM only controls the stopping and starting of its initiators. If a job in a WLM-Managed initiator ends and there is an appropriate job waiting to run, that job will be selected by the initiator - WLM is not involved in the decision to start that job. This behavior is currently the same for both WLM- and JES-Managed initiators.

User control

WLM-Managed initiators are a far more effective way of maximizing batch throughput and system capacity than traditional JES initiators. JES has no understanding of the capacity of the system; all it knows is how many initiators the installation has defined for each class, and whether they are busy or not. If the CPU is completely over-committed, but there are available initiators, JES will happily start more jobs. Equally, if all the initiators are being used, but the CPU is only 50% busy, JES will be unable to start more work to utilize the available capacity. So, in general, WLM-Managed initiators are more effective at exploiting available capacity, and the experiences of customers that exploit them has generally been positive. Note that there are some special cases where JES-managed initiators continue to be more suitable, for example for jobs that must be started immediately they are submitted. These special cases are documented in the IBM Redbook z/OS Systems Programmers Guide to: Workload Management, SG24-6472.
Having said that, however, the installation has very little control over the flow of work in WLM-Managed initiators. It is possible to control the maximum number of WLM-Managed initiators at the MAS level, but not at the individual system level. You can stop all the WLM-Managed initiators on a system, using the SP XEQ JES command, however there is no way to stop just one or more WLM-Managed initiators on a given system.

Possibly a better way to control the flow of batch jobs to different systems in the MAS is by using WLM Scheduling Environments - these are discussed in 5.3.6, “WLM Scheduling Environments” on page 76. However, we recommend that Scheduling Environments are used as well as WLM-Managed initiators, not instead of them. As we noted previously, WLM-Managed initiators are far more effective at optimizing the balance of throughput and utilization than traditional JES-managed initiators.

There are no WLM user exits that allow you to influence the number of WLM-Managed initiators that are started on a system.

**Further information**

For more information on WLM-Managed initiators, refer to:

- IBM Redbook *z/OS Systems Programmers Guide to: Workload Management*, SG24-6472
- WSC White Paper on WLM Managed initiators, available at:  
- *z/OS MVS Planning: Workload Management*, SA22-7602

### 5.3.6 WLM Scheduling Environments

WLM Scheduling Environments provide an easy way to control which systems in a sysplex are eligible to run a batch job. In the batch job you specify a Scheduling Environment name on the job card (with the SCHENV parameter). The job can then only run on systems where that Scheduling Environment has a status of “ON”. The status of the Scheduling Environment is set to RESET when the system is IPLed, and the operator (or automation) then sets the status to ON or OFF using the MODIFY WLM command.

To affect the workload balance, you could switch the Scheduling Environment resource “ON” on the systems which have a lower utilization and “OFF” on the systems which have a higher utilization. There are some limitations in doing this:

- You can only specify one Scheduling Environment per job. So if you already use Scheduling Environments for directing jobs to specific systems (or if you plan to do this in the future), then using them for workload balancing as well can become quite complex - you would need to create rules for your automation specifying which subset of systems in the sysplex are candidates to have a given Scheduling Environment turned ON.
- In order to use this facility to have a tangible impact on utilization it is not feasible to do this manually. This methodology of workload balancing should only be used in conjunction with a mature system automation.
- Your implementation should cater for the importance of the batch jobs associated with each Scheduling Environment: you don’t want to end up with all the high importance jobs running in one system and all the low importance or discretionary jobs running in another.
- Your automation should link the monitoring of available capacity with the turning ON or OFF of Scheduling Environments. You can write some code to use the WLM IWMWSYSQ service to extract this information - a sample program is contained in Appendix B, “Sample IWMWSYSQ code” on page 103.
When altering the status of a Scheduling Environment, you must be conscious of the JES topology. If the JES MAS or complex maps one-to-one with the sysplex, there are no additional considerations.

However if you have more than one MAS or complex in the sysplex and you use the same Scheduling Environment names across the entire sysplex, you need to ensure that every MAS or complex has at least one system where each Scheduling Environment is ON.

### WLM interactions

Scheduling environments must be defined in the WLM service policy. The names can be up to 16 characters long. The turning ON or OFF of the Scheduling Environments is handled by operator commands. There is no function to have WLM automatically alter the status of a Scheduling Environment.

### User control

The status of a Scheduling Environment is only altered by operator command (or by an IPL, in which case the status is set to RESET). Therefore everything in relation to Scheduling Environments is under user control. The considerations relating to how Scheduling Environments can be used to assist with workload balancing were discussed previously in this section.

### Further information

For more information about WLM Scheduling Environments, refer to:

- z/OS MVS Planning: Workload Management, SA22-7602

### 5.3.7 WLM Resource Groups

WLM resource groups are a way of limiting the amount of service the associated service classes can receive, or of guaranteeing a minimum amount of service to the service classes. While resource groups do have a role in workload management, we feel that they are not currently appropriate tools to use in the context of workload balancing. The problem is that resource groups currently work at the sysplex level; the amount of service delivered to the resource group is measured and controlled at the sysplex level. There is currently no way to tell WLM “give this service class x amount of service on SYSA and y amount of service on SYSB”.

As a result, we feel that resource groups should not currently be viewed as one of the tools to use to achieve a balanced workload. If the design of resource groups were to change in the future to let you specify resource consumption targets at the system level, at that point they could become effective tools to help you control where work is executes in the sysplex.

### Further information

For more information on the use of WLM resource groups, refer to:

- z/OS MVS Planning: Workload Management, SA22-7602

### 5.3.8 Tivoli Workload Scheduler for z/OS

Tivoli Workload Scheduler for z/OS (TWS for z/OS) is IBM’s z/OS-based batch scheduling product. From a z/OS perspective, TWS for z/OS normally consists of a Controller address space and one or more Tracker address spaces. Every system that is under TWS for z/OS control would have a Tracker address space. Normally, the Controller address space will submit all jobs for the MAS that it resides in, and one Tracker in every other MAS would submit the jobs for its MAS.
WLM interaction

While TWS for z/OS does have an interface to WLM, that interface is used to increase the service class of critical jobs that are going to miss their deadline. TWS for z/OS does not currently interact with WLM to determine which system a given job should run on.

TWS for z/OS does, however, have the capability, delivered by APAR PK09001, to let you specify a WLM Scheduling Environment on the TWS for z/OS panels that will then be associated with the corresponding job by TWS for z/OS. This mechanism is easier to manage than hardcoding the Scheduling Environment name into the production JCL - mass changes can be effected using one of a number of TWS for z/OS utilities.

User control

The normal mode of operation for TWS for z/OS is to have one CPU workstation per JES2 MAS or JES3 complex. In this case, all the jobs for the MAS would normally be submitted on one of the systems in the MAS. In a JES2 environment, this normally results in a larger percentage of jobs being started on the system the jobs were submitted on, assuming the jobs do not have any affinity and there are available initiators on that system.

It is possible to define a separate CPU workstation for each system in the MAS and specify that given jobs will run on specific of these workstations. In this case, the TWS for z/OS tracker on that system will submit the job, increasing the likelihood that the job will be started on that system. While this could be viewed as a way to more evenly balance the production batch work around the MAS, we do not recommend it for the following reasons:

- It is static. If your configuration changes, meaning that you now have more capacity on CPC B than on CPC A, you must go in and change the TWS for z/OS definitions if you want to change the balance of jobs submitted on CPC B compared to CPC A. It would be preferable if the balance were to adjust automatically with no changes required to TWS for z/OS.
- The use of WLM-Managed initiators is a more dynamic and successful mechanism to balance the execution of batch jobs around the MAS. We recommend the use of WLM-Managed initiators for production batch jobs as the default, accepting that there will be some special cases where JES-managed initiators are more suitable.
- You should consider the use of WLM Scheduling Environments as a better way to control which jobs run on each system in the MAS.

So, to summarize, from a TWS for z/OS perspective, we recommend defining just one TWS for z/OS CPU workstation per JES MAS, and use some mechanism outside TWS for z/OS to influence which system each job runs on.

Further information

For more information on TWS for z/OS, refer to:
- IBM Redbook Maximizing OPC/ESA Throughput, SG24-2130
- IBM TWS for z/OS Managing the Workload, SC32-1263
- IBM TWS for z/OS Customization and Tuning, SC32-1265

5.3.9 DB2 Group Attach Facility

Prior to data sharing, batch jobs had to specify the name of a specific DB2 subsystem in order to be able to use the services of that DB2. However, with the advent of data sharing, people wanted to be able to connect to any member of a data sharing group, as opposed to a specific DB2 instance; this provided the ability to run the job on any system that contained a member of the data sharing group.
To address this requirement, DB2 V5 introduced the Group Attach Facility. The Group Attach Facility allows batch jobs, any IMS dependant region (starting with IMS V7 with APAR PQ42180), and the CICS DFHRCCT table (in CICS TS 2.3, or RDO in CICS TS 3.1) to specify the name of the DB2 data sharing group.

**WLM interactions**
WLM does not play any role in the use of Group Attach Facility. When a job tries to connect to DB2 using a group name, it has already been started on a particular system so it is too late for any workload balancing across systems at this point.

**User control**
We obviously recommend the use of Group Attach Facility as it supports the ability to run your DB2 batch jobs anywhere in the sysplex with no manual intervention. We also recommend the use of Scheduling Environments together with this capability. This allows you to mark a system as being ineligible to start new DB2 jobs if you wish to drain the DB2 subsystem on that system.

It is possible, and becoming more common, to run more than one member of a DB2 data sharing group on a single z/OS system. The default mode of operation for Group Attach Facility is that all jobs that specify the group name will be connected to the first DB2 in the subsystem name table. The problem with this is that it does not provide any workload balancing across the two subsystems. For DB2 V8 customers, a DB2 usermod is available that alters this behavior to instead connect randomly to any member of the group. No effort is made to take goal achievement or session balancing or anything else into account in the decision - the DB2 that is selected is purely random.

**Further information**
For more information on the DB2 Group Attach Facility, refer to:

- *DB2 UDB for z/OS Data Sharing Planning and Administration,* SC18-7417

**5.3.10 DB2 Connect**

DB2 Connect™ Enterprise Edition optionally provides load balancing and fault-tolerance when routing connections to DB2 running in data sharing mode in a sysplex. When connected to a DB2 for OS/390 or z/OS database server running in a data sharing environment, DB2 Connect can spread the workload across the different DB2 subsystems comprising the data sharing group, using information provided by WLM. This support requires DB2 for OS/390 Version 6 or later and that the `SYSPLEX` parameter is specified in the DCS directory of your DB2 Connect definition.

**WLM interactions**

There is only one type of information that WLM provides to DB Connect about the sysplex environment: the amount of available capacity on each system in the sysplex that contains a member of the data sharing group that DB2 Connect is trying to connect to.

This information is then used by the DB2 Connect server to handle the incoming CONNECT requests by distributing them among the sysplex members based on which system has the most available capacity. For load balancing, the list of weighted priority information is obtained during each initial connection. If the DB2 Connect connection concentrator is enabled, this list is also used when determining where to send each individual transaction.
DB2 Connect Enterprise Edition servers also provide fault-tolerance by attempting to connect to an alternate system in the event of a connection failure. An error will only be returned to the application if all known connections have failed.

A profile (or registry) variable DB2SYSPLEX_SERVER enables this feature. If DB2SYSPLEX_SERVER is not present or is set to a non-zero value, then sysplex exploitation is enabled. If DB2SYSPLEX_SERVER is set to zero, then sysplex exploitation is disabled. Setting DB2SYSPLEX_SERVER to a value of zero disables the sysplex exploitation for the server regardless of how the DCS database catalog entry has been specified.

The DB2 registry variable DB2CONNECT_IN_APP_PROCESS can be used to allow clients running on the same machine as the DB2 Connect Enterprise Edition server to exploit the sysplex support. If you are using AIX® then its recommended that the EXTSHM variable be configured as well to exploit the sysplex support.

With the addition of the concentrator, DB2 Connect has the ability to balance the workload at transaction boundaries. The DB2 Connect concentrator must be enabled for this to work. Depending on the version of DB2 on z/OS, different granularities of load balancing will be achieved. If the backend DB2 is DB2 for OS/390 Version 6.1 or later, DB2 Connect will receive updated status from the WLM on each transaction.

User control
The only control you have in relation to the DB2 Connect WLM interface is whether you use it or not, and whether you wish to use it for each transaction or only at connect time. You do not have any control over the format of information WLM returns. Nor can you influence DB2 Connect's decision about where to route the connection.

Further information
For more information on DB2 Connect in general, and specifically about its sysplex and WLM support, refer to:

- IBM Redbook Distributed Functions of DB2 for z/OS and OS/390, SG24-6952
- The chapter entitled “DB2 Connect Sysplex support” in IBM DB2 Connect Quick Beginning for DB2 Connect Enterprise Edition, GC09-4833

5.3.11 DB2 Universal Java Connectivity Connector

The DB2 Universal Java Common Connectivity (JCC) driver FixPak 10, introduced a sysplex workload balancing feature for Java type 4 connections to DB2 on z/OS servers. This feature distributes transactions from a single connection to multiple DB2 members of a data sharing group in a Parallel Sysplex. This is done through reusing transport objects connected to each DB2 data sharing member, and is based on a similar algorithm DB2 Connect uses to exploit z/OS's sysplex capabilities. This feature helps balance the DRDA workload on DB2 servers on a transaction level and also provides DB2 failover to improve application availability.

WLM interactions
When a DRDA application first connects to a DB2 data sharing group through the JCC driver, MVS WLM sends back a server list, along with a list of the DB2 members available to process transactions, their respective service priorities, and their IP addresses and port numbers. The JCC driver then caches this information through the lifetime of the connection and goes through an internal algorithm to calculate which DB2 member is best-equipped or has the highest priority to serve the next DB2 transaction. The priority of the chosen DB2 member will decrease with each use, meaning that other DB2 members with lower priorities will eventually be chosen to serve application requests; workload balancing among all DB2 members is
achieved through this priority calculation. WLM sends the server list information to the JCC
driver every 3 minutes if a connection stays connected for longer than 3 minutes.

**User control**
The only control you have in relation to the WLM interface is whether you use it or not, and
whether you wish to use it for each transaction or only at connect time. You do not have any
control over the format of information WLM returns, nor can you influence JCC's decision
about where to route the connection.

**Further information**
For more information about JCC sysplex workload balancing, refer to:
- *DB2 Universal Database for z/OS Version 8 Application Programming Guide and
  Reference for Java, SC18-7414-03*

### 5.3.12 DB2 DDF

DB2 for z/OS and OS/390's Distributed Data Facility (DDF) is a built-in component of DB2 for
z/OS and OS/390 that provides the connectivity to and from other databases (or servers like
DB2 Connect) over the network. DB2 for z/OS and OS/390 DDF supports two network
protocols, SNA and TCP/IP, as well as two database communication protocols, namely, DB2
Private Protocol and DRDA®.

DDF is DB2's transaction manager for distributed database connections. Unlike CICS or IMS,
connections via DDF are not localized to a few address spaces within z/OS. With DRDA,
connections can come from literally anywhere within the bounds of the SNA or TCP/IP
network that DB2 is operating in.

**WLM Interactions**
DDF calls WLM to classify incoming transactions, however it does not work with WLM from
the perspective of routing transactions. By the time DDF sees the transaction, the executing
system has already been selected. However, DDF supports both VTAM GR and Sysplex
Distributor, and both of these communicate with WLM, so WLM does indirectly have a role in
determining where DDF transactions are executed.

**User control**
From the perspective of controlling where DDF work is routed within the sysplex, there are no
user exits or user controls that you can use. However, because DDF supports Sysplex
Distributor, you should obviously use Sysplex Distributor (with whatever distribution method
you feel is appropriate) to balance the DDF requests across the available DB2 members.
DDF also supports VTAM GR, however because a future release of DB2 Connect may not
support SNA, it is recommended to consider migrating to TCP.

**Further information**
For more information about DDF, refer to:
- IBM Redbook *Distributed Functions of DB2 for z/OS and OS/390, SG24-6952*

### 5.3.13 DB2 Sysplex Query Parallelism

DB2 provides the ability to break large queries into smaller queries that can be run in parallel.
This capability was delivered in three steps: I/O parallelism in DB2 V3, CPU parallelism in
DB2 V4, and sysplex parallelism in DB2 V5. It is the last of these three that we are interested in because this is the one that affects the flow of work across more than one image.

**WLM interactions**
DB2 does not use WLM when deciding how a query should be broken across the members of the sysplex.

**User control**
DB2 bases its decision on how to break up the query on the number of online CPs available to each member of the data sharing group and on the available space in the buffer pools. No allowance is made for how busy or idle the CPs are, nor of the relative speeds - if one DB2 member is executing in a z9 LPAR with 2 online CPs, and another member is executing in an LPAR with 8 z900 online CPs, the z900 DB2 will receive 80% of the query and the z9 DB2 will receive 20% of it, all other things being equal. However, DB2 can adjust the degree of parallelism if communication with the Buffer Manager in each DB2 member indicates buffer pool constraints in that system. DB2 gets the number of online CPs in each image at the start of the query, so adjustments are made if logical CPs are taken on or offline.

There are no user exits or other externals to let you influence DB2's decisions.

**Further information**
For more information about DB2's sysplex query parallelism capability, refer to:
- The chapter entitled “Query Parallelism” in the IBM Redbook *DB2 for OS/390 V5 Performance Topics*, SG24-2213
- The chapter entitled “Improving the response time for read-only queries” in *DB2 UDB for z/OS V8 Data Sharing Planning and Administration Guide*, SC18-7417

### 5.3.14 IMS Shared Message Queue

IMS Shared Message Queue is a very effective way to balance IMS transactions across multiple IMS subsystems in a sysplex. When a message arrives in IMS, the control region checks to see if the message has been marked as being eligible to use the shared message queue. If it has, IMS next checks to see if any suitable local Message Processing Regions (MPRs) are available. If there is one available, the message is sent to that MPR and a copy of the message is also placed in the shared message queue structure in the CF - this ensures that a copy of the message is available to be processed elsewhere should this IMS abend before the message can be processed.

Because of the design of shared message queues, the system that a message is processed on is influenced by the following:
- Which control region the user is logged on to. Because IMS prefers to process the message in a local MPR if possible, there will be a tendency for messages to be processed on the same system the user is logged on to. IMS supports both VTAM GR and Sysplex Distributor for routing logon requests across the available control regions.
- Which system has the most capacity. If a message cannot be processed locally, it will be placed in the shared message queue structure and all interested IMSs will be notified. The IMSs then race to select the message. The IMS running on the system with the most capacity will tend to get to the CF before the other IMSs; this provides a degree of natural workload balancing. As the system with the most capacity selects more messages it will get busier and another will become the system with the most capacity, meaning that it will now tend to get to the CF first, and so on around all the members of the group.
**WLM interactions**

Because messages have a tendency to get processed on the system the user is logged on to, the balancing of logon requests across the IMS subsystems plays an important role in determining where messages will be processed. As mentioned previously, IMS supports both VTAM GR and Sysplex Distributor, so WLM does have a role in determining where the messages will be processed, albeit indirectly.

For messages that are processed from the shared queue structure, WLM really has no direct role. The fastest system or the one with the most available capacity will tend to get more messages. However, IMS does not consult WLM when deciding if it should select a message from the structure or not.

**User control**

Apart from setting up the IMS MPR topology to control how many MPRs on each system are available to process each transaction class, there is little you can do to influence which system will be the one to select a given message. IMS does not provide any exits to give you a role in this process.

However, despite the fact that you cannot directly dynamically control the flow of messages to the different members of the IMS data sharing group, the use of IMS shared message queue is highly recommended because it gives you a solid base of automatically distributed workload. If you can get a good spread of work across all the systems in the sysplex, with little or no ongoing effort on your behalf, you can then fine tune the configuration to meet the aggregation criteria by moving around the workloads that do not support data sharing and therefore do not support dynamic workload balancing.

**Further information**

For more information on planning for and implementing IMS shared message queues, refer to:

- The IBM Redbook *IMS/ESA Shared Queues: A Planning Guide*, SG24-5257
- The IBM Redbook *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908
- The IBM Redbook *IMS in the Parallel Sysplex Volume II: Planning the IMSplex*, SG24-6928
- The IBM Redbook *IMS in the Parallel Sysplex Volume III: Implementation and Operations*, SG24-6929
- *IMS Administration Guide: Transaction Manager*, SC18-7808

### 5.3.15 IMS Group Attach

Just as early releases of DB2 required jobs that wanted to use DB2 services to specify the name of the relevant DB2 subsystem, similarly, IMS used to require that IMS dependent regions specify the name of the IMS subsystem they wished to connect to. However, like DB2, this requirement introduced an affinity between the region and the subsystem whose name it specified.

In order to provide the ability to let a dependent region connect to any member of the IMS data sharing group, IMS introduced the IMSGROUP capability via APAR PQ21039 in IMS V6. This is conceptually very similar to the DB2 Group Attach Facility in that it does not actively play a role in workload balancing, however it does enable dynamic workload balancing as it lets you run the region on any system in the sysplex that contains a member of the associated IMS data sharing group.
WLM Interactions
There is no direct WLM role in the IMS Group Attach capability.

User control
Like DB2, by the time a dependent region gets to use the Group Attach capability, the system that region will run on has already been selected. However, if you have so many BMPs that they could significantly contribute to the sysplex aggregation qualification, you could combine the use of IMSGROUP with WLM Scheduling Environments to manually (or, preferably, using automation) control which systems are candidates to run BMPs.

Further information
For more information about how to define and use the IMS Group Attach facility, refer to:
- *IMS Installation Volume 2: System Definition and Tailoring*, GC27-1928

5.3.16 IMS Connect
IMS Connect provides the ability for a TCP/IP client to communicate with IMS. In IMS Version 9, the IMS Connect function is provided as part of IMS. Prior to IMS V9, IMS Connect was a separate program product (5655-K52).

Clients communicate with the IMS Connect function using TCP/IP, and IMS Connect in turn uses the IMS Open Transaction Management Agent (OTMA) function to communicate with the IMS Control regions using XCF. A sample configuration is shown in Figure 5-2; the IMS Connect address space is represented by the box labelled ICON. You can see that it is possible to run IMS Connect in an image where no IMS subsystem is running - all that is required is connectivity to a subsystem via XCF.

![Figure 5-2 IMS Connect sample configuration](image-url)
WLM interactions
IMS Connect does not use WLM when deciding which IMS instance to send a request to.

User control
The IMS Connect clients have defined “datastores”. The datastores are virtual resource names that IMS Connect translates to IMS Control region names using information passed to it in a control file. The control file specifies (among other things), the datastore ID, the XCF group name, and the XCF member name of the IMS Control region. The IMS Connect User Exit determines which IMS Control region to send the message to. Once the Control region has been identified, the message is passed to IMS via OTMA. If IMS Shared Message Queues are being used, the message may be placed on a shared message queue, from which one of the connected Control regions will collect the message, process it, and place the response back on the message queue. From there the message is collected and passed back to IMS Connect which in turns passes it back to the client.

Between the client and IMS Connect, you can use Sysplex Distributor to spread the incoming IOPCB calls across multiple IMS Connect address spaces, probably spread out across multiple systems in the sysplex. To spread the IMS transactions across multiple IMS Control regions, you must use the IMS Connect User Exit. Because the User Exit can be passed information about all the datastores and the status of the IMS Control regions, you can code the User Exit to either just send the message to the Control region pointed to by the datastore definition, or you can round-robin the request around all the Control regions in that XCF group. If you wish, you could extend the function in the exit to call the WLM IWMWSYSQ service to determine which system in the sysplex has the most available capacity and use that information in the routing decision.

If you do not wish to provide your own code to handling the workload balancing, you could use the User Exit provided by the IMS Connect Extensions product (5655-K48). IMS Connect Extensions does not provide an interface to WLM, however you can hardcode relative capacity weights for your various IMS regions. For example, if you have two IMS Control regions, IMS1 and IMS2, and IMS2 has twice the capacity of IMS1, you could assign a weight of 20 to IMS2 and 10 to IMS1 - IMS Connect would then route twice as many transactions to IMS2 as it would to IMS1. IMS Connect Extensions also provides the ability to define affinities between transactions and particular Control regions. More information about IMS Connect Extensions is available in IMS Connect Extensions User Guide, SC18-7255.

Further information
For more information on IMS Connect, refer to:

- IMS Connect V2.2 Guide and Reference, SC18-7260

5.3.17 IMS Workload Router
The IMS Workload Router (WLR) is a program product, designed to spread incoming transactions across multiple IMS subsystems when using MSC. WLR provides the capability to route workloads based on percentages or weights of inbound workloads. It also provides an affinity mechanism that allows one to specify the target MSC links for transactions with certain characteristics (name, generic name, class, sysid, and so on). It is not sensitive or aware of the CPU consumption of the transactions that it routes.
WLM interactions
There is no communication between IMS Workload Router and WLM in terms of providing information about response times or available capacity on the various members of the sysplex.

User control
A WLR path is a destination or connection from a WLR router IMS system to a given WLR server IMS system. Paths are defined to the WLR router during WLR customization. In general, a WLR path has a one-to-one relationship with a specific IMS MSC MSNAME defined in the IMS SYSGEN for the router system. For uniformity in prioritizing work, assigning affinity destinations, and calculating WLR statistics, the router IMS system has an internal path representation.

Paths are defined to WLR with weights. These weights can be expressed in one of two ways:
- As simple weighted values (for example, if one path has a weight of 1 and a second path has a weight of 5, the second path is 5 times as likely to be used as the first path.)
- As a percent of the router-inbound workload that WLR routes to a path. A path with a weight of 20 should receive 20% of the router-inbound work load.

The CHKGOAL configuration parameter determines how these values are interpreted. If CHKGOAL=OFF is specified, the weights are treated as simple weighted values. If CHKGOAL=ON is specified, the weights are treated as percentages of the router-inbound workload.

While you cannot tell WLR directly to keep each of a set of CPCs x% busy, you can most certainly use WLR to influence CPU consumption in a certain direction. For example, if you have a 500 MSU CPC and a 1000 MSU CPC, you could attempt to keep them evenly utilized by setting up WLR to route 2/3 of the workload to the 1000 MSU CPC and 1/3 to the 500 MSU one as a starting point. If transaction arrival is pretty random, this should keep both CPCs fairly evenly utilized. After monitoring the workload, WLR can then be fine tuned with affinities and weight distributions to achieve the desired result. The weights can be dynamically changed and affinities can be dynamically created/changed via the WLR administrative interface without having to recycle IMS.

Further information
For more information about the IMS Workload Router, refer to:
- The chapter entitled IMS/ESA® Workload Router in the IBM Redbook IMS/ESA Multiple Systems Coupling in a Parallel Sysplex, SG24-4750

5.3.18 CICSPlex System Manager
CICSPlex® System Manager (CICSPlex SM) is a system management tool that enables you to manage multiple CICS regions from a single interface. It provides a dynamic transaction routing (DTR) program that can route eligible transactions from a TOR region to a suitable AOR region, selected at the time the transaction is initiated. CICSPlex SM works with one of two algorithms (queue and goal) to choose which AOR processes any particular transaction.

WLM interactions
When CICSPlex SM runs with the goal algorithm, CICSPlex SM calls the MVS WLM to get the goals associated with a transaction. CICSPlex SM gets the response time goal, but not the importance. CICSPlex SM does not call WLM for every transaction, but builds a
look-aside table containing the response time target for each encountered transaction. When a transaction is initiated, CICSPlex SM first checks to see if the goals for that transaction are already in the table, and only if CICSPlex SM does not find it there does it call the WLM. However, CICSPlex SM refreshes its service class goals tables following WLM policy changes. Note that CICSPlex SM only supports WLM average response time goals - percentile response time goals are not supported.

Using transaction goal information from the WLM policy, together with actual response time information it collects from the AORs, CICSPlex SM selects an eligible TOR to send the transaction to. Note that it has no information about the utilization of each CICS region, nor the system it resides on, so no consideration is given to the available capacity that may or may not exist on each system.

**User control**
The alternative to goal mode is queue mode. When CICSPlex SM runs with the queue algorithm, CICSPlex SM routes the transaction to the AOR with the shortest queue independently of what is specified in the WLM policy. WLM will try to help the CICS regions achieve the goals you gave to CICS transactions, but CICSPlex SM, with the queue algorithm, will try to maximize transactions throughput and standardize response time. Therefore, if your objective is to balance utilization across multiple CPCs, especially when the CPCs are of different capacities or there are differing numbers of AORs on the different CPCs, we recommend you run CICSPlex SM with the goal algorithm.

You can use a mix of goal and queue modes in a CICSPlex, however a TOR can only be associated with one workload, and a workload can only use one mode. So, either all the transactions in a given TOR use goal mode or they all use queue mode.

If you wish to have further control over how transactions are routed, you could provide your own EYU9WRAM exit.

**Further information**
- For more information about workload balancing using CICSPlex SM, refer to:
  - The IBM Redbook *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*, GG24-4286
  - The IBM Redbook *Workload Management for Web Access to CICS*, SG24-6118
  - The IBM Redbook *CICS Web Services On Demand System Management*, SG24-7144
  - CICSPlex SM Concepts and Planning, GC33-0786
  - *CICSPlex SM Setup and Administration Volume 1*, SC33-0784

### 5.3.19 CICS MRO topology

There are many ways to set up a CICS configuration. A traditional one is to have a single CICS region that owns the terminals, runs the transactions, and accesses the files. However, in a sysplex environment, a more flexible and more resilient configuration is to have a number of regions that own the terminals (TORs), a number that run the transactions (AORs), and possibly one or more that access the files (FORs). This latter configuration is more amenable to workload balancing as the transactions can potentially be executed on different z/OS systems on different CPCs, regardless of which CICS region the user is logged on to. CICS supports both VTAM Generic Resources and Sysplex Distributor for distributing the incoming requests across the available CICS regions.

One of the difficulties in implementing a configuration like this is that transactions sometimes have affinities, meaning that two transactions must be run in the same CICS region. For
example, one transaction may save some information away to CICS temporary storage and a subsequent transaction will pick up that information from temporary storage.

There are two ways to handle these affinities:

- You can use features in CICSPlex System Manager to ensure that transactions with affinities are routed to the same AOR.
- You can exploit new capabilities in CICS that move the shared information into the Coupling Facility, thereby ensuring that the information can be accessed regardless of which region or which system the transaction runs in. These features are:
  - CICS Temporary Storage in the CF
  - CICS Coupling Facility Data Tables
  - CICS Named Counter Server

A related feature is CICS Global ENQ/DEQ, introduced in CICS TS 1.3, which uses a definition called ENQMODEL to specify which CICS resources should be externalized to GRS.

Just having an MRO configuration does not automatically provide workload balancing, however it provides the infrastructure that can then be exploited by products like CICSPlex System Manager to balance the CICS transactions across the systems in the sysplex.

**Further information**

For more information on how to set up and exploit an MRO configuration, refer to:

- The IBM Redbook *Parallel Sysplex Application Considerations*, SG24-6523
- *CICS/TS System Definition Guide*, SC34-6428
- *CICS/TS Resource Definition Guide*, SC34-6430

### 5.3.20 WebSphere MQ Shared queues

MQ shared queues support provides the ability to place MQ messages in a shared queue in a CF structure, from where they can be operated on by any member of the shard queue group.

When a queue manager address space starts, it registers with the Coupling Facility, indicating which queues it has an interest in. When a message arrives in any queue, the CF informs all the queue managers that have expressed an interest in that queue. All the queue managers then race to select the message; the first queue manager to react will be the one that gets the message.

When MQ shared queue support was first delivered, the support was limited to non-persistent queues, and to messages that were 63K or less. Also, the administration structure was a single point of failure - if that structure or the CF it resided in failed (and the structure was not duplexed) all connected queue managers would immediately shut down. However, subsequent MQ releases have addressed all these limitations, culminating with WebSphere MQ V6 removing the administration structure as a single point of failure, and increasing the supported message size.

The most common reason to use MQ shared queue support is that it provides better availability than non-shared queues, especially for non-persistent messages. However, an additional benefit is that it provides workload balancing for MQ, sharing the processing of MQ messages over queue managers running on different systems in the sysplex.
**WLM Interactions**

WLM has no role in determining which queue manager will select a given message from the shared queue. The fastest system or the one with the most available capacity will tend to select more messages.

Note, however, that there are two aspects to using the queues. Before a queue manager can select a message from the queue (using the mechanism described previously), the message must be placed in the queue by the queue manager that the user connects to. MQ supports both VTAM GR and Sysplex Distributor to spread incoming connection requests over the MQs in the queue sharing group, and both of these mechanisms use WLM to determine the most appropriate instance to forward the request to. So, while WLM does not have a direct role in the selection of messages from the shared queue, it does have a role in selecting which queue manager will insert them into the queue.

**User control**

There are no exits or other externals provided to control which queue server will process a given message. Obviously you can specify which queue managers are eligible to process a given queue, but you cannot control, on a message by message basis, which manager a given message will be processed by.

**Further information**

For more information on MQ shared queue support, refer to:

- The IBM Redbook *WebSphere MQ in a Parallel Sysplex Environment*, SG24-6864
- The IBM RedPaper *WebSphere MQ Queue Sharing Group in a Parallel Sysplex*, REDP-3636
- *WebSphere MQ for z/OS Concepts and Planning Guide V6*, GC34-6582
- *WebSphere MQ for z/OS System Setup Guide*, GC34-6585

### 5.3.21 WebSphere Application Server

WebSphere Application Server (WAS) is becoming a significant part of many customers z/OS workloads, so it is important to consider it when looking at workload balancing in a sysplex.

There are two aspects to workload balancing in relation to WAS:

- Customers will typically run WAS on a number of z/OS systems. WAS can make use of Sysplex Distributor to distribute incoming WAS requests across the servers in the sysplex.
- Once a request reaches a WAS address space, no further forwarding to another system will take place. However, WAS can optionally communicate with WLM to determine if there is a need to start more servant regions on the same system.

One of the options you have in Sysplex Distributor (in z/OS 1.7 and later) for distributing requests is based on goal achievement. By starting more servant regions, WAS and WLM can improve the goal achievement for a given WAS and thereby influence where incoming requests should be forwarded to.

**WLM interactions**

WAS works with Sysplex Distributor to spread the incoming requests across the available WAS regions. However, once a WAS instance has been selected, the only interaction with WLM in relation to workload balancing is the decision of whether more servant regions are required.
**User control**
There are no user exits or external controls to let you influence which WAS region a given incoming request will be routed to, beyond the control provided by Sysplex Distributor.

**Further information**
For more information on the workload balancing capabilities of WAS on zSeries, refer to:
- The IBM Redbook *WebSphere for z/OS V6 High Availability*, SG24-6850-01

### 5.3.22 Handling affinities

All the previous sections assume that you have the ability to run any work on any member of the sysplex. However, in reality, there are a number of reasons this may not be possible:

- You may have work that still contains affinities that you have not yet been able to remove or circumvent. In this case, the affinities need to be taken into account by the balancing tools (CICSPlex SM and IMS Workload Router, for example).
- You may have some products that are particularly expensive, so you want to minimize the number of LPARs or CPCs those products are executed in. Therefore, any work that uses those products must be routed to the appropriate LPAR. WLM Scheduling Environments is one way to control these jobs.
- You may have work that for business or legal reasons cannot co-exist in the same LPAR, so you are limited to the extent that you can balance this work across the sysplex. However, if you have work with these separation requirements, consideration should be given to whether those workload should even be in the same sysplex. Separating work into different systems in the same sysplex does not provide the same degree of separation as separating when into separate sysplexes.

### 5.3.23 Summary
The previous sections cover all the current IBM products or functions that play an active role in workload balancing. It is likely that only a subset of these will apply to each installation. And we also want to reiterate that even if you were to enable and exploit all these functions with all your applications, it is possible that you will not have the sysplex aggregation configuration that you are aiming for. However, it will be far easier to achieve that configuration if you have work that is already distributed across the members of the sysplex.

If your studies show that your largest sysplex consumes comfortably more than 50% of the used MSUs across all CPCs, we hope that you will be able to use the controls documented in 5.4, “Other related system controls” on page 90 to fine tune the work distribution to meet the qualification criteria. If the largest sysplex is just barely 50%, or below 50%, of the total used MSUs, the suggestions contained in 5.5, “Redistributing workloads and processing” on page 95 may help you meet the criteria with a modicum of effort.

### 5.4 Other related system controls
The previous section described IBM’s products that either distribute work or process the work in the sysplex. The actions and decisions taken by many of those products are affected by the performance and capacity of the systems in the sysplex. In this section we discuss the controls available to you that affect that performance and capacity. If you are able to achieve an environment where all your significant workload supports dynamic workload balancing, we hope that adjusting these parameters should be all that is necessary to maintain aggregation qualification as your configuration and workload mix change over time.
5.4.1 LPAR weights

All the calculations WLM makes to determine the capacity available to an LPAR are based on the relative weight of the LPAR compared to the other active LPARs on that CPC. The “guaranteed capacity” of an LPAR is determined by the relative weight of this LPAR. And the share of unused capacity that this LPAR is “entitled” to is also determined by its relative weight.

Therefore, at a minimum, you should ensure that the weights of all the LPARs in the PrimaryPlex add up to more than 50% of the weight of all the active MVS-based LPARs on each of the CPCs in the PricingPlex. While doing this will not guarantee that the LPARs in the PrimaryPlex will consume more than 50% (they might not have enough work to require that much capacity), it does ensure that if all LPARs on the CPC are 100% busy, then the PrimaryPlex LPARs will be given more than half the MSUs used by MVS-based LPARs.

5.4.2 Hard capping

An extension to LPAR weights in a facility known as “hard capping”. This is a mechanism whereby you can specify that an LPAR cannot consume more than the capacity guaranteed by its weight.

Customers frequently use hard capping as a way of protecting production LPARs from busy or looping development or test LPARs. By placing a cap on the capacity that can be consumed by the other LPARs, you ensure that the production LPARs will always have a certain amount of capacity available to them. As a rule, we do not recommend the use of hard capping: if the production LPARs are busy, their relative weight should ensure that they receive their guaranteed capacity, regardless of how busy the other LPARs may be. Additionally, it is wasteful to have the other LPARs capped at times when the production LPARs are not busy, resulting in those LPARs missing their performance objectives while capacity lies unused on the CPC.

However, from the perspective of workload balancing, there may be situations where the use of hard capping does make sense. If your workload supports dynamic workload balancing, but for whatever reason you still have an imbalance between your CPCs, capping the capacity available to the production LPARs on one CPC may force work to run in LPARs on the other CPCs. For example, if you have two CPCs, and the production LPARs are consuming 85% of the used MSUs on CPC A, but only 40% on CPC B, capping the production LPARs on CPC A to 70% of the capacity of that CPC should result in an increase in the production utilization on CPC B. This assumes, of course, that the workload supports workload balancing, that sufficient spare capacity is available on CPC_B, and that the LPAR weights have been set up appropriately. We should stress that this is not an option that we favor, however if other attempts to achieve a more balanced utilization have failed, this is an alternative that you could try.

Hard capping is enabled on the HMC console. You can dynamically enable and disable capping for an active LPAR, just as you can dynamically alter the weights for an LPAR. However, there is no way to modify these settings directly from a z/OS system without the use of an additional product such as Tivoli System Automation for z/OS.

5.4.3 Defined Capacity

The ability to specify a Defined Capacity (also known as Soft Capping) for an LPAR was introduced in support of subcapacity charging. The idea is that you specify a target number of MSUs for an LPAR on the HMC console - WLM and LPAR working together then ensure that the rolling 4-hour average consumption of the LPAR does not exceed that amount.
When used to control software costs, Defined Capacity is more attractive than hard capping because it allows the current LPAR utilization to exceed the target number of MSUs as long as the rolling 4-hour average has not exceeded that threshold. This would not be possible if you were to use hard capping to limit the CPU consumption of the LPAR.

However, from the perspective of workload balancing, Defined Capacity is not a very effective instrument. This is mainly because Defined Capacity works based on the rolling 4-hour average, whereas sysplex aggregation rules are based on actual utilization. If you are trying to limit the capacity available to an LPAR on a CPC in order to force work to other members of the sysplex, the reaction of WLM with a Defined Capacity will probably be too slow to result in the behavior you are aiming for.

### 5.4.4 Intelligent Resource Director (IRD)

There are two aspects to IRD that potentially affect workload balancing:

- **Vary CPU management**, whereby WLM will bring logical CPs on or offline in an LPAR based on the capacity needs of the LPAR. This feature is intended to increase the efficiency of an LPAR; the fewer logical CPs that are online to an LPAR, the higher is the effective speed of each logical CP.

- **Weight management**, whereby WLM will move LPAR weight from one sysplex member to another member on the same CPC in order to help an ailing workload in the target member.

Both of these functions can be turned on or off independently.

The use of the Vary CPU feature should be approached with care if you are struggling to maintain sysplex aggregation qualification. Remember that WLM takes into account the number of online logical CPs when calculating the available capacity of an LPAR. For users like Sysplex Distributor or VTAM GR, this could result in WLM providing feedback that represents the current, rather than the maximum capacity of the LPAR. As a result, you may find that the work does not flow as you would expect within the sysplex, or reactions to workload changes (from a balancing perspective) could be sluggish. While WLM always attempts to have more logical CPs online than are currently needed, the available capacity calculation will always return a reduced number any time the number of online logical CPs is less than the maximum defined.

The other feature, dynamic weight management, is unlikely to have a significant impact on the CPU consumption for a given sysplex on a given CPC. IRD weight management works by moving weights between LPARs in the same sysplex. Therefore, the total weight of the sysplex, and therefore the guaranteed capacity of the sysplex, on that CPC will remain unchanged even if WLM is moving weights between the LPARs.

For more information on IRD, refer to:

- IBM Redbook *z/OS Intelligent Resource Director*, SG24-5952
- *z/OS MVS Planning: Workload Management*, SA22-7602

### 5.4.5 zAAPs

zSeries Application Assist Processors (zAAPs) are a special type of PU available on z890/z990 and later CPCs. zAAPs are designed to offload Java work from traditional operating system CPs. They are lower cost than traditional OS CPs, and the capacity they provide is not taken into account by IBM when determining the capacity of an LPAR for software charging purposes.
If a significant portion of your workload is Java-based, adding zAAPs to a CPC can be a very cost effective way of providing additional capacity while at the same time potentially reducing your software costs (if you are exploiting subcapacity pricing).

However, you need to consider the impact the addition of zAAPs could have on your sysplex aggregation qualification. Let's say you have a CPC, CPC A, that has two LPARs. LPAR PRDA1 is in the production sysplex and is using an average of 500 MSUs over the 40 prime hours. LPAR DEVA1 is in the development sysplex and is using an average of 400 MSUs over the 40 prime hours. Based on these utilizations, CPC A is determined to be “in” the production sysplex. And of the 500 MSUs consumed by LPAR PRDA1, you determine that 200 MSUs are used by Java.

You are now at the point where you need add more capacity to CPC A. You have a choice between adding CPs or adding zAAPs. Both will provide the equivalent amount of MSUs, however your software bill would go up when the systems start using the additional CP capacity, whereas the bill would be expected to go down when you enable the zAAPs and let the 200 MSUs of Java work run on the zAAPs.

However, consider what would happen from a sysplex aggregation perspective in this situation. Prior to adding the zAAPs, the CPC would be considered to be in the production sysplex because the production sysplex LPARs are using more than 50% of the total used MSUs (500 out of 900). If you add the zAAPs and let all the Java work flow to those PUs, the production sysplex will now only be using 300 MSUs out of a total of 700 used CP MSUs - the production sysplex is now using less than 50% of the used MSUs, and this CPC would now be considered to be in the development sysplex instead, meaning that this CPC could no longer be aggregated with other “production” CPCs. As a result, your software bill could actually increase even though the utilization of the CPs on this CPC has decreased. See 4.3, “Scenario 2 - Spending in order to save” on page 50 for an example of this situation.

This is not to say that zAAPs are a bad idea - they are an excellent way of providing additional capacity in many environments at a very attractive cost. You just need to give some thought in advance to how much work you wish to move from the PrimaryPlex LPARs to the zAAPs. On z890 and z990 CPCs, you do not have a lot of control over this. Using statements in the IEAOPT member, you can specify whether Java work should only execute on zAAPs or if it can execute on either CPs or zAAPs - this is shown diagrammatically in Figure 5-3 on page 94. This diagram reflects changes introduced in APAR OA14131. However, even with this APAR applied, there is no way to say “execute up to this number of MSUs worth of Java work on the zAAPs, and anything above that on CPs”.

Chapter 5. Workload balancing and sysplex aggregation 93
If multiple LPARs are competing to use the zAAP resource, the zAAP capacity will be allocated based on the relative weights of the competing LPARs.

On the System z9, you have more control. First, it is possible to specify two weights for an LPAR. One weight determines the share of the CPs that the LPAR is entitled to. The other weight is the share of the zAAPs that the LPAR is entitled to. So, it is possible, if you wished, to specify that an LPAR should be guaranteed 80% of the CP capacity, but only 50% of the zAAP capacity.

Additionally, it is possible on a System z9 to turn on or off hard capping for zAAPs and CPs independently. So, you could say that the production LPAR is not to have any cap on the amount of CP resource it can potentially consume, whereas it is not to be given more than x% of the zAAP capacity, regardless of whether any other LPAR wants that capacity or not.

To help you determine the amount of Java workload you run in your environment, you can use a tool called the zAAP Projection Tool. This tool can be downloaded from the following Web site:


Another option you have to measure potential zAAP usage is RMF. This new functionality is available as an SPE (APAR OA05731) for z/OS 1.6.

Further information
For further information about defining and using zAAPs, refer to:

- IBM Redbook zSeries Application Assist Processor (zAAP) Implementation, SG24-6386
5.5 Redistributing workloads and processing

If you are reading this section you presumably have determined that you need to move some more work into the PrimaryPlex in order to be sure of comfortably maintaining sysplex aggregation qualification.

As we mentioned previously, our preference is to avoid merging systems into a sysplex of unrelated systems. This generates a lot of work, is not necessarily good for availability, and does not deliver much in the way of systems management benefits.

Our preferred course of action, therefore, is to move workloads from outside the sysplex into the PrimaryPlex. This section discusses some potential low-risk candidates that should be movable without too much effort.

Important: A very important point to remember as you consider your options in this area is that sysplex aggregation is the single most effective way to reduce IBM mainframe software costs. The savings that can be achieved by being able to aggregate a CPC far outweigh any additional cost related to using more MSUs.

To put this another way, if using more z/OS capacity makes the difference between qualifying for sysplex aggregation and not qualifying, in most cases we would expect the net cost to be lower if the increased consumption causes the CPC to qualify for aggregation. For this reason, it is important to, temporarily at least, dispel the normal mindset that more utilization always equates to larger software bills.

For an example, refer to 4.5, “Scenario 4 - More for less?” on page 59.

5.5.1 Batch jobs

The default window that is used for sysplex aggregation qualification is 09:00 to 17:00. And most installations do not start their batch window until some time after 18:00. However, if you find that the load on your production systems is dropping below the 50% threshold at the end of the business day, you may be able to adjust the start time for some of your batch work so that it starts earlier. This has the double advantage of helping you maintain aggregation qualification, and also freeing up some capacity from your batch window, potentially reducing the elapsed times for your critical batch jobs.

5.5.2 Print serving

It is not unusual in large companies to find tens or even hundreds of PC-based printer servers. While the hardware cost of each PC is small, the labor cost of administering and maintaining all those platforms is significant. If you could move the print server function into z/OS (using a product like InfoPrint Server, for example), you could not only save on the costs of looking after all those servers (plus the associated hardware, software, and floorspace costs), but the increased utilization on z/OS may be sufficient to help you maintain aggregation qualification.

5.5.3 Output management

In an effort to control output production and handling costs, most companies have one or more output management products. These allow reports to be viewed online, archived, backed up, and so on. In some cases, these applications are run in non-production systems because they are not viewed as being as critical as the traditional CICS and IMS applications. However, these applications are typically stable, and running them in a production system
does not introduce an unacceptable risk of them adversely affecting the OLTP applications. For this reason, these may be considered as a potential workload that could be moved from outside the PrimaryPlex environment.

5.5.4 SMP jobs

Most customers run their SMP work outside the production environment. However, from the perspective of being a risk to the production systems, there is really no reason those jobs couldn’t equally be run in the production environment. It will mean making the SMP volumes available in the production environment, and potentially some security definition changes. And you need to be confident in your change management processes, to be sure that there is no way the SMP jobs could inadvertently update the wrong data sets. But beyond these concerns, there is no reason not to run the SMP jobs in a production system.

5.5.5 SMF Processing

Like SMP jobs, another workload that many installations run outside the production environment is the processing of SMF data. Not just archiving the data, but also running the performance and capacity planning products against it, user chargeback reporting, and so on. Moving that workload back into the production environment should be relatively easy (there should not be that many data sets involved) and does not introduce any significant risk to the production applications.

5.5.6 Workload consolidation

If moving some or all of the previous workloads into the PrimaryPlex does not place you comfortably above the 50% threshold, you may consider moving work from other platforms into the PrimaryPlex environment. Not all workloads are suited to running in z/OS, but certainly for some, z/OS is just as valid a platform as any other operating system. If you have workloads like this, moving them to z/OS maybe provide a better level of service and lower management costs while helping you lower your zSeries software costs at the same time.

5.5.7 Network ownership

While it is not as common to have separate network-owning LPARs in a TCP network as it was with an SNA network, there are many customers with large SNA networks that still have these network concentrator LPARs.

If you need to move whole LPARs into the PrimaryPlex, LPARs such as these are very attractive candidates. They typically have a very small number of data sets, no database manager, few VSAM files and catalogs, and a small number of RACF profiles and users. And, with a large network, they can consume a considerable number of MSUs.

The alternative - moving the network ownership function into one or more of the PrimaryPlex systems is not as attractive. People typically create a separate network-owning LPAR specifically to insulate the network from the more frequent planned outages that are experienced by "normal" production systems. Given that the objective is to increase the number of MSUs used by the PrimaryPlex, moving the network-owning LPARs intact into the PrimaryPlex may be a little more work, but we believe it is the more desirable option in the longer term.
Handling systems with different time offsets

This appendix is only of interest to installations where some of the systems run with a different local time than other systems within the PricingPlex.

When PLEXCALC detects that different timezone offsets are present within a CPC, it will determine which LPAR is the most highly utilized. This LPAR then becomes the “master” LPAR, and will be so noted on the report unless you override this default behavior by explicitly specifying a “master” LPAR. All the other LPARs on this CPC will have their times adjusted to match the times of this “master” LPAR.

In this way the PLEXCALC report will show simultaneous times on the report. So, if for example, you have a US east coast LPAR and a west coast LPAR, and the east coast LPAR is the most highly utilized, then the 10:00 line on the report will have 10:00 for the east coast LPAR, and the west coast LPAR will be adjusted so that the 7:00 utilization will be reported on this line.

If you want more background to this situation and the considerations for PLEXCALC, read the remainder of this Appendix.
LPARs with different local times

It is possible to have systems in a sysplex that do not have the same local time. In this case, the question may arise about what “real” time is measured for the sysplex aggregation criteria.

First, it may help to very briefly discuss the options available for controlling the time in a z/OS system. We will assume that a common time source (Sysplex Timer or STP) is being used. If you have a sysplex that spans CPCs, you must have a common time source. If your sysplex does not span CPCs, then the whole question of aggregation never arises in the first place.

On the common time source, you have a number of options:

- You can specify the Universal Time Coordinate (UTC - also known as Greenwich Mean Time (GMT)) offset to the actual UTC value. If you do this, you have two options for specifying the offset for the local time:
  - You can specify a local offset on the common time source, and in the CLOCKxx member, specify ETRZONE=YES. This tells z/OS that it is to get the local time from the common time source.
  - You can specify ETRZONE=NO and a TIMEZONE value in the CLOCKxx member. This tells z/OS that the local time is to be obtained by adjusting the UTC time it receives from the common time source with the value from the TIMEZONE keyword.
- On the common time source you can set the UTC to be your local time and not use any offset. This is generally not a recommended option.

From this, you can see that regardless of which option you select, every CPC in the sysplex will have the same UTC value. However the local time can be different in different sysplexes, and even in different members in the same sysplex.

While the majority of installations use the same local time in every system, there are some cases where different systems use different local times. Perhaps one system supports users in Europe, so it is set to Central European Time, and another system supports users on the east coast of the United States, so it would be set to Eastern Standard Time, for example.

If PLEXCALC were to use the local time in its calculations, it is possible, even likely, that the total utilization of all LPARs would exceed 100% of the capacity of the CPC. For that reason, PLEXCALC looks at the utilization of all LPARs on a given CPC for the same UTC rather than the same local time.

As an example, we have a CPC with three LPARs. The LPARs support users in three different time zones, offset by one hour between each LPAR. This results in the utilization shown in Figure A-1 on page 99.
Figure A-1  LPAR utilization for UTC discussion

If you look at the utilization of the CPC using UTC, you would see the utilization shown in Figure A-2 - you will notice that utilization never exceeds 100%.

Figure A-2  Apparent CPC utilization using UTC times

However, if PLEXCALC were to report based on the local times in each LPAR, you would see a different utilization as shown in Figure A-3 on page 100 - this does not reflect the actual CPC utilization.
For this reason, PLEXCALC normalizes the local time for each LPAR back to the Master LPAR that you specify for that CPC. So, if you specified a start time of 0900, and that NYC is to be the Master LPAR on this CPC, PLEXCALC would calculate the UTC corresponding to 9:00 local time in the NYC LPAR, and then use that UTC for calculating the utilization for each LPAR. If you specified that CHI is to be the Master LPAR, it would calculate the UTC corresponding to 9:00 local time in the CHI LPAR, and so on. This allows PLEXCALC to calculate the actual utilization for this CPC regardless of the local time in each LPAR.

So, we have handled the situation where the LPARs in one CPC have different local times. What happens if you have a configuration like that shown in Figure A-4 on page 101. In this case, the production LPAR supporting New York is on CPC A, and the production LPAR supporting London is on CPC B. Both CPCs contain development LPARs for both locations. So, for example, the development sysplex for London can benefit from the available capacity on CPC A while the New York production system is not busy. As a result, the sysplex that each CPC is in depends on what time of day you look at it. At 06:00 New York time, CPC A would be considered to be in the DEVLOM sysplex, and CPC B would be in the PROD sysplex. However, if you were to measure the CPCs at 14:00 New York time, it would appear that CPC A is in the PROD sysplex, while CPC B is in the DEVNY sysplex.
To cater for this situation, PLEXCALC allows you to specify a Master LPAR for each CPC. This Master LPAR determines what UTC time is used for the calculations for that CPC. In this example, you would specify a start time of 09:00 and Master LPAR of PRDNY1 for CPC A, and PRDLON for CPC B. This would result in 14:00 to 22:00 UTC being used as the measurement period for CPC A, and 09:00 to 17:00 UTC being used as the measurement period for CPC B.

Figure A-4  Different time zones for different CPCs
Sample IWMWSYSQ code

The WLM IWMWSYSQ service provides a wealth of performance and capacity information about the sysplex. In relation to workload balancing, there are two immediately obvious uses of this service:

- Providing available capacity information to your system automation tool in order to adjust Scheduling Environments
- Obtaining goal achievement and available capacity information from the user exits of workload balancing products or facilities that do not use WLM themselves

The example shown in this chapter was provided by a customer. This customer uses the code with their automation product to adjust the volume of batch work running on each system. In this particular case, the customer uses it together with Defined Capacities to influence its subcapacity charging; however, the principle applies equally to balancing utilization for sysplex aggregation purposes.
The sample code

The customer that provided this code has a seven-way sysplex. They used this code to balance production batch work across two production systems on each of two CPCs. The customer has implemented data sharing, meaning that any batch job can potentially run on either of the systems in the sysplex. The customer also uses WLM-Managed initiators and Scheduling Environments.

The assembler routine in Example B-1 on page 104 shows you how you can call IWMWSYSQ. As shown in the example, it writes one line for every minute and every system in its joblog. The line contains the unused capacity in all LPARs in the sysplex for the last 60 seconds, the last three minutes and the last ten minutes, as well as the used capacity in the different service classes.

If you copy the program and the provided JCL to your z/OS system, the program runs as a batch job. You will probably want to assemble and link this program into a load library and start it as a Started Task. The parameters in the parm list allow you to specify how often a new line should be written, and how long the started task should run. The first parameter is specified in the format: MMSSTH. The second parameter is for testing purposes only.

The customer that provided this sample program used it to balance their batch jobs evenly across two production systems on each of two CPCs. They started a program on every production system which used the output from the provided sample to decide on which of the production systems jobs should run. They used the 10-minute unused CPU value to identify the system with the lower unused capacity value - the program then turned OFF a scheduling environment on that system, forcing the jobs using that scheduling environment to run on the other CPC (the one with more available capacity). Obviously additional checks were made to ensure that, amongst other things, every scheduling environment was turned ON on at least one system.

Using this methodology, they accomplished a very even usage of their production systems. This is important if your production LPARs are running on different CPCs and you would like to aggregate them. An unevenly distributed batch workload you can contribute to your missing the aggregation criteria, especially if you are very close to the 50% line.

The code provided in Example B-1 is intended to give you a starting point to develop a solution of your own using the IWMWSYSQ service. It is provided on an as-is basis. Further information about the IWMWSYSQ service is available in z/OS MVS Programming: Workload Management Services, SA22-7619.

Example: B-1 Sample code and JCL for calling IWMWSYSQ service

```plaintext
//TTTTTTTTT JOB (6311,00000,5162,SP50574),SYSPROG,MSGLEVEL=(1,1), 00000100
// CLASS=E,MSGCLASS=V,SCHENV=WLM,TYPRUN=HOLD 00000213
//*                                                                     00000300
/*JOBPARM S=ANY                                                         00000400
//*                                                                     00000500
//STEP1  EXEC PGM=ASMA90,REGION=4M,PARM='OBJECT,NODECK'                 00000600
//SYSPRINT DD  SYSOUT=*                                                 00000700
//SYSPUNCH DD  DUMMY                                                    00000800
//SYSLIB   DD  DSN=SYS1.MACLIB,DISP=SHR                                 00000900
//SYSUT1   DD  DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(CYL,(5,1))                00001000
//SYSUT2   DD  DSN=&&SYSUT2,UNIT=SYSDA,SPACE=(CYL,(5,1))                00001100
//SYSUT3   DD  DSN=&&SYSUT3,UNIT=SYSDA,SPACE=(CYL,(5,1))                00001200
//SYSLIN   DD  DSN=&&LOADSET,UNIT=SYSDA,SPACE=(CYL,(1,1)),              00001300
//             DISP=(MOD,PASS)                                          00001400
//SYSIN    DD  *                                                        00001500
PRINT  ON,NOSH 00001600
```
Appendix B. Sample IWMWSYSQ code

```
MODULE QUERYSI

CSECT

AMODE 31

RMODE 24

SPACE 2

R0 EQU 0
R1 EQU 1
R2 EQU 2
R3 EQU 3
R4 EQU 4
R5 EQU 5
R6 EQU 6
R7 EQU 7
R8 EQU 8
R9 EQU 9
R10 EQU 10
R11 EQU 11
R12 EQU 12
R13 EQU 13
R14 EQU 14
R15 EQU 15

*

STM R14,R12,12(R13)
BASR R12,0
USING *,R12
ST R13,SAVE+4
LA R13,SAVE

* L R1,0(R1) STORE EXEC PARM STIMER & CYCLE COUNT
MVC INTV+2(6),2(R1) SAVE INTERVAL TIMER VALUE MMSSTH
PACK KPWORK8,9(S,R1) CONVERT CYCLE COUNTER
CVB R2,KPWORK8
ST R2,CYCLECNT

* OPEN (PRINT,(OUTPUT))
PUT PRINT,HEAD01
PUT PRINT,HEAD02
STORAGE OBTAIN,LENGTH=8192,ADDR=(R11),BNDRY=PAGE,LOC=BELOW
USING SYSI,R11 DSECT FOR SYSTEM INFORMATION AREA
ST R11,SYSINFO

#SYSINFO DS OH
TIME DEC,OUTVAL,ZONE=LT,LINKAGE=SYSTEM,DATETYPE=DDMMYYYY
MVC WORK01,OUTVAL MOVE HHMMSSTH
O1 WORK01+3,X'0F'
MVC WORK02,MASK1
ED WORK02,WORK01
MVC TIME,WORK02+1 TIME STAMP

IWMWSYSQ SYSINFO_BLOCK=SYSINFO,ANSLEN=ANSLEN1,PLISTVER=MAX,
QUERYLEN=QUERYLEN,RETCODE=INFORC,RSNCODE=INFORSN

* CLC INFORC,ZERO
BNE #ERROR1
CLC INFORSN,ZERO
BNE #ERROR2
LH R9,SYSI_INUSE_ENTRIES # OF ENTRIES IN USE/LPAR'S
LR R10,R11 START ADDR. OF SYSI DSECT
AH R10,SYSI_HEADER_SIZE ADD HEADER SIZE OF SYSI
USING SYSI_ENTRY,R10 R10 = SYSTEM ENTRIES

* CLC INFORC,ZERO
BNE #ERROR1
CLC INFORSN,ZERO
BNE #ERROR2
LH R9,SYSI_INUSE_ENTRIES # OF ENTRIES IN USE/LPAR'S
LR R10,R11 START ADDR. OF SYSI DSECT
AH R10,SYSI_HEADER_SIZE ADD HEADER SIZE OF SYSI
USING SYSI_ENTRY,R10 R10 = SYSTEM ENTRIES

TM SYSI_ENTRY_STATUS,SYSI_CAPACITY_INFO_UNAVAIL EQU X'80'
```
BO    #ERROR3                                                  00010113
TM    SYSI_ENTRY_STATUS,SYSI_RESOURCE_CONSTRAINED EQU X'40'    00010213
BO    #ERROR4                                                  00010313

* MVC SYSID,SYSI_SYSNAME MOVE SYSTEM ID.

* LH R8,SYSI_ONLINE_CPU_COUNT NUMBER OF ONLINE CP'S
CVD R8,KPWORK8
MVC CPU_ONL,MASK3
ED CPU_ONL,KPWORK8+6

* L R8,SYSI_CPU_UP     LOAD CPU SU'S PER SECOND
CVD R8,KPWORK8
MVC CPU_SPEED,MASK2
ED CPU_SPEED,KPWORK8+4 CP SPEED IN CPU SERVICE UNITS

* LA R7,SYSI_SUM60      CONVERT ALL CPU SU'S UNUSED,IMP=D-1
LA R6,SYSI_60         FOR THE LAST 60 SECONDS
XR R4,R4
BAL R5,#EDITSU

* LA R7,SYSI_SUM180     CONVERT ALL CPU SU'S UNUSED,IMP=D-1
LA R6,SYSI_180        FOR THE LAST 180 SECONDS
XR R4,R4
BAL R5,#EDITSU

* LA R7,SYSI_SUM600     CONVERT ALL CPU SU'S UNUSED,IMP=D-1
LA R6,SYSI_600        FOR THE LAST 600 SECONDS
XR R4,R4
BAL R5,#EDITSU

* PUT PRINT,LINE01       YES, PRINT
AH R10,SYSI_SYSTEM_ENTRY_SIZE NEXT LPAR
BCT R9,#ENTRIES        ALL LPAR PROCESSED ?

* STIMER WAIT,DINTVL=INTV
L R9,CYCLECNT
BCTR R9,0        SUBTRACT ONE
ST R9,CYCLECNT
LTR R9,R9       ALL CYCLES PROCESSED ?
BNZ #SYSINFO

* STORAGE RELEASE,LENGTH=8192,ADDR=(R11)
CLOSE PRINT

#EDITSU DS OH
AH R7,LAST_ENTRY       LAST ENTRY 6 * 3 FULLWORDS
LA R1,7               SEVEN ENTRIES

#EDITGO DS OH
XR R2,R2
L R3,0(R7)           PROCESS THE ENTRIES
SR R3,R4           SUBTRACT PREVIOUS VALUE
L R4,0(R7)           SAVE VALUE
MH R3,KH010       CALCULATE IN TENTH OF A SECOND
D R2,SYSI_CPU_UP    DIVIDE CPU SERVICE UNITS BY CPU SPEED
CVD R3,KPWORK8
MVC WORK03,MASK4

00010113
00010213
00010313
00011105
00011205
00011305
00011405
00011505
00011605
00011705
00011805
00011905
00012005
00012105
00012205
00012306
00012405
00012507
00012607
00012707
00012807
00012907
00013012
00013107
00013207
00013307
00013407
00013512
00013607
00013707
00013807
00014007
00014107
00014207
00014307
00014407
00014507
00014607
00014707
00014807
00015007
00030705
00030805
00030905
00031005
00031105
00031205
00031405
00031507
00031607
00031707
00031807
00031907
00032007
00032107
00032207
00032307
00032407
00032507
00032608
ED  WORK03,KPWORK8+4                                         00032708
MVC  0(7,R6),WORK03+2                                         00032809
LA   R6,7(R6)                                                 00032907
SH   R7,KH012                                                 00033007
BCT  R1,#EDITGO                                               00033107
BR   R5                                                      00033205
#ERROR1 DS  OH                                             00033306
WTO  '*** RETURN CODE NOT ZERO ***',ROUTCDE=11             00033406
ABEND 4095,DUMP                             00033506
#ERROR2 DS  OH                                             00033606
WTO  '*** REASON CODE NOT ZERO ***',ROUTCDE=11             00033706
ABEND 4094,DUMP                             00033806
#ERROR3 DS  OH                                             00033906
WTO  '*** SYSTEM CAPACITY INFO NOT AVAILABLE ***',ROUTCDE=11 00034006
ABEND 4093,DUMP                             00034106
#ERROR4 DS  OH                                             00034206
WTO  '*** SYSTEM RESOURCE CONRAINED ***',ROUTCDE=11         00034306
ABEND 4092,DUMP                             00034406

PRINT DCB   DDNAME=PRINT,DSORG=PS,MACRF=(PM)                         00037105
SPACE 2                                                        00037405
SAVE   DC    18F'-1'                                                  00037505
WORK01  DC    F'-1'                                                    00037805
WORK02  DC    CL8' '                                                   00037905
WORK03  DC    CL9' '                                                   00038008
MASK1   DC    X'4020202020202020'                                      00038108
MASK2   DC    X'4020202020202120'                                      00038208
MASK3   DC    X'40202120'                                              00038308
MASK4   DC    X'4020202020202120'                                    00038408
KPWORK8  DC    PL8'0'                                                   00038508
KH010   DC    H'10'                                                    00038708
KH012   DC    H'12'              SIZE OF ONE ENTRY (60,180,600 SEC.)   00038808
LAST_ENTRY DC  H'72'              SIZE OF SIX ENTRIES                   00038908
DS   0D                                                       00039008
DC   CL8'TIME VAL'                                            00039108
INTV   DC   CL8'00002998'                                            00039208
  *                  HHMMSTH                                             00039308
OUTVAL  DC   CL16' '          TIME AND DATE RETURNED             00039408
CYCLECNT DC   F'000059'              ------- TEST ONLY               00040005
SYSINFO  DC   F'-1'                                                 00042305
ANSLEN1 DC   F'2048'                                               00042405
QUERYLEN DC   F'0'                                                 00042605
INFORC   DS   F'-1'                      INITIAL VALUE X'FF'     00042705
INFORSN   DS   F'-1'                      INITIAL VALUE X'FF'     00042805
ZERO   DC   F'0'                                                 00042906
  *                                                      00044505
HEAD01 DS   OCL180                                              00044605
  DC   C'1'                                                  00044705
  DC   CL27'IWMWSYSQ: QUERY SYS. INFO.'                      00044805
  DC   CL49' UNUSED/USED CPU IN REAL SEC./ 60 SEC. INTV ->'  00045405
  DC   CL49' UNUSED/USED CPU IN REAL SEC./180 SEC. INTV ->'  00045505
  DC   CL49' UNUSED/USED CPU IN REAL SEC./600 SEC. INTV ->'  00045605
  DC   CL6' '                                                00047505
HEAD02 DS   OCL180                                              00047605
  DC   C' '                                                  00047705
  DC   CL6' TIME'                                             00047805
  DC   C' '                                                  00047905
  DC   CL8'SYSID ' SYSTEM- ID.                               00048005
  DC   C' '                                                  00048105
  DC   CL3'CPS'     NUMBER OF CPU ONLINE FOR THIS LPAR      00048205
| DC    | CL8' CPSPEED'    | SPEED OF CP IN CPU SERVICE UNITS | 00048305 |
| DC    | CL7' UNUSED'    | UNUSED CPU PER MIN.               | / 60 SEC. 00048405 |
| DC    | CL7' IMP=D'     | IMPORTANCE D CPU PER MIN.         | / 60 SEC. 00048505 |
| DC    | CL7' IMP=5'     | IMPORTANCE 5 CPU PER MIN.         | / 60 SEC. 00048605 |
| DC    | CL7' IMP=4'     | IMPORTANCE 4 CPU PER MIN.         | / 60 SEC. 00048705 |
| DC    | CL7' IMP=3'     | IMPORTANCE 3 CPU PER MIN.         | / 60 SEC. 00048805 |
| DC    | CL7' IMP=2'     | IMPORTANCE 2 CPU PER MIN.         | / 60 SEC. 00048905 |
| DC    | CL7' IMP=1'     | IMPORTANCE 1 CPU PER MIN.         | / 60 SEC. 00049005 |
| DC    | CL7' UNUSED'    | UNUSED CPU PER MIN.               | / 180 SEC. 00049105 |
| DC    | CL7' IMP=D'     | IMPORTANCE D CPU PER MIN.         | / 180 SEC. 00049205 |
| DC    | CL7' IMP=5'     | IMPORTANCE 5 CPU PER MIN.         | / 180 SEC. 00049305 |
| DC    | CL7' IMP=4'     | IMPORTANCE 4 CPU PER MIN.         | / 180 SEC. 00049405 |
| DC    | CL7' IMP=3'     | IMPORTANCE 3 CPU PER MIN.         | / 180 SEC. 00049505 |
| DC    | CL7' IMP=2'     | IMPORTANCE 2 CPU PER MIN.         | / 180 SEC. 00049605 |
| DC    | CL7' IMP=1'     | IMPORTANCE 1 CPU PER MIN.         | / 180 SEC. 00049705 |
| DC    | CL7' UNUSED'    | UNUSED CPU PER MIN.               | / 600 SEC. 00049805 |
| DC    | CL7' IMP=D'     | IMPORTANCE D CPU PER MIN.         | / 600 SEC. 00049905 |
| DC    | CL7' IMP=5'     | IMPORTANCE 5 CPU PER MIN.         | / 600 SEC. 00050005 |
| DC    | CL7' IMP=4'     | IMPORTANCE 4 CPU PER MIN.         | / 600 SEC. 00050105 |
| DC    | CL7' IMP=3'     | IMPORTANCE 3 CPU PER MIN.         | / 600 SEC. 00050205 |
| DC    | CL7' IMP=2'     | IMPORTANCE 2 CPU PER MIN.         | / 600 SEC. 00050305 |
| DC    | CL7' IMP=1'     | IMPORTANCE 1 CPU PER MIN.         | / 600 SEC. 00050405 |

**LINE01**

```plaintext
DS OCL180
```

**ASA**

```plaintext
DC CL6' ' ASA
```

**TIME**

```plaintext
DC CL6' ' TIME HHMMSS
```

**SYSID**

```plaintext
DC CL8' ' SMF SYSTEM ID.
```

**CPU_ONL**

```plaintext
DC CL4' ' NUMBER OF CPU ONLINE FOR THIS LPAR
```

**CPU_SPEED**

```plaintext
DC CL8' ' SPEED OF CP IN CPU SERVICE UNITS
```

**SYSI_60**

```plaintext
DC CL7' ' 7 ENTRIES UNUSED IMP=D-1 FOR 60 SEC.
```

**SYSI_180**

```plaintext
DC CL7' ' 7 ENTRIES UNUSED IMP=D-1 FOR 180 SEC.
```

**SYSI_600**

```plaintext
DC CL7' ' 7 ENTRIES UNUSED IMP=D-1 FOR 600 SEC.
```

**END**

```plaintext
QUERYSI
```

```plaintext
CVT DSECT=YES
IWMWYSI
IWMSET
CUT DSECT=YES
IWMYCON
END QUERYSI
```

```plaintext
//STEP2  EXEC PGM=IEWL,PARM='XREF,LET,LST,NCAL',REGION=512K
//SYSLIN DD DSN=&&LOADSET,DISP=(OLD,DELETE)
```

```plaintext
//SYSLMOD DD DSN=&&LOAD,DISP=(,PASS),SPACE=(TRK,(1,1,1)),UNIT=SYSDA
//SYSUT1 DD DSN=&&SYSUT1,UNIT=SYSDA,SPACE=(1024,(80,50))
```

```plaintext
//SYSPRINT DD SYSOUT=*  
```

```plaintext
//GOWLM EXEC PGM=QUERYSI,PARM='010000,00020',TIME=(*,10)
```

```plaintext
//* MMSTH CYCLE
```

```plaintext
//STEPLIB DD DSN=&&LOAD,DISP=(OLD,DELETE)
//PRINT DD SYSOUT=*,DCB=(RECFM=FBA,LRECL=180,BLKSIZ=0)
```

```plaintext
//SYSUDUMP DD SYSOUT=*  
```
Glossary

Your glossary term, acronym or abbreviation. Term definition.

Term1. Term1 definition.
Term2. Term2 definition.
Term3. Term3 definition.
Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this Redpaper.

IBM Redbooks

For information on ordering these publications, see “How to get IBM Redbooks” on page 114. Note that some of the documents referenced here may be available in softcopy only.

- *CICS Workload Management Using CICSPlex SM and the MVS/ESA Workload Manager*, GG24-4286
- *IMS in the Parallel Sysplex Volume I: Reviewing the IMSplex Technology*, SG24-6908
- *IMS in the Parallel Sysplex Volume II: Planning the IMSplex*, SG24-6928
- *IMS in the Parallel Sysplex Volume III IMSplex Implementation and Operations*, SG24-6929
- The chapter entitled IMS/ESA Workload Router in the IBM Redbook *IMS/ESA Multiple Systems Coupling in a Parallel Sysplex*, SG24-4750
- *IMS/ESA Shared Queues: A Planning Guide*, SG24-5257
- *Managing Software for Availability in a Parallel Sysplex*, SG24-5451
- *Maximizing OPC/ESA Throughput*, SG24-2130
- *Merging Systems into a Sysplex*, SG24-6818
- *Parallel Sysplex Application Considerations*, SG24-6523
- *SNA in a Parallel Sysplex*, SG24-2113
- *TCP/IP in a Sysplex*, SG24-5235
- *Using VTAM Generic Resources with IMS*, SG24-5487
- *z/OS Systems Programmers Guide to: Workload Management*, SG24-6472

Other publications

These publications are also relevant as further information sources:

- *CICS/TS System Definition Guide*, SC34-6428
- *CICS/TS Resource Definition Guide*, SC34-6430
- *z/OS Communications Server: SNA Network Implementation Guide*, SC31-8777
- The chapter entitled “z/OS Load Balancing Advisor” in *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- The chapter entitled “TCP/IP in a Sysplex” in *z/OS Communications Server: IP Configuration Guide*, SC31-8775
- *z/OS Communication Server: New function summary*, GC31-8771
- *z/OS Planning for Workload License Charges*, SA22-7506
- *z/OS MVS Planning: Workload Management*, SA22-7602
Online resources

These Web sites and URLs are also relevant as further information sources:

- System z9/zSeries software pricing Web site
  

- Restatement of the sysplex aggregation criteria
  

- Parallel Sysplex Home Page
  
  http://www.ibm.com/servers/eserver/zseries/psa/

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

A
affinities, handling 90
announcements 8, 17, 20
Automated tape sharing and switching 18
available capacity 68

B
base sysplex 6
BronzePlex
capacity upgrade considerations 38
considerations for workload balancing 37
definition of 28
planned outage considerations 34
recommendation 28
unplanned outage considerations 34
workload balancing 38

C
Capacity Backup Upgrade 21
CICS 18, 37
dynamic transaction routing 86
facilities for handling/removing affinities 88
support for Sysplex Distributor 87
support for VTAM GR 87
CICSPlex System Manager 86
cost of ownership 66
CPC upgrade options 38

D
DASD outages 34
Data outages 35
data sharing 25, 34
role in distributing work evenly 25
DB2 37
DB2 Connect 79
DB2 DDF 81
support for Sysplex Distributor 81
support for VTAM GR 81
DB2 Group Attach Facility 79
support in CICS 79
support in IMS 79
DB2 Sysplex Query Parallelism 81
DB2 Universal Java Connectivity Connector 80
Defined Capacity 91
distance considerations 26
DISTMETHOD
BASEWLM 72
ROUNDROBIN 72
SERVERWLM 72
dynamic workload balancing 25

E
Enhanced Catalog Sharing 18

F
Fixed WLC products 6
full-capacity VWLC 11

G
GDPS
and DASD outages 34
description 36
GRS Star 18
guaranteed capacity 91

H
hard capping 91
HSM common recall queue 18
HyperSwap 94, 36

I
IFLs 14
IMS 37
IMS Connect 84
IMS Group Attach 83
IMS Shaed Message Queue
Sysplex Distributor support 83
IMS Shared Message Queue 82
support for VTAM GR 83
IMS TM 18
IMS Workload Router 85
Intelligent Resource Director 92
ISTEXCGR user exit for VTAM GR 70–71
IWMWSYSQ
sample code 103–104
IWMWSYSQ WLM service 76

J
JES2 Checkpoint in the Coupling Facility 18

L
Large Systems Performance Reference 38
licences 9, 12
licencing
defined 9
Linux 14, 50
Load Balancing Advisor 73
LPAR capping 91
LPAR weights 91
M
multiple data centers  30

N
New Application Licence Charging  20
normal mode of operation
definition 20

O
objective
of this document  2
On Off Capacity on Demand  21
One Time Charge products  6

P
Parallel Sysplex
configuration options  3
types  6
planned outage  32–33
in a BronzePlex  34
in a PlatinumPlex  34
PlatinumPlex
advisages  32
attributes of  29
capacity upgrade considerations  38
configuration options  30
considerations for workload balancing  37
definition of  29
planned outage considerations  34
systems management considerations  35
unplanned outage considerations  34
workload balancing  38
PLEXCALC  21–22
required input  23
time considerations  23–24
using to verify sysplex aggregation qualification  25
which systems must be included  21
price per MSU  2, 6
pricing
defined  9
pricing metrics  9
PricingPlex  13
definition  2, 9
example  13, 16
informing IBM  9
PrimaryPlex
definition  9
PSLC  9

Q
qualifying enablement function
definition 10

R
RACF  37
RACF database caching  18
recommendations  7
Redbooks Web site  114
Contact us  xi
rolling IPLs  34

S
sample configurations  46
configuration 1  46
configuration 2  50
configuration 3  14
SCRT tool  20
single system image  28
SmartBatch multisystem processing  18
software licencing  9
software pricing  9
software pricing curve  3
software stack used for sample configurations  46
sub-capacity charging  19–20
comparison to sysplex aggregation criteria  20
submitting reports to IBM  24
Sysplex Distributor
enhancements in z/OS 1.7  72
sysplex aggregation
and the pricing curve  6
background  8
considerations for CBU  21
considerations for OOCoD  21
considerations for zAAPs  19
CPC capacity role in qualification determination  11
hardwa requirements  17
maintaining qualification  8
objective  2, 20
operational requirements  18
qualification  11
rules and criteria  17
sample configurations  12
sample financial benefits  7
software requirements  18
thresholds  8
validation  21
Sysplex Calculator tool  21
Sysplex Distributor  71
IMS Connect  85
interaction with WLM  72
port sharing  72
support in CICS  87
support in IMS  83
support in MQ  89
support in WAS  89
support in DB2 DDF  81
workload balancing options  72
Sysplex Verification Package  7, 23
System Logger LOGREC  18
System Logger Resource Recovery Services  18
System Logger SYSLOG (OPERLOG)  18
systems management  35–36

T
target audience  2
terminology  9
Index

time considerations 23
Tivoli Workload Scheduler 77
  support for WLM Scheduling Environments 78
TSO 18

U
unplanned outage 34
  in a BronzePlex 34
  in a PlatinumPlex 34
utilization threshold
  reasoning behind 8

V
VTAM Generic Resources 18, 69
  DB2 DDF 81
  interaction with WLM 70
  support in CICS 87
  support in IMS 83
  support in MQ 89
  user exit 70
VTAM Multinode Persistent Sessions 18
VWLC 9

W
WebSphere Application Server 89
  support for Sysplex Distributor 89
WebSphere MQ shared message queues 18
WebSphere MQ Shared Queues 88
  support for VTAM GR 89
WebSphere MQ Shared queues
  support for Sysplex Distributor 89
WLM 67–68
  IWMWSYSQ service 76
  relationship to sysplex aggregation 68
  resource groups 77
  Web site 69
  WLM’s objectives 67
WLM Scheduling Environments 76
WLM-Managed initiators 75
workload balancing 4, 16, 25, 37, 66
  algorithms 66
  capacity based 66
  goal achievement 66
  limitations 66
  role in improving application availability 67
  round robin 66

Z
z/OS.e 20
z/VM 14, 50
zAAP
  impact on sysplex aggregation 19
zSeries Application Assist Processors (zAAPs) 92
This Redpaper is aimed at technical management and the Systems Programmers that will have the task of implementing any changes that may arise in the course of adjusting your environment to meet the sysplex aggregation criteria. It provides information that will help you understand and quantify the impact that sysplex aggregation qualification may have on your availability and your IT costs.

Should you decide that changes are required to meet the qualification criteria, it will help you design/create a solution to create a sysplex environment that delivers both the availability and systems management benefits of Parallel Sysplex and the cost savings that can be achieved through the exploitation of sysplex aggregation.