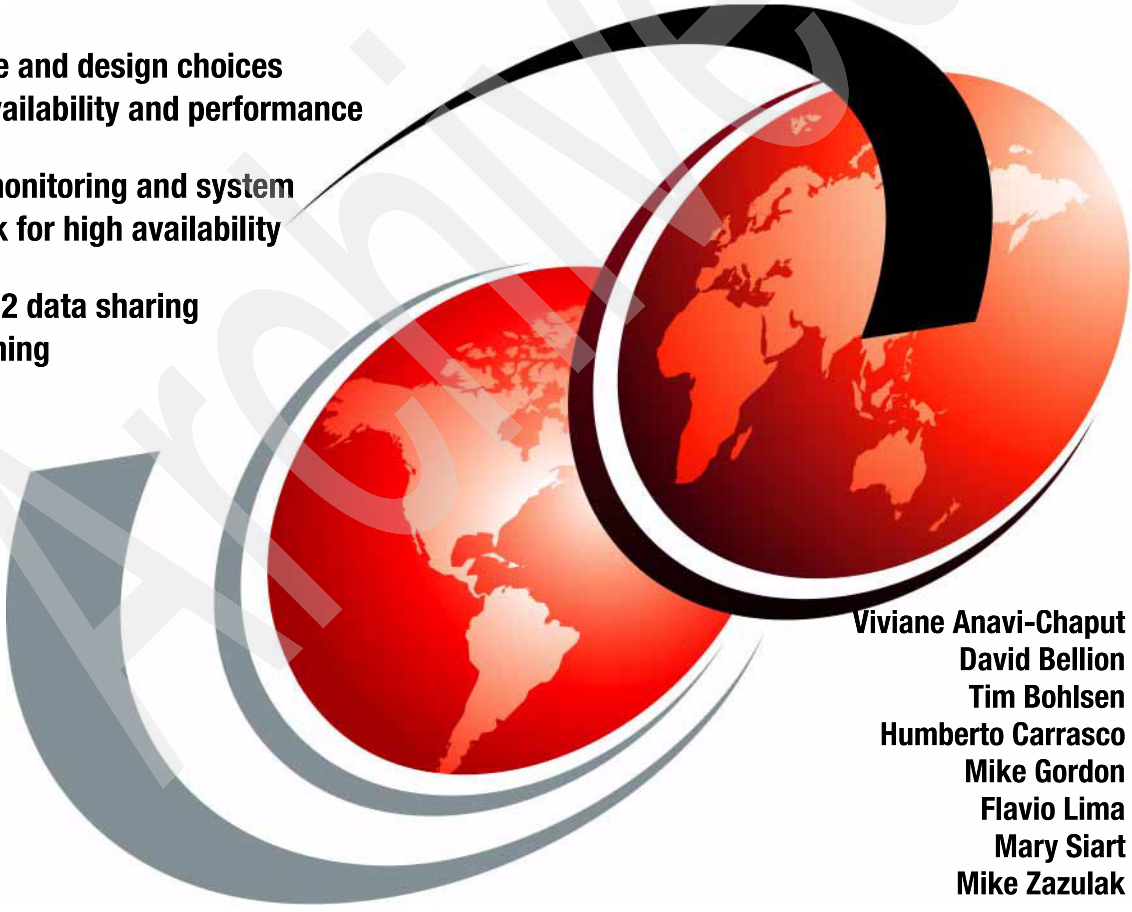
SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing

- Architecture and design choices affecting availability and performance
- Workload monitoring and system healthcheck for high availability
- Sysplex, DB2 data sharing and SAP tuning



Viviane Anavi-Chaput
David Bellion
Tim Bohlsen
Humberto Carrasco
Mike Gordon
Flavio Lima
Mary Siart
Mike Zazulak

Redbooks

ibm.com/redbooks

International Technical Support Organization

SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing

December 2003

SG24-6950-00

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

First Edition (December 2003)

This edition applies to DB2 for z/OS Version 7 and 8, and SAP for DB2 on z/OS Version 6.2.

Contents

Notices

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

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

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

The following paragraph does not apply to the United Kingdom or 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:

AIX®
DB2®
DB2 Connect™
DB2 Universal Database™
DFSMSdss™
DFSMShsm™
DRDA®
ESCON®
eserver ™
FlashCopy®
GDPS®
Hiperspace™
IBM®
MVS™
OS/390®
Parallel Sysplex®
PR/SM™
Redbooks™
Redbooks (logo) ™
RMF™
S/390®
Sysplex Timer®
VTAM®
WebSphere®
z/Architecture™
z/OS®
zSeries®

The following terms are trademarks of other companies:

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

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

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

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

Preface

This IBM® Redbook addresses the challenges posed by monitoring high availability, scalability, and performance in an SAP sysplex data sharing environment.

The book has three parts:

- Part 1 describes the architecture and design choices affecting both SAP availability and SAP performance.

  We introduce the motivations for utilizing a design based on DB2® data sharing. We discuss the principal SAP-DB2 data sharing architecture options and trade-offs used in the industry today and issues that play a role in both high availability and scalability, such as failover design, database connectivity design, workload splitting and load balancing, MCOD, and coupling facility design.

- Part 2 describes important sysplex and data sharing design issues that affect high availability.

  We discuss single point of failure, important failover scenarios and outage avoidance, automation of high availability constructs, and backup and recovery considerations in data sharing environments.

- Part 3 describes SAP performance and scalability considerations in a data sharing environment.

  We discuss performance issues in the order you would approach them at planning and implementation time. First, we discuss tuning the sysplex, which is the base for a well-performing DB2 data sharing system. Next, we discuss tuning the DB2 data sharing system, which is the base for a well-performing SAP system. Finally, we discuss tuning the SAP system.

  We focus on initial planning for performance and monitoring it afterward, and we explain the key points to look for to health-check your system and maintain high performance.

The team that wrote this redbook

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

Viviane Anavi-Chaput is a Senior IT Specialist for DB2, BI, and ERP at the IBM International Technical Support Organization, New York. She writes extensively, teaches worldwide, and presents at international conferences on all areas of Business Intelligence, ERP, and DB2 for OS/390®. Before joining the ITSO in 1999, Viviane was a Senior Data Management Consultant at IBM Europe, France. She was also an ITSO Specialist for DB2 at the San Jose Center from 1990 to 1994.

David Bellion is technical director of Oserve, an IBM Business Partner providing technical support for SAP on z/OS® and OS/390-DB2. He has 22 years of IT experience in the IBM mainframe environment, moving from systems programming in 1992, to work for the past 11 years as an SAP basis consultant. He is an SAP-certified technical consultant for SAP on the OS/390-DB2 platform. He has worked on many SAP OS/390-DB2 implementations at customer sites throughout Europe, including several years with IBM UK supporting their global SAP implementation on OS/390-DB2. He also teaches on several aspects of the subject. He holds a B.A. honors degree in History from Portsmouth University.

Tim Bohlsen is an SAP, DB2, and z/OS Specialist. He has worked in various locations worldwide with SAP on DB2 for z/OS customers during the past six years, as well as instructing for IBM Learning Services on this topic during this time. He has eight years of experience with SAP and 17 years of experience in the IT industry. Prior to working with SAP R/3, Tim was an MVS™ System Programmer in Australia for five years, specializing in large-system performance and automation. He holds an Honours degree in Computer Engineering from Newcastle University.

Humberto Carrasco is a DB2 for z/OS Certified Solutions Expert in Spain. He has seven years of experience in the mainframe environment, giving IBM service support for a variety of DB2 for OS/390 or z/OS customers in Spain. He is an SAP-certified technical consultant on the OS/390-DB2 platform. He holds a degree in Physics from the University Complutense of Madrid. His areas of expertise include DB2 data sharing environments and SAP R/3 on DB2 for OS/390 or z/OS.

Mike Gordon of Gordon Consulting, Inc., is contracting as a consulting solutions architect with IBM SAP e-business Technical Sales - Americas. He has 41 years of experience in the information systems field, 31 of which were spent as an IBM Consulting Systems Engineer. He has 15 years of experience architecting and implementing SAP R/2 and R/3 systems. He holds a Bachelor of Science in Mechanical Engineering from Union College in Schenectady, NY. He has participated in multiple Redbook projects and presented SAP on DB2 for OS/390 and z/OS topics at SAP Tech Ed and Share.

Flavio Lima of Bank of America and of Lima Software Inc. has more than 20 years of experience in data processing and more than 10 years of experience in

MVS, zO/S systems, and DB2 tuning and capacity planning. He assisted in the implementation and tuning of one of the world's largest DB2/SAP databases located in the United States. Flavio publishes technical articles about DB2 tuning pratices in a DB2-related magazine.

Mary Siart is a Technical Sales Specialist for SAP on zSeries® in IBM e-business Technical Sales (eTS), Americas organization. She has 24 years of experience in the IT industry. Prior to rejoining IBM in 2002, she worked for Motorola for 9 years in the database management area. She has been working with SAP on OS/390 since 1996. Her primary focus has been to leverage current and new technologies to develop an infrastructure, for SAP on OS/390 and z/OS, that will provide high availability and continuous operation.

Mike Zazulak is an IBM Technical Sales Specialist for SAP on zSeries in e-Business Technical Sales Support North America. He has been with IBM since 1982. Since September 1998, Mike has been in that position working with SAP R/3 for DB2 UDB for OS/390 and z/OS. His primary responsibilities are supporting the sales team and customers with SAP R/3 for DB2 UDB on OS/390. Prior to this he was a member of the joint IBM/SAP DB2 UDB Porting Team in Walldorf, Germany, where the SAP R/3 was ported to run on DB2 UDB for OS/390 and z/OS.

Special thanks to the following people for the consultancy and technical guidance they provided to this project:

John Campbell
IBM UK

Namik Hrle
IBM Germany

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

Gill Hanlon, Leslie Pendlebury-Bowe
Sony, UK

Brad St Pierre
Accenture, Ottowa, Canada

Kim Andersen
KMD A/S, Denmark

Nick Vallance
OSeRVE, UK

Florence Dubois
IBM, France

Wolfgang Reichert
IBM, Germany

Mark Gordon, JimTeng
IBM US

Thanks to Terry Barthel, Alison Chandler, and Alfred Schwab for their editorial assistance, and Ella Buslovich for her graphics assistance.

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 will team with IBM technical professionals, Business Partners, and/or customers.

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

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

> **ibm.com**/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an e-mail to:

  redbook@us.ibm.com

- Mail your comments to:

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

Part 1

Architecture and design choices affecting availability and performance

Part 1 describes the architecture and design choices affecting both SAP availability and SAP performance.

We introduce the motivations for utilizing a design based on DB2 data sharing. We discuss the principal architecture options and trade-offs used in the industry today, and issues that play a role in both high availability and scalability, such as failover design, database connectivity design, workload splitting and load balancing, MCOD, and coupling facility design.

1

Introduction to SAP high availability and performance architecture

This chapter discusses the motivations for implementing an SAP-DB2 data sharing solution and the building blocks for a continuously available and scalable system.

We describe the following topics:

- Why Parallel Sysplex and data sharing for SAP
- Parallel Sysplex® architecture
- DB2 data sharing architecture
- SAP Sysplex failover architecture

1.1 Why Parallel Sysplex and data sharing for SAP?

There are several motivations for pursuing an SAP implementation based on zSeries Parallel Sysplex and DB2 data sharing. Many customers are deploying SAP applications in support of business-critical operations. Typical business drivers include a desire to run a single global SAP instance, customer and supplier access to Web-based SAP applications around the clock, and support of 24x7 manufacturing and distribution operations. Those business drivers lead to the following IT requirements:

- Near-continuous system availability
- Central processor scalability through horizontal growth

1.1.1 Continuous availability

These applications require what is known as *near-continuous availability*. Availability in this context means the ability of all or some of the end users and applications to access the production databases. The question of what percentage of work can be processed on a continuous basis is an economic decision based on cost and the value of continuous availability to the organization. Although there is no standard definition of near-continuous availability, many large companies have adopted a goal of no more than five to 10 hours per year of planned plus unplanned outages. This translates to an overall systems availability of 99.9%. This contrasts to the historical paradigm of a maintenance window of eight hours every weekend, or 400 hours of outage per year. Other companies are adopting service-level objectives slightly less demanding but still challenging. For example, an objective of a quarterly four-hour maintenance window (16 hours per year of planned outage) requires many of the same design principles as the near-continuous availability objective.

There are four essential building blocks for a continuously available system:

- A design with no single points of failure
- Highly reliable hardware and software components
- Ability to avoid planned outages
- Seamless mechanisms for relocating work loads

One popular definition of continuous availability is:

continuous availability = high availability + continuous operations

High availability

There is much confusion in the industry over the use of the terms *reliability* and *availability*. Reliability is the resilience of a system or component to unplanned outages. This is typically achieved through quality components, internal redundancy, and sophisticated recovery or correction mechanisms. This applies to software as well as hardware. Z/OS has millions of lines of code devoted to correction and recovery. This reliability coupled with a design that eliminates single points of failure at the component level provides what is known in the industry as *high availability*. In fact, this is high *reliability*, or the ability to avoid unplanned incidents. To achieve continuous availability, we strive for a design that has 99.999% reliability.

Continuous operations

The other component of continuous availability is *continuous operations*, the ability to avoid planned outages. This includes the ability to upgrade and maintain the central processors, the operating systems, DB2, and coupling facilities without taking the SAP database down. It also means that DB2 database maintenance must be done while applications are reading and writing data. This requires Online Reorganization, Online Image Copy, and the ability to nondisruptively flash or snap volumes of data. For purposes of this discussion SAP maintenance (such as kernel upgrades, hot packs, and release upgrades) are treated as application maintenance and beyond the scope of infrastructure continuous operations. Requirements in this area have been well-documented with SAP.

1.1.2 Processor scalability

The infrastructure that we have put into place to support near-continuous availability is essential for horizontal processor scalability as well. Historically systems grew *vertically* by adding engines to the machine (also known as a symmetric multiprocessor or SMP or CEC) or by introducing faster processors. This approach limited the size of an SAP system to the largest single SMP or CEC. It also was also constrained by the amount of data and control information that could be held in the primary DB2 address space (the DBM1 address space). The SAP DB2 Parallel Sysplex architecture enables us to overcome these constraints and cluster multiple CECs in a single DB2 data sharing group. This enables horizontal growth of both processor power (MIPs) and virtual storage. Data sharing also gives us another tool in workload management as we can now level multiple workloads across two or more machines.

1.2 Parallel Sysplex architecture

A fundamental building block for both high availability and continuous operations is the notion of clustered database servers operating against a single copy of the data. Figure 1-1 on page 6 introduces a high-level picture of the elements of a zSeries Parallel Sysplex environment.

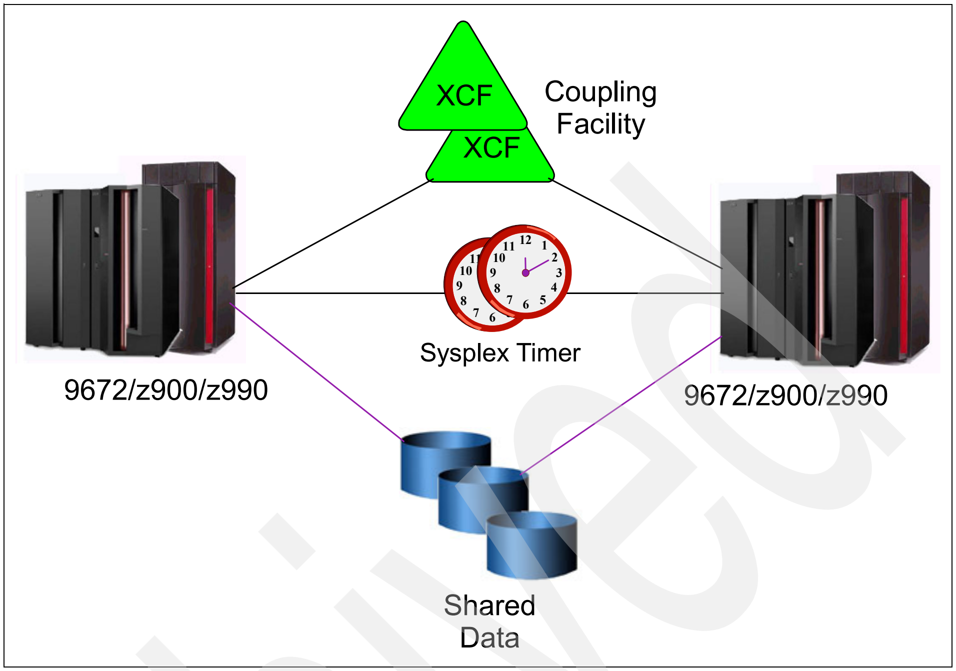


Figure 1-1 zSeries Parallel Sysplex architecture elements

In Figure 1-1 we see that a Parallel Sysplex system is typically made up of two or more computer systems (known as a Central Electronic Complex or CEC), two or more special purpose zSeries computers known as Coupling Facilities (either internal, ICF, or external) for the storing of shared Operating System and DB2 structures between the CECs, two external time sources called Sysplex Timers, sysplex-wide shared data, and multiple high-speed, duplexed links connecting the components. This implementation employs hardware, software, and microcode.

1.3 DB2 data sharing architecture

Figure 1-2 on page 7 completes the picture by laying multiple DB2 data sharing members on top of the Parallel Sysplex infrastructure.

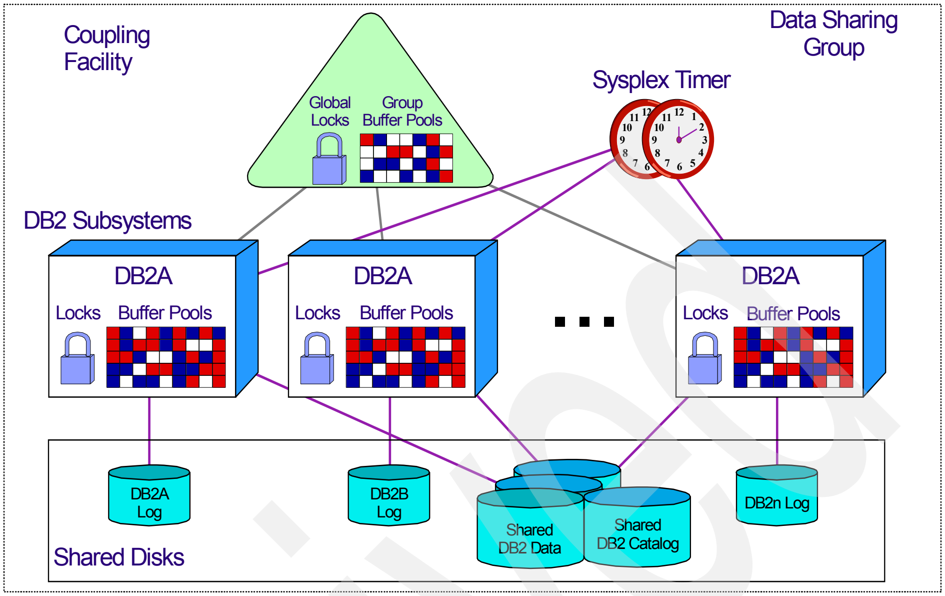


Figure 1-2 DB2 data sharing in a Parallel Sysplex

In this picture we see up to 31 DB2 subsystems (DB2 members) making up a DB2 data sharing group. There is one DB2 data sharing group for each production SAP system (an SAP system instance is also know as an SAP ID or SID). Each DB2 subsystem has its own set of DB2 logs, local buffer pools, and local locks managed by a companion IRLM. The DB2 data sharing group shares the DB2 databases, the DB2 catalog/directory, and the DB2 data sharing structures (SCA, global locks, group buffer pools) stored in the coupling facility.

1.4 SAP Sysplex failover architecture

We complete the DB2 data sharing infrastructure picture with an SAP customization known as SAP Sysplex failover. Figure 1-3 on page 8 introduces the basic building blocks of an SAP Sysplex failover architecture.

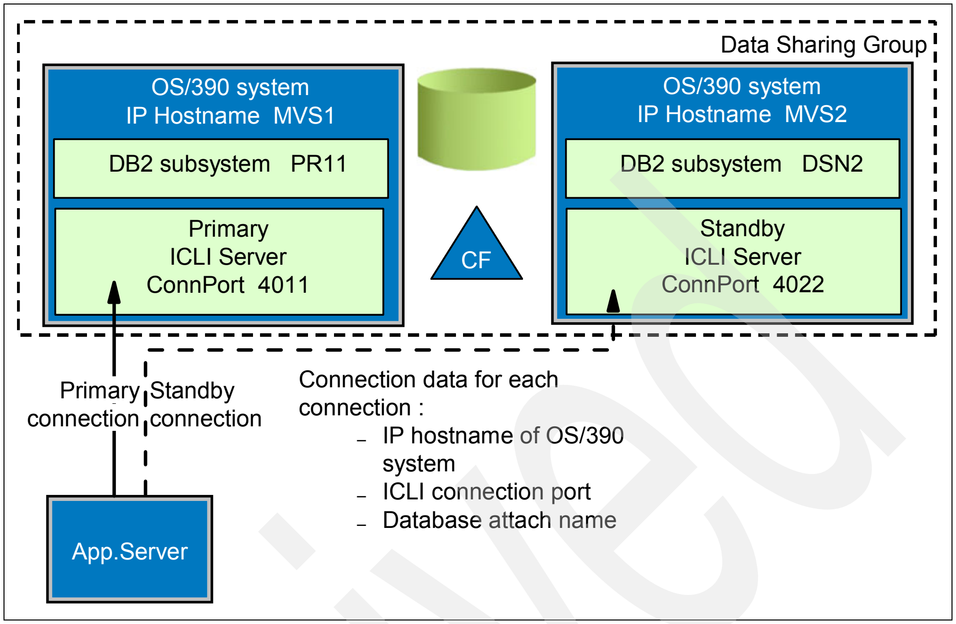


Figure 1-3 SAP Sysplex failover configuration: option 0 example

There are four major data sharing failover configurations providing a highly available environment for an SAP database on DB2 for z/OS:

- Option 0: Single DB2 member with passive (inactive) standby member
- Option 1: Two active DB2 members without passive standby members
- Option 2: Two active DB2 members, each with a passive standby member in same LPAR
- Option 3: Two active DB2 members, each with a passive standby member in a standby LPAR

We discuss these options in Chapter 2, "Architecture options and trade-offs" on page 11.

Figure 1-3 also includes some naming conventions that can ease managing and operating the SAP Sysplex failover configuration.

The three-character SAP instance name (SID name) is illustrated in the figure. We use the SID name (for example, PR1) as the DB2 Group Attach name.

We also introduce the notion of *primary* DB2 members, which normally have application servers attached doing productive work, and *standby* DB2 members, which are normally in hot standby mode with no attached application servers.

The primary DB2 member names are the DB2 Group Attach name plus a digit (for example, PR11), and the standby DB2 member names will be the DB2 Group Attach name plus a letter (PR1A).

This figure illustrates a three-tiered implementation in which each application server has a primary remote SQL server (an Integrated Call Level Interface or ICLI) in the primary DB2 member's LPAR (for example, MVS1) and a standby ICLI in the standby DB2 member's LPAR (MVS2).

Each application server has an SAP control file known as the *Profile*, which contains the z/OS LPAR IP host name, the ICLI well-known connection port, and the DB2 member name (PR11) of the primary DB2 member as well as the same information for the standby DB2. In the event of a planned or unplanned incident, the SAP Database Services Layer (DBSL) in the application server recognizes the need to fail over, looks for Standby information in the Profile, and connects the application server to the standby DB2 member.

2

Architecture options and trade-offs

This chapter explains SAP design considerations in a Parallel Sysplex data sharing environment. We describe:

- DB2 data sharing design options for SAP
- Failover design
- ICLI design

2.1 DB2 data sharing design options for SAP

There are four basic data sharing options for providing a highly available environment for an SAP database using DB2 on z/OS. Historically, we have identified them as option 0 through option 3. Originally, we started counting at option 1. However, after presenting to a customer they proposed another possible configuration that used more minimal resources. Because we had presented the original options to many other customers, we decided not to renumber the options to avoid confusion with following conversations, therefore we added option 0. This should not be a problem for techies who are used to counting from zero!

For review, the options are:

- Option 0: Single DB2 member with passive (inactive) standby member
- Option 1: Two active DB2 members without passive standby members
- Option 2: Two active DB2 members, each with a passive standby member in same LPAR
- Option 3: Two active DB2 members, each with a passive standby member in a standby LPAR

We do not go into much detail about each option's configuration. This is already described in the redbook *SAP R/3 on DB2 for OS/390: Database Availability Considerations*, SG24-5690. We only discuss any applicable performance and monitoring aspects of the choice of options.

2.1.1 Option 0 - Single member with passive (inactive) standby member

This option is chosen most often when high availability is the main concern and the current hardware is sufficient to handle all database SAP requirements identified from an SAP sizing or Insight for SAP report. In other words, your SAP database workload can be handled by one DB2 data sharing member in one CEC.

Under normal conditions (with every component working properly), the passive DB2 member and associated ICLI server should not use any system resources except for that which is required to start each component. Even though the idea of high availability is to eliminate human intervention, system programmers (both z/OS and SAP) should check the status of their systems periodically.

From a z/OS perspective, one of the easiest ways to check the current state interactively is to use the DA screen in SDSF. On this screen, one can view the CPU activity of individual address spaces. All that is required is to look for CPU activity on the passive standby DB2 address spaces or the standby ICLI server address spaces. Another method that is more passive is to direct ICLI server messages to the z/OS system console. This is accomplished by setting the environment variable ICLI_WRITE_TO_LOG to the value of 1. In the event of a failure, the application servers connect to standby ICLI servers and ICLI server message ICLS1300I is sent to the z/OS console. The database attach name is included in the ICLI server message. For this method to work, it is important to identify the DB2 member name in SAP profile parameter dbs/db2/ssid.

From an SAP system programmer (also known as SAP Basis) perspective it is not always possible to get access to a user ID with a TSO segment on the z/OS system to perform such monitoring. For the SAP programmers there is good news: The transaction DB2 has been enhanced, starting with SAP Release 4.6B, to enable an SAP GUI user to initiate a failover of application servers from one DB2 member to another. Implicit in this new functionality is the ability to determine which DB2 member is currently being used by an application server. From the main screen of transaction DB2, click **Installation Parameters**, then click **Database analysis** to see the current state. Click **Switch database host** to move application servers to the other DB2 member. In order to use this functionality, the new IFI data collector must be configured. See SAP notes 426863 and 509529 for details.

2.1.2 Option 1: Two active members without passive standby members

This option is considered most often when one DB2 member running in one CEC or LPAR cannot handle the SAP database workload from a CPU capacity standpoint or the SAP database workload would consume the entire 2 GB of virtual memory in the DBM1 address space. The 2 GB virtual storage limit is true for DB2 Version 7 and before, although DB2 V7 enables you to implement data spaces to take advantage of 64-bit real storage on zSeries machines.

When thinking about configuring DB2 to support more workload, this is most often the first thought that comes to mind. In this configuration, SAP sysplex failover is set up so that application servers will move (connect) from the failing active DB2 member to the other active DB2 member. If the respective machines supporting these DB2 members are sized just to fit the normal workload, then one should expect degraded performance when all of the SAP database workload is concentrated on the surviving DB. This degraded performance could come about due to lack of CPU capacity or lack of memory.

With recent announcements of the new zSeries z990 hardware and the latest version of DB2 software, Version 8.1, both the CPU and virtual storage constraints are being lifted. It might be possible to revert to option 0 and still support increased SAP workload.

The monitoring possibilties of this configuration are essentially the same as with option 0. Monitoring from z/OS can be accomplished with SDSF or the z/OS console. If you had more than one user ID with TSO segment, then you could be logged on to both LPARs and view the DA screen simultaneously.

Actually, it is possible to display the information about all address spaces in the sysplex from one SDSF screen. For ease of use, it would be beneficial to name the address spaces in such a manner as to make them easily discernable from each other.

Monitoring from the SAP side is the same also. When using the new RFCOSCOL-based IFI collector interface (see SAP note 426863), it is possible to execute transaction DB2 from any application server and check the status of any DB2 member.

In the case of failure for option 0, only one LPAR is in use, so there is no increase in database workload. Note that there is the possibility for one subset of application server work processes to be connected to the primary DB2 member and another subset of work processes in the same application server to be connected to the standby DB2 member. There is no real concern, because the workload is still in one LPAR. It only matters for monitoring purposes. It is not possible to see easily which DB2 member an individual SAP work process is communication with.

In general, we do not recommend that this option be used. We recommend option 2 instead. If you decide to use this option anyway, then we remind you to give careful consideration to sizing the hardware properly and configuring Workload Manager (WLM). If you require the same level of performance no matter what state the system is in, then each system should have enough CPU and memory capacity reserved to handle the maximum additional workload on each system. This standby capacity can be very costly if using the resources just for one SAP system. Fortunately, one of the great strengths of z/OS on zSeries is the capability to support multple workloads simultaneously. This is where WLM is important, because it enables you to assign importance to each workload. So in the event of a failover of workload to one surviving DB2 member, WLM can be configured to ensure that the SAP workload receives priority over the other workload, even if it is non-production SAP workload.

If it is non-production SAP workload, then extra definitions in WLM are required for WLM to distinguish between the SAP systems. Those familiar with the SAP on DB2 for OS/390 and z/OS series of planning guides should note that the sample WLM definitions assume that you are running one and only one SAP system per LPAR. All of the service classes begin with the prefix SAP. If you want to mix production and non-production workload or run multiple production workloads in the same LPAR, the sample definitions must be extended to control these workloads. One way is to create services classes for each SAP system.

For example, you could create PR1HIGH, PR1MED, and PR1LOW for SAP production system and DR1HIGH, DR1MED, and DR1LOW for the SAP development system. A more flexible naming strategy would be to put the SAP system name in the service classes.

Figure 2-1 shows how one large company with multiple SAP workloads has selected the data sharing architecture options best suited to each workload.

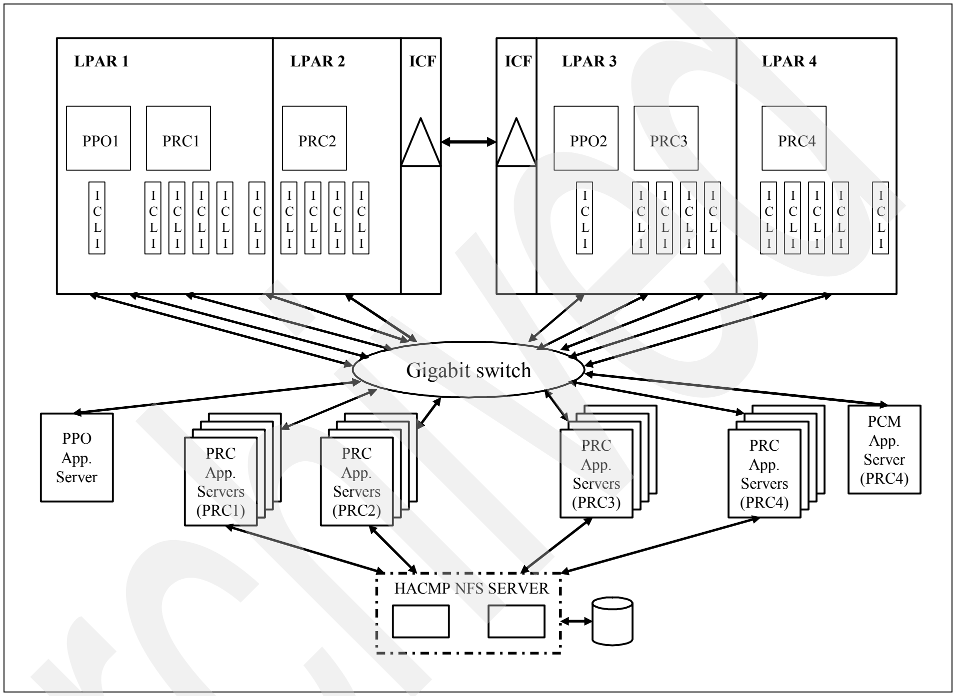


Figure 2-1 Large company using architecture options 0 and 1

This example shows a variation on data sharing options 0 and 1. The production plex has four LPARs spread across two mainframe servers with Emif links between them. Each server has an internal coupling facility defined.

Two production DB2s are running, supporting:

- R/3 Release 4.6C (PRC)
- APO Release 3.1 (PPO)
- CRM Release 3.0 (PCM)

R/3 and CRM are in an MCOD environment. APO has its own DB2 as MCOD is not yet supported for this release of APO.

R/3 runs four-way active data sharing. Each of the 16 application servers has its own ICLI, and the servers are spread evenly across all four DB2 members. As this is a 4.6C system, the old method of Parallel Sysplex failover is defined, with each application server having a standby server. Servers attached to PRC1 fail over to PRC4 and vice versa. PRC2 servers fail over to PRC3 and vice versa. For failover, the application servers share the ICLIs that are in use by the existing application servers of that LPAR.

Each DB2 data sharing member has the capacity to handle the extra load for failover.

APO runs two-way active /passive data sharing. It has its own primary and standby ICLIs.

CRM has its schema in the R/3 database. In effect it is running data sharing option 0, as the CRM server is attached to a single member within the data sharing group, with failover to a second member. CRM has its own primary and standby ICLIs.

With this setup, we can apply maintenance to z/OS and DB2 by controlled failover of the SAP systems during productive operation.

Changes for increased high availability:

- A second gigabit switch (although there is built-in redundancy for all components within the switch).
- Stand-alone enqueue server to replace the SAP CI as the single point of failure. Work is in progress on this.
- CRM in its own data sharing member. If we find that the CRM load affects R/3, or if we need different zparm settings for CRM, we can give CRM its own data sharing member within the PRC data sharing group.

2.1.3 Option 2: Two active members, each with a passive standby member in the same LPAR

Option 2 is really just a variation of option 0. In both options you have an active and standby DB2 member. Option 2 just has more pairs of active and passive members. This option is recommended or required to support any SAP requirement that exceeds the capacity of of a single machine.

Another use of this option would be to isolate SAP business components from each other. This is the logical extension of having separate application servers to run specific SAP modules.

2.1.4 Option 3: Two active members, with a passive standby member in an independent LPAR

Option 3 represents the option 2 solution carried to the next level. The inactive standby data sharing members reside in independent LPARs.

Data sharing architecture option 3 was used in early installations that were memory-constrained by the 2 GB of central storage per LPAR. In this configuration each primary DB2 and each standby DB2 would be in separate LPARs. Availability of z900 64-bit hardware and the 64-bit Real Storage Manager in OS/390 R2.10 and z/OS have effectively eliminated the need for this option, as more than 2 GB of central storage can be made available to a single LPAR.

2.1.5 How many data sharing groups?

A typical SAP core R/3 landscape consists of a development system, a quality control system, a stress test system, and a production system. Optionally, one might decide to have a training system, a technical sandbox, or a production support system.

It is common these days for businesses to roll out the other SAP technology components such as SAP Business Information Warehouse (BW), and Customer Relationship Management (CRM) and so on to support the next generation of mySAP Business Suite solutions. So it is quite common to have separate SAP landscapes for each SAP component.

Whatever SAP components or solutions you are implementing, the total number of SAP systems to build and maintain can add up quickly. It seems that every group involved in implementing an SAP system or solution wants their very own system to work with. Of all those systems, how many should be configured for high availability?

You already may have decided that your SAP production system must be configured to be highly available. Therefore, the production must be configured at the very least to run in DB2 data sharing mode.

What about the non-production SAP systems? The answer is not so easy. It depends on your service level agreement (SLA). Some SLAs require that even the developement system be highly available. It is very costly to have developers that cannot work because the system is unavailable. Whatever your SLA, we recommend that you configure at a minimum one other SAP system for DB2 data sharing in your promote-to-production landscape. This system is where you would test your code or configuration changes to verify that there are no problems related to running them in your data sharing setup. Your production system is not the place you want to learn that your changes do not work with DB2 data sharing.

Which non-production system should be configured for data sharing? It depends on how soon or late you want to test your changes with data sharing. Applications will run just fine with data sharing when doing single-user or component-level tests. The story might be quite different for stress tests. As the number of users running against different systems increases, you might have a bigger potential for resource contention, so we recommend that your other data sharing system is either your quality control system or your stress test system if you have one.

We recommend further that you consider having at a minimum one additional data sharing system in each of your SAP landscapes where your business needs require that you have high availablility for your production system. Each SAP component shares common technology, but there is also non-common functionality. A more important thing to keep in mind is specific landscape configuration work. So it is recommended that you have one additional DB data sharing system per SAP Landscape.

So far we have concentrated on figuring the number of data sharing systems to ensure that the application changes and SAP basis changes do not cause any problems with data sharing. What about the infrastructure changes such as coupling facility changes? What system should the infrastructure group use to test their changes? The infrastructure group should consider building a Parallel Sysplex with a data sharing system that is independent of the production and non-production SAP systems. It is sufficient to have one Parallel Sysplex for the infrastructure group. There is no need nor benefit to have one technical sandbox system per SAP landscape. This approach, while consistent, would be cost-prohibitive.

2.1.6 How many data sharing members?

After you have decided that you need DB2 data sharing, the next question is how many data sharing members are required for each highly available system. The answer to this question depends on the data sharing option you are implementing. The option you choose depends on the sizing estimate for your proposed production system or systems.

For an option 0 production system, you need only two data sharing members per data sharing group. The primary data sharing member does all of the work, and the secondary data sharing member is only a standby. We call this *passive data sharing*. This option is valid as long as your workload does not exceed the SAP rating of the zSeries box or production LPAR. One fact about this configuration that is difficult to live with is that you have a second zSeries box of at least equal

CPU capacity, memory capacity, and I/O bandwidth sitting there using electricity, but not doing any active work. So that leads us to option 1.

For an option 1 production system you have active workload distributed between two or more members of a data sharing group. Assume that you are configuring a two-way data sharing system. If one system fails or becomes unreachable, the workload will be moved to the surviving data sharing member. If you want the system to perform in failover mode with same level of throughput as in non-failover mode, then you must ensure that there be sufficient extra CPU capacity, memory capacity, and I/O bandwidth to handle the failed over work in one zSeries box or LPAR. Basically, you must double the capacity of each zSeries box. A zSeries box fails so rarely that it may not be so important to have all of that extra capacity ready for a failover situation that will rarely happen. In the DB2 V7.1 and lower releases, keep in mind that there is a 2 GB limit on the addressable memory in the DBM1 address space. Every SAP work process that connects to the surviving data sharing member will consume from 1.3 to 2.5 megabytes or more, so you must plan for the maximum number of RRSAF connections per data sharing member.

Another possible option 1 production system could have three data sharing members in a group. If one data sharing system fails with this version of option 1, you have the option to redistribute the workload to one of the surviving members or to redistribute the workload evenly among the surviving members. When making this decision, the main choices are to minimize DB2 inter-systems interest or not overallocate DB2 DBM1 virtual storage. To minimize DB2 inter-systems interest, ideally you would move all of the workload from the failing data sharing member to one of the surviving data sharing members. This might lead to overallocation on DBM1 virtual storage. On the other hand, to prevent possible overallocation of memory, you could evenly distribute the workload among the surviving members. However, this might increase DB2 inter-systems interest between the two surviving members. Which is the best choice to make? It is better to have the system available providing reduced throughput than to over allocate DBM1 virtual storage and risk another abend that would reduce throughput even more. Therefore, we recommend that option 1 systems redistribute the workload evenly among the surviving data sharing members.

To minimize DB2 inter-systems interest and prevent the overallocation of memory in a failover situation, we recommend that you implement option 2 instead of option 1. Option 2 eliminates DB2 inter-systems interest, because all workload from the failing DB2 data sharing member would move to a standby member. Also, no part of the surviving primary data sharing members would not be affected. This standby member could be started, ready, and waiting in the same LPAR as the primary or in another LPAR. Option 2 eliminates the possibility of overallocation of DBM1 virtual storage, because the failover happens to an empty data sharing member that is configured exactly the same as the primary.

In the event that only some of the SAP work processes failover to the standby data sharing member, a corresponding number of SAP work processes would be eliminated from the primary data sharing member. There would be no overallocation of virtual storage, but there could be inter-systems interest, but only for that portion of workload running on those data sharing members.

We recommend that you configure your non-production data sharing system or systems exactly the same way as your production system.

2.2 Failover design

As explained earlier, the notion of standby DB2 members was introduced for several reasons: it is less disruptive to surviving DB2 members (no washing of the buffer pools or dynamic statement caches), and it reduces the need to ensure that there is sufficient DBM1 virtual storage to absorb the movement of a large number of SAP work processes (hence DB2 threads). There are two groups of companies doing SAP DB2 Sysplex Data sharing: those who were motivated primarily by near-continuous availability (secondarily the ability to level load across the multiple CECs required for no single point of failure) and those who had both a scalability and availability objective. The first group typically implements option 0 and frequently is characterized by having many SAP production systems (that is, many production SIDs). Generally they will place half of the production DB2 members in a production LPAR on one CEC and the other half of the production DB2 members in a production LPAR on the other CEC. Each production LPAR will have the standby DB2 members for the other CEC.

Many of the companies pursuing both scalability and high availability have three or more CECs in their sysplex. A typical configuration in a three-CEC sysplex would have a primary DB2 on each CEC with a standby DB2 member residing in each production LPAR that contains a primary DB2, as shown in Figure 2-2 on page 21.

Machine ---------->	CEC1		CEC2		CEC3	
LPAR ------------->	MVS1		MVS2		MVS3	
DB2 member -->	PR11	PR1A	PR12	PR1B	PR13	PR1C
Workload -------->	FICO		SD		PM/MM	
App Srvr Group 1	**primary**			**standby**		
App Srvr Group 2	**primary**					**standby**
App Srvr Group 3			**primary**			**standby**
App Srvr Group 4		**standby**	**primary**			
App Srvr Group 5				**standby**	**primary**	
App Srvr Group 6		**standby**			**primary**	

Figure 2-2 Large company using architecture option 2

In the event of planned or unplanned loss of one of the production environments (for example, CEC1), half of the application servers will reconnect to the standby DB2 member on CEC2 and half will connect to the standby member on CEC3. Although this is more complex than simply moving all of the application servers to one standby DB2, it does offer workload management benefits. Assuming that each of the three primary DB2s were servicing one-third of the total workload, half of this (or one-sixth of the total workload) will be moved to each standby member in the surviving CECs. When coupled with the WLM ability to differentiate priorities (goals, importance, velocity) based on SAP work process type, very high interactive service levels can be maintained with minimized disruption to the surviving CECs. This enables us to minimize the purchase of extra capacity to support failover.

2.3 ICLI design

The Integrated Call Level Interface (ICLI) is an IBM feature shipped with OS/390 or z/OS that provides connectivity between the SAP application server and the DB2 database server. It can be described as a black box that delivers requests from the SAP application servers to the DB2 database server and returns responses from the DB2 database server to the SAP application server. All communications are initiated on the application server side.

The ICLI is used only when the SAP application server runs on AIX®, various versions of Windows® 2000, Solaris, or Linux for zSeries. Said another way, it is used for all application servers that are remote with respect to the DB2 database server. The application server that runs on OS/390 or z/OS UNIX System Services (USS) uses cross-memory services and ODBC to communicate with the DB2 database server.

The ICLI is made up of the ICLI client and the ICLI server. The ICLI client runs where the SAP kernel and Database Services Layer (DBSL) run (such as AIX) and the ICLI server runs in the LPAR where the DB2 database server is running. Each application server instance in your system landscape runs its own instance of the ICLI client. Each of these ICLI client instances can communicate with a single ICLI server instance or multiple ICLI server instances. There is no limitation on the number of ICLI server Instances that can be started in one LPAR. The only requirement is that each ICLI server must have its own port number to distinguish it from the others.

The SAP kernel directs the ICLI client via the DBSL to connect to a particular ICLI server. The kernel does this by passing the host name and port number where the ICLI server is running. So an SAP kernel can communicate with any ICLI server anywhere as long as it can be reached using TCP/IP. The other parameter that is sent in the connect request is the DB2 subsystem ID.

The SAP kernel has an additional function that enables it to provide two or more sets of parameters so that it can communicate with more than one ICLI server. This connectivity flexibility known as Sysplex failover is ideally suited for configuring high availability environments. Note that one ICLI client can talk to only one ICLI server at a time.

2.3.1 How many ICLI servers?

So how do we best configure our ICLI servers to take advantage of the connectivity function and use this sysplex failover function to implement a highly available SAP system and not impact system performance?

For the high availability requirement: As with most high availability systems, the key is to eliminate single points of failure. We can accomplish this with ICLI servers by starting more than one ICLI server per SAP system in order to divide the workload.

To examine how this would work with data sharing option 1, assume that we have two application servers connected to our data sharing system and that each one supports a different SAP module, such as SD and FI. We could configure both application servers to connect through a single ICLI server, but this makes the single ICLI server a single point of failure. To enable the SD users to work independently of the FI users, it would be better to configure two ICLI servers so that a failure of one ICLI server will not affect the connections established through the other ICLI server. This way either the SD users or FI users would still be productive if only one ICLI server fails.

Suppose that the number of SD and FI users is large enough that each module requires more than one application server to support the respective workload. We could configure the two SD application server instances to connect through the one ICLI server assigned to handle SD workload, and we could configure the two FI application server instances to connect through the ICLI server assigned to handle FI workload. It is still true that the SD users are isolated from the FI users, but now you have more of each. Even if only one ICLI server fails, fewer users are sitting idle until the ICLI server problem is resolved and it is restarted.

There is no real limit to the number of ICLI servers that can run in one LPAR, so we start one ICLI server per application server instance. Now the failure of one ICLI server would temporarily affect a smaller percentage of the user community and all business functions could still be operated. All of the users who were forced off during the failure would log on again, and the SAP logon load balancing function would direct the logons to the surviving application server instance that is capable of connectivity to the database server.

2.3.2 Transition from ICLI to DRDA

SAP and IBM have had a continuing long-term objective of making the transition from the Integrated Call Level Interface (ICLI) to the Distributed Relational Data Architecture (DRDA® or DB2 Connect™) open SQL client/server. When the porting project was started in 1995 DRDA was not yet optimized for a high-volume transaction processing environment. The joint SAP and IBM development team chose to build an SAP-specific remote SQL client/server known as the ICLI client and ICLI server. Over the years both DRDA and ICLI have been forced to deal with identical issues: new application server platforms (both hardware and operating systems), new network connectivity's (such as gigabit ethernet), new network stacks (such as TCP/IP), automation issues, and code conversion issues (e.g., Unicode). This has affected both development costs and speed to market. IBM and SAP have indicated that in the DB2 V8 and SAP Web Application Server 6.xx time frame there will be a transition from ICLI client/server to DRDA. At the time of this publication details of the migration path, release dependencies, and design considerations were not available.

3

Workload splitting and logon load balancing

This chapter discusses the following topics:

- Workload splitting
- Logon load balancing
- SAP scheduling and dispatching

3.1 Workload splitting

As we begin to plan our SAP DB2 sysplex data sharing implementation, our goal is to balance the load across the DB2 members and to minimize the cost of data sharing (that is, minimize multisystem interest). The word minimize is chosen carefully as we cannot eliminate multisystem interest, but we can definitely minimize its effect. There are several options for splitting up the workload of an SAP system:

- Based on business processes (e.g., order to cash, purchase to payment, financial and cost accounting) that incidentally align nicely with SAP modules
- Based on geographic regions
- Arbitrary (for example, one-third of users to each of three groups)

Most companies have chosen the business process approach. This approach has the desired attribute that users with common data needs are clustered together. This is a byproduct of a group of users aligning with the same SAP modules (for example, SD, FI, MM). With this approach we have seen data sharing overhead as low as half of the overhead of the other two approaches. Once again, we are not looking for perfection but to moderate the cost of data sharing.

3.2 Logon load balancing

SAP provides an excellent mechanism for implementing workload splitting called logon load balancing. SAP originally developed logon load balancing as an application server high-availability tool. Logon load balancing groups are defined, users are assigned to a group, and each group is mapped to two or more application servers. When a user logs on to SAP, the message server identifies the logon group, looks at the application servers servicing that logon group, and assigns the user to the least heavily used server. The high availability notion came from having two or more servers in each group, which means that if a server failed, the users logged on again and were assigned to a surviving application server in that logon group. In an SAP DB2 data sharing implementation we assign the application servers in a logon group to only one primary DB2 member. We can have multiple logon groups accessing that DB2, but no single logon group should access multiple DB2 members during normal operation. (Failover is a different story.) This provides the benefit of reduced multisystem interest because of the commonality of data interest.

Additional significant technical benefits to logon load balancing include:

- Improved application server buffer hit rates because each server has common data interests resulting in tighter reference patterns

- Higher DB2 buffer pool hit rates with smaller working set requirements or memory available for larger buffers
- Smaller local Dynamic Statement Cache requirement because with common workload on each SAP work process, fewer copies of the SQL statements are required

After you have decided on a logon load balancing strategy, the CCMS transaction ST03 can be used to monitor the workload. Using application aggregation helps in summarizing the views. Additional considerations are discussed in 3.3, "SAP scheduling and dispatching" on page 28.

Appendix A, "Checking SAP data sharing setup" on page 193, contains a check list for verifying the setup of logon load balancing and the other SAP and DB2 customizations described in this section.

3.2.1 Core R/3

The discussion in 3.1, "Workload splitting" on page 26 and 3.2, "Logon load balancing" on page 26 focused on a core R/3 system. This is the environment where the industry has the most experience. As stated previously, most companies have implemented a business process oriented logon load balancing strategy, which at a technical level results in a module orientation. SAP industry solutions have made this approach somewhat more difficult. Several companies with SAP industry solutions have decided to have dedicated interactive servers and dedicated batch application servers. They assign each group to its own DB2 member. As the resulting workloads consume resources in line with the systems sizing, this appears to be an acceptable approach.

3.2.2 BW, CRM, APO

Although several companies have built high availability SAP DB2 Parallel Sysplex environments (typically option 0) for these mySAP.com components, there is little experience with horizontal scalability and active data sharing. In the case of the mySAP Business Information Warehouse (BW), the early implementers believe there is an affinity between collections of users and the info-cubes they most frequently work with. This appears to be a natural approach to setting up logon groups for active data sharing. For mySAP Customer Relationship Management (CRM), it is believed that the largest number of customers will be regional in nature, so logon groups based on region is a sensible way to achieve affinities at least on the underlying customer data. Finally, if the SAP APO planning is to a plant or warehouse level, then the inventory and product data will also have a location-oriented affinity, and a regional approach could work here as well.

3.3 SAP scheduling and dispatching

In this section we discuss SAP scheduling and dispatching customizations that affect performance and the cost of data sharing.

3.3.1 Update (VB) Dispatching

SAP R/3 has long employed what is known as an asynchronous update process. An end user has a conversation with an application server building a document (for example, a customer order); goes through multiple screens that are executed on dialog work processes; reviews and verifies that the document is correct; and approves the work. The dialog work process writes the document to a form of internal processing queue known as the update (or VB) protocol tables (stored as DB2 tables) and alerts the SAP Task Dispatcher that the Update Dispatcher should schedule an asynchronous update work process. Figure 3-1 illustrates this work flow.

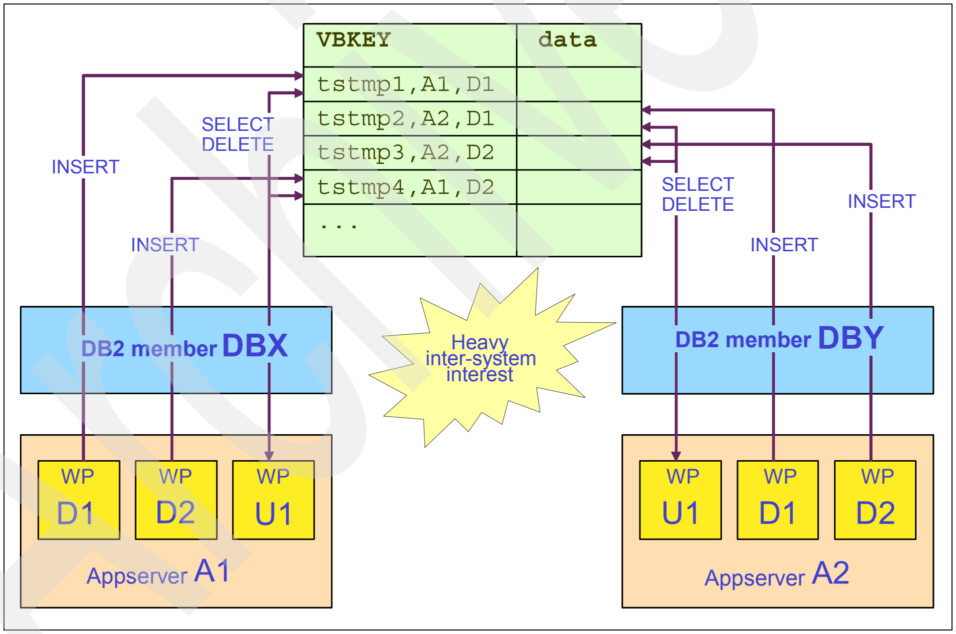


Figure 3-1 Asynchronous update work flow and VB protocol tables

In this figure we see multiple dialog work processes writing (inserting) to the VB protocol tables. These processes are on application servers connected to two different DB2 members. In addition, update work processes are reading (selecting) records to be written to the actual business databases and then, at completion of the update, deleting the records from the VB protocol tables. We have multiple DB2 members reading and writing to a very active set of DB2 tables resulting in a systems hot spot with heavy inter-systems interest.

The first step in addressing this problem is to partition the VB tables based on application server ID. Depending on the update dispatching option chosen, this could eliminate the multisystem interest. (Update table partitioning is further discussed Chapter 16.8.) Figure 3-2 illustrates the effect of VB table partitioning on inter-systems interest.

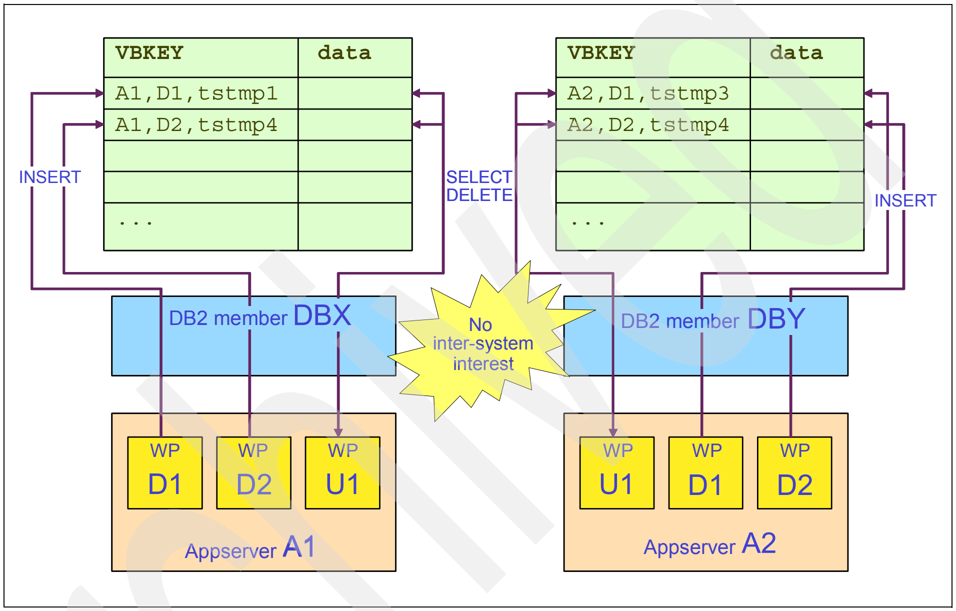


Figure 3-2 Partitioning of VB protocol tables

If we assume that *update local* has been chosen (more about that later), Figure 3-2 illustrates that with the appropriate SAP parameterization the dialog work processes on appserver A1 will only run against DB2 member DBX and that the A1 update work process will also run against member DBX with both accessing the same tablespace partition. A similar situation exists on appserver A2. The result is that there is no inter-system interest (except when the Basis administrators check to see whether all updates have been processed, which touches every DB2 table partition from a single application server and hence single DB2 member. Fortunately, **`pclose`** dries up this multisystem interest after a brief interval).

SAP on IBM DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server Release 6.20 contains a comprehensive description of the steps to be taken in partitioning the VB protocol tables. It is also important to check for current SAP OSS notes.

The final step in implementing update processing is to control the behavior of the SAP update dispatcher. Our objective is configure VB (update) in a way that achieves both load balancing and high availability while ensuring that there is no multisystem interest on the VB protocol tables. This dictates that we have multiple application servers eligible to process any update and that we dispatch the update requests to update work processes connected to the same DB2 member as the dialog/background work processes that initiated the request. Figure 3-3 illustrates that the update dispatcher operates in a round-robin manner.

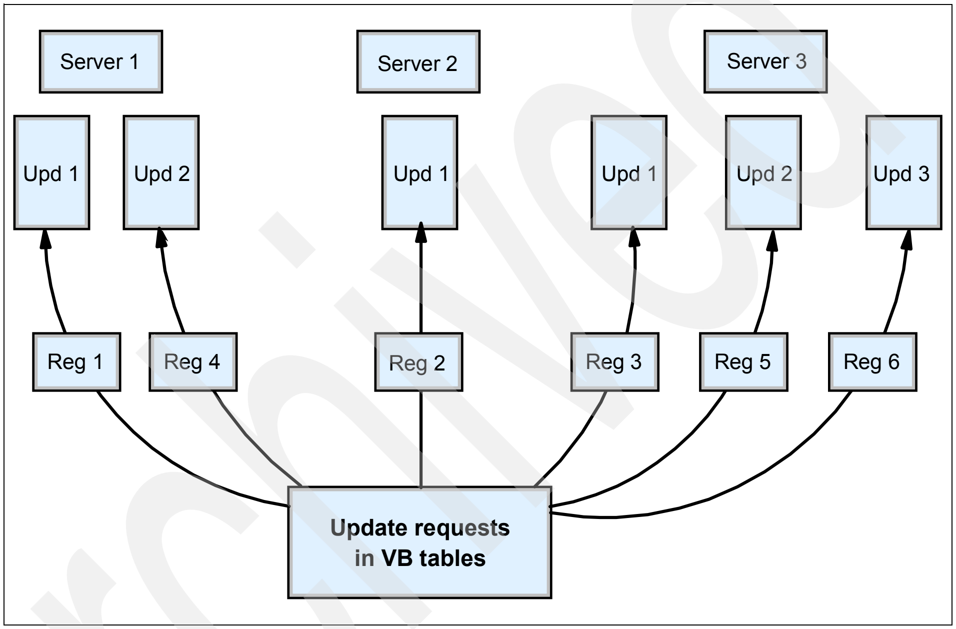


Figure 3-3 Round-robin update (VB) dispatching

In this picture there are three update servers with one to three update work processes each. The first update request is scheduled to Server 1, the second to Server 2, the third to Server 3, and then the dispatcher wraps back to Server 1 with fourth request. In practice this is a very effective process and we typically see update processing evenly spread across the available application servers.

Update dispatching is controlled by the SAP Profile parameter:

```
rdisp/Vb_dispatching = 0 or 1 (default = 1)
```

The simplest dispatching decision to implement is to set the dispatching parameter to 0. This tells the update dispatcher to only use update work processes on the application server that initiated the update request. This is known as *update local*. Unfortunately, although this approach does eliminate multisystem interest on the VB protocol tables, it does not accomplish either the load leveling or availability objective. Update intensive batch programs can overload the update work processes of a single application server while there might be many available update work processes on other servers. Furthermore, if the application server goes down, its update requests will remain unprocessed until that server comes back up. Update local is described in the *SAP 6.2.0 Database Administration Guide*.

Leaving the dispatching parameter set to 1 activates update dispatching. If nothing else is done, the updates will be scheduled across all available servers with update work processes, and heavy multisystem interest will occur. SAP provides a capability called Update Multiplexing, which satisfies all three objectives: load leveling, availability, and elimination of multisystem interest.

Figure 3-4 illustrates a system with two primary (active) DB2 members and a logon load balancing group assigned to the application servers attached to each member.

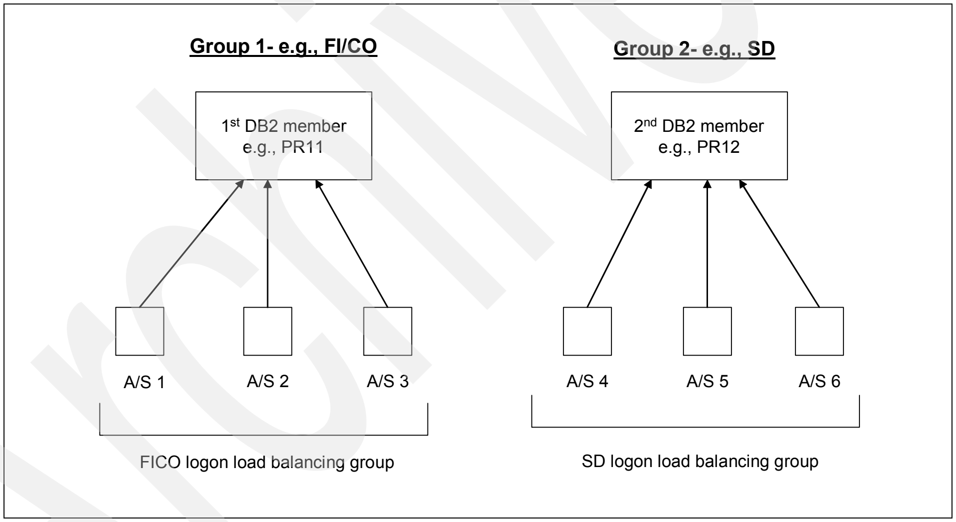


Figure 3-4 Update multiplexing: system with two DB2 members and two load balancing groups

Update multiplexing is created for this system by customizing two SAP Data Dictionary tables: ASGRP and APSRV. In Table 3-1 on page 32 we use the ASGRP table to define a multiplexing group name and relate it to a DB2 member.

Table 3-1 The ASGRP table defining multiplexing server groups

SERVERGRP	DBINSTANCE
GROUP1 - FICO	PR11
GROUP2 - SD	PR12

In this illustration we have defined two multiplexing groups, FICO and SD, with relationships to corresponding DB2 members PR11 and PR12. In Table 3-2 we use the APSRV table to map application servers to an update multiplexing group.

Table 3-2 APSRV table mapping application servers to server groups

SERVERNAME	SERVERGRP
Application Server 1	FICO
Application Server 2	FICO
Application Server 3	FICO
Application Server 4	SD
Application Server 5	SD
Application Server 6	SD

We can now see that the logon load balancing groups illustrated in Figure 3-4 on page 31 are tightly aligned with the update multiplexing server groups defined in Table 3-1 and Table 3-2. We have accomplished all three objectives: load balancing, availability, and elimination of multisystem interest.

3.3.2 Batch classes

With background or batch processing we have two objectives: running the batch at the proper priority and minimizing multisystem interest. Since SAP release 4.5.B, the SAP work process type has been registered with the z/OS Workload Manager (WLM). This has provided the ability to run batch at a lower priority than interactive dialogs. Section 15.3 discusses WLM policies.

Controlling multisystem interest caused by batch jobs has been more problematic. Starting with SAP WAS release 6.10 SAP provides functionality known as batch server groups. Transaction SM61 is used to define batch server groups. Transaction SM36 is used to assign jobs to a server group. This enables us to align the batch server groups with the logon load balancing and update multiplexing groups we created in the previous sections. The Batch Dispatcher schedules work across all available background work processes in a server group, so we achieve the benefits of load leveling and minimizing multisystem interest.

Prior to WebSphere Application Server V6.10, companies employed several strategies to moderate multisystem interest. In these earlier releases there were two options: point a job at a specific application server or allow scheduling across all available background work processes. We know of no one using the first technique as it is labor intensive to maintain, has poor availability characteristics, and requires manual load balancing. Most companies have either suffered the multisystem interest or limited the definition of background work process to the application servers associated with a single DB2 member. SAP operation mode switching helps in controlling the allocation of work processes.

3.3.3 RFC groups

RFC groups should be aligned with the logon load balancing, multiplexing, and batch server groups.

4

MCOD with data sharing

The SAP Multiple Components in One Database function (MCOD) enables more than one mySAP component to be installed into one instance of a Database Management System (DBMS). This chapter includes:

- A brief background discussion of the MCOD function and the various reasons why a customer would choose to use it
- Considerations for running an MCOD SAP system in a DB2 data sharing environment

You can find more details about the MCOD feature in a DB2 for OS/390 or z/OS environment in the redbook *SAP on DB2 Universal Database for OS/390 and z/OS: Multiple Components in One Database (MCOD)*, SG24-6914.

4.1 Basic principles of MCOD

In general there are two broad reasons for choosing to use the MCOD feature:c consistency and consolidation.

The consistency scenario is used where the components installed in an MCOD system have a close relationship in terms of the requirement of absolute point-in-time consistency of data across *all* of these components. Because the data of all components is contained in the one DBMS, the ability to maintain consistency of data in every situation, even in the case of prior point-in-time recovery, is greatly enhanced.

The second major reason for MCOD use is for consolidation. In this case the installation of multiple components into the same DBMS results in a reduction in the total number of DBMSs. This then results in a reduction in the administrative costs of maintaining the number of DB2 subsystems.

4.2 Performance versus high availability

Two major reasons for utilizing the DB2 data sharing feature of DB2 for zSeries are to allow system capacity growth beyond what one LPAR or one zSeries server can provide, and to provide a higher level of system availability through automated failover between DB2 data sharing members. These two reasons are also valid when using MCOD. The MCOD components can either use the same DB2 subsystem to provide the database server layer for the SAP system, or use different DB2 members in the DB2 data sharing group for the different mySAP components.

As discussed in the following section, in an MCOD environment the use of DB2 data sharing allows separate DB2 members for each of the components to make use of the resources of various LPARs and zSeries servers.

4.3 Design choices with data sharing

When using the MCOD feature to place multiple mySAP components into one DB2 subsystem, a major consideration will be whether DB2 is executed in data sharing or non-data sharing mode. Some of the design considerations when choosing the components to be installed into an MCOD DB2 subsystem may in fact drive the choice of whether DB2 data sharing is used.

The use of DB2 data sharing in an MCOD system has some significant advantages. Using DB2 data sharing enables you to run DB2 data sharing members on different LPARs and even different zSeries servers while operating on the same actual set of data.

One of the major advantages of using DB2 data sharing in an MCOD environment is the flexibility to adapt the performance characteristics of different DB2 data sharing members to the requirements of the MCOD component in question. To illustrate the possibilities, a customer may choose to install mySAP R/3 Enterprise (R/3), mySAP Business Information Warehouse (BW), and mySAP Customer Relationship Management (CRM) into one DB2 subsystem in an MCOD environment. Normally these components require specific tuning of DB2 DSNZPARMs to produce the optimum performance to users of the system, and these tuning actions are different for each component. Using DB2 data sharing means that you can adapt the DNSZPARM parameters in different members to the appropriate values, which would not be possible in a non-data sharing DB2 subsystem.

Where different DB2 data sharing members are used for different MCOD components, some of the potential DB2 performance considerations are eased or in fact eliminated. The set of DB2 data each member acts on is limited through the separation of the database objects into distinct subsets due to the unique creator and databases created for each MCOD component. The impact of having multiple dynamic statement caches (DSC) in the separate members is also mitigated, as each set of statements varies according to the mySAP components in question. This reduces the impact of multiple prepares occurring in the different DB2 data sharing members.

5

Coupling facility design considerations

In this chapter we introduce high-level coupling facility design considerations:

- Coupling facility structure duplexing
- Coupling facility sizing

This subject is treated in depth in Part 3.

5.1 Coupling facility structure duplexing

This section discusses mapping the DB2 structures to the coupling facilities in a way that eliminates single points of failure and provides near-continuous availability. We assume that the DB2 data sharing group has at least two coupling facilities available. DB2 provides an efficient implementation of group buffer pool duplexing. This is known as DB2 managed duplexing. The group buffer pool duplexing is done asynchronously and facilitates fast restart in the event of a failure of a coupling facility containing group buffer pool structures. The impact on the cost of data sharing is limited to the path length needed to drive the duplexing operation. This asynchronous mode of operation is in contrast to a synchronous operation such as updating a group buffer pool page in the primary coupling facility. In the case of a synchronous operation the requesting processor engine waits for completion of the operation in what is known as *dwell*. Dwell time is significantly larger than the path length time to drive these operations.

The global LOCK structure is more problematic. This structure can be rebuilt in a reasonable amount of time by the IRLMs servicing this data sharing group. This leads us to place the global LOCK structure in a coupling facility that is external to the CECs that have members of the data sharing group. This is easily satisfied with a standalone coupling facility or an Internal Coupling Facility (ICF) on a CEC that has no active or standby members of this data sharing group. This avoid the double failure trap where a CEC failure takes down the LOCK structure ICF and one of the IRLMs needed to rebuild the LOCK structure. The SCA structure is typically in the same coupling facility as the global LOCK structure.

Coupling Facility Control Code level 12 (CFCC level 12) has delivered several significant capabilities including Systems Managed Structure Duplexing. This raises the question of whether we could avoid the placement considerations raised in the previous paragraph by using Systems Managed Structure Duplexing for the Global LOCK structure. This question is both an economic decision and a performance decision. Systems Managed Duplexing is a synchronous operation with synchronous writes to the primary coupling facility and the duplex coupling facility. In addition the two coupling facilities perform an acknowledgement handshake before responding the requesting CEC. All of this time is dwell time for that CEC's requesting engine. As SAP is a row-level locking application (unlike many legacy applications) we can expect to see lock request arrival rates an order of magnitude more frequently than with legacy applications. Initial estimates are that Systems Managed Global LOCK structure Duplexing will double the cost of data sharing.

Refer to Appendix B, "Systems managed duplexing" on page 197 for an example of system managed duplexing. The redbook *z/OS Version 1 Release 3 and 4 Implementation*, SG24-6581, has a section about system managed duplexing.

5.2 CF sizing

We consider MIPS and DB2 structures sizing.

5.2.1 MIPS

A general rule for CF sizing is that one MIPS of CF capacity should be provided for every 10 MIPS of data sharing application. This means that there should be one MIPS of CF capacity for every 10 MIPS consumed by the ICLIs (get charged for most of the DB2 time) and DB2 address spaces. As many coupling facilities have one or two engines (higher queuing costs) and as the CF operating system lacks the sophistication of the z/OS Workload Manager, we typically strive to run the coupling facilities at lower processor utilizations than the main CECs. Our objective during a sustained CF failover is to maintain utilization no higher than the 50% to 60% range. This dictates that during normal operation we keep utilization in the 25% to 30% range. It is further recommended that the coupling facilities doing productive work have dedicated engines assigned. Use the CF Sizer tool for accurate sizings.

5.2.2 DB2 structures

We discuss memory sizing guidelines for the three primary structures: SCA, global lock, and group buffer pools. It is important to remember that the physical coupling facilities must have enough memory to support the primary structures as well as the rebuild and duplexing operations. In practice the real memory required is twice the structure size when this is taken into account.

SCA and Lock

In most cases we recommend that the SCA be set up with an INITSIZE of 49152 (48 MB) and a SIZE of 65536 (64 MB). Non-production environments can be somewhat smaller. As this structure is used primarily for control and status information, exposure is not great.

The LOCK structure is of greater importance. Generally, a production environment is set up with a minimum of 512 MB. We have seen long-running, update-intensive batch jobs (misbehaved without frequent commits) require up to 1 GB for this structure. The size should be set and monitored in both the QA and production environments.

CFCC level 12 affects this storage in several ways. First, CFCC level 12 implements 64-bit real storage management, which eliminates the concern about being above or below the 2 GB line. Also, expanded addressing requires larger pointers, and we can expect the SCA to grow by 15% and the LOCK structure to grow by 20%. In older CFCC levels these structures live below the 2 GB line.

Group buffer pools

Sizing the group buffer pools involves multiple variables. The variables include:

- The data sharing option chosen (the number of primary and standby DB2 members)
- Whether a common buffer pool template has been used for all members (the usual case)
- The volatility of the individual buffer pools (affects both the percent of backing in the Group Buffer Pools and the number of directory entries)

The architecture option determines the number of local buffer pools to be backed by group buffer pool storage. A reasonable assumption is to count each active DB2 member as one and each standby DB2 member as one-half to account for failover. For example, an option 2 design with two primaries and two standbys would count as two plus one-half plus one-half, or three equivalent members to be backed. Most companies use a standard template for their local buffer pools, so we make that assumption. The final variable is the volatility of the individual buffer pools. A volatile pool is one where the update rate is less than 10% of the requests. These pools require a 40% backing in the group buffer pool per member times the equivalent number of members. Virtual pools with an update rate of less than 10% are considered read-intensive and should have 10% backing in the group buffer pool. Uncategorized pools should be given a backing of 30% (for example, the default buffer pools BP2, BP3). If you are using hiperpools, they receive no backing but do influence the number of directory entries. Typically the directory entries vary from three to 30 entries per virtual pool plus hiperpool page times the number of equivalent members. The update-intensive pools have three entries per page, read-intensive pools have 30 entries per page, and uncategorized pools have 10 entries per page. In pre-CFCC level 12 systems this storage is below the 2 GB line. Typically the aggregate directory storage is an additional 15% to 20% of the group buffer pool storage. CFCC level 12 increases the group buffer pool size by 10%.

Part 2

High availability considerations

Part 2 describes important sysplex and data sharing design issues that affect high availability.

We discuss single point of failure, important failover scenarios and outage avoidance, automation of high availability constructs, and backup and recovery considerations in the SAP data sharing environment.

6

Single point of failure avoidance

This chapter summarizes important considerations in avoiding single points of failure (SPOF) in an SAP R/3 systems design.

We cover:

- SPOF candidates
- Design principles to avoid SPOF

6.1 SPOF candidates

Eliminating single points of failure is key for providing high availabilty to any SAP configuration. SPOFs are hardware or software components in a configuration in which failure provokes the unavailability of the whole system environment. Potential SPOF candidates for an SAP system include the following areas:

- Database server
- Application server
- Disk subsystem

6.1.1 Database server

A discussion about SPOF elimination in the SAP three-tier client-server architecture must start with the database server, which has been the most serious availability exposure in SAP implementations. The SAP architecture options described in Chapter 2, "Architecture options and trade-offs" on page 11, combine DB2 data sharing features and the SAP Sysplex Failover function to effectively eliminate the database server as a single point of failure. Eliminating the database server as SPOF also includes the ability to survive the following failures:

- CEC
- LPAR
- DB2 member
- CF

The redbook *SAP R/3 on DB2 UDB for OS/390: Database Availability Considerations*, SG24-5690, uses architecture option 2 to illustrate SAP high availability implementation considerations, and it documents the testing of a number of failure scenarios.

Regarding the CF single point of failure avoidance, we address the design principles in Chapter 5, "Coupling facility design considerations" on page 39.

6.1.2 Application server

The application server also has a number of potential SPOFs such as:

- Enqueue server
- ICLI server
- Logon load balancing
- Update work processes
- Batch processes

SAP has historically had a single point of failure in its architecture: the *enqueue server*. Because SAP is platform independent and because the industry lacks a referential integrity standard, SAP has provided its own logical object locking mechanism called the enqueue server, which traditionally has been part of what is known as the SAP Central Instance. SAP has addressed the enqueue server SPOF issue in recent releases with the decoupling of the enqueue server and message server from the central instance and with the provision of an enqueue replication capability. When this solution is complemented by the System Automation Facility (or any equivalent automation product) the enqueue server is eliminated as a single point of failure. The redbook *SAP DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation*, SG24-6836, discusses in depth the implementation of the high availability enqueue instance.

The design guidance in 2.3, "ICLI design" on page 21, removes the ICLI server as a single point of failure by providing a primary and a standby ICLI server per application server for use by SAP Sysplex Failover.

Implementation of SAP logon load balancing with at least two application servers in each logon group (3.2, "Logon load balancing" on page 26) ensures that each group of users has at least two paths to the database server.

Section 3.3.1, "Update (VB) Dispatching" on page 28, describes how to set up SAP update dispatching in a way that ensures that in the event of an application server failure there will still be update work processes available to process each group of update requests.

Likewise, 3.3.2, "Batch classes" on page 32, describes the use of SAP batch server groups to provide high availability for classes of batch work.

6.1.3 Disk subsystem

Although disk subsystems have achieved very high levels of availability through internal redundancy and RAID technologies, some companies want complete protection against a disk subsystem failure. Peer-to-peer Remote Copy (PPRC) and Extended Remote Copy (XRC) are two technologies that can provide this protection. Chapter 9, "Backup and recovery architecture in data sharing" on page 75, examines the use of remote copy technology in a disaster recovery scenario.

Also, when setting up DF/SMS Storage Groups for dual logging, there should be sufficient granularity in the design to enable COPY1 and COPY2 to be placed on different disk subsystems.

6.2 Design principles to avoid SPOF

In the late 1970s and the 1980s, significant effort was applied to developing configurations that had no single points of failure. Frequently a methodology called Component Failure Impact Analysis (CFIA) was used to systematically analyze the relationship of infrastructure components to application elements. Over time, the lessons of CFIA were institutionalized in configuration design, best practices, and systems assurance reviews. As a result, it became routine for a system that was being designed for high availability to include redundancy in system elements, such as:

- CECs
- Channel paths
- Coupling facilities
- CF links
- DB2 dual logging
- DB2 dual archiving
- DB2 dual image copies with one for on-site tablespace recovery and one for off-site disaster recovery storage
- CF structures through duplexing or fast rebuild
- Application servers
- Network paths between application servers and the database server (for example, OSA-E adapters with VIPA)

7

SAP Sysplex failover

In this chapter we examine SAP failover triggers, important failover scenarios, and operational impact, and look at an example of planned outage avoidance with concurrent maintenance.

The chapter includes these sections:

- SAP Sysplex failover triggers
- Important failover scenarios and system impact
- Planned outage avoidance example

7.1 SAP Sysplex failover triggers

SAP Sysplex failover can be triggered by the following events:

- DB2 failure:
  - CLI immediately notifies SAP workprocesses
  - Failover initiated immediately
- Any other failure:
  - Packet acknowledgement timeout (Uses TCP/IP parameters)
  - TCPIP Keep Alive timeout (Uses TCP/IP parameters)
  - Failover initiated after timeout interval (See *SAP R/3 on DB2 for OS/390:Connectivity Guide* for timeout behavior details.)
- Impact on user:
  - User waiting for response will receive SQL 000 message and must re-enter the failing transaction.
  - No impact for users not waiting for response.
  - Excecuting batch jobs are cancelled and must be restarted.

7.2 Important failover scenarios and system impact

Table 7-1 summarizes several high-availability failover scenarios and the operational impact.

Table 7-1 Important failover scenarios and operational impact

High Availability Scenario	Effect
Central Processor Complex Failure	No Impact / SQL 0000
OS/390 Failure	No Impact / SQL 0000
DB2 or OS/390 Release Upgrade	No Impact / SQL 0000
DB2 Failure and Automatic Restart	No Impact / SQL 0000
ICLI Failure	No Impact / SQL 0000
ESCON® or LAN Failure	No Impact / SQL 0000
Coupling Facility Failure	No Impact
Coupling Facility Link Failure	No Impact
DB2 Online Reorg Utility	Slow Responses

High Availability Scenario	Effect
DB2 Online Full Image Copy Utility	No Impact
Central Instance Failure	No Impact / HA Enqueue

When reviewing this table refer to 7.1, “SAP Sysplex failover triggers” on page 50 for definitions of the impact. This subject has been thoroughly addressed in these Redbooks:

- *High Availability Considerations: SAP R/3 on DB2 for OS/390*, SG24-2003
- *SAP R/3 on DB2 for OS/390: Database Availability Considerations*, SG24-5690
- *SAP R/3 on DB2 for OS/390 and z/OS: High Availability Solution Using Systems Automation*, SG24-6836

7.3 Planned outage avoidance example

The following figures step through a typical concurrent maintenance (avoidance of a planned outage) scenario. Although the example is the application of maintenance to the DB2 members of a data sharing group, the same steps apply to many other systems activities including applying PTFs to z/OS, upgrading a CEC or z/OS, upgrading coupling facilities, applying microcode, and many other examples. Figure 7-1 shows the initial system state.

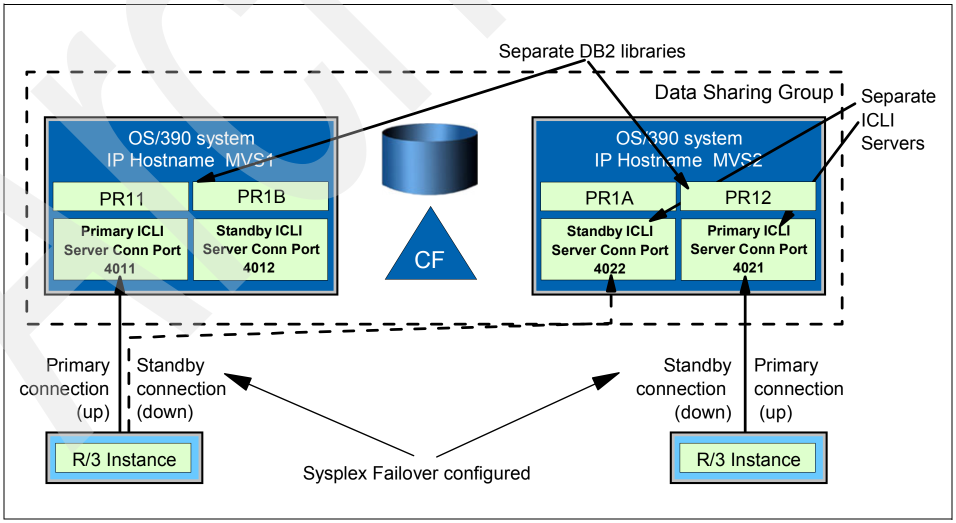


Figure 7-1 Applying service to all DB2 members in an R/3 system

Figure 7-2 starts the scenario.

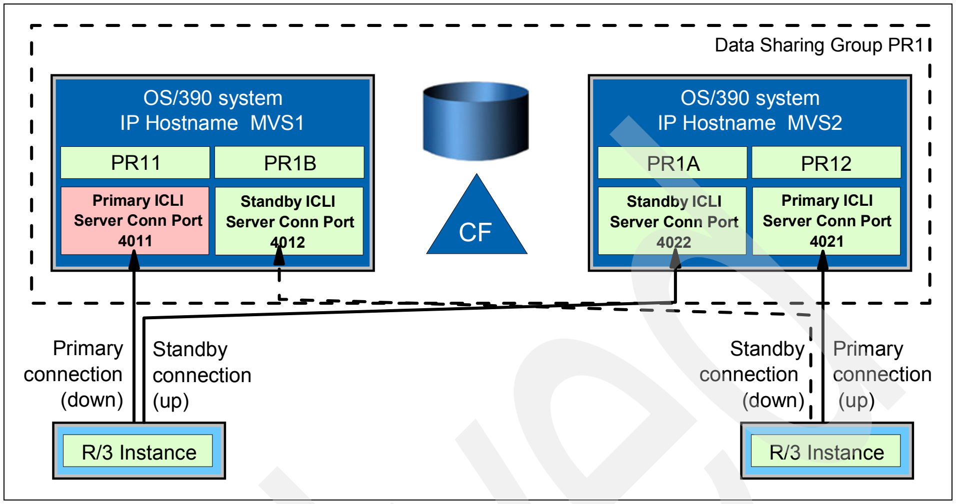


Figure 7-2 Step 1 - stopping the primary ICLI and reconnecting to the standby

Figure 7-2 shows the stopping of the primary ICLI server. Although this example shows the ICLI server being stopped, SAP Web Application Server V6.10 delivered new functionality to provide a graceful failover. Using the transaction or the command SMMS, the state of an instance can be changed (to *passive*). In this state, the application server can continue to process received requests or create requests itself. However, no new requests should be sent to the server. The purpose of this state is to logically remove from the system a server that has to be stopped during productive operation. The server finishes processing its existing requests but does not accept any new requests and can be terminated without having to interrupt any relevant requests such as updates or batch jobs. This state can be changed to the *active* state. This feature is documented in the SAP system documentation for SAP Web Application Server. This capability is complemented by enhancements to the SAP DB2 transaction in Web Application Server V6.xx. This is one company's description of this capability:

"In 6.2.0 Basis, you can cause a failover to a different server. Selecting **DB2 → DB CONNECTION LIST** lists all known DB SERVERS to this instance. Double-clicking the other DB server creates a pop-up that states “Switch Database connections.” Clicking **Yes** moves active threads of this system to another DB SERVER. (This all depends on whether you are using a new failover Connect.ini file in /sapmnt/SID/global or old standby entries in Instance profile. You can list many servers in connect.ini.) Understand that the inactive threads do not move until the values in these profile parameters expire (rdisp/noptime and rdisp/wp_auto_restart). Then the inactive ones will also move. Remember that when the thread takes a commit, it should then move if active. These two parameters can be dynamically changed in RZ11, and they take effect when saved, so you can set them to low values (60 seconds, 59 seconds) and failover and the inactive threads will also move. No need to stop ICLI servers or network connections.

In Figure 7-3 we stop the DB2 subsystem, PR11, and apply service.

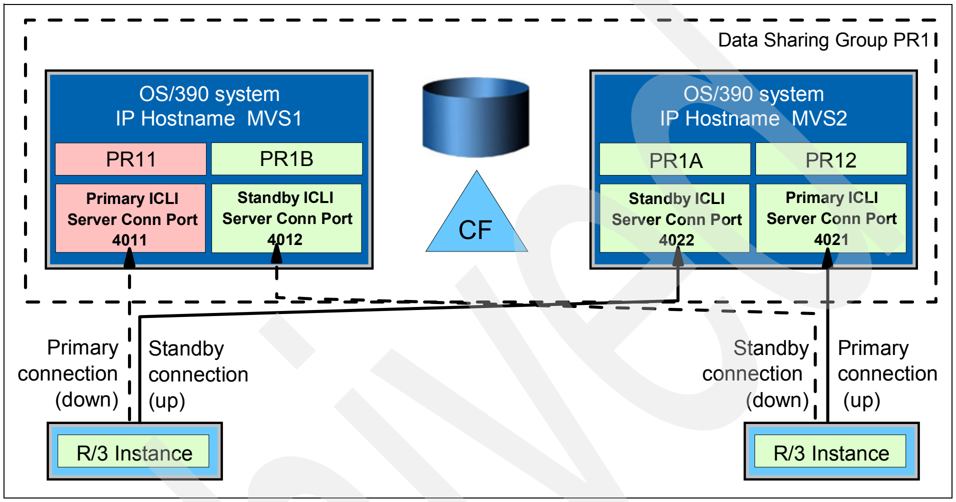


Figure 7-3 Stop the DB2 subsystem and apply service

The scenario continues in Figure 7-4 on page 54 with the restarting of the DB2 subsystem PR11 and the primary ICLI. At this point the application server is still connected to the standby DB2 member PR1A, and work continues with reduced system capacity.

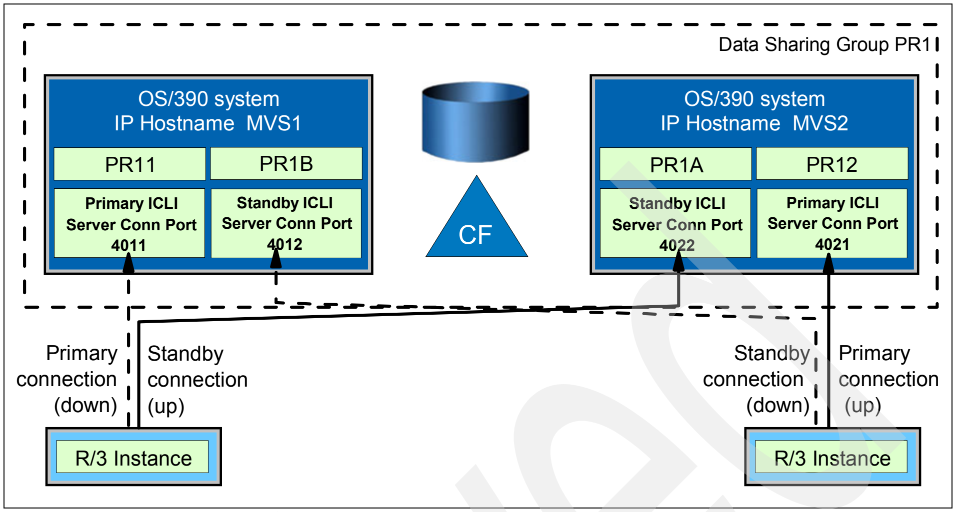


Figure 7-4 DB2 and ICLI restarted

In Figure 7-5, we stop the standby ICLI attached to PR1A, ongoing transactions roll back, and the connections reconnect to the primary ICLI and DB2 member PR11.

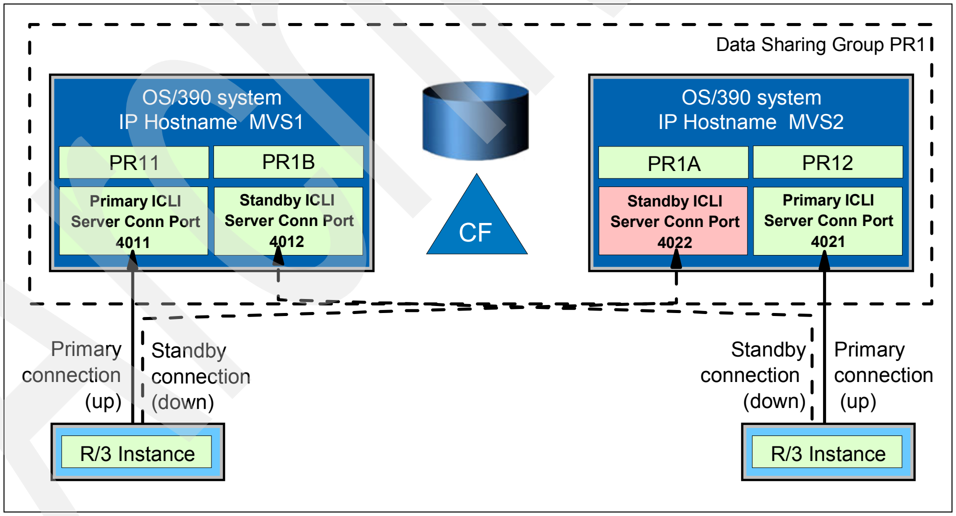


Figure 7-5 Stopping the standby ICLIs

We now stop the standby DB2, PR1A, and apply maintenance. In Figure 7-6 on page 55 we restart the standby ICLI and standby DB2, PR1A, restoring normal operation. The entire scenario can now be repeated for the DB2 members PR12 and PR1B.

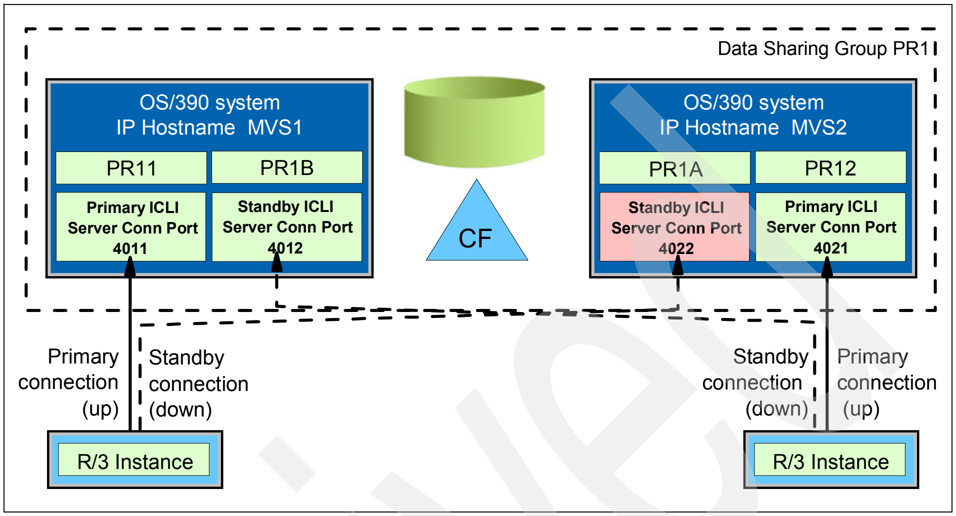


Figure 7-6 Restarting the standby ICLIs and DB2

8

Automation architectures for availability in data sharing

This chapter introduces automation and discusses procedures and processes that should be considered for automation. We discuss:

- Starting and stopping DB2 and SAP:
  - Automatically starting DB2 members
  - Automatically starting ICLI server and SAP
  - Automatic restart of DB2 (restart light to release retained locks)
  - Automatically stopping SAP
- Remote application server failover and failback (graceful failover via DB2 SAP transaction)
- Failover for local application servers running in Unix System Services (USS) in OS/390 or z/OS
- High Availability Enqueue Instance failover and failback
- Nondisruptive system-level copy
- Monitoring involving automation:
  - SAP health check
  - Archive log analysis (create list of objects that changed on this log and identify units-of-recovery that never ended)

8.1 What is automation?

High availability and continuous operation of SAP on OS/390 or z/OS involves implementation of Parallel Sysplex features and many automated procedures. DB2 takes advantage of the features provided in Parallel Sysplex through its data sharing implementation. Other products taking advantage of Parallel Sysplex are TCP/IP through its virtual IP address implementation and USS through its shared HFS implementation. When these high-availability constructs have been implemented, it is necessary to have automation in place to manage their use in support of high availability. Therefore, the high availability and continuous operation infrastructure is a combination of the Parallel Sysplex constructs and the automated procedures for managing them.

Some of the automation is inherent in the implementation of high availability features, but others require the customization of automation tools.The procedures to automate vary from obvious operational tasks such as starting and stopping the DB2 data sharing members, the ICLI servers, and SAP to providing for cycle time reduction when performing a recovery. There are also certain conditions you may want to monitor through automation or an online monitor.

Automation is provided for or supported by various products. The automatic restart manager (ARM) component of OS/390 or z/OS is used to control system actions through the definition of policies. SAP provides a number of application profile settings that are used to implement failover and failback. The IBM product System Automation for OS/390 is used to customize automation for operational procedures and processes.

8.2 Starting and stopping DB2 and SAP

A number of started tasks and processes that must be started for the SAP application to be available for use. First, DB2 must be started. Typically, a DB2 member includes at least five started tasks: MSTR, IRLM, DBM1, DIST, and SPAS. Then the ICLI servers are started, and finally the SAP central instance and application servers. In order to reduce cycle time, automation can be used to manage the start of each component in the appropriate order.

To remove the LPAR or CEC as a single point of failure, it is also important to make sure the different members of the DB2 data sharing group are started on different LPARs. Many installations have automation in place to start the DB2 members on assigned LPARs at their designated time.

When DB2 MSTR is started, the other four started tasks that belong to the DB2 member can be triggered through parameter settings in ZPARM. For IRLM these parameters are IRLMAUT and IRLMPRC. For the DIST started task the parameter is DDF, and for the SPAS started task the parameter is STORPROC. Refer to the appropriate *DB2 Universal Database for OS/390 Installation Guide* for more information about using ZPARM parameters.

When the DB2 member starts normally, a DSN9022I message is written:

```
DSN9022I +ssid DSNYASCP 'START DB2' NORMAL COMPLETION
```

This message can be used as a trigger, through automation, to start the ICLI servers. When the ICLI servers have started normally, automation can be customized to submit a remote execute to the application host to start SAP. The SAP application host can then send messages back to the automation process about whether SAP actually started.

In the Parallel Sysplex configuration there is an option to have separate OS/390 or z/OS system libraries. Many installations choose to do this in order to eliminate a single point of failure and to provide for granularity when migrating to new levels of system software. In order to resolve retained locks by restarting a failed DB2 member, it is important to have all of the DB2 members and their IRLM tasks registered in the subsystem name member of SYS1.PARMLIB (IEFSSNxx) of each LPAR.

Automation should be used to recognize when a DB2 member is no longer available. The reasons can vary, and they include:

- DB2 member failure.
- LPAR failure.
- The LPAR is no longer responsive within the sysplex.

In the case of the abending DB2 member, the surviving LPARs analyze the situation and determine whether the LPAR has also failed. If the LPAR has not failed, the DB2 member should be restarted on that LPAR. If the ICLI servers have not stopped or abended, automation should stop the ICLI servers prior to restarting the failed DB2 member. If the DB2 member starts normally, the ICLI servers will start via automation.

If the LPAR fails, it is important to have the DB2 data sharing member restarted on another LPAR in order to resolve retained locks. In this situation the surviving LPARs would recognize that the LPAR has failed, and the automation policy could be written to direct the restart of the failed DB2 member on a surviving LPAR, using the restart light feature in DB2. In this case, the ICLI servers should not be restarted, because the failing DB2 member is up for just long enough to resolve the retained locks. For more information about using ARM and the DB2 restart light feature, refer to *DB2 Universal Database for OS/390 and z/OS Data Sharing: Planning and Administration* for DB2 Version 7.

It is harder to determine when an LPAR is no longer responsive within the sysplex. Because of this, it has the potential to cause longer application outages due to resource unavailable conditions than a failing LPAR. There is a process in Parallel Sysplex provided by XCF that queries the LPARs to ensure that they are still responsive. If an LPAR is not responsive, it is removed from the sysplex. At this point, automation must recognize that the DB2 members are no longer participating in the sysplex and must be restarted. Again, the restart light feature of DB2 should be used.

When stopping a DB2 member normally, automation should be customized to recognize that the ICLI servers have a dependency on the DB2 member. Automation could then be used to either stop the ICLI servers or alert the operator of the dependency. As in the automatic start, it is possible to have a remote exec sent to the application server host to stop SAP. However, if stopping the ICLI servers is the manner used to force remote application failover, then an exception to this automation process must exist.

8.3 Remote application server failover and failback

Remote application server failover and failback is one of the major infrastructure processes in providing high availability and continuous operation. This provides for high availability during failure of a CEC, LPAR, operating system, DB2 member, or ICLI server. It also provides for continuous operation during maintenance on any of these components. This is accomplished by failing over ICLI threads that are connected to affected DB2 members to other DB2 members. Planned failover can be accomplished by stopping the ICLI servers without automation sending the remote exec to the SAP application host to stop SAP. When using planned failover by stopping the ICLI servers, it is important to remember that any uncommitted work will fail and rollback. So, when executing a planned failover, all SAP batch should be allowed to complete first.

A large part of the automation supporting remote application server failover and failback is controlled by SAP application server profile settings. Prior to SAP 6.1.0, only a single standby host was supported through definitions in the instance profile and environment files. With SAP 6.1.0 and beyond, multiple standby definitions are defined in the CONNECT.INI file located in the global directory.

In SAP 6.1.0 and beyond, failover changed considerably. Failover is now nondisruptive and is managed through an SAP transaction. Failover is initiated by using the SAP transaction DB2. When in the DB2 transaction, selecting DB2 CONNECTION LIST will provide a list of known database servers for a specific instance. After selecting the database server to which the ICLI threads should reconnect, failover can be initiated. The threads are not moved until a commit point has been reached. If an SAP batch process is taking frequent commits, the failover is transparent. After a commit point is reached, the process moves over to the new database server and the SAP batch continues to run. Refer to SAP OSS Note 402078, *DB2/390: New failover support from Release 6.10* for more information.

More information can also be found about establishing remote application server failover in the *SAP R/3 on DB2 for OS/390* (or z/OS) *Planning Guide* for the specific SAP version.

8.4 Local application server failover and failback

Local application servers run in the UNIX System Services (USS) component of OS/390 or z/OS. With SAP 6.x, the connection is now an RRSAF connection. There is no interim process or started task, such as ICLI, used to facilitate the connection to DB2. The failover solution for local application servers only removes the DB2 member as a single point of failure. If the CEC, LPAR, or operating system fail, then there will be a disruption to the work running in the local application servers.

As is the case with most DB2 data sharing groups, each DB2 member for the database server is running on a separate LPAR. The application servers running in USS are running on the same LPAR as the DB2 data sharing member to which they are connected. So if you have USS application servers connected to each member, then you have USS application servers running on each LPAR.

In order to facilitate connection to another member of the DB2 data sharing group, there must be another DB2 subsystem running on each LPAR where USS application servers are running. If this DB2 subsystem is not a member of the database server DB2 data sharing group, then this system is used as a gateway. When connected to the gateway DB2 system, the SQL request from the USS application servers will be treated as a remote request to another member of the database server running on a different LPAR. DRDA protocol is used and the communications database will have to be set up in support of this connection. It will also be necessary to bind the SAP packages on the remote DB2 subsystem.

Failover of the USS application servers requires the connect.ini file. This file must be stored in the SAP global directory, and each instance of this system needs access to the directory. As an example:

```
connecti.ini

[ihsapfc]

CON1=SGE2_on_ihsapfc
```

```
[SGE2_on_ihsapfc]

SSID=SGE2
HOST=ihsapfc
PLAN=OBCM620

[D7A0_on_ihsapfc]

SSID=SGE2
HOST=ihsapfc
PLAN=OBCM620
LOCATION=D7A0
```

If the connection to SGE2_on_ihsapfc is broken, then the secondary connection is used: D7A0_on ihsapfc.

Controlled failover and failback is managed through the DB2 SAP transaction.

For more information about implementation, refer to the appropriate *SAP on DB2 UDB for OS/390 and z/OS Planning Guide* or the *SAP High Availability Guide*.

8.5 High availability enqueue failover and failback

Availability of the SAP central instance is critical in order to keep an SAP system running. If any component of the central instance fails it would cause a disruption to the SAP system.

In order to provide high availability, the SAP central instance has been broken into standalone server components: enqueue server, message server, gateway, and syslog collector, and are grouped together as an *enqueue instance*. The independence of the components enables more efficient recovery for each component and provides better performance of the enqueue server. These components share the same instance directory and instance profile. However, they can be started, stopped, and recovered independent of each other. None of these components requires access to the DB2 database. This new high availability infrastructure component may be referred to as High Availability Central Instance, High Availability Enqueue Instance, or SAP Central Services. It is available for SAP 46D and higher.

The *virtual IP address* (VIPA) feature of TCP/IP is used in this solution to provide for IP takeover for the enqueue server components. In order to keep the setup of TCP/IP and SAP profiles small, all of the components use the same VIPA. The application server instances connect directly to the enqueue server via the VIPA.

To provide for high availability and continuous operations, along with transparent failover of the enqueue server, there is a new component called the *enqueue replication server*. This is also a stand-alone component, and it connects to the enqueue server. When connected, the enqueue server sends replication data to the replication server. The replication data is stored in a shadow enqueue table that is maintained in shared memory, and it is used to rebuild the tables and data structures for the enqueue server during failover.

The NFS server is also provided failover through the use of VIPA. The number of NFS servers required to provide high availability to the SAP landscape is an installation configuration decision. There may be a requirement or desire to isolate production from test and development. Consideration should be given to planned failovers in support of performing maintenance activities.

Automation must be customized to start and stop all of the enqueue instance components, the enqueue replication server, and any NFS servers.

The replication server runs on a separate system from the enqueue instance. So with the new enqueue instance plus an enqueue replication server, SAP central services is being removed as a single point of failure, and the CEC, LPAR, and operating system are also removed as SPOF. If the CEC, LPAR, operating system, or enqueue server fails, the system where the replication server is running will take over.

- The VIPA is taken over.

  The automation process that accomplishes this is a feature of TCP/IP. The automatic takeover of the IP address by another system is established when this IP address has been defined as a VIPA.

- The components of the enqueue instance are restarted.

  The most important components are the enqueue server and the message server. The automation for the component restarts must be created using automation tools. The automation for this solution is goal-driven in that the behavior of the systems and application components are defined. Dependencies between the components are understood, as are which conditions trigger what actions.

  Therefore, the automation must be able to recognize when the enqueue instance needs to be restarted. This would include failures of the CEC, LPAR, operating system, or enqueue server.

  – After it has been determined that the restart of the enqueue instance is required, the automation process must restart the enqueue server on the LPAR where the enqueue replicator server is running.

  – The new enqueue server will communicate to the enqueue replicator server and determine that this is an enqueue server failover, and it will rebuild the enqueue tables and data structures from the replicator server's shadow tables.

  - Automation also must be in place to restart the message server and, if applicable, the gateway and syslog collector tasks.

- The application servers reconnect to the enqueue server and the message server.

  The enqueue server failover is transparent to the application. The enqueue locks are preserved and transactions continue to run.

- Through automation definitions, the enqueue replicator server should be stopped on this LPAR and restarted on a different LPAR.

This high-availability SAP central services infrastructure can also be used to provide continuous operations during maintenance activities. Maintenance on the CEC, LPAR, or operating system would require any enqueue instances running there to be failed over. This would require a procedure to initiate an enqueue instance failover.

SAP Central Services failover is different from application server failover. Both are required in order to provide high availability and continuous operations. The application server failover is used to ensure that the servers always have a connection to the DB2 database, and the enqueue instance failover ensures that the enqueue server and its enqueue table are always available. The automation for the enqueue instance also ensures the restart of the message server and, if applicable, the gateway and syslog collector.

For more information, refer to *SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation*, SG24-6836.

8.6 Nondisruptive system-level copy

Nondisruptive system-level copy is another high-availability infrastructure process. To obtain a system-level copy without stopping or quiesceing the SAP application takes planning, automation, and the ability to perform hardware-assisted copies. The hardware-assisted copies are also referred to as volume-level copies, but a hardware-assisted copy is more than a volume-level copy. It is an extremely quick way of creating a logical copy of the source volume. The IBM feature ESS FlashCopy® is a hardware-assisted copy product.

Even though the SAP application is not going to be stopped or quiesced, it is still necessary to provide the ability to ensure consistent data when this system-level copy is used for recovery or as the source in a homogeneous copy. The best way to do that is to suspend update activity. The SET LOG SUSPEND command in

DB2 acquires the log-write latch, and no update activity takes place until DB2 can continue to write to the logs output buffer. Additionally, when this command is issued, the log buffers are externalized to DASD, and the suspend RBA is recorded in the BSDS as the HIGHEST WRITTEN RBA. While the update activity is suspended, the hardware-assisted copies of the DB2 system and all SAP DB2 objects can be obtained. After the copies are complete, each member's update activity must be resumed with the SET LOG RESUME command.

This seems simple, but concerns regarding data sharing, ensuring a complete system-level copy, and recovery must be addressed:

- The SET LOG SUSPEND/RESUME is not a group-level command. This means that the command must be issued to each member of the group.
- This command only suspends use of the logs output buffer, not the data buffers. To avoid data being written out to DASD during the copy process, it is necessary to force a system checkpoint prior to the suspend. To force a DB2 member in the data sharing group to perform checkpoint processing, the command SET LOG LOGLOAD(0) must be issued. This is especially important for pagesets with 32K pages because it takes multiple physical IOs to externalize their pages. This lowers the probability of a 32K page write only being partially complete during the volume-level copy. Again, this is not a group-level command.
- Most installations experience frequent changes in the number of DASD devices used in an SAP environment. It is important that all devices are included in the copy process.
- Typically, hardware-assisted copies are obtained not only to speed up the copy process, but to help reduce cycle time during recovery. If certain conditions exist in the DB2 group when the copy is taken, other recovery steps may be required when using the copy in a recovery process. To avoid additional recovery actions when using the system-level copy in a recovery:
  - Stop any OS/390 or z/OS utility batch jobs for this DB2 data sharing group from running during the hardware-assisted copy job.
  - Ensure that no utilities are running at the time of the copy. This can be accomplished by using the DISPLAY UTIL(*) command.
  - Ensure that no pagesets are in a restricted status by using the DISPLAY DB(*) SPACE(*) LIMIT(*) RESTRICT command.

In order for all of the status checks to be completed and the appropriate commands to be issued to all members of the DB2 data sharing group, then performing the copy quickly and resuming the update activity to each member, automation is required. The automation should be included in the job that is performing the copy.

8.6.1 Nondisruptive system-level copy job flow

Step 1

A process should be designed using scripts or REXX execs to rebuild the hardware-assisted copy control cards at the beginning of the job. The input to this process is the updated list of DASD devices for this SAP system. This is to ensure that all required volumes are included in the copy process. If DASD packs have been added or deleted since the last time the control cards were created, this process should refresh the control cards.

Step 2

If using FlashCopy, create a REXX exec to ensure that none of the volumes being copied are currently in a FlashCopy relationship. This is to ensure the use of fast replication for the copy.

Step 3

When designing the automation for the following actions, make sure the automation process is writing out the responses to the commands to a log or data set that can be used for problem identification.

- Identify which members are active in order to determine to which members to issue commands”

  ```
  DISPLAY GROUP
  ```

- If you have determined that you do not want to allow utilities to be running while you are obtaining the copy, stop all stored procedures:

  ```
  STOP PROC(*.*) ACTION(QUEUE)
  ```

  This is not a group command, and it must be issued to each active member of the group. ACTION(QUEUE) causes any process that is trying to use the stored procedures to wait instead of being terminated. Think about increasing the ZPARM parameter STORTIME for each member to a long enough time to allow queued processes to wait for the copy to complete and the stored procedures to be started. For purposes of documenting the success of this command, include a DISPLAY PROC(*) command.

- Again, if you have determined you do not want utilities to run while you obtain the copy, ensure that no utilities are registered in SYSUTILX:

  ```
  DISPLAY UTIL(*)
  ```

  If there is a utility registered, have the automation process pass a return code or stop the job in a manner that will be noticed.

- Ensure that no pagesets are in a restricted status:

  ```
  DISPLAY DB(*) SPACE(*) LIMIT(*) RESTRICT
  ```

The automation process can be designed not to be concerned with pagesets in copy pending. Ensuring that pagesets are not in a restricted status avoids additional recovery actions when using this hardware-assisted copy for recovery or in a homogeneous copy. If there is a pageset in a restricted status other than copy pending, pass a return code or stop the job in a manner that will be noticed.

– Force a DB2 system checkpoint to each active member of the group:

```
SET LOG LOGLOAD(0)
```

This is not a group command and must be issued to each active member. Be sure that the automation process checks for successful execution of the command.

– Suspend the update activity for each active member of the group:

```
SET LOG SUSPEND
```

This is not a group command and must be issued to each active member. Be sure that the automation process checks for successful execution of the command.

Step 4

Run the copy step by executing the program or procedure provided by the hardware vendor to perform hardware-assisted copies. This step uses the control cards created in the first step of this job. It should run extremely fast. Depending on how many devices are included, it should not take more than a few minutes.

If this step does not finish normally or passes an unacceptable return code, go directly to the step that resumes the logs and starts the stored procedures.

Step 5 (optional)

Run DSNJU004 to print all of the members' BSDSs.

Be sure that the output is written either to the joblog or to a data set that can be referenced at a later date. This can be useful in performing a homogeneous copy.

Step 6

This is another automated step. In this step the logs for all active members will be resumed and the stored procedures will be started. To ensure that this step will always be executed, code the COND=EVEN parameter on the EXEC card. Again, be sure that the automation process is writing the responses to the commands to a log or data set that can be used for problem identification.

– Identify which members are active in order to determine to which members to issue commands:

```
DISPLAY GROUP
```

– Resume the update activity for each active member of the group:

```
SET LOG RESUME
```

This is not a group command, and it must be issued to each active member. Be sure the automation process is checking for successful execution of the command.

– If you stopped the stored procedures, start them with:

```
STA PROC(*.*)
```

This is not a group command, and it must be issued to each active member. Make sure that the automation process is checking for successful execution of the command.

If you have determined that you do not want DB2 utility jobs for this data sharing group to run while you obtain the copy, then you need to have a way to hold or cause these jobs to wait until the hardware-assisted copy has completed. Remember to release the DB2 utility jobs or enable them to run when this job has completed.

This system-level copy process provides a nondisruptive copy that can be used in system-level local recovery to the suspend point or to a point in time. The target packs of the copy can be dumped to tape and sent off-site for disaster recovery. This hardware-assisted copy can also be used as a source in a homogeneous copy.

See 9.1.6, “Volume-based backup: online” on page 83 to read about using this type of copy in a point-in-time or disaster recovery.

Refer to *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287 for more about using this system-level copy as source in a homogeneous copy.

8.7 Automation for monitoring

SAP application high availability and continuous operation can also be viewed as reduced cycle time on outages or application-unavailable situations. The quicker the recovery of the system or resolution of the resource unavailable condition, the more time the SAP application is available for use.

A number of processes can be automated in order to facilitate a quicker recovery, or recognition of a resource-unavailable condition. Also, monitoring critical DB2 resource components for conditions that could cause DB2 to fail or impact its ability to service the workload should be considered.

8.7.1 Archive log process

Typically, performing the recovery of an SAP system means recovering the whole system. This would include the DB2 directory and catalog and all of the SAP DB2 objects that have changed. The reason for only recovering the objects that have changed is to reduce cycle time on the recovery and make the SAP application available for use. This can be accomplished by interrogating the contents of each archive log for log records that indicate change to pagesets. The time to interrogate the archive logs is prior to any required recovery. If this is not done ahead of time, the recovery cycle time is extended. This must be done throughout the day and be ready for use during recovery.

Automation can be used to provide the interrogation process each time a DB2 member completes archiving an active log. The output of this process must be placed in a data set that can be used as input to REXX execs used to create recovery jobs. Upon completion of an archive task, DB2 writes out a message indicating that the offload is complete:

```
DSNJ139I +ssid LOG OFFLOAD TASK ENDED
```

Automation can be triggered by this message to execute a process that creates a list of objects that have log records on that log that indicate that the pagesets have been changed.

Step 1

Use the DSN1LOGP program to create a summary of log records. The input is the archive log. The control card is SUMMARY(ONLY). The output has to go to a sequential data set that can be used as input to the next step.

Step 2

This step is a compiled REXX exec. The input is the sequential data set that was created in step 1. The REXX exec interrogates the DSN1LOGP output for messages indicating change to pagesets. These messages are DSN1151I, DSN1152I, DSN1160I, and DSN1162I. The most recent occurrence of a change log record for a given pageset is written to the output file. The output for each record written should contain such information as date/time stamp, start and end RBA, start and end LRSN, and transaction status. The output is another sequential data set.

Step 3

This step is a REXX exec. The input is the sequential data set that was created in step 2. The output of this step is a new member in a partitioned data set (PDS). There is also an INDEX member that is updated with information about the new member being added to the PDS. The member name is the sequence number of the archive log (Annnnnnn). In this manner, it is easier to determine which members should be used as input to the REXX exec creating recovery jobs during the recovery process. This REXX exec is also used to identify which members can be deleted. Most installations have established a length of time for keeping archive logs. By reviewing the information in the INDEX member, the members that have expired can be identified and deleted.

Step 4

This step uses a utility such as IEBCOPY to compress the PDS in order to reclaim space.

It is important to remember when using this PDS as input to an image copy recovery process that the current active log should be archived and allowed to go through this process. If a PDS member for the current active log does not exist in the PDS, then pagesets that have changes recorded on that log may not be included in the list of objects requiring recovery.

Neverending unit of recovery

There have been incidents of an ICLI thread still existing in DB2 that is no longer connected to an SAP application server process. On occasion, one of these threads has a unit of recovery (UR) that will require a backout when the thread terminates. It does not show up as a long-running (UR), because it is not updating. However, after the thread is terminated, it will need to back out. If this thread has been out there for an extended period of time, it may require an archive log that no longer exists in order to complete the backout.

A process can be created to interrogate the most current member in the archive log PDS created in step 3. This process should look for URs that are not committed, that do not have an ENDRBA or ENDLRSN, and that have a date/time stamp that is greater than a specified time frame, such as 24 hours. If this situation exists, the appropriate people should be notified. In some cases, this can be resolved with a planned failover of the application server. In extreme situations, the DB2 member may have to be forced down. CANCEL THREAD will not work because the application server process is no longer connected, and there is no activity in the thread.

8.7.2 Resource unavailable

In a data sharing environment, resource unavailable is probably one of the most common perceived application outage situations. Even though the DB2 member is available, if a highly required object is not available, it looks like an application outage to the end user. A good example of this is when SYSLGRNX is not available for an extended period of time.

"Starting and stopping DB2 and SAP" on page 58 has a discussion about using ARM and the restart light feature in DB2 for resolving retained locks. One of the situations described is when an LPAR is unresponsive. The intent of the automation is to recognize when there are DB2 members no longer participating in the Parallel Sysplex and restart them in order to resolve their retain locks. Best case is when this all happens prior to another member actually failing a user process because of resource unavailable. However, if for some reason this does not happen, automation can be customized to trigger the DB2 restarts when the resource unavailable message is written.

The DB2 member that is experiencing the resource unavailable condition will write out DSNT501I messages with REASON 00C900BE. If this data sharing group has more than two members, the DISPLAY GROUP command can be used to determine which members have failed.

8.7.3 Long-running units of recovery

The impact of a long-running unit of recovery (UR) is really felt during DB2 system restart. If a UR did not take frequent commits and the DB2 member failed, the backout of that UR could extend the period of time for restart. Also, it would extend the period of time resources are unavailable to other members. The use of postpone recovery or no backout during restart is not recommended in the DB2 systems where SAP is running. Therefore, the solution must be preventive rather than reactive.

Automation should be used to capture long-running UR conditions and report them to the appropriate people. This is controlled by the ZPARM parameter URCHKTH. The value for this parameter indicates how frequently a UR is expected to commit in terms of DB2 system checkpoints. If the DB2 system finds that a UR has not committed within this number of checkpoints, a message (DSNR035I) is written to the log.

Prior to DB2 Version 7, it was hard to use this message to trigger any type of action because the DB2 system checkpoint was dictated by LOGLOAD. Depending on the fluctuation of workload in a given DB2 system, checkpoints could be taken every few seconds to every few hours. With DB2 V7, checkpoint frequency is stated in terms of minutes by using the CHKFREQ PARM parameter. The recommendation is to use 10 minutes. Now, an installation can determine a reasonable length of time for a UR to run without committing. Identification of long-running URs can be triggered by using the URCHKTH and CHKFREQ parameters settings, and automation can be customized to either take action or notify the appropriate people.

8.7.4 SAP health check

Automation or monitoring can be used to indicate when there is an exception to the established SAP environment. Most installations have defined the norm for their SAP environments and, therefore, what is considered an exception. If it is important to know when these exceptions have occurred, automation could alert operations or the appropriate people.

- If the ICLI threads from a particular application server host are expected to always connect to a particular DB2 member, this can be monitored through the interrogation of the output from a DISPLAY THREAD command.
- The status of the DB2 members can be monitored by interrogating the output of the DISPLAY GROUP command. The LPARs, on which each member is running, are also displayed in the output of this command.
- Some DB2 members may only be used in support of application server failover. It is important to know whether the application server host is able to communicate, via the TCP/IP connection, to the ICLI server that is used to provide the thread connection into this DB2 member. Automation can be used to ping the IP address of the ICLI server from the application server host periodically throughout the day.
- Automation can be used to issue R3trans to the database server to ensure that the application server has a database connection. This could also be done periodically throughout the day.
- Automation can be used to provide a warning if the archive process is not keeping up with the full active logs. The output from the DISPLAY LOG can be interrogated for percentage of full logs needing to be archived, and as to whether the offload task is available:

  ```
  FULL LOGS TO OFFLOAD = 20 OF 24
  ```

  This would indicate that there is less than 20% of available active log. Automation could alert operations to take whatever actions the installation has defined, such as checking tape drives, notifying DB2 systems support, and so on.

  ```
  OFFLOAD TASK IS (AVAILABLE)
  ```

  If the offload task is available, and *full logs to offload* is not equal to zero, then there is a problem. Automation should notify the appropriate people.

- Virtual storage in DBM1 is a problem for a number of installations. DB2 V7 has a new IFCID 225 record. This IFCID maps out the users of DBM1's virtual storage and reports how much each is using. By turning on statistics class 6, this is recorded at whatever interval the installation has established for statistic records to be created. This can be activated through the ZPARM parameter SMFSTAT.

  Tracking the growth in virtual storage use in DBM1 can be accomplished by using standard tools or monitors that use SMF records as input.

  For information about virtual storage tuning, refer to the *SAP6.2.0 Database Administration Guide, Virtual Storage Considerations*.

9

Backup and recovery architecture in data sharing

In this section we discuss the backup and recovery issues a DB2 installation must consider when moving from a non-data sharing to a data sharing environment. We focus on how our usual SAP backup recovery procedures can be affected when we start working in a data sharing environment and the adjustments we must make on these procedures. We also discuss Disaster Recovery and Homogeneous System Copy in an SAP environment.

This chapter includes these sections:

- Data sharing backup-recovery considerations
- Disaster Recovery considerations
- Homogeneous System Copy considerations

9.1 Data sharing backup-recovery considerations

In this section we start with a brief description of the new recovery environment introduced by DB2 data sharing. We cover how DB2 data sharing recovers data, we analyze the modifications to the current models of backup and recovery procedures that apply to SAP databases when moving from non-data sharing to data sharing, and we describe the new possibilities of DB2 Version 8 that address the SAP recovery requirements in a data sharing environment.

DB2 data sharing introduces new features that enable database recovery from failures across multiple DB2 data sharing members:

- Log record sequence number (LRSN)
- Logical page list (LPL), which is also valid for a non-data sharing environment
- Group buffer pool recovery pending (GRECP)
- SCA structure
- LOCK structure
- Damage assessment processing (DAP)

These features are introduced to cover data sharing requirements. DB2 uses all of these features at recovery time, and they have changed the way DB2 performs recovery from different failures in a data sharing environment.

9.1.1 Data sharing recovery environment

In a data sharing environment, the member subsystems maintain separate recovery logs. Each manages its own active and archive log data sets and records those in its own bootstrap data set (BSDS). The shared communications area (SCA) in the coupling facility contains information about all members' BSDSs and log data sets. In addition, every member's BSDS also contains information about other members' BSDS and log data sets in case the SCA is not available.

In accordance with other operational procedures, the changes introduced by DB2 data sharing to database recovery are mostly internal to DB2, and they have little impact on the existing tablespace recovery procedures. However, the scope of the recovery process is at a data sharing group level, and updates made by all members must now be considered. Consequently, DB2 has to process logs from all members, and it must be able to sequence updates to a single page across all members in the DB2 data sharing group. The recovery process can be performed on any member.

Therefore, it is clear that relative byte addresses (RBAs) cannot be used to sequence the log records from multiple members of the data sharing group. Each DB2 member has its own log. The RBA of a member has no relationship with the

RBAs of other members. The rate at which log RBAs are advanced is related to the intensity of updates occurring at the individual member. For example, some members in a data sharing group can only be used for special purpose, such as parallel batch runs or query support. Their log RBAs are likely to lag behind the RBAs of other members, where updates occur all the time.

The LRSN is used to sequence log events such as pageset updates from different members. RBAs are still used within a single member. The LRSN is based on the time of day, which is obtained using the STORE CLOCK instruction. It is a 6-byte value that is equal to or greater than the time-of-day time stamp value truncated to 6 bytes. When a page is updated in a data sharing environment, the LRSN is stored in the page header. Because members generally run on different machines, they must have synchronized time and therefore consistent LRSN values. This function is provided by the sysplex timer.

It is important to keep in mind that a unit of recovery (UR) can execute on a single member. All log records related to one UR are written to the log of the member where the UR executed. A single UR cannot write some of its updates to one member's log and other updates to another member's log, although different URs, executing on different members, can concurrently update the same pageset or partition.

In a data sharing environment, the DB2 catalog and directory are shared among all members. This means that every new DDL or bind originated by one member is immediately visible for the rest of the members. Any tablespace or partition can be recovered from any member. However, the RECOVER utility executes on a single member.

As in a non-data-sharing environment, key information for recovery is stored in the data sharing group catalog and directory, in the bootstrap data sets, and on the log. The DB2 catalog and directory are expanded to track member-specific information and to achieve log synchronization. In a data sharing group each member maintains its own active and archive logs. A member must have read access to other members' BSDSs and logs, but they cannot write to them.

An important aspect to consider when enabling data sharing in a DB2 environment is the media used to store archive logs. Recovery processes that require archive logs from multiple members must allocate these archive data sets simultaneously. This means, in case those data sets are stored on tape, that we need the same number of tape units available to perform the recovery. This can be a source of problems in recovery situations.

Note: For data sharing it is recommended, not only for availability but also for performance reasons, to avoid using tape archive logs for data recovery.

For more about the DB2 data sharing recovery environment see *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935.

Another good reference for understanding the basic principles of recovery in data sharing is Chapter 2, “Database Recovery Environment,” in *DB2 on the MVS Platform: Data Sharing Recovery,* SG24-2218.

Next, we discuss two kinds of data recovery situations that can happen with data sharing:

- Tablespace recovery
- Recovering pages in the logical page list (LPL)

9.1.2 Tablespace recovery

The procedures for data recovery are fundamentally the same for data sharing as for non-data-sharing. Data sharing involves one catalog, but there are now many logs and BSDSs. In addition to disk and cache controllers, a new medium is introduced: the coupling facility. This adds a possible point of failure and requires appropriate recovery procedures. In planning for data sharing, it is important to consider having more than one coupling facility. Should a structure failure occur, recovery for the SCA and LOCK structure can proceed automatically if a second coupling facility is available.

As in non data sharing, full image copy is used as the base for the tablespace recovery. When a member performs recovery of an object, it will review all SYSLGRNX entries from all members for the object being recovered. Working in data sharing mode, SYSLGRNX contains starting and ending LRSN values in addition to RBA values for each member and the member ID to which each range belongs.

DB2 can access the logs from other DB2 systems in the group and merge them in sequence. The log record sequence number (LRSN) uniquely identifies the log records of a data sharing member. The LRSN is always incremented for log records that pertain to the same page. There are never duplicate LRSNs for the same page, but LRSNs may be duplicated in log records of members on different pages.

Figure 9-1 on page 79 illustrates how database recovery works in a data sharing group. Each member’s log participates in the recovery process. The member that performs the utility gets access to the group log environment.

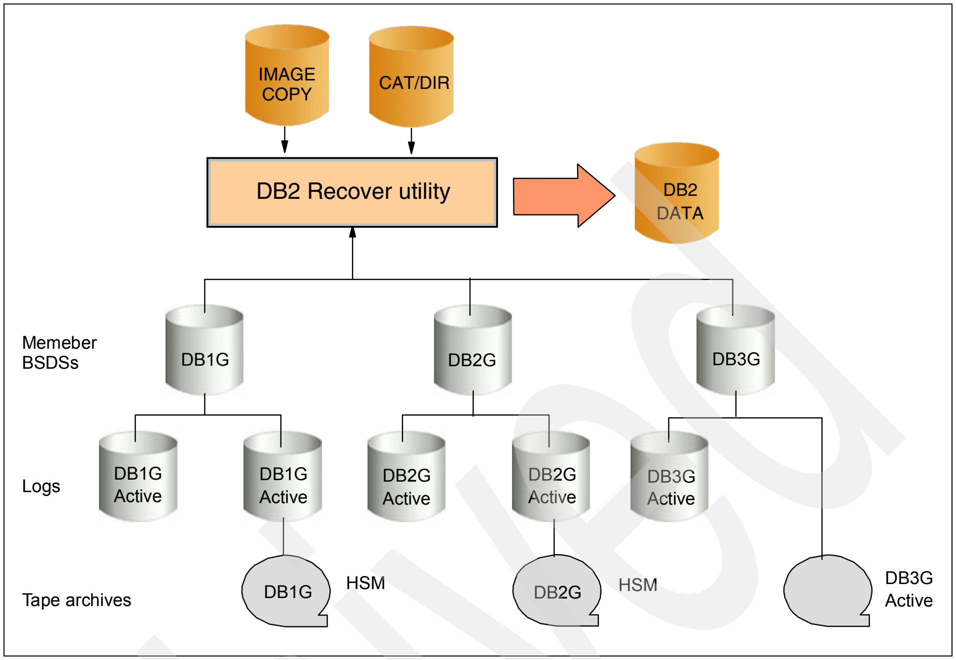


Figure 9-1 Database recovery in a data sharing group

The efficiency of the log apply process can be greately enhanced using the fast log apply (FLA) feature. This feature appeared in Version 6, and it is also used in data sharing during DB2 restart and during START DATABASE for LPL and GRECP recovery. The process is able to sort log records so that pages that are to be applied to the same page or same set of pages are together. Then, using several log apply tasks, it can apply those records in parallel.

In order to enable fast log apply you must provide enough storage using the LOG APPLY STORAGE field of installation panel DSNTIPL. This storage is allocated in the DBM1 address space. If virtual storage is not a problem in the DBM1 address space, it is recommended during recovery to increase the storage available to fast log apply by setting the ZPARM parameter, LOGAPSTG, to the maximum of 100.

When operating in a data sharing group, ability to recover can be hindered by a failed member holding retained locks. You must remove the retained locks by restarting the failed member before you can proceed with pageset recovery. New restrictive states such as LPL and GRECP can have an impact if you perform a logical partition recovery on a nonpartitioning index, so you must remove them first. The recovery process relies on applying changes to a page in the same sequence as they originally occurred. Over a period of time, the same page is likely to be updated by URs running on different members. All changes are externalized to the member's log at commit time. In support of data sharing, log records are expanded with new fields in their headers.

More information about data recovery in data sharing and log considerations can be found in *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935.

9.1.3 Recovering pages on the logical page list

DB2 responds to transient disk read and write problems by placing pages in the LPL. If DB2 cannot determine the reason for a page read or write error, the page is recorded in the LPL. In a data sharing environment, the LPL also contains pages that could not be read or written for must-complete operations because of some problem with the group buffer pool.

Typically only write problems result in LPL pages. Read problems typically result in resource unavailable conditions. The only time a read problem can result in LPL pages is when the read fails during must-complete processing. The read or write operation mentioned can be to DASD or to the group buffer pool.

The LPL is maintained for each data set of a pageset; indexes or partitions have separate lists. The LPL is kept in the database exception table (DBET) in the SCA. For fast reference, the DBET is also cached by each member. When adding pages to the LPL, they are logged as non-UR-related DBET REDO log records.

Some common situations that result in adding entries to the LPL are:

- Error for must-complete operations

  DB2 finds errors when reading a page during restart or rollback processing.

- Force at commit write failure

  After a commit, DB2 must write updated pages to the group buffer pool. In case of problems the page is placed in LPL.

- Group buffer pool castout read failure

  The group buffer pool castout process reads updated pages from the group buffer pool and writes them to DASD. If the read request fails, the requested pages are added to the LPL.

- Restart with DEFER option

  When a member fails while holding pages pending to write to DASD or unresolved units of recovery, DB2 needs access to the data set to apply the changes during restart. If you restart this failing member, and the restart option for the pageset indicates the DEFER option, the pending pages are converted to LPL entries.

DSNB250E is the message that DB2 issues when adding a page to the LPL. This is an important message that should be caught in the system and analyzed. Apart from taking the appropriate action to resolve, it can reveal other important problems.

In some cases, DB2 can automatically recover pages on the logical page list when group buffer pools are defined with AUTOREC(YES), the default. However, there are many situations where pages are put on the LPL that require you to do manual recovery. There are several ways to do this:

- Start the object with access (RW) or (RO). This is the most common method of recovery, and in most cases all that is required.
- Run the RECOVER utility on the object. This method should be used when simply starting the object does not successfully recover the LPL pages.
- Run the LOAD utility with the REPLACE option on the object. This assumes an acceptable copy of the object exists that is current and consistent with the other data objects in regard to application referential integrity.
- Issue an SQL DROP statement for the object. This assumes that the object is no longer needed or can be recreated.
- Use the utility REPAIR SET with NORCVRPEND. This can leave your data in an inconsistent state.
- Use START DATABASE ACCESS(FORCE). This can leave your data in an inconsistent state.

DB2 V8 brings two new enhancements for performing LPL recovery:

- Automatic recovery of LPL pages: To avoid manual intervention for LPL recovery through the START DATABASE command or the RECOVER utility, in most cases DB2 automatically initiates an LPL recovery processor to recover pages as they are added to the LPL.
- Less-disruptive LPL recovery: The LPL recovery processor (by way of the START DATABASE command or the new automatic LPL recovery feature), makes a write claim instead of a drain on the object that is being recovered. As a result, good pages in the object are available to SQL users, and performance is improved because the claim is less disruptive than a drain. In Version 7, the whole tablespace is inaccesible during the recovery.

None of the items in this list work if there are retained locks held on the object. You must restart any failed DB2 that is holding those locks.

9.1.4 Data sharing impact on SAP recovery procedures

We must consider some modifications in order to prepare SAP database recovery procedures to run in a data sharing environment. You will find a good summary of the required changes in the SAP OSS Note 83000, *DB2/390: Backup and Recovery Options*.

This note describes the backup and recovery options that should be implemented in the SAP on DB2 for OS/390 environment. The note is valid for both the data sharing mode and the DB2 normal mode. When it is necessary, SAP OSS Note 83000 addresses specific actions that must be taken when working in data sharing environments.

It is not in the scope of this book to discuss extensively all possible backup and recovery scenarios for SAP on DB2 environments. For a complete description, refer to OSS Note 83000. In the following scenarios we discuss only the main issues affecting backup and recovery as you move your SAP system to data sharing:

- Object-based backup: online and offline
- Volume-based backup: online
- Establishing a group level point of consistency
- Recovery to any prior point in time

9.1.5 Object-based backup: online and offline

This option has no specific data sharing considerations. Full or incremental image copies with SHRLEVEL CHANGE for backup online (concurrent read/write access to the data), or SHRLEVEL REFERENCE for backup offline. Read access to the data can be scheduled by any member of the group during the offline backup, which meets the requirements of the installation.

From an availability point of view, it is recommended that you schedule DB2 administration utilitiy jobs on multiple members. If all of the DB2 administration utility jobs run on one member, the catalog table SYSUTILX will never get inter-DB2 R/W interest and, as a consequence, this member will get an exclusive P-lock at a page set level. If this member fails, this exclusive P-lock will be retained by this member until it is restarted. This will cause a resource unavailable condition on other active members.

For this reason, the Group Attachment Name (GAN) support for generic access to the DB2 members becomes invaluable. Combining GAN support with a Workload Manager Batch Scheduling environment could be established to distribute the DB2 administration utility jobs (for example, jobs generated using SAP transaction DB13) on the most available DB2 member, according to LPAR resources availability and performance objectives.

As a reminder, we offer a list of other considerations that apply to running utilities in data sharing environments:

- DISPLAY UTILITY is a group scope command.
- A running utility can only be terminated on the same running MVS image. A stopped utility can be terminated from any active member in the data sharing group.
- A stopped utility can be restarted in any member of the group.

9.1.6 Volume-based backup: online

Online volume-based backups require availability of a disk subsystem capable of generating very fast volumes copies. Many options are available from different disk vendors. One of the options is FlashCopy with IBM Enterprise Storage Server ESS/SHARK. Prior to DB2 Version 8, these backups are not registered in DB2. In order to obtain a consistent copy of all volumes containing DB2 system or data objects, the copy must be taken after DB2 update activity has been suspended. This is accomplished with the DB2 command SET LOG SUSPEND, which sets a drain on the output buffer for the DB2 log.

In data sharing terms, this command is only member scope, which means that the command only effects the one member on which the command has been issued. Therefore, to achieve a real suspension of the update activity across the whole data sharing group, it is necessary to issue this command on all active members of the group.

Prior to the volume-level copies, it is important to ensure that particular statuses and activities do not exist in the DB2 data sharing group in order to avoid delays at restart time following a recovery action of the database. In particular, no utilities should be active, no pagesets should be in a restricted status, and no long-running units of recovery (batch without frequent commits) should be running during the backup process. In the case of a running utility or a pageset in a restricted status, there may be a recovery of one or more objects required after system restart. If a long-running unit of recovery was running during the volume-level copy, then backout processing may extend the system restart time. The installation parameters URCHKTH and URLGWTH, used in conjunction with LOGLOAD or CHKFREQ, detect long-running units of recovery and issue warnings of a workload not committing in the established period of time for your installation.

In a non-data sharing environment, issuing SET LOG SUSPEND would trigger a DB2 system checkpoint. This is not the case in data sharing. In order to force a DB2 member in the data sharing group to perform checkpoint processing, the command SET LOG LOGLOAD(0) must be issued. It is important that checkpoint processing take place prior to issuing the SET LOG SUSPEND in order to externalize the DB2 data buffers to DASD. This is especially important for pagesets with 32K pages because it takes multiple physcial IOs to externalize their pages. This lowers the probability of a 32K page write only being partially complete during the volume-level copy. Again, this reduces an eventual restart delay. The commands SET LOG LOGLOAD(0) and SET LOG SUSPEND are not group-level commands so they must be issued to each member of the group. There will be pending writes in group buffer pools that will not be externalized to DASD. This is not a problem and will be handled during DB2 group restart.

These are the actions that DB2 initiates with the SET LOG SUSPEND command in a data sharing environment:

1. Force out log buffers.
2. Update the high-written RBA in the BSDS.
3. Hold the log-write latch to suspend updates to the log output buffers.
4. Echo back high-written RBA and last system checkpoint RBA in a DSNJ372I message.

Whenever a SET LOG SUSPEND is issued on a DB2 system, upon successful completion the following message is written to the LPAR syslog, DB2 MSTR message log, and the console:

Example 9-1 Format of message DSNJ372I

```
*DSNJ372I  -DB7X DSNJC09A UPDATE ACTIVITY HAS BEEN  606
 SUSPENDED FOR DB7X AT RBA 0008E300CBD5, LRSN 0008E300CBD5, PRIOR
 CHECKPOINT RBA 0008E2EEA6A6
 DSN9022I  -DB7X DSNJC001 '-SET LOG' NORMAL COMPLETION
```

Keep in mind that volume-level backups are of no use unless it is certain that update activity throughout the DB2 data sharing group has been suspended. Therefore, it is recommended that an automated procedure be put in place to guarantee that this message has appeared for all active members in the group before starting the backup.

We recommend setting up an automated process for the whole backup procedure in order to follow these steps in all of the active members of the group. The specific IBM product for this implementation is System Automation for z/OS. More details about this topic can be found in Chapter 8, “Automation architectures for availability in data sharing” on page 57.

After successful execution of the command, in each member:

- Shared reads are allowed.
- Updates are not allowed.
- Buffer pool contents are not flushed.

- Group buffer pools (GBP) are not flushed.
- Write I/Os and castouts are still allowed.

An SAP end user will notice after triggering this command that saving data takes more time than expected, but querying data proceeds as usual.

At all times we can verify the log activity in DB2 with the DIS LOG command. If log activity is suspended, the output in Example 9-2 appears.

Example 9-2 Output display log command when log activity is suspended

```
-dis log
DSNJ370I =DBK4 DSNJC00A LOG DISPLAY
CURRENT COPY1 LOG = DB2V610K.DBK4.LOGCOPY1.DS01 IS 9% FULL
CURRENT COPY2 LOG = DB2V610K.DBK4.LOGCOPY2.DS01 IS 9% FULL
H/W RBA = 000000A0D662, LOGLOAD = 100000
FULL LOGS TO OFFLOAD = 0 OF 6, OFFLOAD TASK IS (AVAILABLE)
DSNJ371I =DBK4 DB2 RESTARTED 09:37:59 APR 21, 2000
RESTART RBA 00000001D000
DSNJ372I =DBK4 DSNJC00A UPDATE ACTIVITY HAS BEEN SUSPENDED FOR DBK4
AT RBA 000000A0D662
DSN9022I =DBK4 DSNJC001 '-DIS LOG' NORMAL COMPLETION
```

Now the fast volume copy must be invoked from one of the systems. In case of FlashCopy, you can use DFSMSdss™, TSO, Web interface, or script.

For this kind of backup, it is very important to have volume independency between all of the components of the DB2 subsystem. In other words, the active logs and BSDSs of all DB2 members should be on volumes separate from any DB2 directory and catalog objects (VSAM data sets) or SAP DB2 objects (VSAM data sets). Different ICF catalogs should be created for the DB2 system data sets and objects and the SAP DB2 objects. These ICF catalogs should be on one of the volumes with either the DB2 directory and catalog objects or the SAP DB2 objects. This automatically includes them in any volume-level copies. This is even more important if you want to implement PIT recovery without restoring the LOG and BSDS data sets from the volume-level backup.

After the volume-level copy has finished, the updating activity must be resumed in each member of the group using the SET LOG RESUME command.

SET LOG RESUME will:

1. Resume logging and update activity.
2. Release log-write latch.
3. Issue the DSNJ373I message.
4. Delete the DSNJ372I message from the console.

9.1.7 Establishing a group-level point of consistency

Getting a point of consistency in the database is becoming less important due to the fact that point in time recovery must be set at a system level (the whole database is restarted at an specified LRSN and in this way DB2 gets consistency) and because a point of consistency implies a degree of unavailability. Even so, for an installation that can afford this cost, getting a point in which the database is consistent may help in certain situations. Dealing with full image copies obtained with share level change (in which case you cannot recover TOCOPY), it is good to have a daily point where you know that all of your data is committed.

In a data sharing environment, it is even more difficult to get a point of consistency because with all members sharing the same database, this must be a coordinated situation.

As is stated in SAP note 83000, there are different ways of getting the database quiesced:

- ARCHIVE LOG MODE(QUIESCE) TIME(n)
- QUIESCE utility
- START DATABASE ACCESS(RO)
- STOP DB2 MODE(QUIESCE) in all members

Any of these methods also work with data sharing. The first method is preferred. Issuing this command from one of the members provokes all of them to start waiting for all transactions or jobs to commit and draining new units of recovery in the group.

As in non-data sharing, it is recommended that you set the TIME parameter of this command just below the IRLM timeout parameter to avoid cancelling transactions.

9.1.8 Recovery to the current state

Normally, we recover to the current state after some set of data has been damaged, leaving a number of tablespaces in an unavailable state. Typical examples are DASD problems or a failed reorganization. The RECOVER utility discussed earlier in this chapter is usually used. If you are not recovering to current and you need to recover to a prior point in time, you cannot specify TORBA for a PIT recovery. In that case you have to look for an LRSN and specify TOLOGPOINT in the utility control statement.

Most often this will not be the case because recovering a subset of tablespace to a previous point in time, leaving the rest of the database in the current state, goes against the SAP requirement of considering the whole database as a consistency unit of recovery.

9.1.9 Recovery to a previous point in time

When planning for this kind of recovery after moving to data sharing, there are some important points to consider.

The main consideration applies when enforcing a point of consistency to the database with a conditional restart to an arbitrary prior point in the current log or when restoring the whole volume-based backup of the DB2 environment including logs and BSDSs.

First, be consistent within the group. All members of the data sharing group must be restarted to the same point in time to ensure that the database is left in a truly consistent state.

In data sharing, use an LRSN as a common point of conditional restart for all the members instead of an RBA. The DSNJU003 utility enables creation of a conditional restart control record (CRCR) that specifies an LRSN as a parameter. A CRCR should be defined for each member as in Example 9-3.

Example 9-3 Conditional restart record for restarting conditionally in data sharing

```
   CRESTART CREATE ENDLRSN=0008E300CBD5
```

In order to find a valid LRSN, in most installations it is possible to convert a time stamp into STCK format. In some cases enabling data sharing introduces a delay in the LRSN, so it does not match the time stamp. In any case, it is always possible to use DSNJU004 to print BSDS information. From the checkpoint queue section we can match an LRSN with a target point in time.

Example 9-4 Checkpoint queue section from DSNJU004 output

```
CHECKPOINT QUEUE
                       14:28:31 OCTOBER 15, 2002
0      TIME OF CHECKPOINT      14:20:33 OCTOBER 15, 2002
       BEGIN CHECKPOINT RBA           0A05351409BA
       END CHECKPOINT RBA             0A05354CC6EF
       END CHECKPOINT LRSN            B861C340ABA0
0      TIME OF CHECKPOINT      14:19:28 OCTOBER 15, 2002
       BEGIN CHECKPOINT RBA           0A0533939C8D
       END CHECKPOINT RBA             0A053396939C
       END CHECKPOINT LRSN            B861C30286E9
```

Using the new DB2 Version 7 parameter CHKFREQ in minutes ensures that for a given number of minutes there will be a checkpoint that may serve as a reference.

One of the major components of DB2 data sharing is the coupling facility. There are a number of structures in the coupling facility that provide for data integrity across the members of the data sharing group. These structures include the Systems Communication Area (SCA), the LOCK structure, and all of the group buffer pools (GBP). When all of the members of a data sharing group are shut down, the SCA and the LOCK structure remain allocated, but the GBPs are de-allocated. In certain failure situations, a GBP may remain allocated with a failed-persistent connection. Also, there is information about the GBPs stored in the BSDSs of the data sharing members.

There will be information in the BSDSs, logs, and coupling facility structures that represent the state of the system at the time the members were stopped or abended.

So, when using conditional restart, the current state of the data sharing group as recorded in the BSDSs, logs, and coupling facility structures will not match the state of the system at the conditional restart point. In order to be consistent, this information must be rebuilt at restart time by deleting the structures before restarting the members with the MVS command SETXCF FORCE. Then DB2 can perform a group restart.

Before you can force the deallocation of the LOCK structure, all connections must be forced out first. If DB2 abnormally terminated and a GBP is retained in the coupling facility with failed persistent connections, these connections must be forced out as well. In the case of GBP, when a failed persistent connection is forced, it automatically deallocates the GBP structure. If these structures are not purged before restarting, when using a conditional restart the pages resident in the structure could be considered valid for the DB2 members. This could lead to data inconsistencies.

Important: In situations when a conditional restart is performed or data and logs are restored from a previous system backup, it is important to delete the DB2 structures in the coupling facility and let DB2 perform a group restart.

As specified in *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server 6.20*, one recovery option is to recover the whole data sharing environment to the time when a volume-based backup was obtained. In this case a conditional restart of the members is not necessary. The information for restarting the various DB2 members is stored in their BSDSs and logs. If the volume-level copy was taken when the DB2 members were suspended, the HIGHEST WRITTEN RBA for each member is equal to the suspend RBA for each member. Be sure to restore all that is needed for restarting: DB2 and SAP databases, logs from all members, BSDSs from all members, and ICF catalogs. After the structures in the coupling facility are deleted, a group restart brings the DB2 data sharing group to a consistent state based on the volume-level backup. This method is valid under the assumption that you can afford to lose the activity since the backup.

If a volume-level copy is being used in a point-in-time recovery, consider:

- Volume independence was established.
- Only the volumes containing the DB2 directory and catalog and the SAP DB2 objects are restored.
- All of the logs created between the volume-level copy point and the restart point are registered in the BSDS as either active or archive logs.
- The active logs on DASD are the ones registered in the BSDS.
- Some certain number of archives prior to the volume-level copy point are available.
- The image copies of the DB2 directory and catalog taken prior to the volume-level copy are available.
- Image copies of all SAP DB2 tablespaces taken prior to the volume-level copy are available.
- After restart, perform the normal 'to current recovery' for the DB2 directory and catalog. This uses the image copies taken prior to the volume-level copy. The recovery uses image copies instead of LOGONLY because there is no way to guarantee that the HPGRBRBA in the DB2 directory and catalog tablespaces is current. If performing a LOGONLY recovery, there is reasonable probability that one or more of the DB2 directory and catalog tablespaces will need an archive log that is very old, and a recovery using the image copy would be required.
- After identifying which SAP DB2 objects need recovering, use LOGONLY recovery. Prior to DB2 V7, there was no way to guarantee that the HPGRBRBA in the SAP tablespaces was current. Potentially, the LOGONLY recovery would require an extremely old archive log, once again requiring the recovery to be performed using an image copy. With DB2 V7, by using the CHKFREQ ZPARM parameter combined with the DLDFREQ parameter, you can ensure that the HPGRBRBAs for all SAP DB2 tablespaces and indexes are current. Then, LOGONLY recovery will not require archive logs that may have expired.

  The HPGRBRBA for each updated object will be updated on every *n*th checkpoint. The value *n* is based on the DLDFREQ value. If DLDFREQ=5, the HPGRBRBA should be updated every hour for each object being updated. Not all HPGRBRBAs are updated during the same checkpoint.

Instead only a percentage is updated at each checkpoint. That percentage is established by (1/DLDFREQ)*100. So, if DLDFREQ=5, then 20% of the updated objects will have their HPGRBRBAs updated during one checkpoint. If CHKFREQ is set at 10 minutes, then all objects being updated should have their HPGRBRBAs updated once an hour.

In summary, to recover to an arbitary prior point in time using conditional restart, follow the next steps in a data sharing environment:

1. Create a list of tablespaces and indexes that need to be recovered.

   When using object-based backup, these tablespaces are those that changed since the target LRSN. The rest have not changed since the target point and therefore they do not need to be recovered.

   When using volume-based backup, only those page sets that have been modified between the backup and the target point should be recovered with LOGONLY. The rest are already at the target point.

   Refer to *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server 6.20* for a detailed explanation and foro using DSN1LOGP as the basis to prepare the list. There are also considerations that affect to tables dropped or created in between and REORG LOG(NO).
2. Stop all data sharing group members.
3. Copy BSDSs and LOGs that contain LRSNs bigger than the target recovery point.
4. Look for a target LRSN at which all members will conditionally restart.
5. Use DSNJU003 to create a conditional restart record ENDLRSN for all members.
6. Delete all data sharing group structures in the coupling facilities.
7. If restoring from volume-based backup, flashback just the volumes that contain databases. If you are recovering from image copies, be sure to have all image copy data sets ready.
8. Update each member's system parameters and specify DEFER ALL.
9. Restart all members (group restart). New structures must be allocated.
10. Working from one member, recover the DB2 catalog and directory to the current point in time following the specific instructions for this type of database. See Chapter 2-19, "Recover," in *DB2 Universal Database for OS/390 and z/OS: Utility Guide and Reference*, SC26-9945.
11. Recover all tablespaces identified in the first step to the current point in time.
12. Recover or rebuild the indexes on recovered tablespaces.

13. Reinstate RESTART ALL in members' system parameters.

14. Start SAP and perform verifications.

15. Take a new full database backup.

Refer to *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server 6.20* for a detailed description not limited to data sharing.

9.1.10 New utilities in DB2 V8 for online backup and point-in-time recovery

DB2 UDB for z/OS Version 8 provides an easier and less disruptive way for fast volume-level backup and recovery. This utility greatly simplifies backing up systems such as SAP in which the number of database objects in use, as well as recovery requirements, make volume-based backups the most efficient option.

The total solution provided by this utility is dependent on the DFSMShsm™ in z/OS V1.5 and a disk system that provides hardware-assisted volume-level copy. In order for the backup to be registered, the disk system has to write to the DFSMShsm API. IBM ESS disk systems using FlashCopy takes full advantage of this solution. Even so, it is possible to take advantage of some of its features with other disk models and fast copy solutions.

As noted in the previous chapter, one of the challenges of the current online volume backup solutions is the need for coordination between DB2, using the SET LOG SUSPEND command, and the mechanism for triggering the FlashCopy. Currently, the physical copy is not registered in DB2, hence out of DB2's control for later use as a recovery point or as a registered copy to be used in a point-in-time recovery. The procedure for obtaining a system-level copy using FlashCopy must ensure that all SAP system volumes are included and data consistency can be enforced. The procedure for recovering an SAP system using the flashcopied backup, must ensure that:

- All volumes are correcty restored.
- There is a process to identify which SAP DB2 objects (pagesets) require recovery.
- There is a process for generating the recovery jobs.

In DB2 V8 new utilities have been developed integrating DB2 and the fast volume copy capability. Now system-level backups using the fast volume-level copy are managed by DB2 and DFSMShsm, which work together to support a system-level point-in-time recovery. Thus, suspending the DB2 log will no longer be necessary.

Next, brief descriptions are listed for the new DB2 utilities with special considerations for data sharing:

- BACKUP SYSTEM
- RESTORE SYSTEM

Backup system

This utility obtains a complete copy of a DB2 system. There are two kinds: DATAONLY and FULL. DATAONLY is used to obtain a FlashCopy of the data objects, whereas FULL is used to flashcopy data objects and DB2 logs and BSDSs. FULL is designed to restore the DB2 environment to the backup stage.

This utility requires z/OS R5 support for HSM COPYPOOLS. A COPYPOOL is a new SMS construct representing a set of SMS storage groups that are copied together in a single DFSMShsm invocation. A COPYPOOL is limited to 256 SMS storage groups and can be defined with a version attribute in order to keep up to 15 backup versions.

Each DB2 system has two SMS COPYPOOLs:

- DATA COPYPOOL (DSN$location_name$DB)
- LOG COPYPOOL (DSN$location_name$LG)

LOG COPYPOOL is not needed for BACKUP SYSTEM DATAONLY.

Each SMS storage group, including a COPYPOOL, must have a dedicated new type of storage group: COPY TARGET, which is used to hold volume copies of DASD defined in the COPYPOOL.

When the BACKUP SYSTEM utility is used, DB2:

- Suspends 32K writes for objects created before NFM.
- Suspends data set creation, deletion, rename, and extensions operations.
- Prevents data set from pseudo-close.
- Records the Recover Based Log Point (RBLP) in DBD01 with a logscan start point
- Invokes DFSMShsm to take a FlashCopy of ‘DB’ COPYPOOL. HSM uses DSS COPY to copy the volumes in the COPYPOOL.
- For Backup System Full, DB2 invokes DFSMShsm to take a FlashCopy of the ‘LG’ COPYPOOL.
- The Copy is registered in the BSDS of the submitting member.
- Resumes the quiesced activities.

Finally a volume copy of the COPYPOOL is registered by DB2 in BSDSs and by HSM with an associated RBLP inserted in DBD01.

Restore system

This utility is needed to recover the system to an arbitrary point in time.

It may use, as input, copies from BACKUP SYSTEM FULL or DATAONLY. If the recovery does not require the restore of the log backup copies, then use the DATAONLY option of this utility.

As in every recovery, there are two phases:

1. RESTORE phase: Recover the data volumes from the latest backup version prior to the arbitrary point in time.
2. LOG APPLY phase: Apply log records to recover the database to that arbitrary point in time.

When the recovery point in time, and therefore a target LRSN, has been determined, the first thing to do is to deallocate the data sharing group structures in the coupling facility. Afterward, the truncation target LRSN must be established on all active members using CRESTART CREATE SYSPITR=end-lrsn.

3. All members must be restarted.
4. From one of the members issue RESTORE SYSTEM.

When a single DB2 or a data sharing member is conditionally restarted using SYSPITR, the system enters into a System Recover Pending mode. In this status, DB2 automatically uses DEFER ALL, FORWARD = NO and ACCESS(MAINT) when restarting. DB2 establishes consistency during restart by using the log and recreating the data sharing group coupling facility structures.

In order to downgrade the System Recover Pending status, you must submit the RESTORE SYSTEM utility.

When it is issued, DB2 asks HSM for the COPYPOOL version that was taken by BACKUP SYSTEM prior to the specified point-in-time recovery point. Subsequently, DB2 performs the log apply function.

During log apply phase, DB2:

1. Reads the DBD01 header page to retrieve RBLP and the log scan starting point.
2. Applies log recovering objects in parallel and using Fast log apply (FLA).
3. Detects creates, drops, extends, and LOG NO events. Objects are marked in RECP or RBDP state.

RESTORE SYSTEM can handle fast volume copies obtained with SET LOG SUSPEND and without BACKUP SYSTEM starting in z/OS V1.3 . In this case the LOGONLY option must be specified.

9.2 Data sharing considerations for disaster recovery

Another important aspect to consider, after deciding to enable data sharing for SAP on DB2 on OS/390 or z/OS, is the need to introduce changes in the disaster recovery procedures to acommodate the new configuration. We describe the most important concepts and different options for implementing a disaster recovery strategy with data sharing, from the traditional method to the most up-to-date implementation.

The options for implementing a disaster recovery strategy with data sharing are essentially the same as the options in non-data sharing environments. However, some new steps and requirements must be addressed.

Detailed descriptions of disaster recovery options can be found in *DB2 UDB for OS/390 and z/OS V7 Administration Guide*, SC26-9931. Specific information about data sharing is available in *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935. For SAP, good references are *SAP R/3 on DB2 for OS/390 Disaster Recovery*, SG24-5343, and documentation about Split Mirror Backup/Recovery Solutions can be found at:

`http://www.storage.ibm.com/hardsoft/diskdrls/technology.htm`

9.2.1 Configuring the recovery site

The recovery site must have a data sharing group that is identical to the group at the local site. It must have the same name and the same number of members, and the names of the members must be the same. The coupling facilities resource manager (CFRM) policies at the recovery site must define the coupling facility structures with the same names, although the sizes can be different. You can run the data sharing group on as few or as many MVS systems as you want.

The hardware configuration can be different at the recovery site as long as it supports data sharing. Conceptually, there are two ways to run the data sharing group at the recovery site. Each way has different advantages that can influence your choice:

- Run a multi-system data sharing group.

  The local site is most likely configured this way, with a Parallel Sysplex containing many CPCs, MVS systems, and DB2s. This configuration requires a coupling facility, the requisite coupling facility channels, and the Sysplex Timer®.

The advantage of this method having the same availability and growth options as on the local site.

- Run a single-system data sharing group.

  In this configuration, all DB2 processing is centralized within a single, large CPC such as an IBM z800 or later zSeries processor. With even a single CPC, a multi-member data sharing group using an internal coupling facility must be installed. After the DB2 group restart, all but one of the DB2s are shut down, and data is accessed through that single DB2.

Obviously, this loses the availability benefits of the Parallel Sysplex, but the single-system data sharing group has fewer hardware requirements:

- The Sysplex Timer is not needed, as the CPC time-of-day clock can be used.
- Any available coupling facility configuration can be used for the recovery site system, including Integrated Coupling Facilities (ICFs).

With a single-system data sharing group, there is no longer inter-DB2 R/W interest, and the requirements for the coupling facility are:

- A LOCK structure (which can be smaller)
- An SCA

Group buffer pools are not needed for running a single-system data sharing group. However, small (at least) group buffer pools are needed for the initial startup of the group so that DB2 can allocate them and do damage-assessment processing. When it is time to do single-system data sharing, remove the group buffer pools by stopping all members and then restarting the member that is handling the workload at the disaster recovery site.

9.2.2 Remote site recovery using archive logs

Apart from these configuration issues, the disaster recovery procedural considerations do not greatly affect the procedures already put in place for a single DB2 when enabling data sharing. All steps are comprehensively documented in Chapter 4 of the *DB2 UDB for OS/390 and z/OS V7 Administration Guide*, SC26-9931.

The procedure for DB2 data sharing group restart at the recovery site differs in that there are steps ensuring that group restart takes place in order to rebuild the the coupling facility structures. In addition, you must prepare each member for conditional restart rather than just a single system.

To force a DB2 group restart, you must ensure that all of the coupling facility structures for this group have been deallocated:

1. Enter the following MVS command to display the structures for this data sharing group:

   ```
   D XCF,STRUCTURE,STRNAME=grpname*
   ```

2. For the LOCK structure and any failed-persistent group buffer pools, enter the following command to force the connections off of those structures:

   ```
   SETXCF FORCE,CONNECTION,STRNAME=strname,CONNAME=ALL
   ```

   With group buffer pools, after the failed-persistent connection has been forced, the group buffer pool is deallocated automatically.

   In order to to deallocate the LOCK structure and the SCA, it is necessary to force the structures out.

3. Delete all of the DB2 coupling facility structures by using the following command for each structure:

   ```
   SETXCF FORCE,STRUCTURE,STRNAME=strname
   ```

   This step is necessary to clean out old information that exists in the coupling facility from your practice startup when you installed the group.

Following is a conceptual description of data sharing disaster recovery using the traditional method of recovery based on image copies and archive logs.

First, be sure to have all of the information needed for the recovery. The required image copies of all the data objects will be the same, but now all the BSDSs and archive logs from all members must be provided using one of three options:

- **`Archive log mode(quiesce)`**

  As previously explained, this command enforces a consistency point by draining new units of recovery. Therefore, this command is restrictive for providing continuous availability but, under successful execution, it gets a groupwide point of consistency whose LRSN is specified in the BSDS of the triggering member.

- **`Archive log mode(group)`**

  With this command, members of the group are not quiesced in order to establish a point of consistency, but all of them register a checkpoint for their log offload. Because we are going to conditionally restart all the members of the group, we must find a common point in time on the log in order to provide for consistency throughout the group. We will have to find the lowest ENDLRSNs of all the archive logs generated (see message DSNJ003I), subtract 1 from the lowest LRSN, and prepare the conditional restart for all members using that value.

Attention: Make sure that all members of the group are active when you archive the logs. If you have a quiesced member whose logs are necessary for a recovery base at the disaster recovery site, you must start that member with ACCESS(MAINT) to archive its log.

- **`Set log suspend`**

  If we plan to use a fast volume copy of the system, we must remember that the suspend command does not have group scope, so it must be triggered in all group members before splitting pairs or performing FlashCopy.

At the recovery site, it is important to remember that each member's BSDS data sets and logs are available. Also, the logs and conditional restart must be defined for each member in the respective BSDS data sets. The conditional restart LRSN for each member must be the same. Contrary to the logs and BSDS data sets, the DB2 Catalog and Directory databases, as with any other user database, exist only once in the data sharing group and only have to be defined and recovered once from any of the active members.

Also, DSNJU004 and DSN1LOGP have options that allow for a complete output from all members.

After all members are successfully restarted, if you are going to run single-system data sharing at the recovery site, stop all members except one by using the STOP DB2 command with MODE(QUIESCE). If you planned to use the light mode when starting the DB2 group, add the LIGHT parameter to the START command listed above. Start the members that run in LIGHT(NO) mode first, followed by the LIGHT(YES) members.

You can continue with all of the steps described in Section 4.7.17, "Remote site recovery from disaster at a local site," in *DB2 UDB for OS/390 and z/OS V7 Administration Guide*, SC26-9931.

9.2.3 Using a tracker site for disaster recovery

A DB2 tracker site is a separate DB2 subsystem or data sharing group that exists solely for the purpose of keeping shadow copies of your primary site data.

No independent work can be run on the tracker site. From the primary site, you transfer the BSDS and the archive logs, then the tracker site runs periodic LOGONLY recoveries to keep the shadow data up-to-date. If a disaster occurs at the primary site, the tracker site becomes the takeover site. Because the tracker site has been shadowing the activity on the primary site, you do not have to constantly ship image copies. The takeover time for the tracker site can be faster because DB2 recovery does not have to use image copies.

Tracker site recovery

Using DB2 for z/OS V8, we can take advantage of the new utilities to perform tracker site recovery. These steps can be used:

- Use Backup System to establish a tracker site.
- Periodically send active, BSDS, and archive logs to tracker site (PPRC, XRC, FTP, or tapes).
- Send image copies after load/reorg log(no).
- Each tracker recovery cycle:
  - Run RESTORE SYSTEM LOGONLY to roll database forward using logs.
  - Use image copies to recover objects that are in recover pending state.
  - Rebuild indexes that are in rebuild pending state.

More information about setting up a tracker site and recovery procedures can be found in *DB2 UDB for OS/390 and z/OS V7 Administration Guide*, SC26-9931, *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935, and *SAP R/3 on DB2 for OS/390: Disaster Recovery*, SG24-5343.

9.2.4 GDPS infrastructure for disaster recovery

GDPS® stands for Geographically Dispersed Parallel Sysplex. It is a multisite application that provides the capability to manage:

- The remote copy configuration and storage subsystems
- Automated Parallel Sysplex tasks
- Failure recovery

Its main function is providing an automated recovery for planned and unplanned site outages. GDPS maintains Multi-Site Sysplex, in which some of the MVS images can be separated by a limited distance (currently not recommended more than 20 km). GDPS follows the sysplex specification of being an application independent solution.

The primary site contains some of the MVS Sysplex images supporting some of the data sharing group members, and the primary set of disks. These are the disks that support all DB2 activity coming from any DB2 member of the group. At the secondary site, there are active syplex images supporting active DB2 members working with the primary set of disks. There is also a secondary set of disks, which are mirror copies from the first site.

GDPS supports two data mirroring technologies:

1. Peer to peer remote copy (PPRC) in which:
   - The mirroring is synchronous.

   - GDPS manages secondary data consistency and therefore no, or limited, data is lost in failover.
   - The production site performs exception condition monitoring. GDPS initiates and executes failover.
   - Distance between sites up to 40 km (fiber).
   - Provides for both: Continuous availability and Disaster recovery solution.
2. Extended remote copy (XRC) with:
   - Asynchronous data mirroring.
   - Limited data loss is to be expected in unplanned failover.
   - XRC manages secondary data consistency.
   - GDPS executes Parallel Sysplex restart.
   - Supports any distance.
   - Provides only a disaster recovery solution.

The following is an example of multi-functional disaster recovery infrastructure using GDPS and PPRC to provide all the elements of a backup and recovery architecture, including:

- Conventional recovery, to current and to a prior point in time
- Disaster recovery
- Fast system copy capability to clone systems for testing or reporting
- A corrective system as a “toolbox” in case of application disaster
- Compliance with the high availability requirements of a true 24x7 transaction environment based on SAP

This configuration is prepared to support very stringent high availability requirements in which no quiesce points are needed, and the need for SET LOG SUSPEND is avoided using the command to freeze GDPS. In this way, data backup is obtained without production disruption. No loss of transactions and data is encountered, even during split mirror phase. The infrastructure provides for a corrective system as a snapshot of production that can be obtained repeatedly throughout the day.

The components of this example solution are IBM @serverzSeries, z/OS Parallel Sysplex, DB2 for z/OS data sharing, GDPS with automation support, IBM ESS Shark disk subsystems with PPRC/XRC and FlashCopy functionality, and SAP/IBM replication server for application high availability.

Figure 9-2 on page 100 shows the GDPS solution landscape.

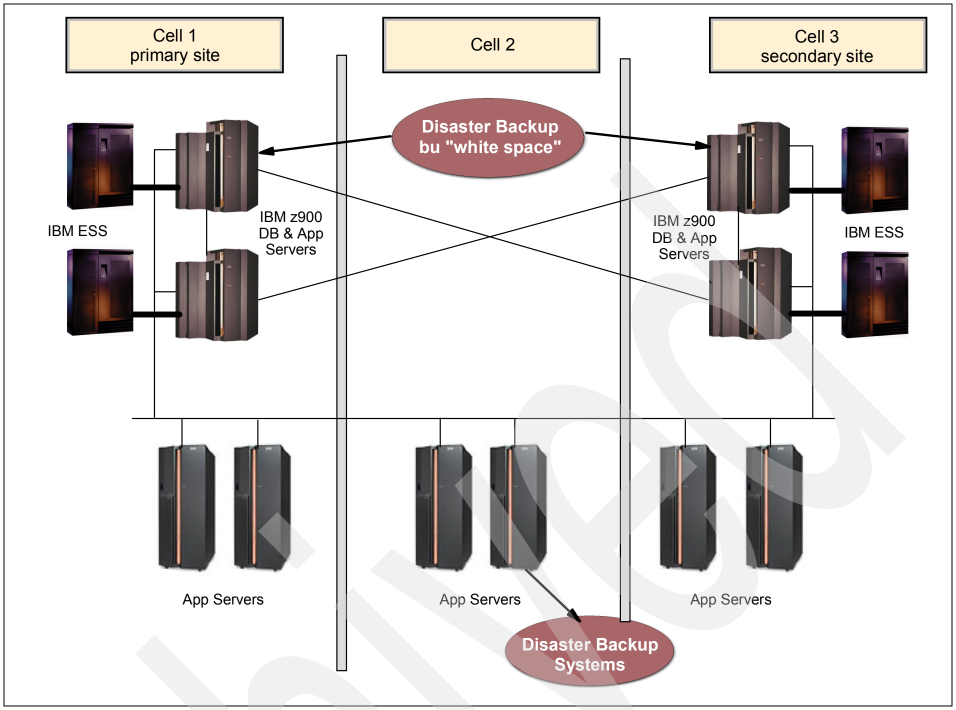


Figure 9-2 High availability with GDPS configuration example

This configuration is made up of two sites and three cells. (Cell 2 is where the corrective system is started.) The three cells are totally encapsulated and safe against floods, earthquakes, and so on. The distance between cell 1 and cell 3 should be about 20 km based on GDPS recommendations. Both cells belong to the same sysplex and keep members of the same data sharing group. Cell 2, on the other hand, is out of the sysplex in order to keep the same DB2 data set names for the corrective system. In future versions of FlashCopy, this will not be a requirement.

ESS primary and active set of disks is located on the primary site and, using PPRC, they are mirrored to the secondary site. All ESS FlashCopy activity takes place on the secondary site. The design keeps symmetry between both sites, having the same ESS disk capacity on each site. Therefore, if one site is not available (disaster, maintenance), the other is able to provide an alternate backup process using SET LOG SUSPEND.

This infrastructure uses the GDPS `freeze` command to suspend all data storage operations temporarily, and initiates FlashCopy on the secondary site. The mirror is split until the end of FlashCopy.

Figure 9-3 illustrates the process of obtaining a non-distruptive volume backup.

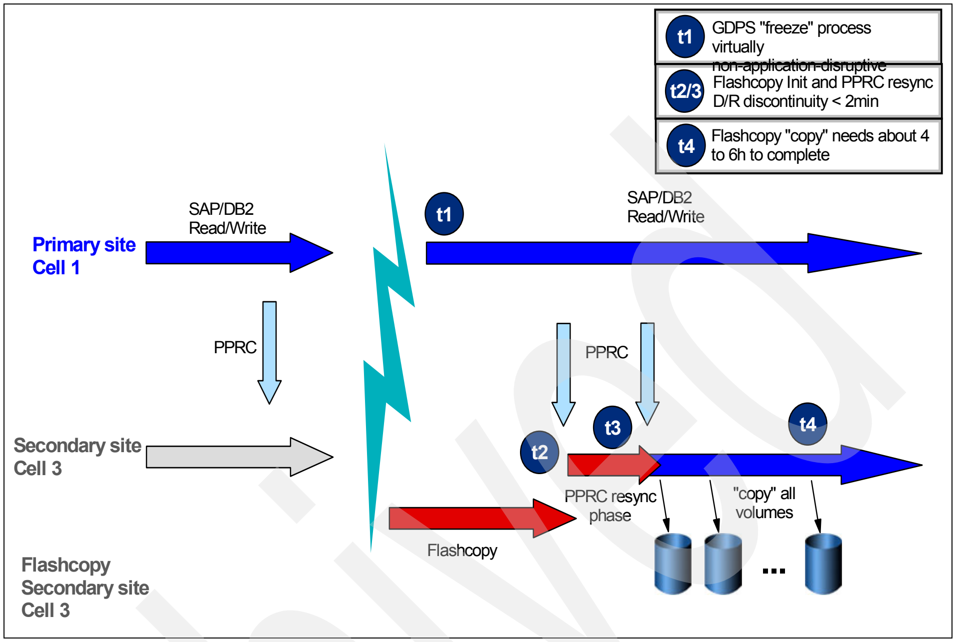


Figure 9-3 Process of obtaining a nondisruptive volume backup

Unlike the DB2 log suspend method, t1 in the GDPS freeze process is just a moment, not a duration. The **freeze** command may keep the primary site volumes frozen for one second. During this time frame DB2 looks stopped and PPRC is split between both sites. Immediately, activity continues normally on primary site while, on the secondary site, the initial FlashCopy phase is taking place.

At the end of the FlashCopy initial phase (t2), PPRC synchronizes the volumes on both sites.

> **Important:** Even with this configuration, there is a possibility of getting 32k-page set in recovery pending after restoring from backup. (See SAP OSS Note 363189.) In this case a recovery from image copy is still needed to reset status.

Between t1 and t3 (several minutes for big databases) there is a possibility of losing transactional data in the event of disaster failure. The reason is that the mirroring is not active during this interval.

One way to solve this problemis to exclude the second active log copy of all members (in the primary site) from this mirroring, and enable some kind of backup of this active log. A recovery using this backup could provide transactional data until the last moment.

9.3 Homogeneous system copy in data sharing

Under normal conditions, sooner or later every SAP installation finds the need to perform an efficient homogeneous system copy (HSC). Customers use SAP R/3 homogeneous system copy for various reasons:

- Application testing and quality assurance
- System function test
- Production maintenance
- Reporting
- Data mining
- Training

SAP R/3 supports two methods for performing an HSC:

- Using SAP R/3 export/import tools
- Using database-specific tools

The SAP R/3 export/import procedure uses a standard SAP-supplied transaction to export the data from the source database to a flat file and then import the data into the target database. This process is not recommended for large production SAP environments. Typically, it is used with small systems that are being used in pilot projects or some development efforts. The time it takes to accomplish the export-import process with large production systems makes this process prohibitive.

Therefore, we focus on the second method, which is the most commonly used in SAP installations. Our starting point will be the standard procedure for DB2 for OS/390 or z/OS described in Chapter 4 of *SAP R/3 Homogeneous System Copy, Release 4.6C SR2* documentation. Here, we discuss the procedural changes required to support a database server that has enabled DB2 data sharing.

The aim of this section is to present concepts, not to be exhaustive in the steps sequence. For a complete review of the procedure, reference the detailed steps and considerations, including data sharing, given in *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287.

9.3.1 Planning for homogeneous system copy in data sharing

When planning for homogeneous system copy for a source system that is a data sharing group, consider the following issues:

- What is the DB2 data sharing configuration of the target system?
- Which method are you going to use to obtain the copy?

It is not uncommon to find in some installations that the production DB2 system has been configured for high availability, while the non-production DB2 systems have not. Usually this is done to conserve resources. There could be instances of a non-production system being non-data sharing or, if it is data sharing, not having the same number of members as the production system. In this case, we could find a different group configuration between source and target system.

However, if it is determined for availability reasons to obtain the source system copy using online processes (fast copy volume solution and set log suspend), the target system configuration has specific requirements for facilitating group restart and retained lock resolution. If the source system DB2 data sharing group is going to be quiesced and stopped while obtaining the copy, the requirements on the target configuration are not as stringent.

Review of HSC in non-data sharing

In order to understand the implications of the issues involving source and target systems configuration and whether the source system copy is obtained online or offline, we first must review the normal homogeneous system copy method for non-data sharing to non-data sharing.

The HSC method is based on copying the whole DB2 system from one environment to the other. If the copy is *offline* we must stop the SAP environment and DB2 system prior to the copy process. If the copy is *online*, we must perform a SET LOG SUSPEND, take the fast volume copy and SET LOG RESUME to continue running, or use the BACKUP SYSTEM utility in DB2 V8.

At some point there must be a step to rename the data sets to the target environment HLQ. This rename can be done:

- During the DFDSS logical copy if using the offline method
- With DFDSS and an intermin LPAR if using an online copy
- With an ISV tool
- With ESS shark disks, using the new features of FlashCopy at data set level

In the copy we must include the following data sets:

- DB2 log data sets
- DB2 BSDS data sets
- DB2 system data sets

- SAP system data sets
- (Optionally) SMPE target libraries

Now, assuming that all of the procedures, parameter libraries, and MVS definitons have been established for the target DB2 environment, prepare the start of the DB2 target system.

In a non-data sharing to non-data sharing HSC, the source system BSDS data sets can be copied into the target system BSDS data sets and used for restart of the target system. However, the VCAT alias and the active log data sets must be changed to the target system's VCAT and active log data set names. The modifications can be performed with the stand-alone utility DSNJU003. The only other modification that might be required is the DDF information. There is no requirement for a conditional restart card.

The restart of the target system varies depending on whether the source system copy was obtained online or offline. Restarting from an online copy requires access to the DB2 catalog and directory and SAP tablespaces in order to recover any outstanding units of recovery or externalize unwritten pages that existed at the time of the log suspend. At the time of target system restart, the VCAT stored in the DB2 catalog tables SYSSTOGROUP, SYSTABLEPART, and SYSINDEXPART is still the VCAT from the source system. To avoid access to the source system's VSAM data sets, you must restart the target system with DSNZPARM DEFER ALL.

During the restart of the target system from a source system offline copy, there should not be any units of recovery to resolve or unwritten pages to externalize. However, it is still recommended to start with DEFER ALL to ensure that the target system does not try to open any of the source system VSAM data sets.

After the DB2 target system has restarted, the temporary workspace tablespaces for the target system must be defined and created. Then all of the DB2 steps necessary to alter the VCAT alias, in all of the defined stogroups, must be performed. After the VCAT alias has been altered to the VCAT for the target system, DB2 opens the VSAM data sets for the target system, instead of the VSAM data sets for the source system.

Finally, perform the subsequent technical actions for SAP. Refer to steps 6 to 14 of the *SAP R/3 Homogeneous System Copy, Release 4.6C SR2* documentation.

Requirements for data sharing

Data sharing introduces the following changes to the procedure:

- Coupling facility structures information cannot be included in the source system copy. For online copy, some committed data pages in the group buffer pools will have to be recovered in the target system.

- BSDSs cannot be exported with the copy because it contains data sharing group information that cannot be changed.
- To move from data sharing to non-data sharing, or to a data sharing group with a different number of members, perform a cold restart. This is only possible when using an offline copy of the source system's database. This means that all members were quiesced and stopped prior to the copy being obtained.

9.3.2 Designing homogeneous system copy in data sharing

In order to apply the modification to the procedure introduced by the data sharing conditionings, we consider two cases:

- Data sharing to data sharing (with the same number of members) copy
- Data sharing to non data sharing copy

In either case, when the target system is also data sharing group there is no other option than performing a target system group restart to allocate new structures in the coupling facility. Therefore, preparation steps must be performed to assure good CFRM structure definitions and enough space in the coupling facility for the structures.

Data sharing to data sharing

We now describe the two copy possibilities: online and offline.

Online copy design considerations

During the online copy, all active DB2 members of the source data sharing group are suspended. When this copy is used to restart a DB2 data sharing group at the target, an equivalent number of DB2 members must be restarted at the target system to resolve held locks. This is similar to having a DB2 data sharing member abend. To resolve held locks, that member must be restarted.

In order to support group restart via the coupling facility, it is necessary to have the same number of members in the target system as in the source system. However, not all of the members in the target system have to be configured as robustly as a member that actually supports a workload. In other words, the active logging configuration must be sufficient to support group restart and nothing else. The configuration to support group restart consists of each target member having BSDS data sets, and the current active log from the source member available and registered in the target member's BSDS.

It may not be necessary to restart all members in the target system. If a member, or members, of the source system were quiesced and stopped at the time of the copy, these members will not need to be restarted in the target system. However, all active source members must be restarted. This is required in order to resolve local locks held by an active member. The members that are restarted will read the BSDS and the registered active logs of the members that are not restarted and will perform group restart for these peer members.

The restart process can use active or archive logs from the source system. If the restart of the target system is done using active logs from the source system, APAR PQ46138 is required. The active log configuration for each member of the target data sharing group can be different from that of the source system members and different from each other.

Many things can be changed in the BSDS via the change log inventory utility (DSNJU003). However, the information about the data sharing group and its members cannot be changed, so it is necessary to keep all BSDSs, belonging to all members, of the target data sharing group intact. That means that we do not use the BSDSs from the source system to perform the restart of the target system. However, there is information in the source system BSDSs that must be recorded in the target system BSDSs in order to accomplish the restart in the target system. Depending on whether the restart is being done with the active logs versus the archive logs, the required BSDS information will vary. This information may include some, but not all, of the following items:

- The suspend LRSN, to be used as the conditional restart LRSN
- The checkpoint taken just prior to the suspend
- The archive log containing the suspend LRSN and the checkpoint
- The active log containing the suspend LRSN and the checkpoint
- The highest written RBA

To ensure the successful use of this information during the restart of the target system, consider creating a skeleton BSDS. See “Creating the skeleton BSDS” on page 17 of *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287.

Offline copy design considerations

During the offline copy, all members of the source data sharing group are stopped. There should not be any outstanding units-of-recovery, and all data pages in the virtual buffer pools should have been externalized (written to disks). In other words, all data managed by the source system is quiesced and consistent.

The process is similar to the online copy procedure except that there is no log suspend and resume. On the contrary, the copy is made with DB2 group stopped, and the BSDSs print log map from each source member should be obtained while the group is stopped. With this information we define the restart of the target DB2 data sharing group. The restart process should be faster, for there are no page sets to recover.

Data sharing to non-data sharing

This DB2 system cloning configuration involves moving the data from a DB2 data sharing group to a non-data sharing DB2.

This step is similar to disabling data sharing in one DB2 environment. There is no other way than performing a cold restart. For this reason the database must be copied in a state of consistency, which can only be achieved with offline copy.

Because the target system is non-data sharing, the DB2 system is managed by RBA and not LRSN. The target system original BSDS and active logs are used. The information required to perform the cold start would be registered in the BSDSs of the target system.

As an example, suppose our source DB2 system has a two-member data sharing group. The target system is a non-data sharing DB2. The highest used LRSN of our source system could be used as the restart RBA of our target system. Example 11-5 shows the highest used LRSN in the source system.

Example 9-5 BSDS print of source DB2 with LRSN

```
TIME OF CHECKPOINT 18:00:08 JUNE 18,2001
BEGIN CHECKPOINT RBA 0012F391263C
END CHECKPOINT RBA 0012F391448C
END CHECKPOINT LRSN B6016DA1E435
```

Example 3-2 shows the cold start at the target system with the source LRSN used as target start RBA.

Example 9-6 Highest used LRSN (source) to RBA (target)

```
//ACTLOG EXEC PGM=DSNJU003
//STEPLIB DD DISP=SHR,DSN=DSN610.SDSNLOAD
//SYSUT1 DD DISP=OLD,DSN=DB2V610B.BSDS01
//SYSUT2 DD DISP=OLD,DSN=DB2V610B.BSDS02
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
//SYSIN DD *
CRESTART CREATE,STARTRBA=B6016DA1F000 ,ENDRBA=B6016DA1F000
/*
```

As previously noted, testing environments with all of the details to plan and prepare the procedures and recommendations can be found in *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287.

Part 3

Performance considerations

Part 3 describes SAP performance and scalability considerations in a data sharing environment in the order you would approach them at planning and implementation time.

We build a bottom-up case for good SAP performance as illustrated in Figure 9-4 on page 110. A well-performing sysplex is the base for a well-performing DB2 data sharing system, which in turn is the base for a well-performing SAP system.

We discuss:

- Sysplex performance: performance issues relevant to ensuring the good performance of your syspex platform
- Data sharing performance: performance issues relevant to ensuring the good performance of your DB2 data sharing system
- SAP performance: performance issues relevant to the good performance of your SAP system on the DB2 data sharing platform.

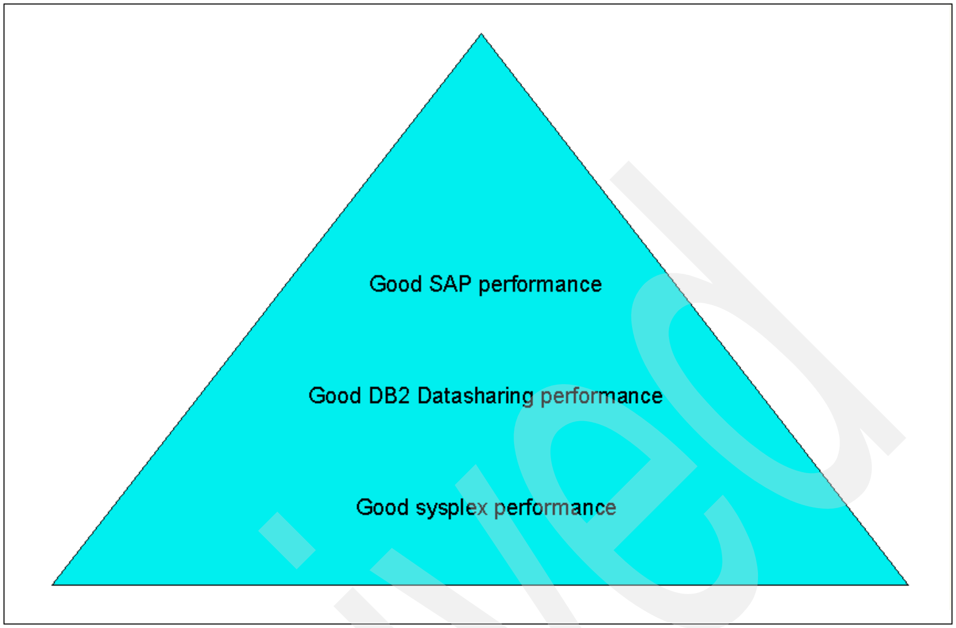


Figure 9-4 bottom-up approach to good SAP performance

We primarily focus on methodology and refer you, when necessary, to publications where you can find more detailed discussion about these subjects.

10

Sysplex performance

This chapter discusses sysplex performance and monitoring issues relevant to SAP data sharing environments. A sysplex that is set up correctly and performing well is the basis for good performance in your DB2 data sharing system. We describe:

- What to measure: defining a baseline
- CF performance monitoring and tuning
- Sysplex workload monitoring
- WLM policies
- RMF™ Monitor III
- RMF Postprocessor reports
- RMF spreadsheet

10.1 What to measure: defining a baseline

To ensure that tuning efforts are producing the expected results, keep a baseline for a before-and-after comparison. This baseline may be comprised of:

- Coupling facility processor utilization
- Average synchronous and asynchronous service time
- Percentage of changed requests

10.2 CF performance monitoring and tuning

This is usually not a DB2 topic, but it is important to have a basic understanding of coupling facility (CF) performance tuning concepts in order to have a properly tuned DB2 data sharing group.

Four items need to be monitored:

- CF processor utilization
- CF storage utilization
- CF link performance
- CF signalling path length

Monitoring coupling facility utilization and performance can be achieved by using RMF Spreadsheet Reporter, which is described in 10.6, “RMF Spreadsheet Reporter” on page 123.

10.2.1 CF processor monitoring

CF processor utilization must be kept under 50 percent because synchronous service time starts to degrade with higher utilization. High sync service time may have a performance impact because the requestor’s processor has to wait for the request to complete. Besides reducing CF efficiency, it also increases CPU utilization in the z/OS side.

Our recomendation is to keep CF utilization under 30 percent if you are in a single-CP coupling facility, in case of failover and one CF has to absorb the workload from a failing CF.

To obtain information about CF utlization, run an RMF Postprocessor report with option SYSRPTS(CF).

10.2.2 CF storage utilization

CF level 12 introduces 64-bit mode support and removes the 2 GB limitation that existed for some DB2 structures. CF level 12 provides the following advantages:

1. 64-bit CFCP with CFLEVEL12 and OS/390 R10 to overcome the 2 GB control storage limit and provide for very large LOCK1 structures (because of modified record list requirement)
2. 64-bit real with OS/390 R10 and any z/OS operating system to support dataspace buffer pools and provide VSCR above line in DBM1
3. 64-bit virtual support with z/OS R3 and DB2 for z/OS V8 to provide even more VSCR
4. Reduced data sharing overhead, using WARM/RFCOM with z/OS R4, CFLEVEL12, and DB2 V8
5. SLPIT Recovery (BACKUP SYSTEM utility and RESTORE SYSTEM utility without LOGONLY option) with z/OS R5 and DB2 V8

Special care also must be taken in anticipation of failover scenarios. The surviving CF must have enough capacity to absorb the structures coming from a failing CF.

10.2.3 CF link performance

Service time is monitored from the moment an exploiter issues a CF request to the moment z/OS receives the command return. Service time is recorded in microseconds for each structure used by each system. As CF links can be shared between two or more LPARs in the same CPC, it is possible some contention exists.

When such a contention occurs, one request is processed and the other is rejected. This situation gets registered as ‘Path busy’. It is important to check the total percentage of delayed request in the Subchannel Activity section from an RMF CF report.

If the total path busy is less than 10% of all requests, this indicates link contention. Consider dedicating CF links to the CF from each partition or adding more shared links to the CF.

Each CF link is capable of supporting a certain number of concurrent CF operations. The maximum number of concurrent CF operations is the total number of subchannels (buffers) defined to a CF.

When all subchannels are busy performing operations to the CF, then newly incoming requests are delayed. This is known as a *subchannel busy* condition.

For SCA and group buffer pool requests, z/OS will queue these delayed requests and process them in first-in-first-out order as soon as one of the active operations ends. For a LOCK structure, z/OS will generally spin and wait for one of the active requests to finish and then re-issue the request.

In summary, for SCA and group buffer pools, increased subchannel busy may lead to request queuing, thus affecting performance. For the LOCK structure, high subchannel busy leads to higher CPU utilization and diminished CF efficiency.

If the total subchannel busy is less than 10% of all requests, you may have to add another CF link between the CF and the z/OS partition.

We recommend monitoring service times. The zSeries family of processors introduced three additional links: ISC-3, ICB-3, and IC-3. These links must be configured in Peer Mode. Peer Mode supports CF between z900 servers and provides both sender and receiver capability on the same link. Peer links provide up to seven expanded buffer sets (subchannels), which in reality becomes as many as 14 buffer sets, as both receiver and sender traffic travel in the same link.

Each link type has its own characteristics. They facilitate improved channel efficiency and service times in the processing of CF requests. But it is beyond the scope of this book to provide more in-depth details on CF links.

Service times vary accordingly to the hardware involved. Figure 10-1 shows the guidelines for service times for the z/900 series.

```
Synchronous service time:
        GBPs            LOCK1, SCA
ICP     30 mics         20 mics
CFP     65 mics         40 mics

Asynchonous service time:
        GBPs            LOCK1, SCA
ICP     120 mics        100 mics
CFP     150 mics        120 mics
```

Figure 10-1 Service times for z/900 series

Starting with z/OS 1.2, IBM introduced a new algorithm to decide whether to convert synchonous requests to asynchronous requests. This new function monitors CF service times for all structures (CACHE, LIST, and LOCK) and, based on new internal thresholds, determines which requests would be more efficiently executed asynchronously. Thresholds are different for simplex and duplex requests and for Lock and non-Lock requests.

For more insight, review Washington System Center Flash 10159 at:

http://www-1.ibm.com/support/docview.wss?uid=tss1flash10159

10.2.4 CF signalling path length monitoring

z/OS systems within a sysplex must pass messages to each other constantly. These signalling paths are heavily used when IRLMs need to negotiate locks in a data sharing environment.

When Global Lock contention is detected, Global Lock Management via XES/IRLM uses message passing and is heavily dependent on fast XCF communications. Global Lock Management is synchronous with respect to the application. The lack of enough XCF signaling resources to accommodate Global Lock contention management traffic can be detrimental to DB2 performance. It is best to overconfigure XCF signaling resources.

We have observed that low-weight LPAR partitions can get worse performance for anything dependent on asynchronous operations and XCF messaging (such as Global Lock contention). There are several spots they can slow down:

- The z/OS dispatcher must run to see an asynchronous completion. The LPAR may not have any logical CPs for MVS1D dispatched when the asynchronous operation completes, so MVS1D is very late in noticing.
- Several SRBs must run to complete the asynchronous operation, post XCF, and then wake up IRLM. LPAR can take away the logical processor running any of this work whenever it feels like it. This could also slow down lock contention on other partitions as they sometimes have to wait for MVS1D to respond to them

Transfer time between LPARs can only be seen by issuing these commands:

- D XCF,PI,DEV=ALL,STATUS=WORKING
- D XCF,PI,STRNM=ALL

The recommended transfer time is less than 2000 microseconds (or 2 milliseconds). These IRLM negotiations are synchronous, so keeping them under 2000 microseconds is very important.

It is recommended that these two commands be issued automatically every hour.

To monitor signalling paths you must run an RMF Postprocessor with REPORTS(XCF) control card. For XCF tuning guidelines, refer to IBM Washington Center Flash 10011 at:

http://www-1.ibm.com/support/docview.wss?uid=tss1flash10011

This flash provides a step-by-step road map to ensure that transfer time between LPARs is under 2000 microseconds.

Figure 10-2 on page 116 shows an example of an RMF Postprocessor output from option REPORTS(XCF).

```
 ----- BUFFER -----      ALL
TO        TRANSPORT  BUFFER        REQ   %    %    %    %     PATHS     REQ
SYSTEM    CLASS      LENGTH        OUT  SML  FIT  BIG  OVR  UNAVAIL  REJECT
3W02      DEFAULT    16,316    119,266  100    0   <1  100        0       0
          DEFSMALL      956    129,157    0  100    0    0        0       0
3W03      DEFAULT    16,316    197,102  100    0   <1  100        0       0
          DEFSMALL      956    242,876    0  100    0    0        0       0
3W05      DEFAULT    16,316    183,354  100    0   <1  100        0       0
          DEFSMALL      956    206,937    0  100    0    0        0       0
                            ----------
TOTAL                        1,078,692
```

Figure 10-2 RMF Postprocessor output from REPORTS(XCF)

10.3 WLM policies

We recommend that you configure SAP to use the Workload Manager (WLM) to help you manage the performance of your SAP system and achieve good overall system performance for your sysplex. SAP workloads can be managed by the WLM in goal mode, which enables you to assign different levels of importance to the various types of work running in your SAP system. You should also manage your other system components such as VTAM®, IRLM, RRS, and the DB2 address spaces using the WLM in goal mode.

Detailed instructions for defining your SAP WLM policy and parameterizing your SAP system to exploit WLM can be found in the IBM publication *SAP on DB2 UDB for z/OS: Planning Guide (SAP Web Application Server 6.20)*, SC33-7959.

In this section we discuss how you might configure your WLM policy to reflect your sysplex system configuration. Depending on whether you have application servers running under UNIX System Services (USS), application servers running on external UNIX/Linux/NT servers, or a heterogeneous environment consisting of both, you will have to configure your WLM policy differently to achieve the desired results. In a sysplex environment there can be additional considerations. You may want to configure your WLM policy to enable you to give preference to different workloads on different members of the sysplex.

When configuring your WLM in goal mode, follow the recommendations in *SAP on DB2 UDB for z/OS: Planning Guide (SAP Web Application Server 6.20)*, SC33-7959, to ensure that your system-wide settings for your VTAM, IRLM, RRS, and DB2 address spaces are correct. Additional information for configuring WLM can be found in SAP OSS Note 396720. Do not forget to assign the ICLI server address space to a valid service class, because this address space must be able to manage the ICLI threads. We recommend placing it in a service class with a velocity above the SAP threads, but below DB2 address spaces.

When configuring your SAP service classes you must decide, based on your system requirements, which types of SAP workload (dialog/batch/update/spool) will get preference. None of the SAP service classes should have a velocity equal to or above the system address spaces, as this may result in serious performance problems.

When the Central instance is under USS, you should be careful to give higher velocity to the MSGSRV, DISPATCHER, and the ENQUEUE workload service classes. These should be placed above the other SAP workloads, but below the velocity of the DB2 address spaces and the ICLI server if you are running it.

You should be aware of how the ICLI server threads and the SAP USS processes cost the work they do in DB2. These threads connect to DB2 via RRS using cross-memory services, and as a consequence the work they do in DB2 on behalf of the SAP work process is determined by the execution velocity of the enclave they belong to. This is because with cross-memory services, when the ICLI thread or the SAP USS process is connected to DB2, the work performed by the DB2 thread will be controlled by and charged to the TCB (task control board) of the ICLI thread or the SAP USS process. You will therefore see the CPU resources allocated to the ICLI server or the SAP USS threads and not the DB2 address spaces. See Figure 10-3.

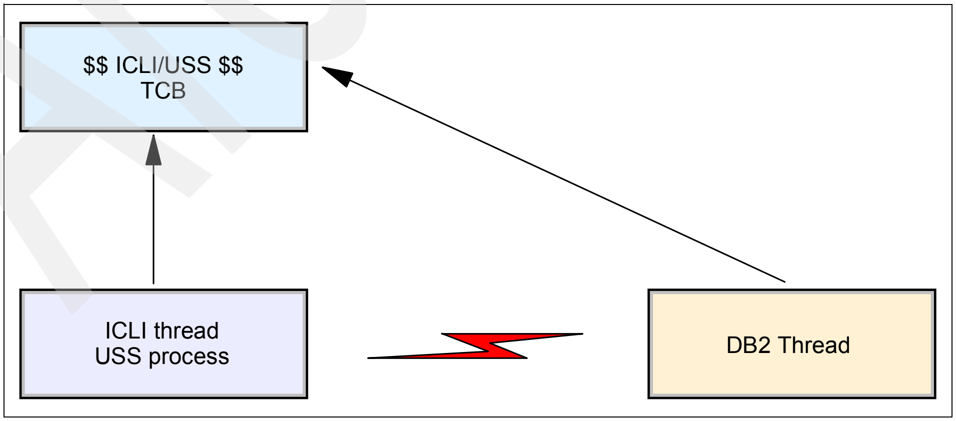


Figure 10-3 TCB of ICLI thread or SAP USS process controls velocity of DB2 thread

Remember that there is only one active WLM policy in a sysplex. However, using the granularity of the WLM service class attributes you can create a sophisticated WLM policy to manage your SAP workload across the sysplex. You can allocate the same SAP workload types from different SAP application servers to different service classes and velocities within the sysplex. By including service classification attribute *Host name*, which maps to the qualifier type SPM, you can create distinct classifications for the same SAP workload types from different SAP application servers.

One example of using this capability would be if you wanted one member of the sysplex to favor SAP batch processing over dialog processing. At the same time, all other members of the sysplex should favor SAP dialog processing over SAP batch processing. You would create a service classification rule for the SAP application server that connects to the member of the sysplex you want to favor batch processing.

Figure 10-4 shows an example of the WLM service classification to achieve this.

```
                    Modify Rules for the Subsystem Type      Row 1 to 1
 Command ===> ____________________________________________  SCROLL ==

 Subsystem Type . : SAP         Fold qualifier names?   Y  (Y or N)
 Description  . . . WLM definition for SAP R/3 6.20

 Action codes:   A=After     C=Copy        M=Move     I=Insert rule
                 B=Before    D=Delete row  R=Repeat   IS=Insert Sub-
                                                              More =
           --------Qualifier--------               -------Class-----
 Action     Type       Name     Start                Service     Repo
                                          DEFAULTS:  SAPAS       SAP
  ____    1  UI        ICLIRUN  ___                  SAPAS       ____
  ____    2    SPM       SAPHOST1 ___                SAPAS       ____
  ____    2    TN        GENERIC  ___                SAPAS       ____
  ____    2    TN        DIALOG   ___                SAPMED      ____
  ____    2    TN        BATCH    ___                SAPLOW      ____
  ____    2    TN        UPDATE   ___                SAPHIGH     ____
  ____    2    TN        UPDATE2  ___                SAPAS       ____
  ____    2    TN        SPOOL    ___                SAPLOW      ____
  ____    1  UI        ICLIRUN  ___                  SAPAS       ____
  ____    2    SPM       SAPHOST2 ___                SAPAS       ____
  ____    2    TN        GENERIC  ___                SAPAS       ____
          2    TN        DIALOG                      SAPLOW
```

Figure 10-4 WLM service classification example

Batch work from SAP application server SAPHOST1 runs in the service class SAPLOW. Batch work from the SAP application server SAPHOST2 runs in the service class SAPMED. If the SAP application server SAPHOST2 is connected to the sysplex member where you want to favor SAP batch processing, then on this member it will run with a higer velocity than the SAP dialog processing.

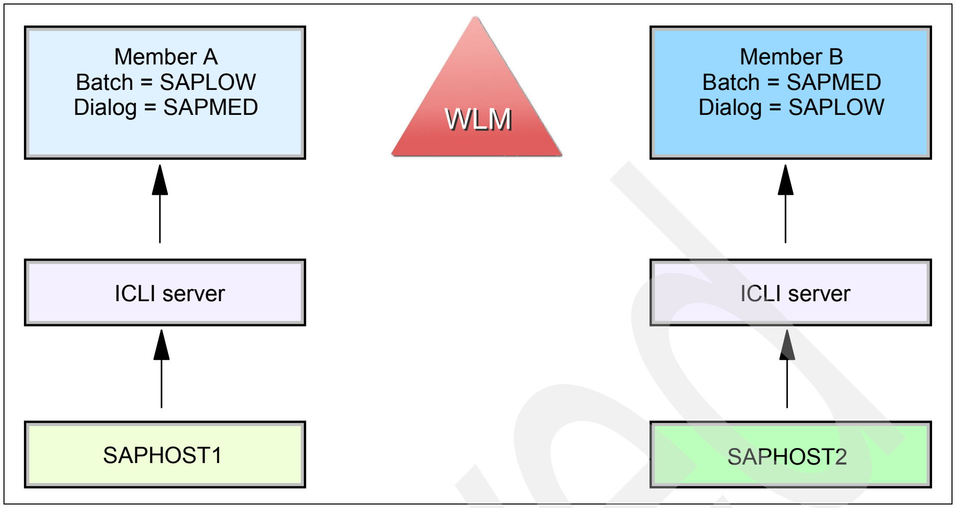


Figure 10-5 Impact of WLM policies in failover situations

If you exploit this granularity in your WLM setup and base it on SAP application server host names, then you should beware of the implications in a failover situation. In Figure 10-5, if SAPHOST2 fails over to another member of the sysplex, then batch work from SAPHOST2 will run with the same velocity as dialog work. You may want to prepare a WLM policy that you can use in a failover situation.

10.4 RMF Monitor III - Work Delay Monitor

The RMF Work Delay Monitor, otherwise known as RMF Monitor III, provides very useful reports for identifying delays in the system components that have an impact on your SAP workload performance.

- CF activity overview

  From your ISPF RMF Monitor III menu, select **S SYSPLEX** → **5**

  This displays an overview of your coupling facility activity. Check that the CPU activity is no higher than 50% and that there is free storage.

- CF paths

  From your ISPF RMF Monitor III menu, select **S SYSPLEX** → **6**

  This displays availability and performance of the paths to the CF. Check that all defined paths are available and that the delay % is not too high.

- Delay Report

  From your ISPF RMF Monitor III menu, select **1** → **4**

  This takes you to the Delay Report. A shortcut to this report is to enter `DLY` from the command line on any RMF Monitor III screen.

  Delay Report is a very useful report for identifying delays to your SAP system workload components.

  If any of the DB2 address spaces on the LPAR are suffering delays due to resource shortages, you can see the delays from the Delay Report screen.

  If you are running an SAP application server on the LPAR and you have made your SAP work processes WLM managed (see SAP OSS Note 396720), then you can see an entry for each WLM enclave. If an enclave is delayed you can see the reason for the delay from this screen.

  If you are running your ICLI server as WLM managed (see *SAP on DB2 UDB for z/OS: Planning Guide (SAP Web Application Server 6.20)*, SC33-7959), you will see an entry for each WLM enclave for your ICLI server. The enclaves are doing work in DB2 for the external SAP application servers. If any of these enclaves are delayed you can see the reason for the delay from this screen.

  The two main types of delay you are likely to see are PRC and DEV:

  – PRC

    PRC indicates a delay for processor resources. If you have not set up your WLM service classes correctly, less-important work may be getting preference for CPU resources over and above your DB2 or SAP workloads. CPU processor delays in your ICLI server will result in high database response times for application servers connected to this ICLI server because the SQL executed in DB2 is controlled by the TCB of the ICLI thread. In extreme cases you may also see connectivity problems between your application servers and the database.

    Delays in your SAP central instance will result in poor response time for work running on that instance. If the critical message server, dispatcher, and enqueue tasks are delayed, the result will be poor response time for the whole SAP system.

    One cause of PRC delays is that the LPAR is reaching its allowable CPU limit. Check the PR/SM™ weightings and the CPU activity for the machine. The CPU activity can be checked using the RMF Postprocessor reports described in 10.5, “RMF Postprocessor reports” on page 122. You can also see the machine CPU activity in real time from the RMF Monitor III. To see the machine CPU activity from Monitor III:

    From your ISPF RMF Monitor III menu, select option **1** → **3**

This option shows the CPU activity for the whole machine and for the LPARs defined on the machine.

– DEV

This indicates a delay for a DASD or tape. For your DB2 address spaces, you may see DASD device delays. This can be caused by any of several reasons. The most common is that there are too many active tablespace data sets on a single volume. This can be related directly back to your SAP database response times. If your SAP single record statistic analysis or your SAP SQL trace analysis identifies high access times for a table, this may be caused by device delays in DB2 for the volume.

In the DB2 system overview you might also see high I/O suspend times.

You can investigate further by selecting the resource device delay report:

- From your ISPF RMF Monitor III menu, select **3** → **2**

  This shows the volumes causing delays to the current workload. You can get further detail on the data sets that are being accessed on the volume.

- From your ISPF RMF Monitor III menu, select **O** → **4** → **Report=DSNV**

  Now enter the volser of the volume you want more detail about.

  Return to the ISPF RMF Monitor III menu.

- From your ISPF RMF Monitor III menu, select **3** → **3B**

  This displays the detail data set delay report for that volume, showing which data sets (and therefore which tablespaces) are on the volume that are being accessed.

Device delays can be reduced with the Parallel Access Volume facility (PAV). PAV, in either dynamic or static form, can make a big difference in terms of IO response times by significantly reducing IOSQ time while queuing on the unit control block (UCB). PAV involves creating aliases for the UCB to reduce queuing and to push it down where possible to the DASD controller. Use of PAV can significantly reduce (but not eliminate) the need for careful data set placement.

If you cannot use PAV then you should consider reducing IO contention by careful data set placement. Even if you have implemented PAV you should still think about controlling data set placement to achieve optimal performance and to use your DASD resources to their full potential. Data set placement can be achieved by implementing SMS policies to ensure separation of data across volumes.

Another way to reduce IO delays is to partition the tablespace to spread the I/O across volumes. Even if you have PAV you should still consider controling the data set placement for the partitions by using SMS policies or manual methods. Keep in mind that OS/390 z/OS queues I/O activity at the UCB for the volume, despite the fact that you may be using new technology DASD hardware.

10.5 RMF Postprocessor reports

RMF Postprocessor reports provide historical perfomance data for the SAP data sharing environment. This section highlights some of the reports that can be used to analyze performance problems in an SAP data sharing environment. This is not an in-depth discussion of RMF reporting but, for SAP basis consultants who want to investigate performance in a data sharing environment, it is useful to be aware of the additional OS/390 and z/OS reporting facilities.

Futher documentation on how to run these reports and read the output can be found in the RMF Report Analysis manual for your operating system release.

- CPU Activity Report

  This report is useful for investigating CPU resource delays in your system. It can provide historical information about logical and physical processor utilization in your syplex.

- Coupling Facility Activity Report

  This report provides useful information about each coupling facility attached to your sysplex. If your previous investigations lead you to belive that performance problems are due to pool data sharing performance, this report provides useful information about the activity in the coupling facility.

- XCF Activity Report

  This report provides information about the activity of a cross-system coupling facility. This is useful if you supect that poor performance may be due to poor data sharing performance.

- Device Activity Report

  In this report, it is important to ensure that PAVs are being properly distributed as needed. Either dynamic or static form can make a big difference in terms of I/O response times by significantly reducing IOSQ time while queuing on the UCB queuing.

  PAV involves creating aliases for the UCB to reduce queuing and to push it down where possible to DASD controller. Use of PAV can significantly reduce (but not eliminate) the need for careful data set placement.

10.6 RMF Spreadsheet Reporter

RMF Spreadsheet Reporter is a set of applications comprised of Excel macros that can be downloaded to your workstation. RMF Spreadsheet Reporter extracts information from a server mainframe RMF Postprocessor report and presents this data graphically. This tool can be downloaded for free from:

http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/rmftools.htm

RMF Spreadsheet Reporter consists of four applications:

- Collector
- Extractor
- Converter
- Several spreadsheets

After installing the product, you might need to customize the provided skeleton JCL in C:\RMFPP\PROGS\rmfpp.jcl. In some cases, you may want some customization that is not part of Collector.

RMF Spreadsheet Reporter enables you to submit an RMF Postprocessor job from your workstation to a server mainframe through TCP/IP.

Collector

This module enables you to provide all information to build the job using the skeleton JCL mentioned above. You can choose the date and time, duration interval, and the type of reports. At job submission time, it provides two options for receiving the listings: Immediate and Deferred.

- Immediate

  Immediate starts your job and waits for its conclusion. If you are reading SMF buffer data from RMF, your job likely will end in less than 10 minutes as it reads data that requires no sorting. Be careful, because the default FTP value for JESPUTGETTO is 600 seconds. This parameter specifies how long an FTP-job-submitter can wait on the job until it is finished.

- Deferred

  Deferred is more appropriate when you need to read from SMF tape files and sort it. Your job will be submitted, and when it ends you can have the output sent back to your workstation via FTP.

Extractor

After the result set from your batch job is sent to your workstation, you can run the Extractor. It creates a Report-Work-Set from the dowloaded file.

Converter

This takes the output from the Extractor and converts the RMF reports to a spreadsheet format.

Spreadsheets

The Excel macros are the core of RMF Spreadsheet Reporter. There are several type of macros, but for our tuning purposes we recommend these:

- Summary report: A graphically represented summary report for CPU utilization and DASD I/O rate
- DASD activity report: Graphically represented detailed I/O report
- Workload Activity Trend in goal mode report: Graphically represented detailed report on Service Class and more
- Coupling Facility trend report: Very useful, graphically represented report on CF utilization, number of requests, service times for all LPARs, percentage of changed request due to path-busy, subchannel-busy, detailed data for all CF structures and more

RMF Spreadsheet Reporter is a great jump forward in assisting the whole mainframe server (such as z/OS and DB2) in the areas of creating baselines, tuning, and tracking performance.

11

Data sharing performance

This chapter discusses DB2 data sharing performance and monitoring issues that are relevant to SAP implementations in a sysplex environment. A well-configured data sharing environment is the basis for good performance in an SAP system. In this chapter, we describe:

- What to measure: defining a baseline
- Local buffer pool performance
- Global buffer pool performance
- Dynamic SQL Cache
- Lock performance
- DB2PM statistics batch report
- Tablespace partitioning
- SAP update table partitioning
- Configuring for query parallelism
- Managing DBM1 virtual storage
- Configuring DB2 sort pool and workfiles
- DB2 backup policies: avoiding disruptive backups

11.1 What to measure: defining a baseline

Having a baseline is very important. This is the only way to ensure that improvements derived from DB2 system tuning can be quantified.

When we tune buffer pools our final objective is to reduce the number of pages read in or I/O traffic. In addition to being quantifiable, I/O savings can be compared to MIPS consumption through CPU time savings. Buffer pool hit ratios may be good performance indicators, but I/O traffic is a better tracking metric.

We have used such a baseline for trend reporting as well. The suggested baseline can be a 24-hour period updated daily. At the beginning of the week, this baseline, which can be in a spreadsheet, can be published to all involved and to management. It can be a powerfull decison support tool.

The baseline can be comprised of:

- Total number of steps
- Total number of dialog steps
- Average database response time
- Total number of synchronous I/O reads for all local buffer pools
- Total number of pages read sequentially for all local buffer pools (pages read via seq. prefetch + pages read via list prefetch + pages read via dynamic prefetch)
- Global and Local Dynamic SQL Cache hit ratios
- Global locking contention rates: Real and False
- Average CPU consumption

The first three items can be obtained directly from the SAP application, with transaction ST03. The next three come from a DB2PM Statistics report, and the last two items can be obtained from an RMF Postprocessor CPU report.

In large environments, it can be time-consuming to browse different performance reports on a daily basis and act proactively. This kind of spreasdsheet gives an idea of the overall health of our DB2 systems.

For instance, any 20% increase in I/O traffic (sync or sequential) for more than a day or two warrants a more detailed investigation, even if the average DB reponse time is not being affected.

This baseline is also good for spotting changes in the application behavior that you may not be aware of otherwise. For example, sometimes new SQL challenged code is promoted into production without DB2 systems group being made involved. This can be detected by the increased I/O traffic caused by this change.

11.2 Local buffer pool performance

One of the most important points when designing DB2 buffer pools for an SAP environment is to group similar objects by to their characteristics, such as:

- Large workset, highly updated, random access
- Small workset, highly updated, random access
- Large workset, little updated, random access
- Small workset, little updated, random access
- Highly updated, sequential access and
- Little updated, sequential access

The guidelines for such buffer pool design are well documented in *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server 6.20*.

When tuning DB2 buffer pools, the primary goal is to reduce or eliminate I/O by finding needed pages in the local buffers and minimizing synchronous single page I/O.

The recommendations for buffer pool tuning are geared toward local BPs with a mixed load of random and sequential pages.

You can find the lowest I/O rate for a buffer pool through an estimated page residency time.

Plotting buffer pool miss rates against buffer pool size results in a queuing (exponential) curve. The objective is to avoid getting below the arc of the cursor, and the recommendation for virtual pools (vpools) plus hiperpools is to get you well beyond the arc and onto the flat part of the curve. The two general rules are:

- Pages should be resident in the virtual pools for at least a minute.

  or

- Pages should be resident in the combination of virtual and hiperpools for approximately 15 minutes.

Take several reasonably large, typical sample intervals of 15 minutes each and perform these two calculations:

- Vpool residency = (# of vpool buffers / hiperpool writes per second)

  The result should be approximately 60 seconds. This formula can tell how long a page can be in a virtual pool before it is written to a hiperpool.

If the average page residency is much greater than 60 seconds, you can cut back on vpool buffers and give more to hiperpool buffers (at least two hiperpool buffers for each vpool). To get the hiperpool writes, add up the synchronous and asynchronous data mover hiperpool writes. This result tells you the number of pages moved out of the vpools by the least recently used algorithm.

- Total (vpool + hpool) residency = (# of vpool + hpool buffers / pages read per second).

  This combined result should be approximately 15 minutes, assuming that you have vpools and hiperpools.

  If you are not using hiperpools, just calculate residency by using the formula:

  `vpool residency = (# of vpool buffers / # of pages read)`

  `Pages read` means the total number of pages read via sequential prefetch + number of pages read via list prefetch + number of pages read via dynamic prefetch.

11.3 Group buffer pool performance

In this section we discuss:

- Caching pages in the group buffer pool (GBP)
- Calculating GBP hit ratios
- Castout efficiency

11.3.1 Caching pages in the GBP

There are mainly two requirements for a page to be cached in the GBP:

- The pages must have been updated in a local buffer pool.
- There must be inter-DB2 interest in the page, and at least one member has read/write interest.

When a page set or partition becomes GBP-dependent, all updated pages in the local buffer pool are moved to the corresponding GBPs. Updated pages in the local pool are not always written synchronously to the GBP. Prior to Version 8, all GBP page writes were CF synchronous operations. Pages were written synchronously with respect to the application using *force at commit* protocol at commit. Pages were written asynchronous to the application using CF synchronous operations due to VDWQT/DWQT thresholds being reached, or system checkpoint becoming due, or as part of page p-lock negotiation. Updated pages were also written synchronously during index leaf page split. In V8, DB2 will use new WARM CF commands (asynchronous CF command, multiple pages). Index leaf page split will also exploit WARM, and the number of GBP writes will be reduced to two.

Changed pages can be written to the GBP before the updated transaction is commited when:

- One of the local buffer pools thresholds (DWQT or VDWQT) is reached.
- The local buffer pool is short of available pages.
- A system checkpoint is issued. Such an event can clean out a local buffer pool before the commit.
- The same page is required for update by another system. This page writing into the GBP is part of page P-lock negotiation.

When such pages are are written to the GBP, all copies of those pages that are cached in other members' buffer pools are invalidated. This process is called cross-invalidation (XI). The next time the application references those pages, they must be loaded from the GBP (or DASD) in a synchronous request.

Another main point to consider is *directory reclaims*. Group buffer pools contain directory entries that are used to maintain information about all pages in all local buffer pools.

Thus, having suffcient directory entries in the GBP optimizes cache structure usage.

However, without suffcient directory entries in the GBP, a new reference to a page in a local buffer pool requires that the oldest, least recently used directory entry in the GBP be reclaimed, provided that the old entry is not associated with a dirty or changed page in the GBP.

11.3.2 Calculating GBP hit ratios

With the above information in mind, the formula to calculate group buffer pool hit ratio could be:

```
(SYN.READS(XI)-DATA RETURNED / (SYN.READS(XI)-NO DATA RETURN +
SYN.READS(XI)-DATA RETURNED))
```

SYN.READS(XI)-DATA RETURNED means that a synchronous CF read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid. The page existed in the group buffer pool and was returned.

SYN.READS(XI)-NO DATA RETURN means a synchronous CF read was issued because a local virtual buffer pool or hiperpool had a page marked as invalid and no page was found in the group buffer pool. The recommended hit ratio is 70% or higher, provided that the GBP has no problems with directory entries being reclaimed due to cross-invalidations.

Another recommendation is to check periodically for cross-invalidations due to directory entry reclaims.

Automate the issuance of the command every four hours:

```
DIS GBPOOL(*) GDETAIL(INTERVAL) TYPE(GCONN)
```

Figure 11-1 shows an example of how the response to the DISPLAY GROUPBUFFERPOOL command can be recorded in the MSTR job log.

```
DSNB788I  -D3WP   CROSS INVALIDATIONS
                DUE TO DIRECTORY RECLAIMS               = 0
```

Figure 11-1 GROUPBUFFERPOOL command display

11.3.3 Castout efficiency

Castout is the process of flushing pages from the group buffer pool to DASD. Keep in mind that there is no physical connection between the CF where group buffer pools reside and DASD. Therefore, the castout process involves reading the page from the group buffer pool into DB2's private buffer (not part of the buffer pool storage) and flushing the page from the private buffer to DASD.

Castout occurs when:

- The number of updated pages for a castout class queue exceeds a class threshold (CLASST) value.
- The total number of updated pages for a group buffer pool exceeds a group buffer pool threshold (GBPOOLT) value.
- The group buffer pool checkpoint is initiated.
- There is no more inter-DB2 read/write interest in the page set.
- The group buffer pool is being rebuilt, but the alternate group buffer pool is not large enough to contain the pages.

Group buffer pool class castout threshold (CLASST)

Each group buffer pool contains a fixed number of castout class queues. DB2 internally maps updated pages that belong to the same page sets or partitions to the same castout class queues. Because the number of castout class queues is limited, it is possible that more than one page set or partition will be mapped into the same castout class queue. This internal mapping scheme is the same across all sharing subsystems.

When DB2 writes changed pages to the group buffer pool, it determines how many changed pages are on a particular class castout queue. If the number of changed pages on a specified castout class queue exceeds the threshold, DB2 casts out a number of pages from that queue.

The default group buffer pool class castout threshold is 10, which means that castout is initiated for a particular page set or partition when 10 percent of the group buffer pool contains changed pages for the class.

Group buffer pool castout threshold (GBPOOLT)

This threshold determines the total number of changed pages that can exist in the group buffer pool before castout occurs. The default group buffer pool castout threshold is 50, so when the group buffer pool is 50% full of changed pages, castout is initiated.

You can enforce the trickle writing of data by lowering the CLASST and GBPOOLT thresholds to 1 percent and 20 percent, respectively.

When monitoring group buffer pools, make sure that CLASST is being hit much more often than GBPOOLT.

Unlock castout

The UNLOCK CASTOUT counter should always be signicantly less than the PAGES CASTOUT counter. If It is not, then the castout write I/O is not being done efficiently. Look at the PAGES CASTOUT and UNLOCK CASTOUT fields in a DB2PM Statistics report. DB2 usually includes more than one page in the lock request to write pages to DASD. Therefore, UNLOCK CASTOUT should be less than or equal to PAGES CASTOUT. It will be much less if multiple pages are written per I/O. This ratio is a good indicator of how well the castout process is performing.

The thresholds for GBPCHKPT, GBPOOLT, and CLASST prior to DB2 V8 were too high. In V8 they are set at 4, 25, and 5, respectively.

We recommend aggressive monitoring of messages such as DSNB319A/325A/228I/327I (out of storage in GBP).

We also recommend use of XES Auto Alter if DB2 APAR PQ68114 is applied as an effective co-req.

XES Auto Alter has two distinct and maybe competing algorithms (in this order):

1. Structure Full Avoidance
2. (Entry/Directory) Reclaim Avoidance

FULLTHRESHOLD considerations (assuming current structure size is less than SIZE) are:

- If both number of elements (DB2 pages) and entries (directory) exceed FULLTHRESHOLD, then the structure size can be increased. If either one or the other (but not both) is above threshold, structure size is NOT increased and “juggling” between entries and elements occur.
- Recommend settings:
  - SIZE=INITSIZE+30%
  - FULL THRESHOLD=90
  - MINSIZE=INITSIZE.

11.4 Dynamic statement cache

The local dynamic statement cache (DSC) contains the user (thread) copy of the prepared SQL statements and is kept in the DBM1 address space. The ZPARM MAXKEEPD controls maximum size across all threads and is expressed as number of statements. It is enforced as each user thread reaches commit. Coupled with KEEPDYNAMIC=YES, user copy is preserved across commit boundary.

Our recommendation is to set MAXKEEPD initially at 12000 and monitor the local dynamic cache hit ratio closely. As this may consume virtual storage within DBM1, we suggest lowering it to 8000. By doing so the workload will be pushed to the global dynamic statement cache.

The global dynamic statement cache contains the skeleton copy of prepared SQL statements and is held in DBM1 inside the EDM pool or in an EDM data space. It is implemented by CACHEDYN=YES.

We recommend moving the global dynamic cache to a data space. This saves considerable storage within DBM1. We also suggest reviewing APAR PQ73624 before moving the global dynamic cache to a data space.

If your monitor does not provide local and global dynamic cache hit ratios, use these formulas:

- To calculate the local DSC hit ratio:

  ```
  PREP_FROM_CACHE / (PREP_FROM_CACHE + IMPLICIT_PREPARES)
  ```

  Local DSC hit ratio should be greater than 90%.

- To calculate the global DSC hit ratio:

  `(PREP_STMT_CACHE_INSERTS - PREP_STMT_CACHE_REQUESTS)/PREP_STMT_CACHE_REQUESTS`

  The global DSC hit ratio should be greater than 95%.

Information about Dynamic Cache hit ratios can be found in the DB2PM Statistics report, shown in Figure 11-2.

```
DYNAMIC SQL STMT            QUANTITY  /SECOND  /THREAD /COMMIT
--------------------------  --------  -------  -------  ------
PREPARE REQUESTS             7110.00    11.85    28.55    2.00
  FULL PREPARES              1766.00     2.94     7.09    0.50
  SHORT PREPARES             5344.00     8.91    21.46    1.50
GLOBAL CACHE HIT RATIO (%)     75.16      N/A      N/A     N/A

IMPLICIT PREPARES               0.00     0.00     0.00    0.00
PREPARES AVOIDED                0.00     0.00     0.00    0.00
STMT INVALID (MAXKEEPD)         0.00     0.00     0.00    0.00
STMT INVALID (DDL)              0.00     0.00     0.00    0.00
LOCAL CACHE HIT RATIO (%)        N/C      N/A      N/A     N/A
```

Figure 11-2 DB2PM Statistics report: DSC hit ratios

11.5 Lock performance

Reducing the amount of locks taken by an application improves the lock performance. In a data sharing environment, locking tuning should be done both at local and global levels.

In general, locking problems can be prevented by coding frequent commits into the applications to release the locks and avoid contention. Problems usually arise in user-written programs that do not have an appropriate commit frequency.

A high number of updates and deletes with few commits can accumulate a high number of locks. Currently IRLM can manage up to 8,000,000 locks concurrently. When this threshold is reached, additional requests for locks will terminate abnormally with SQL code -904, resource unavailable.

We recommend coding `PC=YES` in the IRLM address space. This will allow up to 8 MB locks. DB2 V8 no longer supports PC=NO. IRLM V2R2 will run in 64-bit mode and support more locks.

11.5.1 DB2 locking mechanisms

DB2 provides mechanisms for limiting the number of locks that are concurrently held. This is achieved using the system parameters NUMLKUS and NUMLKTS, and the tablespace attribute LOCKMAX.

The system parameter NUMLKUS sets a limit on the number of locks that any individual DB2 thread can hold. When that limit is reached the program that accumulated these locks terminates with sqlcode -904. The maximum value for NUMLKUS is 2097152 (make sure that DB2 APAR PQ52651 has been applied) and it is recommended as an initial, first-cut value. Setting a lower value for NUMLKUS helps detect offending programs earlier, and it is especially recommended for test systems. Nevertheless, in most production systems (except Retail component) a lower value for NUMLKUS is acceptable, but it should not be lower than 500,000.

The system parameter NUMLKTS sets the default for the tablespace attribute LOCKMAX.

If the number of locks on a particular tablespace exceeds the LOCKMAX value, these locks will be replaced by a single tablespace-scope lock (note that the LOCKMAX is enforced on a per-thread, per-tablespace basis). This process is called lock escalation, and it can eliminate the cases of exhausting the IRLM and CF LOCK structures.

On the other hand, lock escalations do not necessarily address all lock contention problems. They can lead to deadlocks but it is very likely that the deadlock victim (the process that needs to back out) is a process for which the backout is least expensive. This addresses the issue of long backouts caused by the IRLM or CF LOCK structure exhaustions.

Another challenging aspect of lock escalations is in coming up with an optimal value for LOCKMAX. This is very customer-specific and depends on the particular workload and available central storage resources. If set too high LOCKMAX allows the transactions and reports to hold too many locks, which leads to the problems described earlier. If set too low, it causes too-frequent lock escalations. For example, with too-low LOCKMAX, it might happen that an application process that acquired more than the LOCKMAX locks triggers a lock escalation, which in turn could not be executed successfully due to other applications holding non-compatible locks. After the timeout period the escalation triggering application process would receive a -913 timeout and it would need to roll back. Had the LOCKMAX value been higher, the application might have completed without lock escalation.

With NUMLKUS set to 2097152, we recommend an initial, first-cut value of 1000000 for LOCKMAX. This value can be adjusted by the customer depending on the site-specific considerations and NUMLKUS setting described above.

The current SAP databases are installed with either LOCKMAX=0 (meaning lock escalation is disabled) or LOCKMAX=3000000. That will change in the next SAP release (6.20 and above), where the tablespaces will be created with LOCKMAX=1000000. For the existing installations identify your critical tablespaces and ALTER them by changing LOCKMAX accordingly. Alternatively, you can alter all of the SAP tablespaces to, for example, LOCKMAX=1000000.

There are a number of ways to find these critical tablespaces even before they show themselves during abnormal terminations due to IRLM or CF storage shortage conditions.

- Monitor the maximum number of locks that are held by individual DB2 threads (transaction ST04, Thread Activity, Locking Activity). Identify those threads that show a large number in that field and find out which transactions and reports have been serviced by the associated work processes. The transaction SM66 displays which tables are being accessed by which work process.

  If DB2 PM is available, you can use the Accounting Trace Exception reports to identify such DB2 threads more easily. You should use this monitoring technique not only to find out which tablespaces should have their LOCKMAX adjusted, but also to identify the transactions and reports that requested and held so many locks. The best practice is to change these applications (e.g. by inserting commits if the application logic allows it) to reduce the number of locks held. For SAP-written programs open an SAP problem message and for the user-written programs talk to the application developers.
- Monitor fields Executions, Getpages, Rows Processed and Rows Examined by the individual update, delete, and insert statements reported in the ST04 Cached Statements Statistics panels. This could give you a hint which tablespaces need the special attention.
- Monitor the number of changes per table in the ST10 transaction. Again, it can serve as a hint where to go next.
- Monitor alerts (SAP Alert Monitor) raised due to encountering long-running, non-committing units of recovery. They do not necessarily hold a large number of locks, but they are definitely worth investigating.

Apart from enabling lock escalation, you should ensure that the IRLM and CF are sized to maximize the number of locks that can be concurrently held.

For the IRLM, specify `PC=YES` in the IRLM start up procedure and set the IRLM region size to zero.

11.5.2 Global locking

Conceptually all locks taken in a data sharing group are global locks. However, not all locks must be propagated to the LOCK structure in the coupling facility.

The number of locks that must be propagated to the CF LOCK structure for a page set or partition is determined by the number of DB2 members interested in the page set and whether their interest is read or write. The CF LOCK structure consists of two parts:

- Lock table. Also known as hash table, this is made up of many lock table entries (also called hash classes). When a lock require by a member needs to be made known to other members, it is registered in the Lock table.
- IRLM relies on the System Lock Manager (SLM) component of z/OS (also called XES) Inter-system locking services. Various IRLM lock levels can ONLY map to one of two XES lock levels. IRLM IS and S locks map to XES -S lock. IRLM U, IX, SIX, and X locks map to XES-X lock.

  For example with two members holding IX locks on same table space. Both IRLM IX locks map to XES-X locks - hence lock conflict. XES detects contention and global lock contention processing invoked by IRLM to finally determine that IX is really compatible with IX and to grant lock request. This can be a problem with RELEASE(COMMIT) as used by SAP. There are some changes in DB2 V8 which will help. Firstly, IRLM will now maps IX L-locks to XES S and will grant IX parent L-locks locally when only IS or IX L-locks held on the object.

  Parent L-locks still sent to LOCK structure but never cause contention for common conditions. IX remains incompatible with S because S parent L-locks must now map to XES X. There will be the additional overhead of global contention processing to verify that a pageset S L-lock is compatible with another pageset S L-lock. But this is a rare case.

  The majority of cases are IS - IS, IS - IX, and IX - IX and hence there will be overall performance benefits. Secondly, Child L-lock propagation will no longer based on the parent L-lock. It will be based on cached (held) state of pageset P-lock. If pageset P-lock negotiated from X to SIX or IX, then child L-locks are propagated. This will provide reduced volatility. If P-lock not held at time of child L-lock request, child lock will be propagated.

  Parent L-locks no longer need to be held in retained state after DB2 failure e.g., a pageset IX L-lock no longer held as a retained X lock. This is an important availability benefit.

Locking components

In a data sharing environment, lock information is held in three components:

- IRLM

  IRLM contains the most detailed lock information.

- XES

  XES is an important locking component. All lock information passed to XES is stored in XES. XES recognizes only two types of locks: S and X.

- CF LOCK structure

  The CF lock table also recognizes only two types of locks: S and X.

Lock contention types

There are three types of lock contentions:

- False contention

  False contention takes place when the same lock table entry is used to represent more than one resource. This inconsistency happens when the lock table is not sized properly and the number of lock table entries is inadequate. The global lock manager XES comes into play by getting information from all other XESs in the group that have locks for the resource.

- XES contention

  If XES determines that false contention does not exist, it checks whether XES contention applies. XES only recognizes two types of locks modes: X and S. If an XES tries to register an X-mode lock and this type of lock is already registered by another member, it has to check with IRLMs to find out exactly what type of lock IRLM requires, as it could turn out that IRLM really needs an IX-mode. At this point XES passes control to the IRLM contention exit to resolve the conflict. The exit checks the real need of IRLM and if it is not a real contention, it is called XES contention.

- Global contention

  Global contention includes all types of contention.

When contention (XES, false, or global) occurs, one of the XESs is known as the global lock manager to resolve the contention.

Throughout false and XES contention, there is extensive XCF signalling traffic between IRLMs to resolve the conficts. This emphasizes the need to have XCF message traffic well-tuned, as mentioned in 10.2.4, “CF signalling path length monitoring” on page 115.

Monitoring lock contention rates

Global contention rate under 3% is great. A consistent Global contention rate above 5% warrants an investigation.

False contention rate should not be more than 1.5% of all requests. The final goal is to minimize XCF signalling traffic to resolve lock conflicts.

To calculate Global and False contention rates we recommend that you use RMF and not DB2PM because the new z/OS 1.2 algorithm that converts synchronous CF requests to asynchronous CF requests has also caused DB2 Instrumentation Facility to inflate the False Contention rate.

To calculate Global and False contention rates run RMF Postprocessor with option SYSRPTS(CF). In the COUPLING FACILITY STRUCTURE ACTIVITY section, look for the LOCK structure data as shown in Figure 11-3:

```
REQ TOTAL     2933K (A)
-CONT          456K (B)
-FALSE CONT     88K (C)
```

Figure 11-3 Coupling facility LOCK structure activity

REQ TOTAL represents the number of sync and async requests to this LOCK structure.

CONT represents the number of lock requests that resulted in contention.

FALSE CONT represents the number of requests that resulted in false contention.

To calculate Global contention rate, the formula is B/A. In the example above, it would be 15.5%.

To calculate the False contention rate, the formula is C/A. In the case above, that would be 19.3%.

A noteworthy comment is that the above formula will assist you if you have a uniform workload across all members of the group. However if that is not the case, you might need to check Global & False contention on a LPAR basis.

Monitoring LOCK structure storage

To help monitor the efficient use of CF LOCK structure storage, you can use the -DISPLAY GROUP command as shown in Figure 11-4 on page 139.

```
SCA   STRUCTURE SIZE:  80128 KB, STATUS= AC, SCA IN USE: < 1 %
LOCK1 STRUCTURE SIZE: 120064 KB
NUMBER  LOCK ENTRIES:  33554432 (A)
NUMBER  LIST ENTRIES:    174921 (B),LIST ENTRIES IN USE:1887(C)
```

Figure 11-4 DISPLAY GROUP command showing the CF LOCK structure

(A) is the maximum number of lock entries possible for the lock table, (B) is the maximum number of modify lock list entries, and (C) shows how many of those list entries are in use.

The coupling facility LOCK structure size limits how many locks can be held concurrently. CFLEVEL 12 provides a certain relief because it enables the LOCK structure to reside beyond the previous storage limitation of 2 GB.

Storage allocated by the LOCK structure is comprised of two parts:

- Lock hash table

  This is also called the lock table or the hash table. The hash table is where DB2 members register interest in a resource. The hash table is comprised of many table entries or hash classes. A lock needed by a DB2 member is registered in the hash table when that member needs to make it known to other members in the group. If this hash table is too small, locks might be represented by one single value. This introduces False Contention, as two locks for two different resources might be represented by the same value.

- Record list entires (RLE).

  Also known as Modified Lock List (MLL). DB2 members of a data sharing goup store run-time information about modified locks owned by each member in the RLE. The information contained in the RLE is used to create retained locks when DB2 member abends. Therefore, if a lock is to be preserved across a subsystem abend, it must be stored in the RLE. The maximum number of modify locks related to database update processing that can be concurrently held and propagated to the CF is limited by the size of the RLE section. When 90% of the space allocated for the RLE is used, new modify lock request will abnormally terminate with -904. As in the IRLM storage exhaustion, long and disruptive backouts would follow.

For the CF LOCK structure, the sizing considerations are more complicated.

The actual number of modify locks that can be held in a given RLE section size is dependent on the CF level and the level of OS/390. For more details, see 0.0.1, “Coupling Facility Level (CFLEVEL) Considerations” at:

http://www.s390.ibm.com/pso/cftable.html

Starting with z/OS 1.2, IBM has introduced a new z/OS algorithm to decide whether to convert synchonous requests to asynchronous requests. This new z/OS function monitors CF service times for all structures (CACHE, LIST, and LOCK) and, based on new internal thresholds, determines which requests would be more efficiently executed asynchronously. Thresholds are different for simplex and duplex requests and for Lock and non-Lock requests. (For more details, review Washington System Center Flash 10159 at:

http://www-1.ibm.com/support/docview.wss?uid=tss1flash10159

11.6 DB2PM statistics batch report

DB2PM reports come usually with two layouts, short and long, at a member or group level.

We should first concentrate on the long version of the report and, when we are familiar with its capabilities, we can then switch to the short version.

The member-level report header displays information such as location, group name, member ID, subsystem ID, and time interval.

In the HIGHLIGHTS section, besides information about date, time interval, and which member ID is being reported on, the important displays are the number of GETPAGES and synchronous pages read.

The EDM POOL section presents detailed information about EDM POOL utilization. It is important to ensure that there is at least 20% available space to accommodate unforseen workload growth.

The LOG ACTIVITY section helps monitor how well logging activity is performing. Pay special attention to the UNAVAILABLE OUTPUT LOG BUFF field. If it has any number but zeros, it may be time to increase OUTBUFF in the zparm member.

The DYNAMIC SQL STMT section gives the global and local cache hit ratios.

DBM1 STORAGE STATISTICS (MB) section helps monitor DBM1 virtual storage consumption.

The sections on member buffer pool information assist in tuning and monitoring the buffer pool configuration.

The next sections have information about group buffer pools at a member level.

At the group level, a DB2PM header displays location, group name, and the interval.

If using z/OS 1.2 and later, you may have to disregard the DATA SHARING LOCKING section. DB2PM is wrongly inflating False Contention Rate and they have announced dropping support for this item. You may find False Contention Rate information accurately displayed in RMF.

Read 11.3, “Group buffer pool performance” on page 128 for the necessary information on a group level to tune your group buffer pool configuration.

These reports can be quite lengthy, however DB2PM is flexible enough to allow customizing the reports to eliminate unnecessary fields or a whole section. You can also change the order in which sections and fields are presented.

To customize a report you may need to copy member DGOJINIT from the *.SDGOSAMP library into your personal JCL library.

Next, customize the DGOJINIT member by locating the high_level_qualifier field and updating it with your local DB2PM high-level qualifier.

You will have to allocate a PDS library with LRECL=80, BLKSIZE-6400.

Then, go to TSO option 6 and issue the EX MY.DSN(DGOJINIT) command. A window will pop up with six options that represent the various reports that can be customized.

Place the recently created PDS library in the DPMPARMS data set field. Now you can start changing the DB2PM reports layout.

The new layouts will be saved in the library specified in the previous step, and you must insert it into the DB2PM execution JCL in ddname DPMPARMS.

By customizing DB2PM reports you can have a better presentation and save space in the JES output queues.

11.7 Tablespace partitioning

In this chapter we discuss reasons for using tablespace partitioning in an SAP data sharing system. Detailed information about partitioning tablespaces can be found the *DB2 UDB for z/OS Administration Guide* for your DB2 release. SAP also has a database administration tool that can be started with SAP transaction code SE14. This tool can be used to partition tablespaces. Documentation about the SAP tool can be found in the SAP manual *SAP R/3 Database Administration Guide: DB2 UDB for OS/390*, which is available for download from the SAP Web site (which requires a password for access) at:

http://service.sap.com/instguides

11.7.1 Partitioning to improve tablespace management

In both non-data sharing and data sharing environments you may have to partition large tablespaces due to their size and predicted growth. In both of these environments, partitioning a tablespace makes it manageable for housekeeping utilities such as Reorg, Runstats, and Image Copy.

11.7.2 Partitioning to avoid I/O contention

In addition to the data management and housekeeping issues already mentioned, one of the main driving forces to implement partitioning is to avoid I/O contention for heavily used tables. This is achieved by partitioning the data and placing it on separate I/O paths and volumes.

You can choose a partitioning key that evenly spreads your data around the partitions in a linear manner. This type of partitioning often uses a random artificial key column. For example, in the SAP core banking solution a Global Unique Identifier (GUID) is generated in the key of critical tables. This key is used to partition, and results in even spread of rows across the partitions. You can then benefit from reduced I/O contention on the partitioned tables.

When choosing this type of linear partitioning key, make sure you create enough partitions to hold the predicted growth in data without the partitions becoming too big. You should also consult your application developers to ask if there is any benefit from clustering the data in any particular ascending or descending order using another column in the key. Note that in DB2 V8 the partitioning key must also be the clustering key.

When choosing linear partitioning you should benefit by reducing your I/O contention by spreading your data. However, you may not be able to control member affinity for the batch jobs or transactions accessing the data. This may result in cross-member interest in the group buffer pools. This cross-member interest can be reduced by using certain tablespace settings:

- MAXROWS=1

  SAP uses row-level locking, with lock escalation for selected tablespaces. If you have some small tables that may have a lot of cross-member insert interest (which can often be the case with SAP), then you may be able to reduce cross-member interest by using the MAXROWS=1 option in your tablespace. You review the table and its use carefully before making this change. MAXROWS=1 allows only one row per data page, which significantly reduces the possibility of data page P-lock contention during insert. Documentation on this parm can be found in the DB2 SQL reference manual for your release of DB2.

- MEMBER CLUSTER

  If multiple batch jobs that are running on different members are updating the same clustered pagesets to make inserts, then they will get increased P-lock negotiation for these pagesets. To help reduce this, the MEMBER CLUSTER option for the tablespace definition results in each member getting a different space map page, so that each member is inserting into different areas of the tablespace. This results in significantly reduced P-lock negotiation. The disadvantage is that the data is no longer clustered; however, you have to decide whether this is acceptable in return for reducing insert times. If time and resources allow it, you can always reorganize the tablespace after the inserts to cluster the data again, ready for reading. Documentation for this parm can be found in the DB2 SQL reference manual for your release of DB2.

- TRACKMOD=NO

  Space map contention between members can be reduced farther by using the tablespace option TRACKMOD=NO. This stops DB2 from inserting rows into the space map pages for GPB-dependent objects, and therefore reduces P-lock contention for these space map pages. These space map pages are used to optimize the incremental image copies. The consequence is that incremental image copies will run slower because they now have to scan the pagesets to identify changes that have been made. Documentation for this parm can be found in the DB2 SQL reference manual for your release of DB2.

Another method of partitioning is to choose a partitioning key that results in each partition being filled in turn as the data is created. With this type of partitioning you may be able to parameterize your jobs to operate separately on each partition. This is because the key may based on an account number, a customer number, or a processing period.

When you consider the separation of partitions to different I/O paths and volumes, you may wonder why this is necessary if you have the latest disk hardware. The reason this is still important is that OS/390 and z/OS still control access to the volume at the UCB level. There will be queuing and contention at the volume UCB. If the Parallel Access Volume enhancement (PAV) is implemented, then this contention will be reduced significantly.

In an SAP environment, if you do not have PAV you can control the placement of the partitions you create with your SMS policies. SAP DB2 objects have a clearly defined naming convention. For more information about the SAP database layout and naming standards we refer you to the SAP publication *SAP R/3 Database Administration Guide: DB2 UDB for OS/390*, which is available for download at:

http://service.sap.com/instguides

An example of the SAP-defined naming convention for DB2 objects can be seen in Figure 11-5 on page 144.

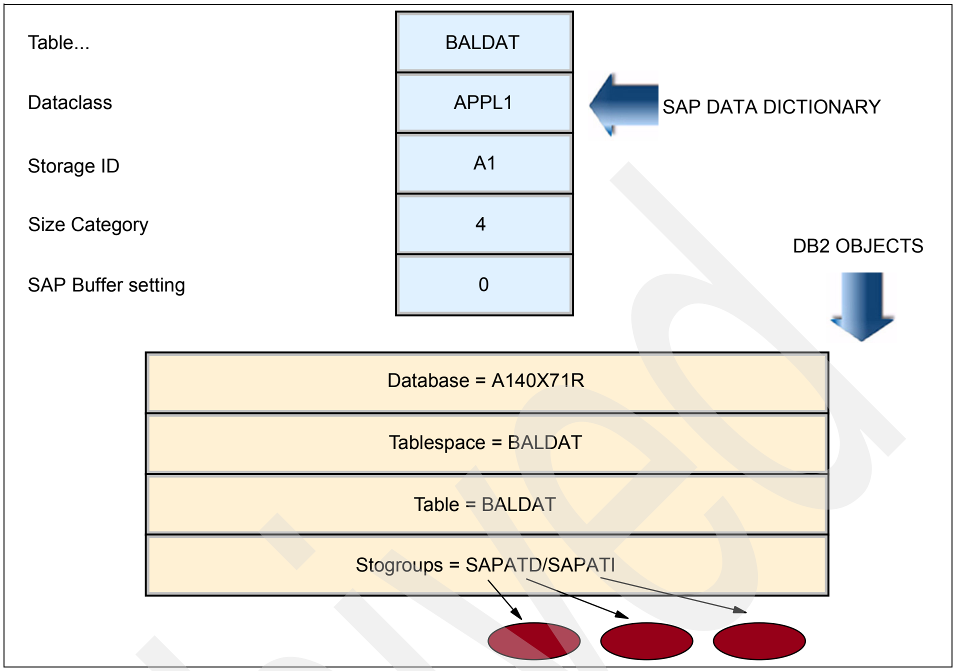


Figure 11-5 SAP naming convention for DB2 objects

Figure 11-6 shows how the DB2 database name is constructed from the SAP data dictionary information for a table.

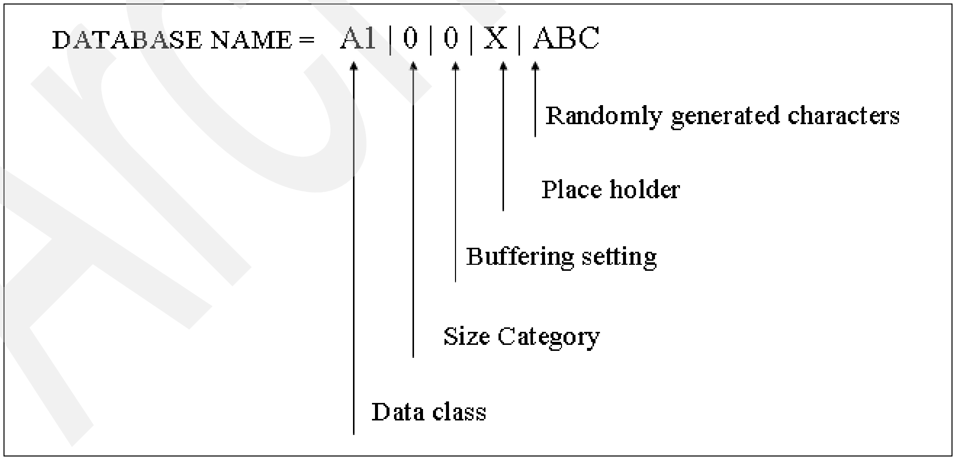


Figure 11-6 DB2 database name

Figure 11-7 on page 145 illustrates the VSAM data set name that results from the SAP data dictionary definitions. When partitioning you can use this naming convention to place the partition pieces for the tablespace in index on the desired volumes.

```
Command ===>                                                          Scroll ===> PAGE

Command - Enter "/" to select action                      Message            Volume
-------------------------------------------------------------------------------------
               BCXYT.DSNDBC.A140X71R.BALDAT.I0001.A001                          *VSAM*
               BCXYT.DSNDBD.A140X71R.BALDAT.I0001.A001                          ADD100
```

Figure 11-7 VSAM data set naming convention for SAP data sets

11.7.3 Improving I/O performance for nonpartitioning indexes

If your partitioned tablespace also has a nonpartitioning index or indexes, these could be an I/O bottleneck if you have concurrent SAP processes making a lot of inserts, deletes, or updates into the table.

You may see high insert and update times in the SAP single record performance statistics (transaction STAD). To reduce the insert and update times for your applications you should consider using the PIECESIZE option for these nonpartitioning indexes. This is especially relevant to nonpartitioning indexes that support large tables, but it can help performance on any partitioned or nonpartitioned table. Figure 11-8 illustrates how a nonpartitioned index can be a performance bottleneck.

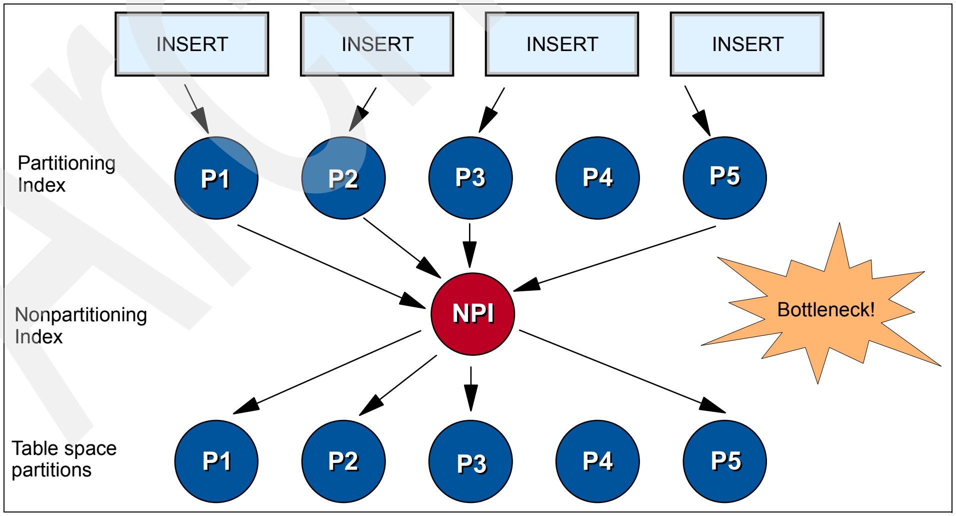


Figure 11-8 Nonpartitioning index bottleneck without PIECESIZE option

By using the PIECESIZE option you can split up your nonpartitioning indexes into several physical data sets. These data sets can then be placed on separate volumes and I/O paths to reduce the I/O contention. This helps balance the I/O across the data sets of the index, as illustrated in Figure 11-9.

For further considerations about choosing a PIECESIZE, consult the DB2 SQL reference manual for your DB2 release.

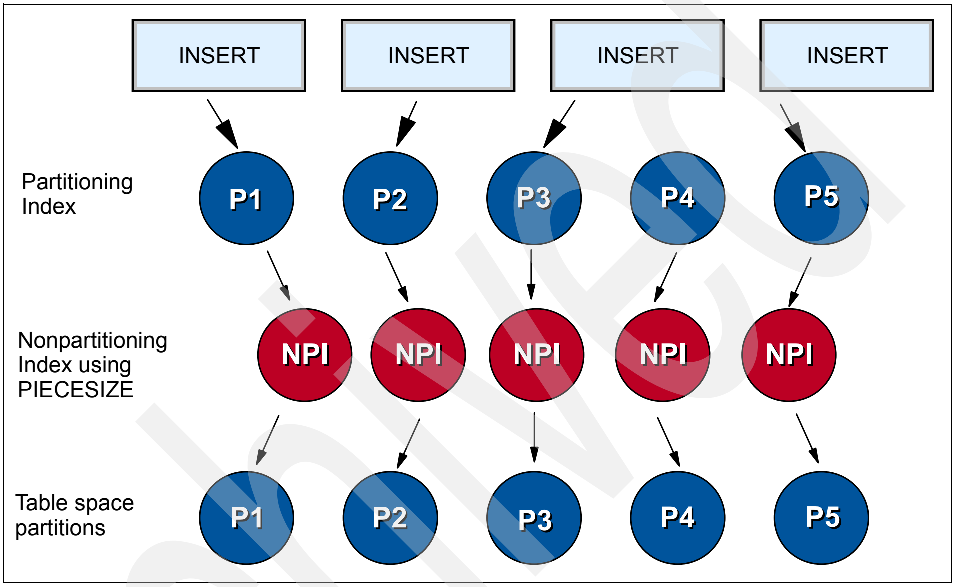


Figure 11-9 Nonpartitioning index with PIECESIZE option

11.7.4 Partitioning to distribute workload around the members

In addition to reducing I/O contention, with careful partitioning you can also distribute your online and batch workload around the data sharing members with the use of affinity routing. Affinity routing of users to the member whom you want to be able to access a particular partition avoids the overhead of lock propagation and group buffer pool dependency. Figure 11-10 on page 147 illustrates how affinity routing can work in an SAP data sharing environment.

When choosing a suitable set of tables and a suitable partitioning key you must consult your application support team. To find the tables that are used by your online users and batch jobs you can analyze the SAP single record performance statistics using transaction STAD. If the SAP system parameter stat/tabrec is set on, then these records show you which tables are used by the job. stat/tabrec = 5 is a good value to start with. An SQL trace of the transaction/batch job using SAP transaction ST05 can also be done to determine which tables are being accessed.

With this method of partitioning, after you have defined your partitions you can benefit from affinity routing for your batch jobs, buffer chosen partitions in the local buffer pools on different data sharing members, and then parameterize your jobs to process data from individual partitions. The jobs can then be scheduled on application servers connected to the data sharing member that has that partition in its local buffer pools.This avoids intersystem interest and enables you to direct your workload.

This second method of partitioning is also applicable to online activity if the partitioning key is related to a group of users. Once again you can benefit from affinity routing of users to data sharing members (for example, users who work with a specific company code or who work with a specific cost center). After you have defined your partitions, using SAP logon load balancing you can direct users to the application servers connected to the data sharing members that hold the partition with their data in the local buffer pool. This method avoids intersystem interest with the data, as illustrated in Figure 11-10.

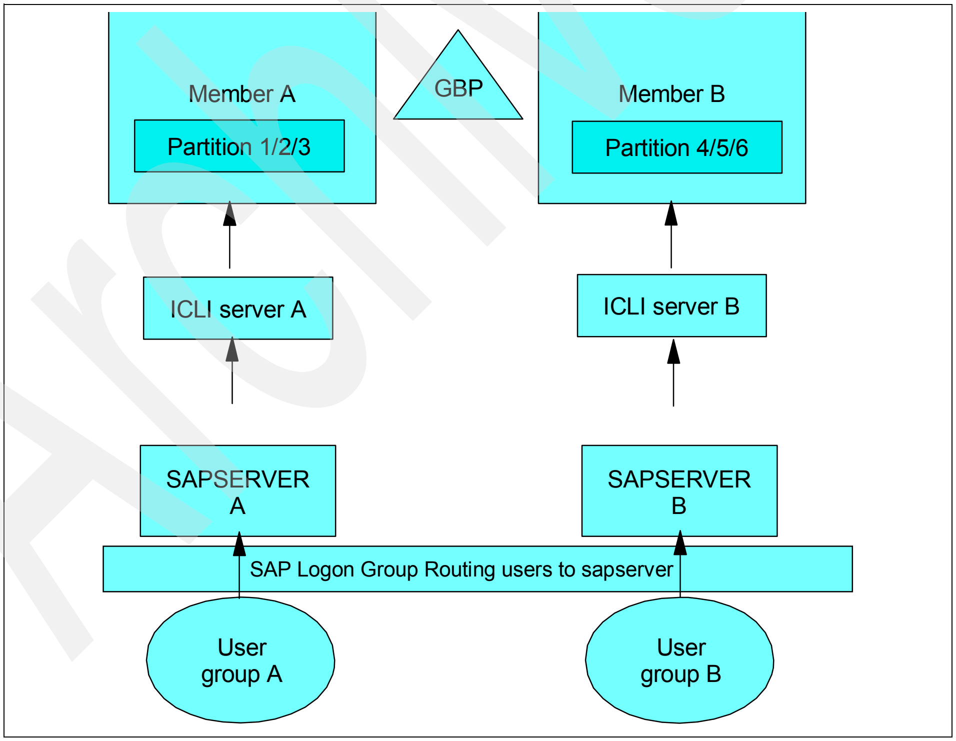


Figure 11-10 Using affinity routing tor SAP workloads

11.8 SAP update table partitioning

SAP R/3 has a particular technique that is used to effectively de-couple a dialog (online user) process from the actual updating of tables in the system. This effectively allows an SAP logical unit of work (SAP LUW) to span multiple database logical units of work (DB2 LUW) without forcing the transaction to commit its work at an inappropriate time.

This technique is most commonly called the SAP Update Protocol (or VB Protocol after tables involved), and it is effected by a set of database tables as well as work processes (application server tasks). In the case of data sharing, the activity that results requires some proactive tuning activity to minimize the effects of cross-system interest in the tables involved. This process is described in detail in the SAP document *DBA Guide for SAP on DB2 for OS/390*.The information in this book should be read in conjunction with this SAP document.

The required tuning is a combination of SAP Basis and DB2 activity. The outcome of the tuning is that for the tables VBLOG, VBDATA, and VBHDR (hereafter referred to as the “VB Tables”), a partition is created corresponding to each SAP application server in the system. All new or changed records in the VB tables caused by dialog work process activity on any application server result in records being inserted into or updated in the VB tables for the partition specific to that application server.

Subsequently, the records created by the application server will be read and deleted by other processes, called update work processes. These are configured so that the update work processes for a group of application servers attached to a DB2 member of a DB2 data sharing group execute only in application servers attached to this DB2 member. In this way all activity on pages in any given partition will be limited to application servers attached to that data sharing member, hence cross-system interest is eliminated.

11.8.1 Analyzing the effectiveness of update table partitioning

After the table partitioning of the VBHDR, VBMOD, and VBDATA tables has been accomplished as discussed, a few techniques can be used to determine whether the result is effective. One is to issue a DB2 display DB(*xxxxx*) LOCKS against the database, where *xxxxx* represents the database name containing the tablespaces. This should indicate no cross-system interest.

Each partition and its associated tablespace should be opened by one DB2 data sharing member, so checking open data sets for each member should reveal that only the appropriate partitions are in fact open on the members.

A notable exception in a data sharing environment is when transaction SM13 is used to examine the update protocol tables. Using SM13 to look for broken or stalled updates in an SAP system is normal and considered good practice. However,using the transaction results in all partitions being read by one data sharing member (the member used by the application server where SM13 is run). The use of SM13 in an active data sharing environment where update table partitioning is in place should be limited to times when activity is minimal. This results in cross-system interest on the tables being restricted to a short period of time. The associated locks will naturally be removed, and the data sets for partitions not normally opened will be pseudo-closed.

11.9 Configuring for query parallelism

SAP can benefit from query parallelism for read-only queries. To enable SAP DB2 systems to choose parallelism, use the following settings:

- Set DSNZPARM PARAMDEG

  For example, set this to be one less than the number of processors in your system.

- Set DSNZPARM CDSSRDEF

  Set this value to 1 to enable SAP dynamic SQL to dynamically enable parallelism with the SET CURRENT DEGREE = ANY statement.

- Work Files

  Make sure that the number of work files you have is at least as large as the number of parallel tasks you plan for.

- Use the PIECESIZE feature to get parallelism against nonpartitioning indexes.
- Configure your buffer pools to ensure that you have adeqate resources to parallelism. VPPSEQT = 0 turns off all parallelism.

You can can monitor for the degree of parallelism you are achieving using the following facilities:

- The EXPLAIN function in SAP or from DB2 can be used to analyze the degree of parallelism you get in an SQL statement.
- The DB2 command DISPLAY THREAD displays parallel tasks. The status field contains PT for parallel tasks immediately after originating thread.
- The DB2 command DISPLAY BUFFERPOOL(*) DETAIL also provides information about whether parallelism is occuring for the buffer pool contents.

The following IFCIDs can also help to monitor parallelism:

- Statistics trace record
  - IFCID 2 System statistics
  - IFCID 202 Buffer pool attributes
- Accounting trace record
  - IFCID 3 Accounting trace
- Performance trace record
  - IFCID 201 Buffer pool alter
  - IFCID 221 Parallel group execution trace
  - IFCID 222 Parallel group elapsed time
  - IFCID 231 PArallel operations
  - IFCID 237 SET CURRENT DEGREE trace

11.10 Managing DBM1 virtual storage

For many years, the 31-bit addressability of MVS, and then OS/390, has required various techniques for tuning, to work around the fact that only 2 GB (actually, somewhat less than this) is available to an address space. This has led to features such as Dataspace and Hiperspace™, which are exploited by DB2 for OS/390 and z/OS as a means of mitigating the effect of the 2 GB addressability limit. In this chapter we discuss the use of these facilities specific to an SAP environment.

There are three major contributing environmental factors:

- How much central (real) storage is available in the LPAR?
- How many DB2 subsystems are run in this LPAR?
- What other applications will require memory resources in this LPAR?

The overriding factor in (real and virtual) storage management is the initial condition of how much storage is allocated to the LPAR in consideration. Obviously where there are multiple LPARs involved, there is also a consideration of how much storage to allocate to each LPAR from the physical memory available on the S/390® or zSeries server. The allocation choice is somewhat beyond the scope of this book, but the general philosophy that follows (that of making the optimum use of available resources by moving resources to the source of greatest benefit) also applies to the question of LPAR storage.

Another issue to be considered is how much storage will be consumed by non-DB2 tasks in this LPAR. It is widely recommended by SAP and IBM that production SAP systems should reside in an LPAR that contains only production, as opposed to development, testing, or other activities. It is possible that some of these other tasks will be unrelated to the SAP system's DB2 subsystem; that is there may be other non-SAP production work running in the LPAR. The storage requirements of such tasks are beyond the scope of this book. If the requirements are known, they can be effectively subtracted from the available storage to be allocated. For non-production systems, there are several possibilities for the number of LPARs and how many and what types of SAP systems will be run in each LPAR.

System tasks also consume a significant amount of memory resource. The requirement for actual real storage, as opposed to virtual storage, depends on how much memory the system tasks need, which is not paged to auxiliary storage due to inactivity of the pages.

Finally, after taking into account all other possible consumers of resources in the LPAR, we have an idea of how much storage we can allocate to the OS/390 or z/OS resident components of an SAP system. These components are the DB2 subsystems and possibly OS/390 application servers. When running multiple DB2 subsystems for different SAP systems, the allocation of memory between them will be a factor of the workload to be run within the subsystems and the performance levels expected in the systems.

Therefore, the process of tuning starts with an understanding of the available resources in the system in question. Also, you need to understand the concept of Virtual Storage constraint, which can ultimately lead to subsystem failure through getmain failures, as opposed to Real Storage constraint, which can lead to performance problems through paging and auxiliary storage shortages. The process of tuning that follows these understandings can then take place in a structured manner, leading to a proposed first configuration, and then subsequent refinement of this configuration through controlled changes and measurement. With a knowledge of how much real storage DB2-related tasks are able to occupy, we can then decide which of the available virtual storage constraint relief facilities it is appropriate to use.

SAP Note 162923 also provides a discussion of Virtual Storage constraint relief and tuning, and it should be read in conjunction with this section when considering and performing tuning.

11.10.1 Managing SAP work processes

The approach that should be taken with the number of work processes is one of minimization. That is, the less that can be configured to achieve the required throughput, the better the overall result will be. One of the key reasons is that DB2 consumes a finite amount of storage for each thread it has, and each SAP work process causes a DB2 thread to be created. The thread storage allocated is in the DBM1 address space, which is shared with many other resource types, and hence reduces the space available for these other resources.

Not having enough work processes of a given work process type to perform the workload results in SAP reporting the performance impact as *wait time* in ST03 statistics. Depending on the type of work process, this may or may not be a major issue. In a case where the application server wait time has some other issue contributing to the problem (for example, it is CPU-constrained or there is some other underlying problem causing poor performance), adding more work processes may not be the answer. This action may simply serve to move the statistical time to other areas, while the performance remains the same.

If the requirement for more work processes occurs at a particular time, it may be possible to address the problem using SAP operation modes to reallocate existing work processes to alleviate the problem, without necessarily creating more work processes.

11.10.2 Monitoring work process utilization

Consider reducing work processes if:

- After the SAP application server has been active for some time, and when you click the **CPU** button in SM50, there are work processes with zero or close to zero CPU time.
- The ST03 statistics for this work process type indicate minimal wait time during a suitable period of time.

If both of these criteria are met, it may be possible to reduce the number of work processes and closely monitor the results by adjusting the appropriate rdisp/no_wp_*xxx*, where *xxx* is the work process type:

- DIA - Dialog
- BTC - Batch
- UPD - Update
- UP2 - Update2
- SPO - Spool

Consider increasing work processes if:

- ST03 indicates significant wait time (< 5-10% of total response time).
- Using SM50 and clicking **CPU** indicates that all work processes have non-zero CPU time against them.
- SM50 CPU stats indicate that no more work processes can be stolen either permanently or through SAP operation modes.

Additionally, ensure that the performance of the existing workload is optimal and that there is no underlying performance problem causing the workload in question to occupy work process for longer than necessary. This can also consist of CPU (or other resource) overload on the application server. In this case, adding more workload to an overloaded system will have no benefit, and in fact probably reduces overall throughput.

As discussed in the previous Virtual Storage analysis, the number of threads in DB2 is a key factor in how much storage is used in DBM1 address space, so minimizing the threads allows allocation of these resources to other pools, and that improves the overall performance of the subsystem.

11.10.3 Managing the DSC size

With the advent of the EDM Pool data space available from DB2 V6 and subsequent releases, the process of managing DSC resources has been greatly simplified. The EDM Pool data space is enabled through DSNZPARM EDMDSPAC, and it defaults to 40MB if an EDMPOOL size greater than 40 MB is specified. By allocating DSC pages in a data space, the allocated pages no longer have the detrimental effect on the DBM1 address space of consuming a large number of EDM Pool pages to achieve adequate Dynamic SQL performance.

SAP relies heavily on the performance of Dynamic SQL, so effective caching is extremely important, as it reduces the number of times DB2 has to consume the resources required to perform a full prepare. If an EDM Pool data space is utilized, which is highly recommended, then it is also recommended to make the EDM Pool data space large enough to contain all prepared Dynamic SQL statements. This is because it is probably more resource-effective to page in a DSC page related to the data space from auxiliary storage than it is to prepare the statement again.

The number of pages used for DSC can be monitored through IFCID *nnn*, which is easily accessible through transaction DB2/ST04: first click the **DB2 Subsystem** button, and then the **EDM Pool Activity** button on the following screen. This is shown in Figure 11-11 on page 154. Divide the number of pages by 1024 to relate the displayed number of pages in use to the EDMDSPAC parameter.

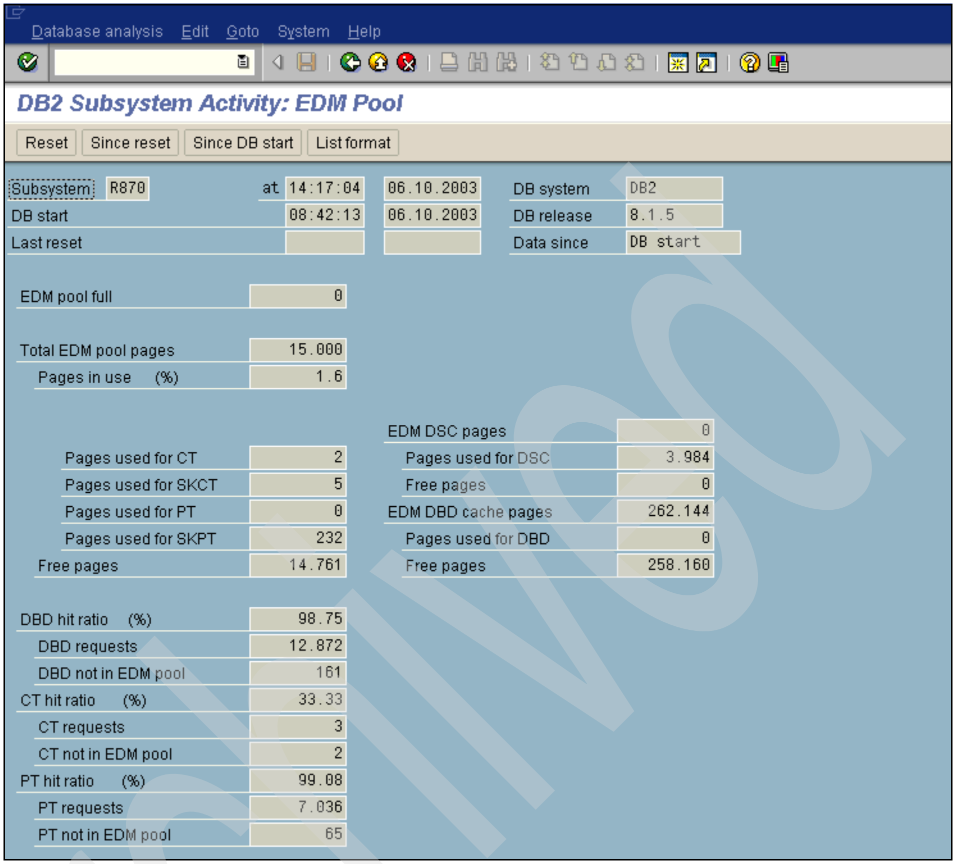


Figure 11-11 ST04/DB2 EDM activity

An approach to finding an appropriate value for EDMDSPAC is to initially set it high, for example 300-400 MB, and periodically monitor the number of pages used for DSC during a time of normal activity, such as taking daily snapshots. The number of pages should eventually stabilize at a peak value and, subsequently, reducing EDMDSPAC to 5-10% more than this will result in an appropriate level.

The most appropriate time to check the number of DSC pages used is immediately prior to shutting down the SAP system following a significant period of uptime. It should be noted that within the statement cache, SQL statements exist that probably will not be executed again prior to DB2 stopping, such as initial reads of SAP buffered tables. However, things like application server restarts may result in them being run again, and the minimal cost of having them paged out to auxiliary storage is considered a small price to pay for possibly avoiding DB2 prepares.

By having all SQL statements in DSC cache, there are also some benefits to be gained by IFCID 318 monitoring of DSC statement performance as discussed in 12.4.2, “Dynamic Statement Cache analysis” on page 176, as any given statement can subsequently be located in the cache.

11.10.4 Managing open data sets

The number of open data sets in a given SAP system is somewhat outside the control of a DB2 System Administrator or SAP Basis Administrator, and it is primarily a result of the SAP modules or components employed in the SAP system. Generally, the target is to allow DB2 to open the required data sets through normal SAP imposed activity, that being to satisfy the flow of SQL requests ariving in the form of Dynamic SQL.

Periodically some other activity may occur in the system that results in other data sets being opened that are not usually used in the normal course of SAP activity. These can include:

- SAP Client Copy being executed. An SAP Client Copy is a process that replicates rows in a significant number of tables, and this typically results in data sets being opened that normally remain closed.
- DB2 utilities such as RUNSTATS being performed, which again may be run against tablespaces that normally remain closed, as the Administrator is performing a complete job by updating statistics on all tables.

11.10.5 Monitoring storage

DB2 Version 7 introduced a new feature to evaluate the use of virtual storage by the Database Services Address Space, which is commonly known as the DBM1 address space. IFCID 225 was introduced, allowing monitors using the Instrumentation Facility Interface (IFI) to obtain a more accurate picture of actual virtual storage utilization than calculations for the ZPARMs affecting storage utilization. The DB2 monitoring functions built into transactions ST04 and DB2 enable this monitoring to be performed in SAP.

Figure 11-12 on page 156 shows an example of this transaction. In this case the subsystem is DB2 Version 7, and the SAP system is SAP R/3 Enterprise with a 6.20 SAP kernel.

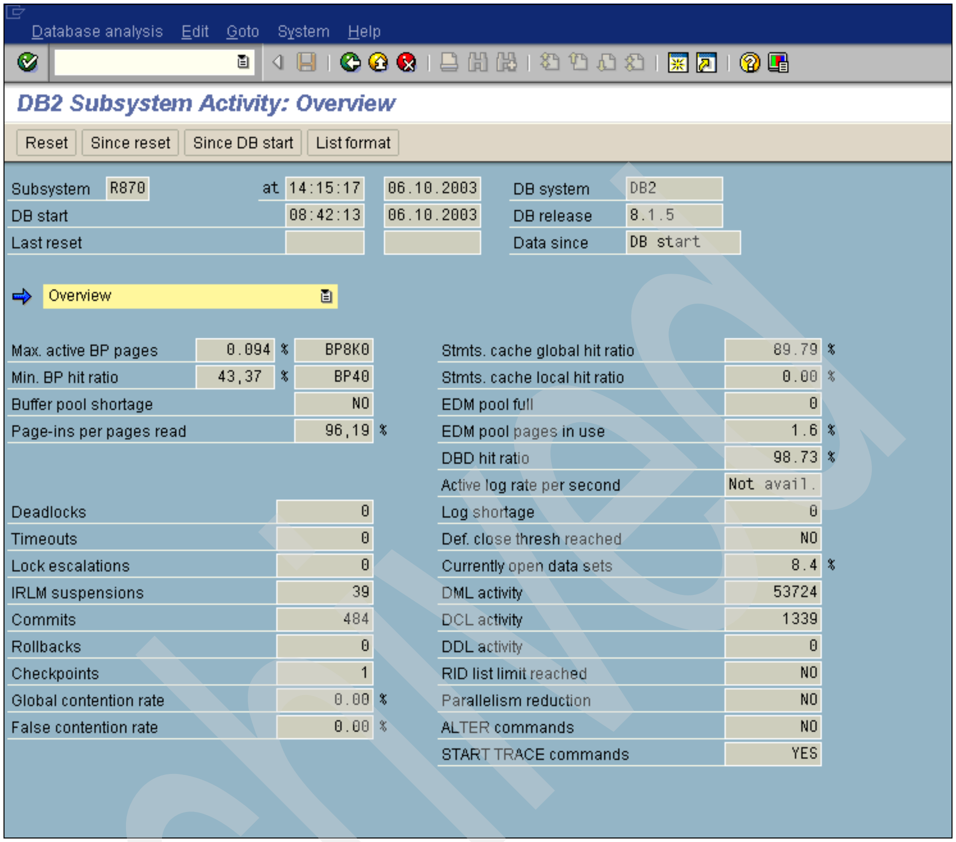


Figure 11-12 Transaction DB2, Subsystem Activity, Virtual Storage

You can see both the overall storage usage of the address space and the components that make up this total. This should closely resemble the estimate of virtual storage usage made through performing the calculations on active DB2 parameters. It is valuable both to validate the current snapshot and to monitor the trend of virtual storage utilization. In most cases, the main objective is to ensure that an adequate storage buffer is maintained to prevent an outage caused by DB2 Subsystem failure in the case of a getmain failure.

11.10.6 SAP-related enhancements in DB2 Version 7

In this section we describe the DB2 V7 functions that are beneficial to SAP R/3 on OS/390 and z/OS.

- Asynchronous INSERT preformatting

  DB2 improves the performance of insert operations by asynchronously preformatting allocated but unformatted data pages. When a new page is used for an insert, that page is close to the end of the formatted pages, and allocated but unformatted space is available in the data set—DB2 preformats the next range of pages.

- UNION and UNION ALL operators

  The scope in which UNION and UNION ALL operators can be specified has been expanded. The CREATE VIEW statement, the INSERT statement, the UPDATE statement, the DECLARE GLOBAL TEMPORARY TABLE, nested table expressions in a FROM clause, and the subquery predicate are changed to allow a fullselect where a subselect was used in previous releases.

  SAP BW exploits UNION in views.

- DB2 Restart Light

  In data sharing environments, the new LIGHT(YES) parameter of the START DB2 command lets you restart a DB2 member in light mode. Restart-light mode means that a DB2 data sharing member restarts with a minimal storage footprint and then terminates normally after DB2 frees retained locks.

  Restart-light mode is intended for a cross-system restart in the event of an MVS system failure. The reduced storage requirement makes it possible to temporarily restart a DB2 data sharing member on a system that might not have enough resources to start and stop DB2 in normal mode. Releasing the locks with a minimum of disruption promotes faster recovery and data availability. For example, applications that are running on other DB2 members have quicker access to the data for which the failed member held incompatible locks.

  You can also use restart-light mode in conjunction with the MVS Automatic Restart Manager (ARM). To have a DB2 data sharing member automatically restarted in light mode when system failure occurs, you must have an appropriately coded ARM policy. ARM does not restart the DB2 member again after a light restart is performed; the member terminates normally for the light restart.

- Online system parameters

  In SAP environments utilizing DB2 in a 24x7x52 mode, the need has been growing for online update of the major DB2 system parameters.

With DB2 V7 the new -SET SYSPARM command is introduced to dynamically reload the DSNZPARM load module. All parameters of the DSN6ARVP macro can be changed, and a large number of parameters from the DSN6SYSP and DSN6SPRM macros can be changed as well.

Table 11-1 lists the DB2 subsystem online parameters that are of special interest to SAP, with the recommended values for SAP systems with DB2 V7.

Table 11-1 DB2 subsystem online parameters of special interest to SAP system

ZPARM	SAP recommended value	ZPARM	SAP recommended value
CTHREAD	Site specific	CHKFREQ	10-15 min
EDMPOOL	60000	URCHKTH	1
EDMDSPAC	100000	URLGWTH	100 KB
EDMBFIT	No	Timeout (via MODIFY PQ38455)	600 sec
MAXRBLK	100000	UTIMOUT	3
CONTSTOR	Yes	RETLWAIT	1
DSMAX	6000	NUMLKUS	0
PCLOSEN	5	RELCURHL	Yes
PCLOSET	10 min	SEQCACHE	Seq
DSSTIME	5 min	SEQPRES	No
STATIME	30 min	CDSSRDEF	1
SYNCVAL	30 min	PARAMDEG	Number of CPUs
STARJOIN	N0 (except for BW)		

- Checkpoint frequency in minutes

  With DB2 V7, the checkpoint frequency parameter is enhanced to optionally enable you to specify a range of minutes instead of a number of log records. Both options are available at install time and can be changed dynamically via commands.

  This feature is useful in environments where the logging rate varies. You can maximize system performance by specifying the checkpoint frequency in time to avoid the performance degradation due to many system checkpoints taken in a very short period of time because of high logging rate. We recommend that you set CHECKFREQ to a value between 10 and 15.

- Long-running UR warning

  Prior to DB2 V7, the warning for long-running unit of recovery (UR) was based on the number of checkpoint cycles to complete before DB2 issues a warning message for an uncommitted UR. But the number of checkpoints depends on several factors, which may not include the long-running job.

  With DB2 V7, the warning mechanism is additionally based on the number of log records written by an uncommitted UR. The purpose of this enhancement is to provide notification of a long-running UR that may result in a lengthy DB2 restart or a lengthy recovery situation for critical tables. The warning message is repeated each additional time the threshold is reached.

  The value for written log records in the message is cumulative and indicates the number of log records written since the beginning of the UR. If statistics trace class 3 is active, an instrumentation facility component identifier (IFCID) 0313 is also written.

  The UR log write check threshold is set in the DB2 parameter load module DSNZPARM (DSN6SYSP URLGWTH) at install time. The value may be modified using the -SET SYSPARM command. We recommend that you use URCHKTH = 1 and URLGWTH = 100.

11.10.7 SAP-related enhancements in DB2 V8

DB2 Version 8 includes dozens of changes in SQL, improving family consistency in many cases and leading the way in others. Many barriers that had been limiting customers are now removed, including: using 64-bit memory, providing consistent table and column name lengths, allowing 2 MB SQL statements, 4096 partitions, and three times the log space.

Key performance enhancements deliver better family consistency and run many times faster. Being able to make database changes without an outage, such as adding a partition, is a breakthrough for availability. Improvements in Java™ function, consistency, and integration with WebSphere® make z/OS a much better platform for Java. Expansions to security enable row-evel granularity, helping with the security issues of Web-related applications. Many of these enhancements also help in key vendor applications such as PeopleSoft, SAP, and Siebel.

In this chapter we introduce the main enhancements available in New Function Mode that are applicable to the SAP application. These enhancements are grouped into categories based on the area of impact to the SAP application and are broadly associated to the following categories:

- Architecture
- Usability, availability, and scalability
- Performance

- Tools and administration
- Backup and recovery

This is not a complete list of all DB2 Version 8 features. A more comprehensive view of DB2 Version 8 features can be found in *DB2 UDB for z/OS Version 8 Technical Preview,* SG24-6871. The features of DB2 UDB for z/OS Version 8 that are of particular relevance to SAP are discussed in detail in *DB2 UDB for z/OS V8: Through the Looking Glass and What SAP Found There,* SG24-7088. Some highlights from these Redbooks are included in the following sections.

64-bit virtual storage

OS/390 R10 and z/OS have provided the 64-bit *real storage* addressability needed to scale in real memory addressing. While OS/390 R10 has the ability to run in either 31-bit mode or 64-bit mode on a zSeries, z/OS only runs in a 64-bit mode real storage environment. z/OS V1.2 and later releases provide *virtual storage* exploitation of the addressing range above 2 GB.

Basically, R10 has provided initial z/Architecture™ real addressing support (up to 128 GB of central storage) and the support for 24-bit, 31-bit, and 64-bit applications.

z/OS 64-bit real storage support has provided significant and transparent reduction of paging overhead, now only to disk, and real storage constraint relief for workload limited by the previous maximum support of 2 GB of real storage. The elimination of Expanded Storage support has been handled by z/OS with minimal customer impact while reducing memory management overhead. For more information about 64-bit real exploitation visit the z/OS migration Web site:

`http://www.ibm.com/servers/eserver/zseries/zos/installation/`

In z/OS V1.2, IBM has delivered the initial *64-bit virtual storage* management support. With the new z/OS 64-bit operating environment, now an application address space can have 2 to the power of 64 (or 2^{64}) virtual addresses with backing by real storage as needed. Figure 11-13 on page 161 gives a pictorial representation of the evolution of the memory management from the 24-bit to the 31-bit to the 64-bit support.

DB2 V8 enables existing 31-bit DB2 applications (including those written in Assembler, PL/I, COBOL, FORTRAN, C/C++, and Java), and future DB2 applications to transparently benefit from the 64-bit virtual storage support in DB2. The benefit is derived from the efficiencies with the local availability of data made possible through 64-bit data addressability. DB2 z/OS V8, which exclusively runs in 64-bit mode, can only execute on IBM @serverzSeries hardware running z/OS V1R3 or later. DB2 V6 and V7 already support 64-bit real storage addressing for data space buffers, providing improved scalability and performance in a zSeries processor running in 64-bit real mode. Using 64-bit real provides a significant storage relief for most customers.

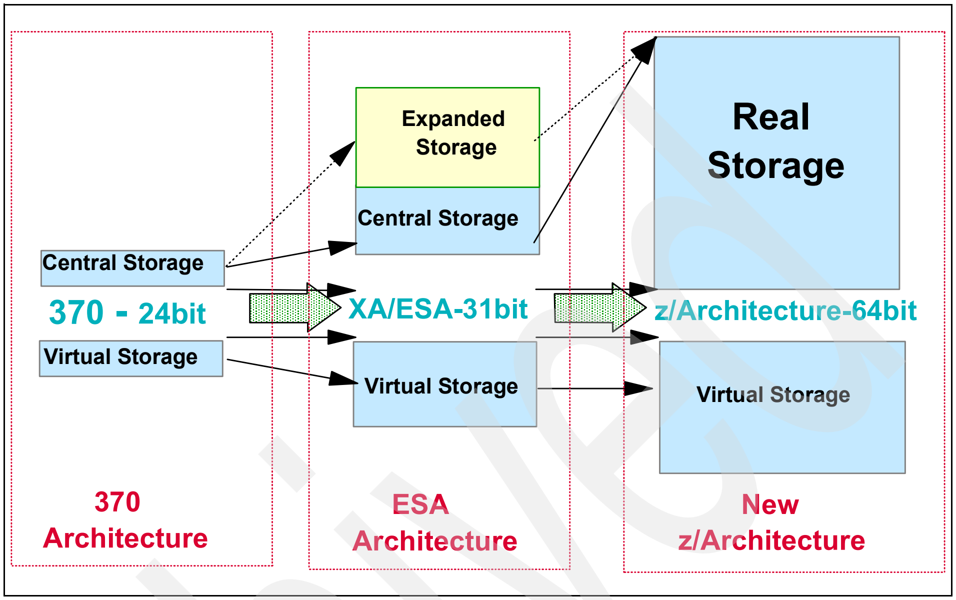


Figure 11-13 Evolution of z/OS memory architecture

Over the years, virtual storage usage has grown dramatically in the DB2 DBM1 address space. This storage growth has been fueled by larger workloads, new functions, faster CPUs, and larger real storage available on mainframe processors. The latter, in particular, has enabled customers to run workloads that in the past would have been limited by paging overhead. Hence, 64-bit virtual addressing largely improves the scalability of DB2.

Automatic space management

Automatic Space Management is enabled with the new DSNZPARM of MGEXTSZ, which can be set to either YES or NO. (The default is NO.) That is, after moving to DB2 Version 8, the new functionality is disabled initially, and space management occurs as it previously did in DB2 Version 7.

When enabled by setting MGEXTSZ, in certain situations DB2 automatically adapts the size of each subsequent secondary allocation requested. It does this in such a way that it follows a sliding scale that guarantees that the object will reach its theoretical maximum data set size (DSSIZE) prior to reaching the maximum number of extents allowed for a VSAM data set of 255.

One of the greatest challenges in administering SAP systems is that every customer varies in the usage of the supplied package. In some cases these differences are great, such as the difference between a customer who installs SAP Enterprise and implements an HR/payroll system, and another customer who implements a Sales and Distribution system. Even customers implementing the same or a very similar set of SAP modules will make choices in the implementation process that result in a different subset of database objects being actively utilized, both in terms of data volumes and activity on this data.

11.11 Configuring DB2 sort pool and workfiles

SAP applications, especially SAP BW, can perform many sorts in DB2. The speed of these sorts is a very important factor in the performance of the application. Therefore you should take care to configure your DB2 sort pool and the work file tablespaces used for sorting for maximum efficiency.

You should configure your zparms SRTPOOL to make your DB2 sort pool the maximum size allowed by your DB2 release. However, remember that this sort pool is not a shared memory area in the DBM1 address space. Each DB2 thread allocates a sort pool of this size if it requires it, so this may have a significant impact on your DBM1 virtual storage.

Next, begin by deciding how many workfile tablespace data sets you can afford to allocate, bearing in mind that each workfile tablespace data set should be placed on its own disk volume and I/O path to avoid I/O contention. We recommend a minimum of five workfile tablespace data sets for each data sharing member. As a starting point these tablespaces should be sized at 2GB and each tablespace should be the same size. You should place these tablespaces on shared disks, because if a query uses sysplex parallelism, the controlling member of the data sharing group may need to read from these tablespaces. Placing the tablespaces on shared disks also keeps a data sharing member connected to its work files if it fails over to another LPAR.

After you have decided how many workfile tablespaces to define, you must place these tablespaces into their own local buffer pool. This buffer pool should not be backed by a group buffer pool. For best use of the sort data sets, the buffer pool should be as large as possible within your DBM1 virtual storage constraints; otherwise it will be a limiting factor in how many sort data sets DB2 can use.

If you are not using a hiperpool when making the buffer pool settings you should set the DWQT and VDWQT thresholds to 90% to avoid writing to the DASD too early. However, if you set these thresholds to these high values you run the risk of flooding the I/O subsystem with large amounts of pages to be written when the threshold is reached. If you see this happening, then consider leaving the thresholds at their lower default values to prevent writing large amounts of pages to the disk when they are reached.

If you are using a hiperpool you should set the DWQT and VDWQT thresholds low to force the pages asynchronously out to disk. After the pages have been written to disk they will be marked clean and therefore eligible for migration to the hiperpool. When they are in the hiperpool they can be brought in during the merge phase of a sort without any I/O.

If I/O does occur in the sort process, then DB2 will use sequential prefetch to bring the pages back into the buffer pool. So if the buffer pool is constrained for sequential prefetch, then sequential prefetch will be disabled and performance will suffer. You should monitor your work file buffer pools to ensure that sequential prefetch is not being disabled.

You should monitor your sort buffer pool activity to determine whether the performance-inhibiting thresholds are being reached. Keep in mind that processing activity in these buffer pools is almost all sequential.

If you are doing large sorts, and after tuning the buffer pool you still see significant I/O activity on your workfile volumes, then you should consider allocating this buffer pool to a data space. This is possible if you are at DB2 Version 6 and above and you are using a 64-bit zSeries machine and OS/390 2.10 or z/OS. Before you use data spaces, ensure that you have enough central storage to hold them. If you can not hold the data space in central storage, then their use is not recommended. If you are not using a 64-bit zSeries machine, consider allocating the buffer pool to a hiperpool.

The best scenario of all is to try to avoid excessive sorts made by the application in DB2. In SAP the appearance of the ABAP statement *order by* after a select from the database results in a sort in DB2. If your organization is developing its own code, try to influence the developers to use a different sort method after the rows have been brought back from the database to the application server. This will transfer the load from the database server to the application server. If the data to be sorted is already in a good sort order then the sorts will be shorter, so check cluster ratios of your data and reorganize them regularly if required.

The efficiency of sorts can be analyzed in the DB2PM SQL activity reports. The sort summary trace record IFCID 0096 can be used to analyze your sorts and is included in performance trace class 3.

For more criteria on which to base the number and size of workfile tablespaces, consult the DB2 administration guide for your DB2 release.

11.12 Backup policies: avoiding disruptive backups

Chapter 9, “Backup and recovery architecture in data sharing” on page 75, discussed backup policies for SAP in a DB2 datasharing environment. In this section we briefly discuss the performance implications of the various backup methods. When analyzing performance issues, be aware of the performance implications of the DB2 backup method that your DBA chooses for your SAP data sharing environment. All DB2 z/OS and OS/390 SAP environments (and not specifically data sharing environments) present unique challenges to designing backup and recovery policies. When faced with a recovery of SAP objects in the database, you have to recover the entire database as a single logical entity, due to the nature of the data model in SAP. This is especially challenging if you have to provide prior point-in-time (PIT) recovery capability, and you have to design a backup strategy to support this.

DB2 online backups are one way to perform nondisruptive backups. Even if you are performing hardware volume-based backups, you must perform image copies as well. When considering the performance of your SAP data sharing environment, bear in mind that using the usual non-SAP DB2 techniques for performing backups and securing a point of referential integrity may not be suitable. Using the quiesce utility after an online backup or the `archive log mode(quiesce)` command may have a severe performance impact on your system. These methods may even lead to severe disruption of your system resulting in transaction failures for the users.

In an SAP DB2 environment, the quiesce utility and `archive log mode(quiesce)` command may never be able to finish. In a moderately busy SAP system there will always be at least one logical unit of work (LUW) active, and you will be lucky to get a quiet point in which the quiesce utility can quiesce all of the many thousands of objects in your database. You will in fact be running several concurrent quiesce utility jobs on later releases of SAP because of the amount of objects in the database. The quiesce utility can be especially disruptive and can cause SQL -904 errors as it quiesces many tablespaces that other LUWs must access. The `archive log mode (quiesce)` command is not disruptive, but it may never complete.

So if your business requires the capability to recover to any prior PIT you need to design a recovery strategy based around a non disruptive backup method. This is essential to ensure performance and to eliminate disruption in your system.

Recovery of the database to the current or most recent state is not a problem. However, for prior PIT recovery, this presents challenges. Without a quiesce point to give a point of referential integrity, you must be able to chose your LRSN as a prior PIT and then generate jobs to recover only those objects that have changed between the prior PIT and the present time. You will be using your online image copies and your log files. Alternatively, you could use the DB2 conditional restart method. However, this is also complicated and even with this method you only want to recover your changed objects. To be able to execute either of these methods you need to develop your own tool to automate and speed up the process. Probably the easiest option is to obtain a tool that has been developed specially to perform prior PIT recovery on an SAP DB2 database from a vendor.

If you do not rely on the software method of performing a PIT recovery, then you can use the hardware facilities of the newer disk technology to take regular backups, then use these as recovery points. Again, this can be disruptive to SAP system performance. To take the backup, you must suspend the DB2 activity using the DB2 suspend logwrite command. Depending on how long this takes, activity in the system is stopped, causing performance problems and disruption to the users. Some hardware vendors' disks can also take a backup of your system without the need for the DB2 activity to be suspended. However, you should consult your vendor for advice.

> **Important:** Remember that you may still have to perform a software-based recovery of your system to get back to your chosen prior PIT if it is not close to the time of one of your hardware backups.

Therefore, when considering the performance of your SAP data sharing environment, talk with your DBAs about how they intend to provide prior PIT recovery without affecting performance or disrupting the running of the system.

12

SAP performance

In this chapter we provide a suggested route map for approaching SAP performance problems reported by your SAP users. We describe more specifically how to analyze causes of performance problems that are specific to data sharing.

We discuss these topics:

- Performance problem analysis route map
- Route map for server-wide and system-wide performance analysis
- Route map for process-specific performance analysis
- ST04 and DB2 performance monitoring tools
- STAT and STAD statistics records
- ST05 SQL trace
- ST10 statistics analysis

For further discussions on SAP performance you may also consult the following IBM whitepaper:

- *Tuning SAP / DB2 / zSeries*
  http://www-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100287
- *SAP BW Query Performance with DB2 on zSeries*
  http://www-1.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/WP100322

12.1 Performance problem analysis route map

When users complain about poor performance, you need to follow a route map or procedure to capture the relevant information necessary to begin your analysis and follow it through to a conclusion.

Figure 12-1 illustrates a series of steps required to capture the information you need to begin your analysis.

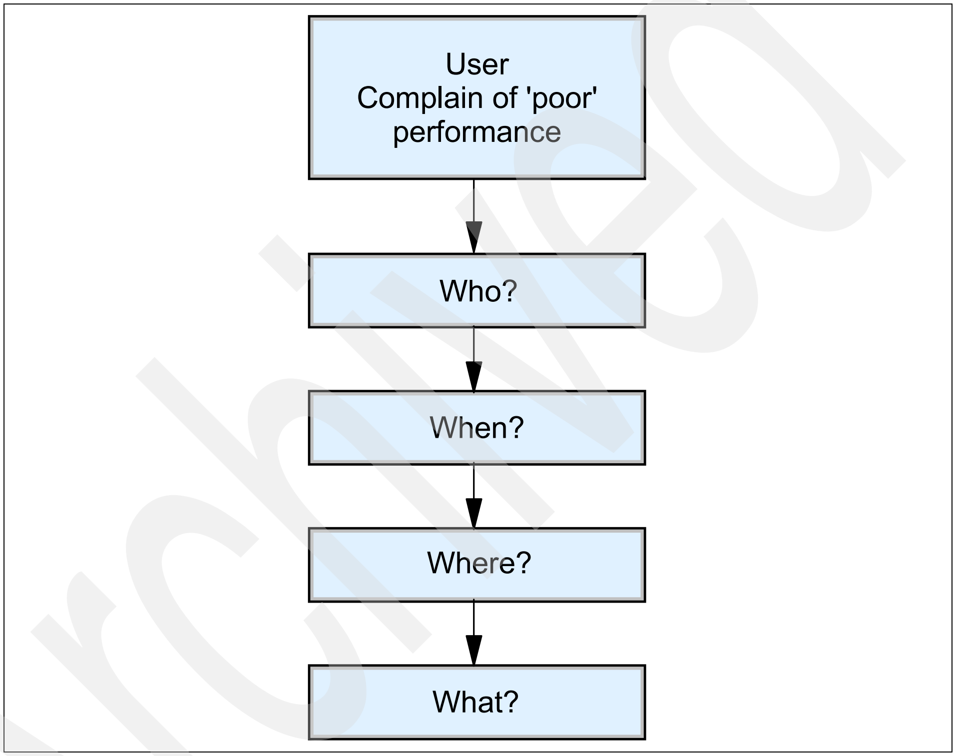


Figure 12-1 Information required to begin your investigation

When users complain of poor performance, it is essential to establish the facts:

- Which users are complaining? (You need to know their user IDs.)
- On what date and time did the users have the problem?
- Which server was the user was logged on to or which server was the batch executing on?
- Most important, what transaction, reports, query of batch job was the user running that suffered the poor performance?

Instruct your help desk or operations center to capture at least this information when users complain of poor performance.

12.2 Route map for server-wide and system-wide performance analysis

After establishing the facts, you can begin your analysis. If many users are complaining of poor performance or system hangs, then this may be a system-wide problem. If all users on one particular server are complaining of poor performance, then the problem may be only with that server.

Figure 12-2 illustrates a suggested route map for analysis of server-wide and system-wide performance problems.

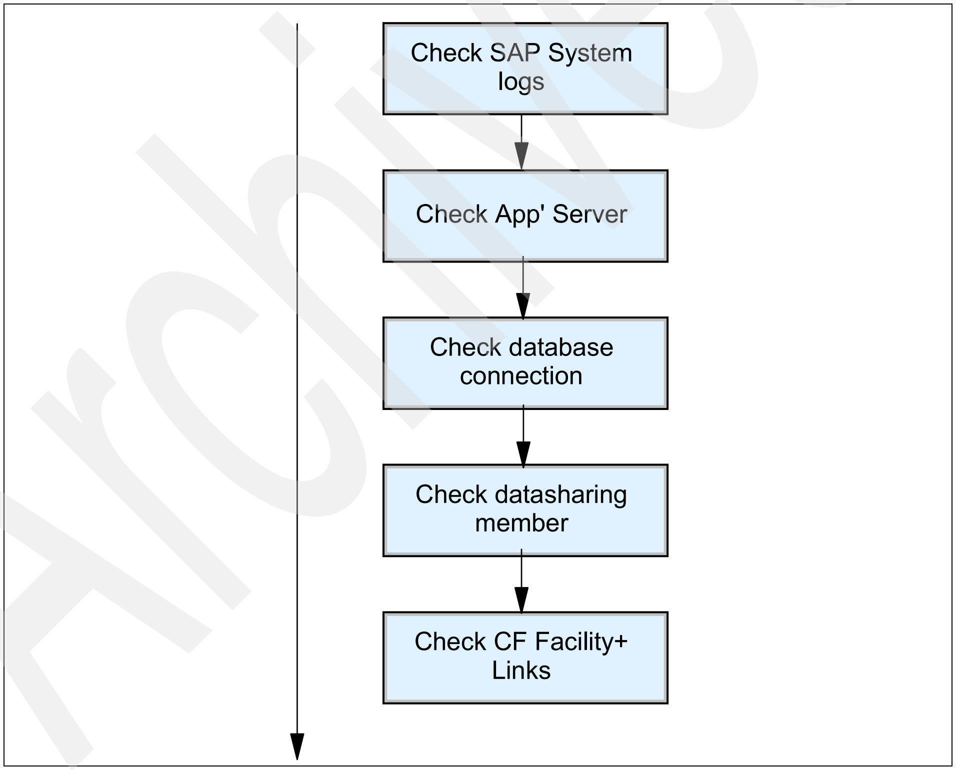


Figure 12-2 A route map for server-wide and system-wide performance analysis

We look at each step in a little more detail:

- Check SAP system logs.

  Use transaction SM21 to check your SAP system logs for any error situations that may be causing performance problems. Use the menu path **Systemlog** → **Choose** → **All remote system locks**.

- Check application server health.

  If there are no obvious errors on the SAP system logs, begin checking the health of the application server(s):

  – Use transaction ST06 to check the CPU consumption. From the menu path **Goto** → **Current data** → **Snapshot** → **Top CPU processes** you can determine whether a user transaction or report is looping and using all of the available CPU.

  – Check the health of the SAP buffers using transaction ST02.

  – If applicable, check the health of the update process on the server using transaction SM13.

  – Check the SAP locks using transaction SM12. Look for a large number of locks held by an individual user.

- Check the database connection.

  If there are no problems with the servers in question, confirm that the database connection from the server to the DB2 data sharing member is performing well:

  – Check the ICLI network connection by using SAP transaction DB2, selecting the **Performance Tuning** tab, then **ICLI ping** to test the network response time.

    This should show a response time of less than 2 milliseconds to send 500 bytes of data. If this response time is high (above 10 ms) then there is a problem with the link.

    High ICLI response times cause database response times for the servers to increase.

    Continue the investigation by asking the network support group to investigate the problem.

  – Check the ICLI server logs for errors.

  – Check the performance of the ICLI server on the data sharing member or members. You can use the RMF work delay monitor Monitor III to investigate for any delays with the ICLI server. The use of RMF Monitor III is covered in 10.4, “RMF Monitor III - Work Delay Monitor” on page 119.

- Check the data sharing member.

  Performance analysis on OS/390 z/OS is beyond the scope of this book, but there are some basic checks that you can perform:

  – Check the OS/390 z/OS system logs for any error situations that may cause problems on the LPAR.

  – Check the DB2 MSTR joblog for any error situations in DB2 that may be causing performance problems.

  – Check the LPAR CPU utilization and look for looping address spaces consuming CPU.

  – Use RMF Monitor III to check your SAP and ICLI workloads for any delay. See 10.4, “RMF Monitor III - Work Delay Monitor” on page 119 for more information about this.

  – Check the health of the data sharing subsystem on the member.

  For more guidelines about what to check in the DB2 data sharing member, refer to Chapter 11, “Data sharing performance” on page 125.

  You can perform the following basic checks:

  – Check the health of the data sharing complex and its members

    From the OS/390 z/OS console, issue this DB2 command where *pppp* is your command prefix:

    ```
    -pppp DISPLAY GROUP DETAIL
    ```

    Check for any members that are quiesced or have failed. They may be holding locks that cause systems to hang.

    You can also issue this command from SAP by using the transaction DB2 and selecting the **Performance Tuning** tab, then **Tools** → **DB2 commands**.

    You can then create the command, save it, and execute it.

  – Check for DB2 locks using SAP transaction DB2 and selecting the **Performance Tuning** tab, then **DB2 subsystem activity** → **Lockwaits**.

    The output shows any holding and waiting SAP processes in the DB2 subsystem. The display also tells which database the locks relate to.

  – Check for retained database locks.

    From the OS/390 z/OS console, issue this DB2 command where *pppp* is the command prefix:

    ```
    -pppp DISPLAY DATABASE LOCKS
    ```

    An `R` in the LOCKINFO column indicates that a retained lock is being held and by which member of the data sharing group.

This causes users waiting for locks on that database to hang.

- Check the DB2 subsystem activity for indicators of potential problems using SAP transaction DB2 and selecting the **Performance Tuning** tab, then **DB2 subsystem activity** → **Overview**.

- Check the coupling facility and links.

  For more guidelines about what to check with the coupling facility and links, refer to Chapter 10, “Sysplex performance” on page 111.

  However, you can perform the following basic health checks:

  – From the OS/390 z/OS console issue the following command

    ```
    D XCF,STR
    ```

    This displays the status of your structures.

  – From your ISPF RMF Monitor III menu, select **S SYSPLEX** → **5**.

    This displays an overview of coupling facility activity. Check that the CPU activity is no higher than 50% and that the storage has not been all used.

  – From your ISPF RMF Monitor III menu, select **S SYSPLEX** → **6**.

    This displays availability and performance of the paths to the CF. Ensure that all of your defined paths are available and that the delay percentage is not too high. Refer to Chapter 10, “Sysplex performance” on page 111 for more information.

12.3 Route map for process-specific performance analysis

If only a specific group of users is reporting performance issues. then a system-wide performance problem is unlikely. If you can determine that this is not a general system-wide performance problem, then you should perform a process-specific performance analysis.

Figure 12-3 on page 173 illustrates a suggested route map for process-specific performance analysis in an SAP data sharing environment.

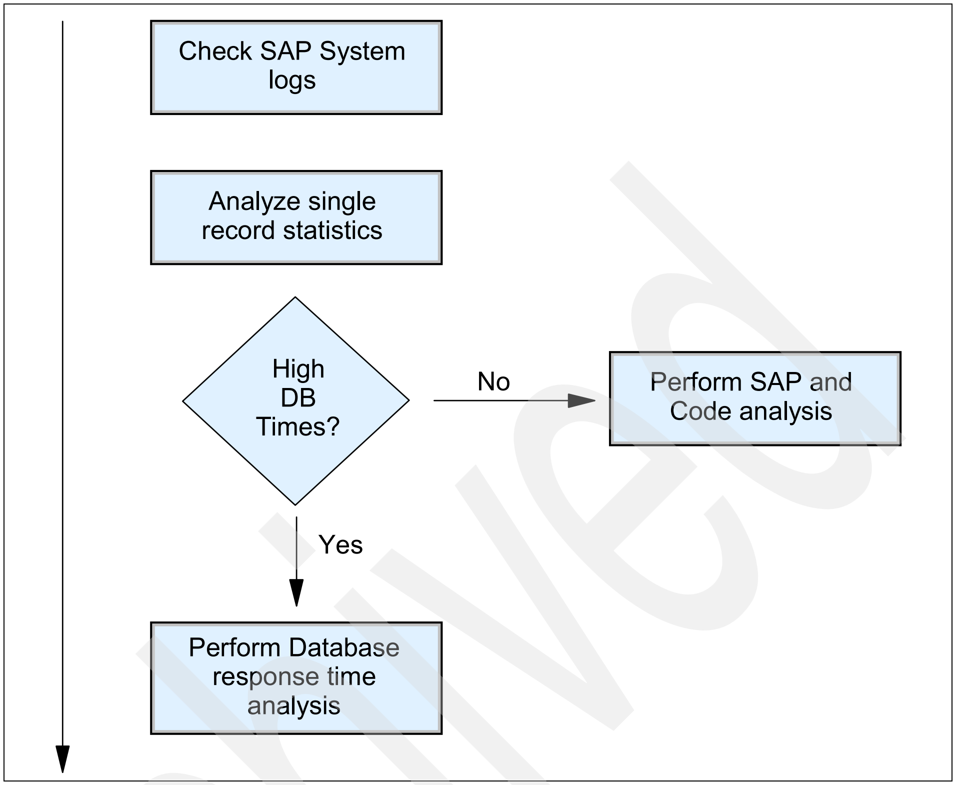


Figure 12-3 Route map for process-specific performance problem analysis

We look at each step in a little more detail:

- Check SAP system logs.

  Use transaction SM21 to check your SAP system logs for any error situations that may be causing performance problems. Use the menu path **Systemlog** → **Choose** → **All remote system logs**.

- Analyze single record statistics.

  The single record statistics provide a breakdown of the response time for the transaction dialog step, the report, or the batch job. Select transaction **STAT** or **STAD** to select the single record statistics.

  Refer to 12.4, “ST04 and DB2 performance monitoring tools” on page 174, for more details on using these transactions.

  Depending on your analysis of the single record statistics, proceed with the next step of your analysis.

- If the database component of the response time is *not* the most significant contributing factor, then continue your analysis of the SAP application and code. We will not cover this in this book because we are concentrating on issues specific to data sharing.
- If the database component of the response time *is* the most significant contributing factor, then continue your analysis of the database response time. We concentrate on this analysis with additions for a data sharing environment.
  - Analyze the table access times in the STAT/STAD record:
    - Check the buffering of the most "expensive" tables in SAP or DB2. Should they be buffered in SAP or DB2?
    - Is the user using the correct SAP application server according to your affinity rules? Should the user be logged on to an application server that connects to a different data sharing member where the expensive tables are buffered in DB2?
  - Perform an SQL trace using SAP transaction ST05. Arrange for the user to repeat the process and analyze the trace output:
    - Identify the most expensive SQL statements and use the Explain facility to analyze the access paths. If required, add or modify indexes to improve access paths. Ensure that Runstats has been run against the object and that the DB2 catalog information is up-to-date.
  - If access paths are good but access times are still high, analyze the data sharing performance to check for factors in data sharing that may contribute to poor database times:
    - Consult Chapter 10, "Sysplex performance" on page 111 for sysplex performance analysis.
    - Consult Chapter 11, "Data sharing performance" on page 125 for data sharing performance analysis.

12.4 ST04 and DB2 performance monitoring tools

SAP transactions ST04 and DB2 contain a number of analysis tools that are very valuable in an SAP on DB2 for OS/390 system, and many of these pertain particularly to a DB2 data sharing environment. Essentially, transaction DB2 is a superset of transaction ST04, with a similar screen layout and some additional tools. Most important, there are three tabs of additional tools that execute other relevant administrative transactions or ABAP programs. As such, transaction DB2 becomes a very valuable navigation point with a wide variety of tools available without requiring the user to memorize all of the related transaction codes and ABAP program names. Throughout this section, we refer to transaction DB2 for consistency, where the actual function being described may also exist in transaction ST04 or other transactions as well. Figure 12-4 shows the initial screen on transaction DB2.

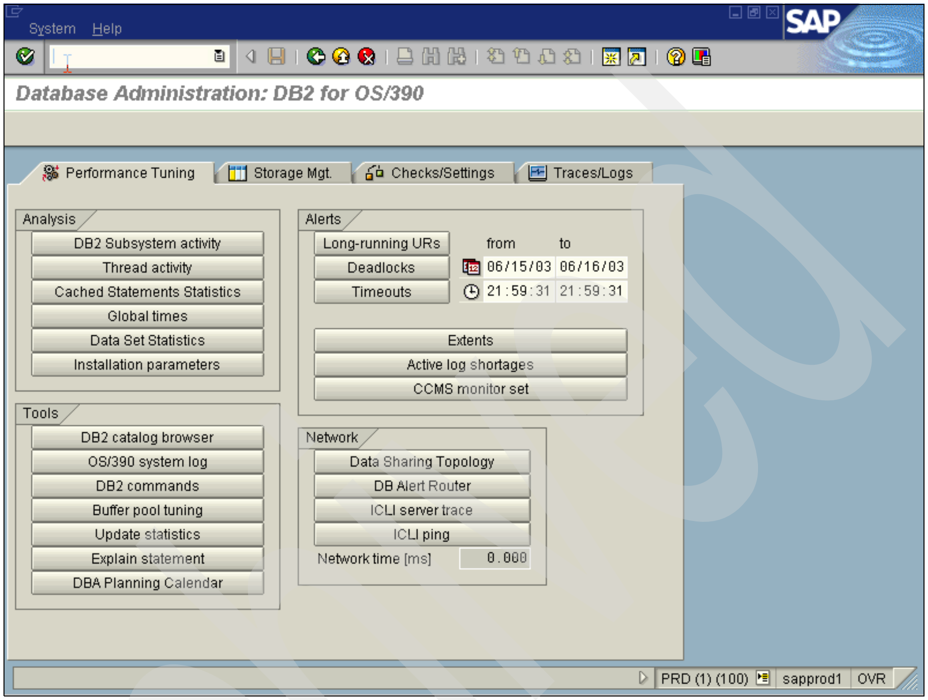


Figure 12-4 Transaction DB2 initial screen

Some of the most effective performance analysis tools begin with data obtained from the functionality of transaction DB2. In some cases, most of the analysis is possible from the functionality provided in transaction DB2; in other cases, you may need to obtain further information from other sources to complete the investigation and subsequent action. Most of the functions in transaction DB2 and described here are also described in detail in the SAP manual *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server 6.20*: This manual should be referred to along with this redbook while performaing analysis.

We now discuss these areas of transaction DB2 functionality:

- Installation checks
- Dynamic Statement Cache analysis
- DB2 subsystem activity

12.4.1 Installation checks

This function is found as a button labelled Installation Checks on the Checks / Settings tab of transaction DB2. It is also accessable directly as transaction DB16. This performs a check dynamically of the currently active DSNZPARMs of DB2 against a set of recommended values maintained by SAP and IBM. The values checked can be seen and adjusted using transaction DB17. This ensures that all of the critical and important values are set appropriately in the DB2 subsystem. Note that in a DB2 data sharing environment, each member has its own DSNZPARM values so should be checked separately by running this check on an application server attached to this member.

12.4.2 Dynamic Statement Cache analysis

The overall concept of dynamic statement caching and related DSNZPARMs are discussed in 11.4, "Dynamic statement cache" on page 132. In a DB2 data sharing environment, it is important to remember that statement caching takes place completely separately on each DB2 member, meaning that SQL statements are prepared separately on each DB2 member. It also means that the contents of the statement cache and the associated statistics will be different across each DB2 member. This fact can actually lead to valuable information about the overall activity in the system, and the distribution of activity between the differing DB2 data sharing members.

Figure 12-5 on page 177 shows an example of a screen for Dynamic Statement Cache where no filtering has been specified after the Cached Statement Statistics button on transaction DB2 has been pressed.

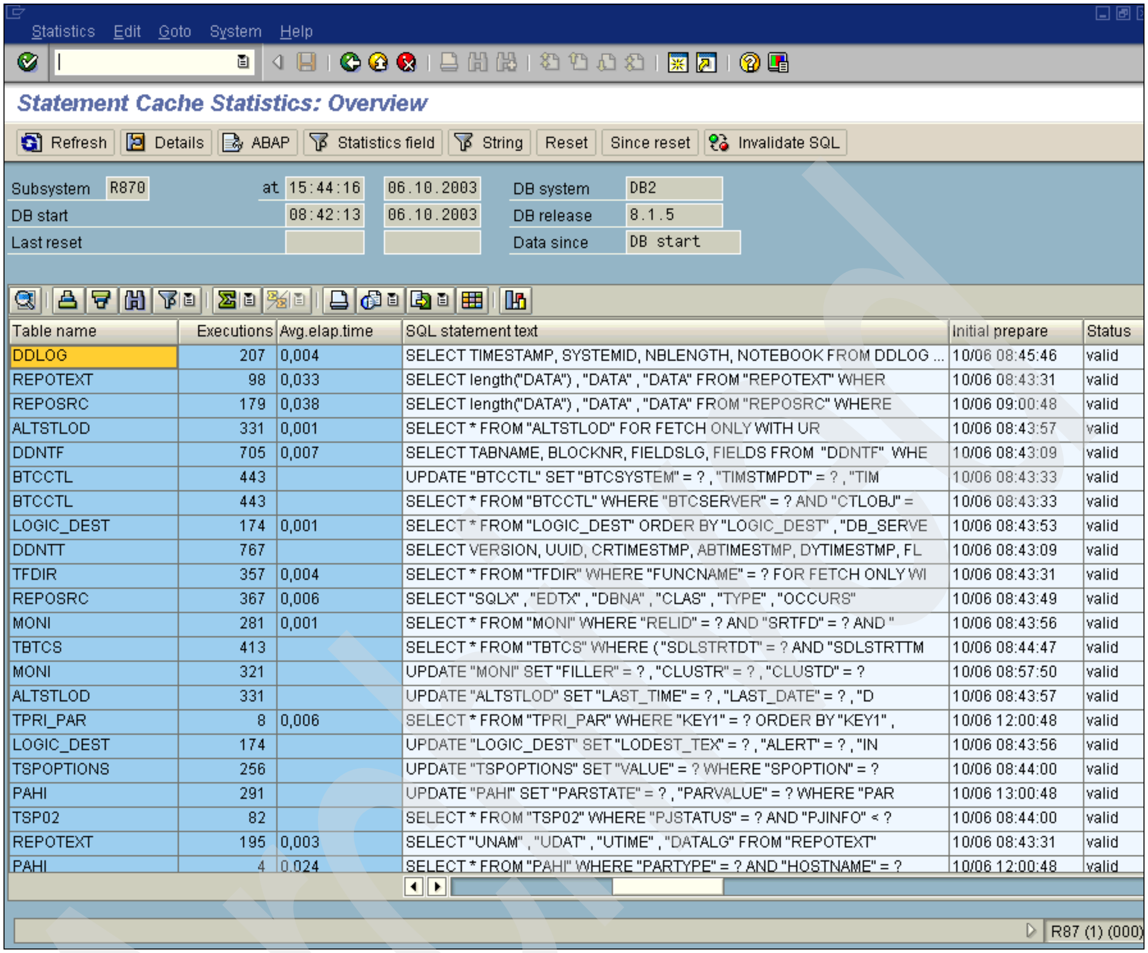


Table name	Executions	Avg.elap.time	SQL statement text	Initial prepare	Status
DDLOG	207	0,004	SELECT TIMESTAMP, SYSTEMID, NBLENGTH, NOTEBOOK FROM DDLOG ...	10/06 08:45:46	valid
REPOTEXT	98	0,033	SELECT length("DATA") , "DATA" , "DATA" FROM "REPOTEXT" WHER	10/06 08:43:31	valid
REPOSRC	179	0,038	SELECT length("DATA") , "DATA" , "DATA" FROM "REPOSRC" WHERE	10/06 09:00:48	valid
ALTSTLOD	331	0,001	SELECT * FROM "ALTSTLOD" FOR FETCH ONLY WITH UR	10/06 08:43:57	valid
DDNTF	705	0,007	SELECT TABNAME, BLOCKNR, FIELDSLG, FIELDS FROM "DDNTF" WHE	10/06 08:43:09	valid
BTCCTL	443		UPDATE "BTCCTL" SET "BTCSYSTEM" = ? , "TIMSTMPDT" = ? , "TIM	10/06 08:43:33	valid
BTCCTL	443		SELECT * FROM "BTCCTL" WHERE "BTCSERVER" = ? AND "CTLOBJ" =	10/06 08:43:33	valid
LOGIC_DEST	174	0,001	SELECT * FROM "LOGIC_DEST" ORDER BY "LOGIC_DEST" , "DB_SERVE	10/06 08:43:53	valid
DDNTT	767		SELECT VERSION, UUID, CRTIMESTMP, ABTIMESTMP, DYTIMESTMP, FL	10/06 08:43:09	valid
TFDIR	357	0,004	SELECT * FROM "TFDIR" WHERE "FUNCNAME" = ? FOR FETCH ONLY WI	10/06 08:43:31	valid
REPOSRC	367	0,006	SELECT "SQLX" , "EDTX" , "DBNA" , "CLAS" , "TYPE" , "OCCURS"	10/06 08:43:49	valid
MONI	281	0,001	SELECT * FROM "MONI" WHERE "RELID" = ? AND "SRTFD" = ? AND "	10/06 08:43:56	valid
TBTCS	413		SELECT * FROM "TBTCS" WHERE ("SDLSTRTDT" = ? AND "SDLSTRTTM	10/06 08:44:47	valid
MONI	321		UPDATE "MONI" SET "FILLER" = ? , "CLUSTR" = ? , "CLUSTD" = ?	10/06 08:57:50	valid
ALTSTLOD	331		UPDATE "ALTSTLOD" SET "LAST_TIME" = ? , "LAST_DATE" = ? , "D	10/06 08:43:57	valid
TPRI_PAR	8	0,006	SELECT * FROM "TPRI_PAR" WHERE "KEY1" = ? ORDER BY "KEY1" ,	10/06 12:00:48	valid
LOGIC_DEST	174		UPDATE "LOGIC_DEST" SET "LODEST_TEX" = ? , "ALERT" = ? , "IN	10/06 08:43:56	valid
TSPOPTIONS	256		UPDATE "TSPOPTIONS" SET "VALUE" = ? WHERE "SPOPTION" = ?	10/06 08:44:00	valid
PAHI	291		UPDATE "PAHI" SET "PARSTATE" = ? , "PARVALUE" = ? WHERE "PAR	10/06 13:00:48	valid
TSP02	82		SELECT * FROM "TSP02" WHERE "PJSTATUS" = ? AND "PJINFO" < ?	10/06 08:44:00	valid
REPOTEXT	195	0,003	SELECT "UNAM" , "UDAT" , "UTIME" , "DATALG" FROM "REPOTEXT"	10/06 08:43:31	valid
PAHI	4	0,024	SELECT * FROM "PAHI" WHERE "PARTYPE" = ? AND "HOSTNAME" = ?	10/06 12:00:48	valid

Figure 12-5 Dynamic Statement Cache analysis

Note that if you see no statements when pressing the Cached Statement Statistics button, it is possible that you have not enabled the IFCID 318 trace to capture this data. This is started by the following DB2 command:

```
-START TRACE(P) CLASS(30) IFCID(318) DEST(SMF)
```

This command can be executed from the DB2 Commands section of the DB2 transaction in SAP. It is recommended that this trace be started automatically after a DB2 subsystem start and is left running for the life of the DB2 subsystem. As it cannot be started by a ZPARM, some automated mechanism will have to be implemented to issue the command to DB2 after it has started normally.There is a minimal overhead involved with running this trace, but it is considered trivial relative to the value of the statistics for tuning that are obtained.

12.4.3 DB2 subsystem activity

The transactions ST04 and DB2 provide several monitoring features related to overall DB2 subsystem activity. These are viewed by pressing the **DB2 Subsystem Activity** button on the DB2 transaction screen shown in Figure 12-4 on page 175. After pressing this button, the screen appears as shown in Figure 12-6. This screen presents the major highlights of the subsystem activity. A pull-down menu of other activity screens, which shows more detailed information, is found near the top of the screen. Note that the screen may appear different for different releases of SAP and DB2. For this example we use a DB2 Version 8 system, and SAP R/3 Enterprise Version 4.7 on a V6.20 SAP kernel.

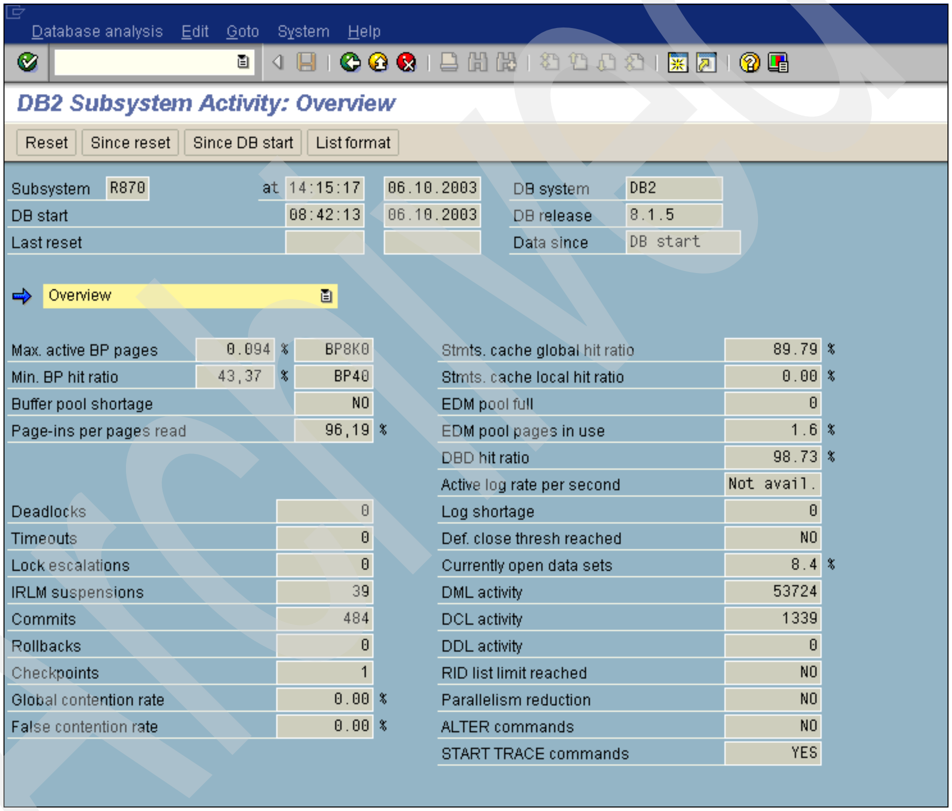


Figure 12-6 DB2 Subsystem Activity overview

More detail of the data available in this transaction is contained in the document *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server 6.20*. We do not attempt to duplicate that information here. As an example of more detailed information, Figure 12-7 on page 179 shows the detailed breakup of the content of the EDM pool.

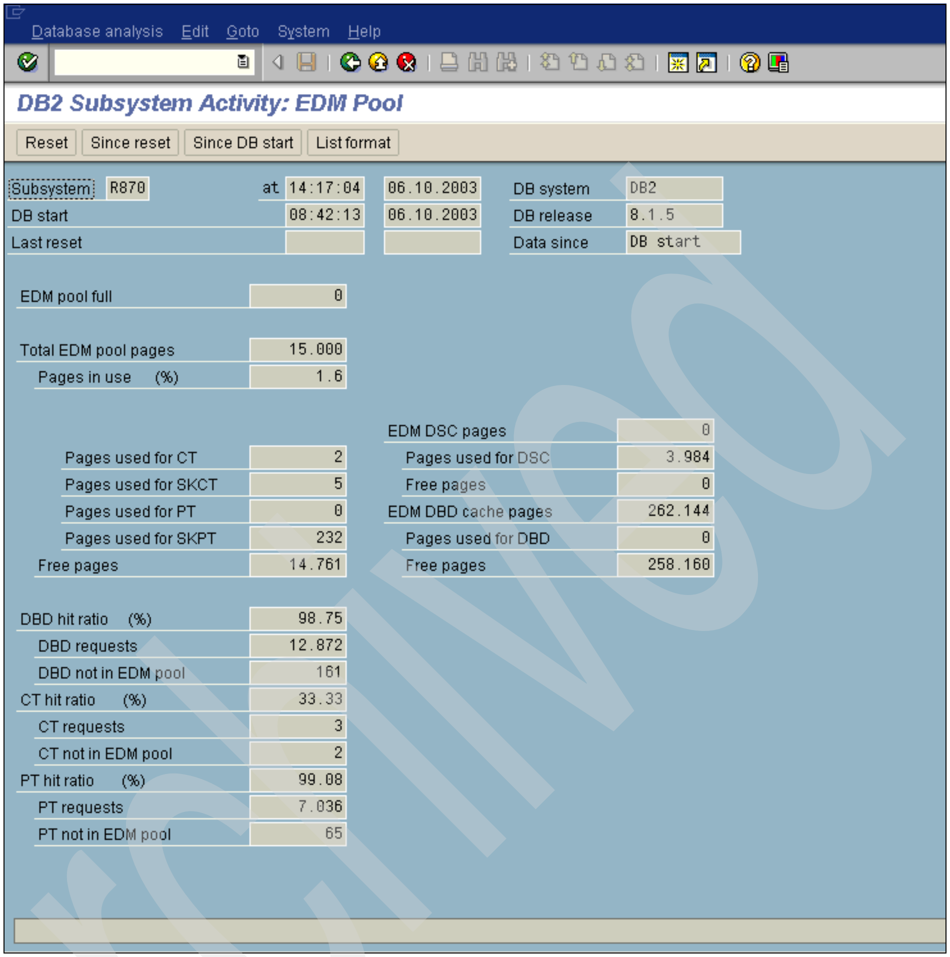


Figure 12-7 EDM Pool Activity screen

12.5 STAT and STAD statistics records

SAP transactions STAT and STAD enable the analysis of individual dialog step statistic records (STAT records) to gain furhter understanding of the nature of the performance of the SAP system. At the completion of every step (which can include dialog, batch, spool, update, and so on) SAP writes a corresponding record in a file on the application server, detailing the response time characteristics of the step. Subsequently, these statistics are analysed by SAP jobs to become the statistical data behind transaction ST03, which is commonly used for SAP performance analysis.

Transaction STAT operates on the stat file of the application server where it is executed, whereas transaction STAD searches the stat files of multiple application servers via RFC calls to those servers. As such, STAD is a superset of transaction STAT, and it commonly is used for gaining an overall picture of the SAP system in question, as well as searching for records where the application server of execution is unknown. Transaction STAT is used when the search of stat records is to be limited to the application server in question. When displaying results, both transactions give the same data from the same source, namely the stat file.

These individual stat records can be used to determine various performance problems that can be remedied by a variety of means. It should also be noted that because individual stat records are the lowest granularity of statistics, they are subject to one-off effects in the system that result in spikes affecting single transactions. Generally, before starting to conduct more thorough efforts to improve performance, it should be determined it the problem in question is more than a one-time occurrence. Additionally, the information about whether the problem is seen on all application servers in an SAP system or is limited to one or a few of the application servers is valuable in planning a course of action to determine the solution required.

Figure 12-8 on page 181 shows the initial data in a STAT record and includes the response time and the factors contributing to it.

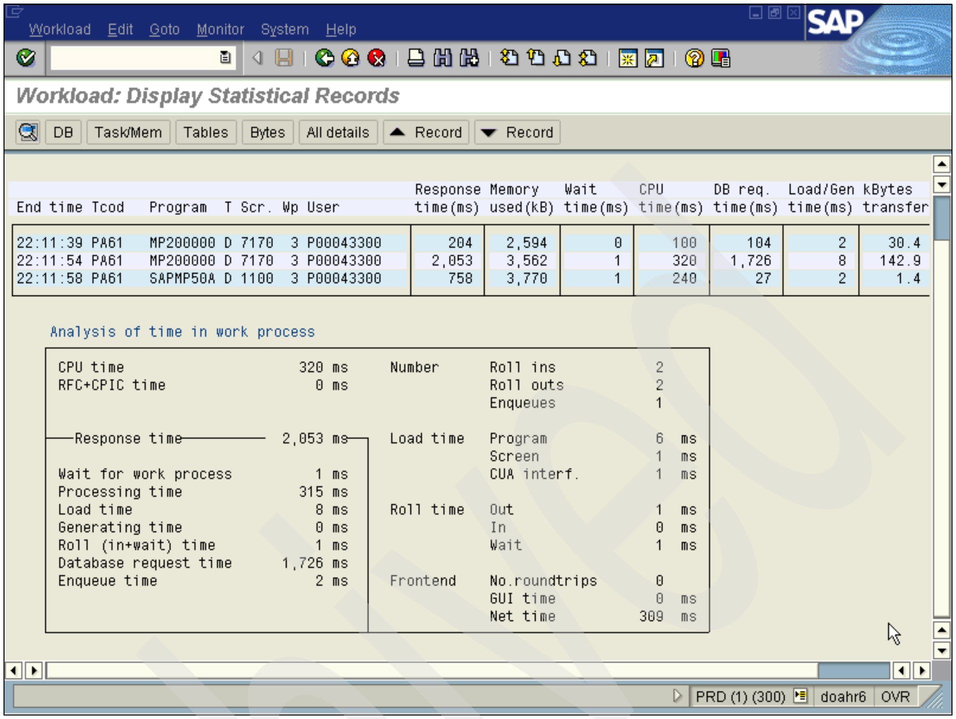


Figure 12-8 Format of stat record: description of fields

The analysis of performance and tuning is a detailed subject, but we limit our discussion to the topic of database performance. From the initial STAT record detail, click the **DB** button at the top of the screen to display the kinds of details shown in Figure 12-9 on page 182.

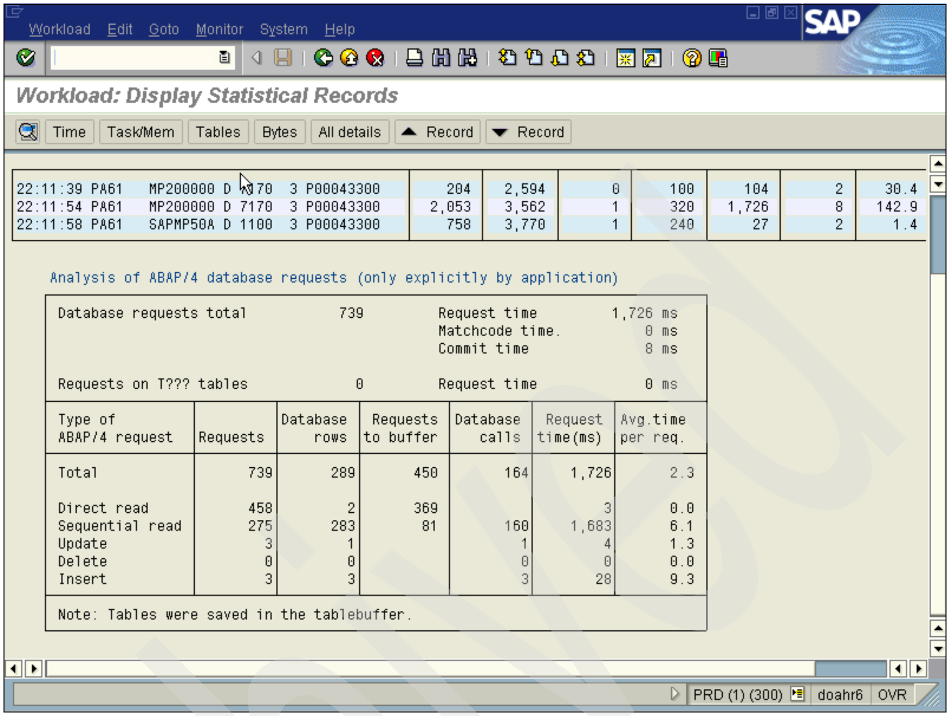


Figure 12-9 STAT record showing database detail

The database detail shows the summation of all activity the transaction has performed as a result of SQL statements contained in the program. The Requests column indicates the total number of SQL statements executed and their types (such as read/update). This includes access to SAP buffered tables that may not result in DB2 activity. The Database Calls column is a summary of all requests that passed through to the database (in our case, the DB2 subsystem). The average response time per request is also shown. Note that these statistics are the summation of all requests for all tables by the program.

SAP has the ability to collect table-level statistics through the use of the stat/tabrec instance parameter. Setting this parameter to a value other than its default of 0 (zero) results in some overhead on the application server, but the data captured can be extremely valuable in database performance analysis. When this parameter is enabled (a value of 5 is most commonly used, as it generally results in sufficient data), an additional button is seen on the STAT record detail screen. Pressing this button, **Tables**, produces the data shown in Figure 12-10 on page 183. With stat/tables=5 implemented, the top five table contributors to the database component of the response time are captured, and their performance statistics are displayed.

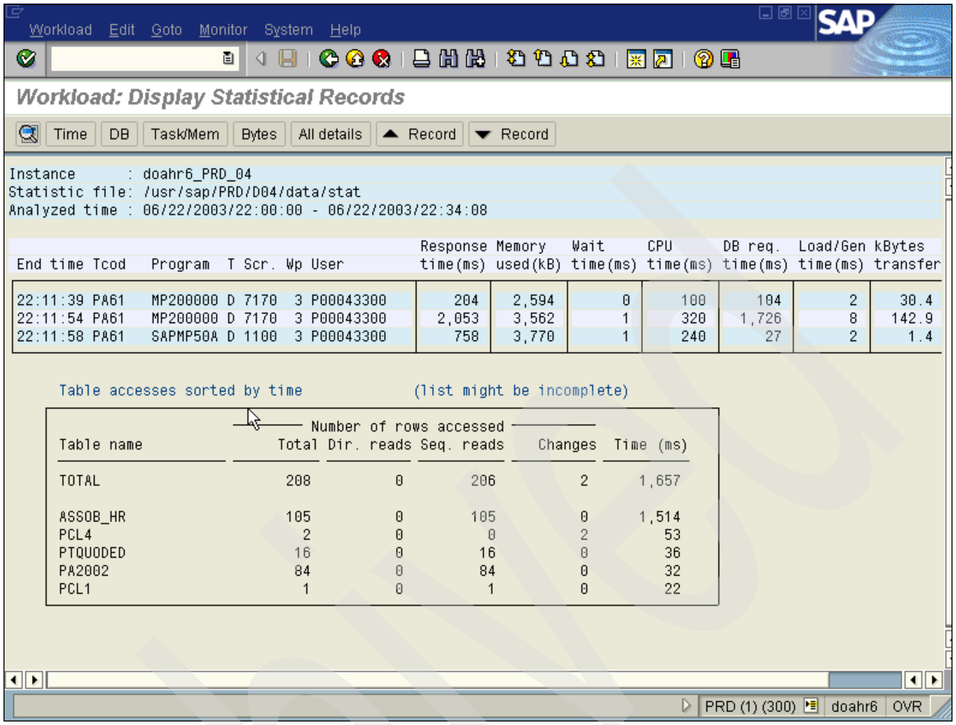


Figure 12-10 STAT record showing individual table records

If this data reveals that a problem is not specific to one of a small number of tables, it can point to overall problems. If the high times are specific to tables with update activity it may point to problems with logging or local or global locking issues. If the problem is specific to one or more tables, further analysis can be applied, such as that described in 12.4.2, “Dynamic Statement Cache analysis” on page 176 or 12.6, “ST05 SQL trace” on page 183.

12.6 ST05 SQL trace

SAP transaction ST05 is a general tracing tool provided as part of a standard SAP system. It enables a user to switch on a flag that causes SAP to write detailed information to a file about the nature of steps involved in processing a transaction or program. Subsequent analysis of the stored information gives a much more detailed picture of the actual execution of the program and, most important, some of the performance measurements taken during the trace. Transaction ST05 trace is particularly important when determining database-related performance problems, and the details captured can be very valuable to obtain solutions to various problems.

As we shall see, it is not directly possible to determine performance problems related to DB2 data sharing when using ST05 tracing analysis, but it is nonetheless a very valuble tool to determine the likely cause of a problem. In some circumstances it leads to further analysis to determine a cause, which may be related to the data sharing environment.

Figure 12-11 shows the basic format of an SQL trace captured by transaction ST05.

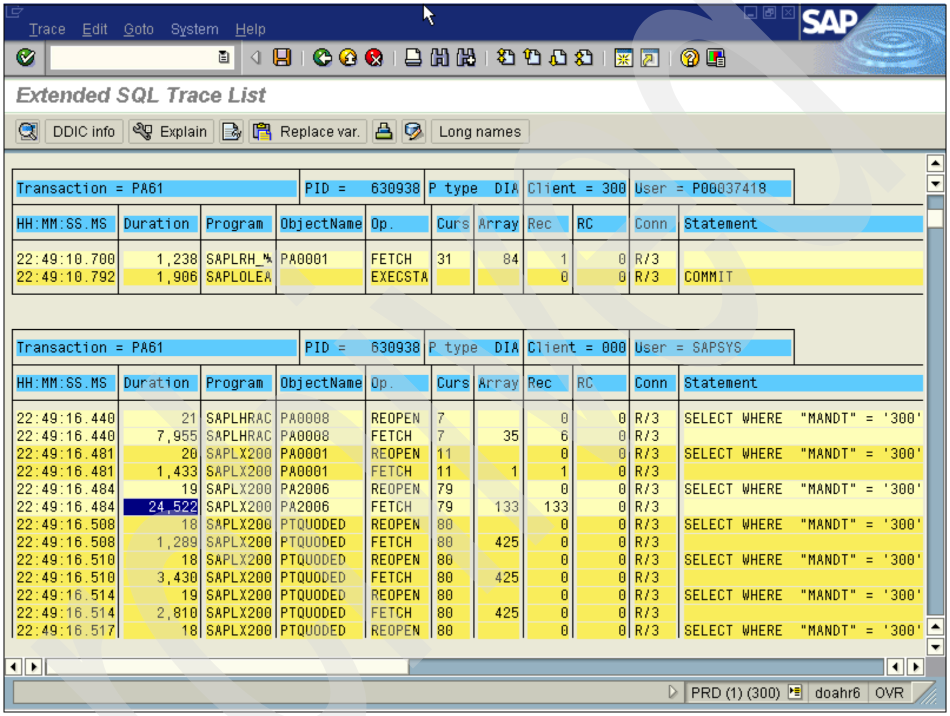


Figure 12-11 ST05 SQL trace listing

The use of transaction ST05 to capture the trace is described in detail in the online help for the transaction. A user of ST05 must have a detailed idea of what is being analyzed. By selecting the Trace for User option on the initial screen, it is possible to trace for individual SAP users, SAP transactions, and specific tables. When a suitable trace is performed and listed, the following notes apply to its analysis.

From the SQL trace list, it is possible to see the SQL statement, predicate variables used, the return code and number of records returned, and finally the response time to achieve the result.

Important: When analyzing individual SQL statement response times with ST05, note that in the case of non-OS/390 application servers, the individual response times include the network response of the request. It is possible that performance problems between the application server and database server can result in seemingly poor response times, which are in fact related to the network, not DB2. Always look for a consistent and reproduceable result before proceeding with database analysis.

By clicking on an SQL statement then clicking the **DDIC** button, the user can branch into the SAP dictionary for the database object, as Figure 12-12 shows. This displays the indexes available on the table, their fields, and associated details. This information, when combined with the data obtained from explaining the statement as described in Figure 12-14 on page 187, can in appropriate cases be used to make decisions such as changing DB indexes or ABAP code changes to improve performance.

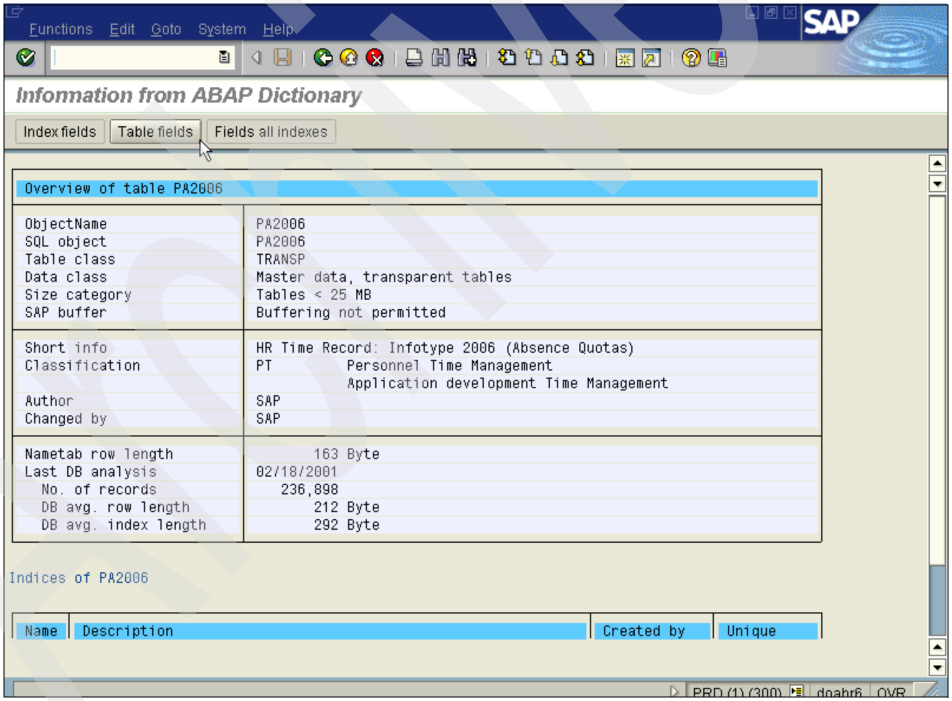


Figure 12-12 ST05 DDIC information display

Perhaps the most valuable action that can be taken from an ST05 SQL trace is to branch to the actual ABAP program that issued the statement. This information may not be available through any other method in performance analysis. The detail displayed by clicking on an SQL statement then clicking the ABAP program icon is shown in Figure 12-13.

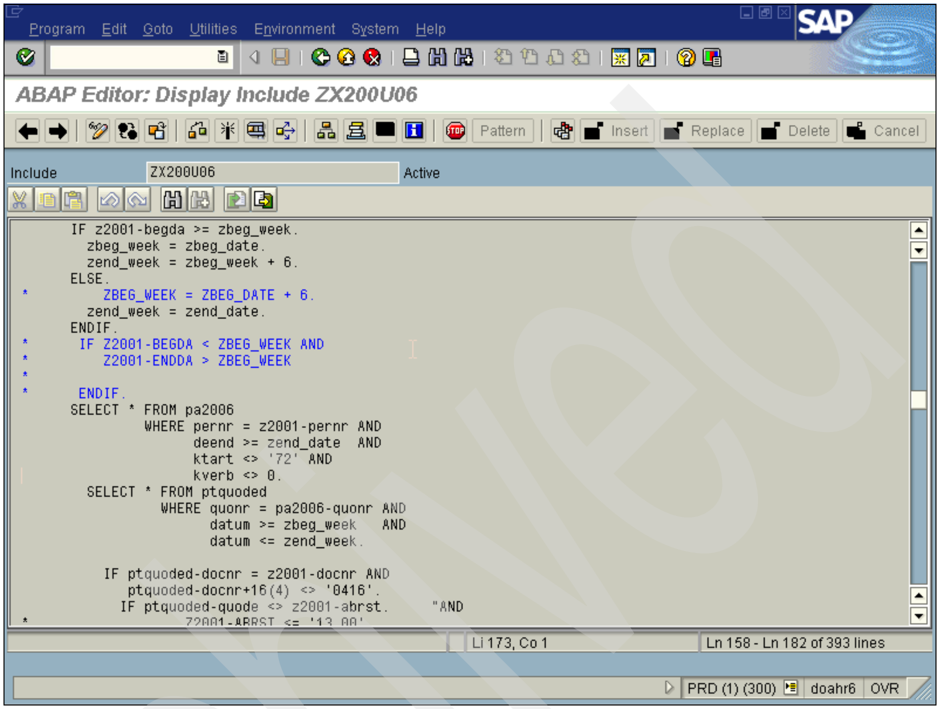


Figure 12-13 ST05 ABAP program display

By understanding which program issued the statement, and with which predicate variables, you may be able to gather information to solve a problem several ways. This may include adding an index to increase selectivity or changing the program to issue a better-performing SQL statement.

The code in Figure 12-13 shows a situation in which two embedded select statements exist that can be rewritten as a join between the two tables in question. In this case, the performance of the individual SQL statements themselves was deemed acceptable, but analyzing the code enables us to speed up processing, removing overhead and a potential bottleneck in a DB2 data sharing environment.

By selecting an SQL statement and clicking the Explain button, the access path of the statement in question is analyzed by DB2, and the screen in Figure 12-14 on page 187 is displayed.

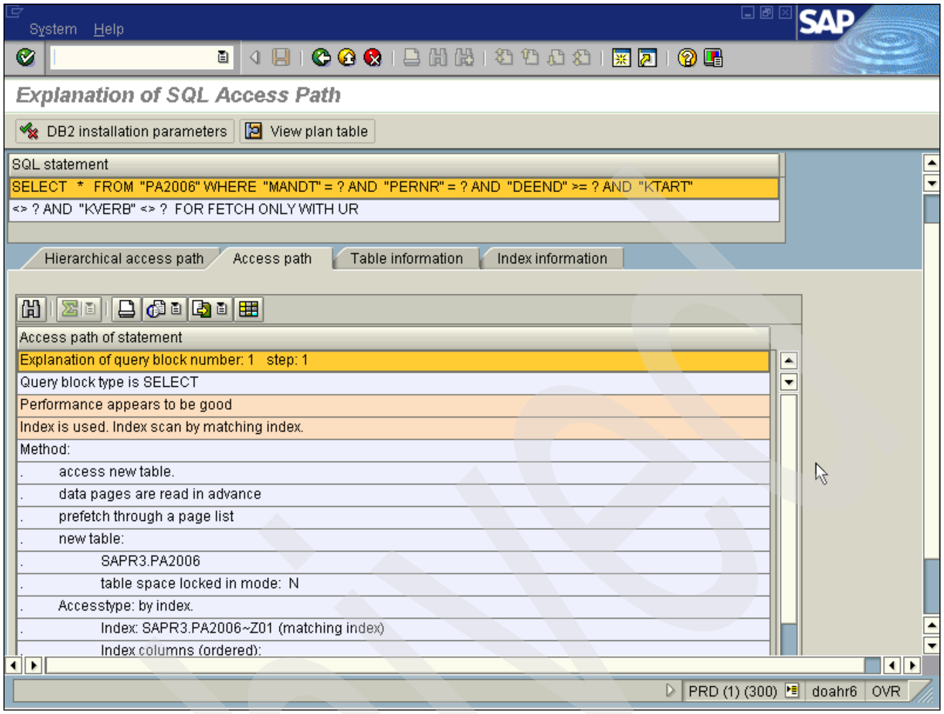


Figure 12-14 ST05 SQL statement Explanation display

> **Tip:** To ensure that the latest version of the Explain feature of SQL statements on an SAP on DB2 for OS/390 system is displayed, be sure that your SAP system has the latest SAP support transports applied, as detailed in the SAP OSS Note specific to your system's SAP version.

The explanation in Figure 12-14 shows which method the DB2 optimizer used to process the statement in question. Note that there are many other methods available to gather this information, but this can be a convenient method when using ST05 to trace a program. In addition to indicating the path used, the transaction enables you to view information about the tables involved, indexes on those tables, current DB2 statistics on those objects, and indications of whether updated RUNSTATs or a REORG is suggested on these objects.

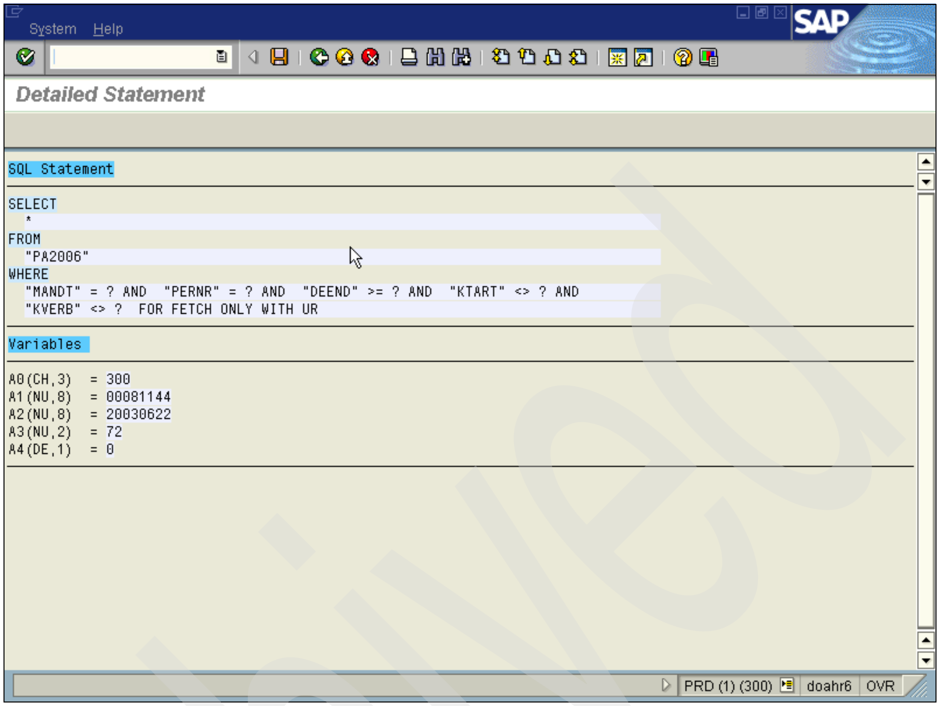


Figure 12-15 ST05 SQL statement detailed display

Finally, by clicking the Statement Detail button, it is possible to display the statement in question, as well as the actual variables specified in the predicate for this execution, as shown in Figure 12-15.

12.7 ST10 statistics analysis

SAP automatically collects statistics related to database activity that are very valuable for analysis in a DB2 for OS/390 or z/OS environment, and especially where DB2 data sharing is in use. Statistics on per-table and per-application server levels are accumulated by the database interface and periodically externalized in the form of records into table MONI into record type TB. These are accumulated into daily, weekly, and monthly records and kept for the period of time specified in *Residence times of statistical data* in transaction ST03. It is also accumulated both on a per-application server form and a total record, which is the summation of activity of all application servers in the corresponding period.

This data is valuable for many different tuning reasons. It can identify problems with SAP buffering, can point to problems with ABAP code logic, and, most important, is a useful source of data for DB2 buffer pool tuning. Additionally, using some post-processing of the data stored per application server, it can provide insight into potential inter-system table interest, and the effectiveness of logon group balancing in a DB2 data sharing environment.

A basic methodology of analyzing ST10 data follows, as well as some of the conclusions that may be reached and suggested courses of action to further determine whether a solution is possible.

ST10 initial analysis

A good starting point for an installation that has been in production phase for a reasonable length of time (usually three months to allow post-implementation abnormal activity to subside) is to analyze a prior month of table access data. This can be achieved by starting transaction ST10 and selecting the radio buttons **all tables** for Tables, **previous month** for Time frame, and **all servers** for Server. This is displayed in Figure 12-16. A pop-up dialog displays all months for which data is stored. Select a month in which activity was typical.

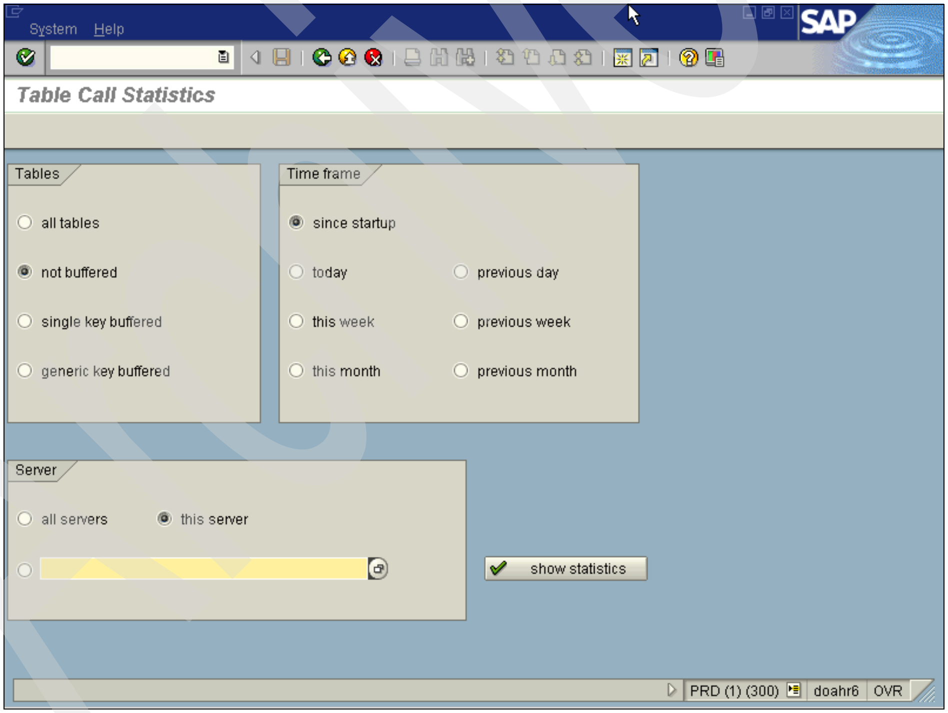


Figure 12-16 ST10 Startup selection screen

Initially the list displayed in ST10 will always be sorted by table name. For our purposes of statistical analysis, we need to sort this table on other columns. To sort the table on a column, click one value in the column to highlight it, and click the **Sort** button on the toolbar.

Figure 12-17 shows a production system for which statistics for a typical month of activity have been displayed and sorted by ABAP/IV Processor Requests.

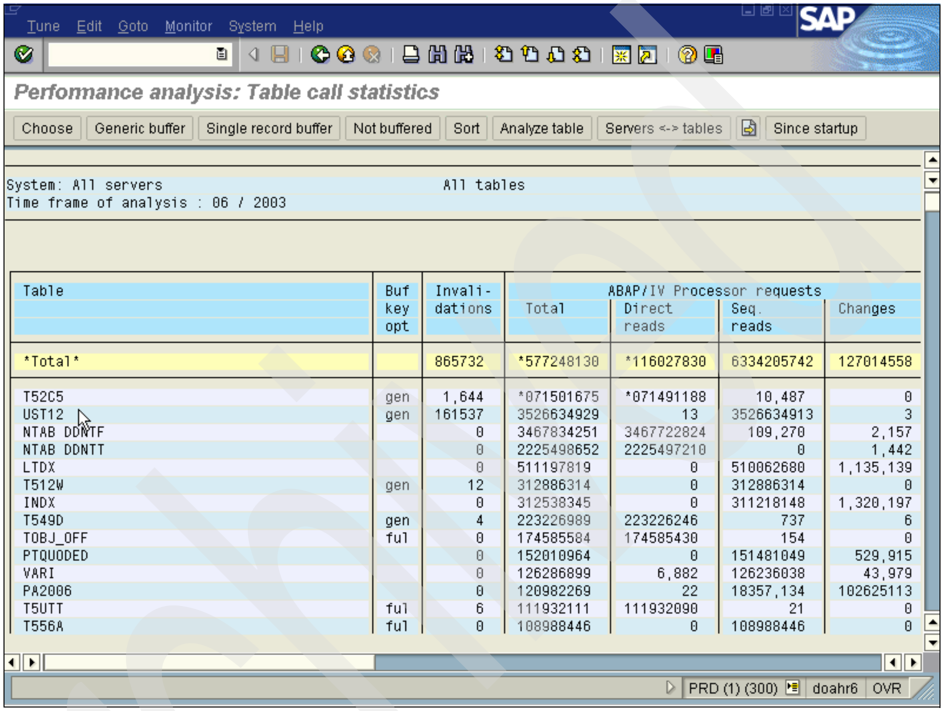


Table	Buf key opt	Invali-dations	ABAP/IV Processor requests Total	Direct reads	Seq. reads	Changes
Total		865732	*577248130	*116027830	6334205742	127014558
T52C5	gen	1,644	*071501675	*071491188	10,487	0
UST12	gen	161537	3526634929	13	3526634913	3
NTAB DDNTF		0	3467834251	3467722824	109,270	2,157
NTAB DDNTT		0	2225498652	2225497210	0	1,442
LTDX		0	511197819	0	510062680	1,135,139
T512W	gen	12	312886314	0	312886314	0
INDX		0	312538345	0	311218148	1,320,197
T549D	gen	4	223226989	223226246	737	6
TOBJ_OFF	ful	0	174585584	174585430	154	0
PTQUODED		0	152010964	0	151481049	529,915
VARI		0	126286899	6,882	126236038	43,979
PA2006		0	120982269	22	18357,134	102625113
T5UTT	ful	6	111932111	111932090	21	0
T556A	ful	0	108988446	0	108988446	0

Figure 12-17 ST10 data sorted by Total Requests

Sorting on ABAP/IV processor requests: total

This is a summation of every time an ABAP program has processed an SQL request listed according to table. As the most heavily used tables in an SAP system usually are, by design, SAP buffered, a majority of the most used (say, 50-100 tables) are in fact buffered as indicated in the Buffer Key Option. Note that NTAB DDNTF (dictionary field text) and NTAB DDNTT (dictionary table text), which normally feature in the highest-used tables, are in fact buffered despite having a blank value in the option column. You should be able to verify that the highest-used tables, say the top two screens of the list, are being served by SAP buffering, with the exception usually of a few functional tables that are not SAP buffered. You can determine this by seeing that the DB activity in the Calls column is significantly lower than the ABAP Processor Requests column for SAP buffered tables.

Sorting on ABAP/IV processor requests: changes

Gathering a statistically relevant set of data and sorting on the change column produces a valuable source of information from which to understand the tables that may become involved in intersystem interest in a DB2 data sharing environment. While this is not a solution in itself, an analysis of the tables involved, and their place in the system in terms of usage by transactions and programs, can provide insights into how to reduce effects of intersystem interest. It may be possible through means such as SQL tracing, outlined in 12.6, “ST05 SQL trace” on page 183, to identify transaction-causing updates and isolate the running of these transactions to one DB2 member by SAP logon load balancing.

After particular tables have been identified, it is also possible through Dynamic Statement Cache analysis to try to identify the SQL statements in question touching the table and further understand possible tuning actions.

When a table is identified with significant updating activity and is buffered by SAP, this can identify a problem that may have extreme consequences in a DB2 data sharing environment. When an SAP buffered table is updated, it updates the underlying DB2 table and triggers a process by which all application servers will read the data again from the DB2 table if programs subsequently access it. If the table is heavily used and updated frequently, this can result is a large amount of reading and updating activity, which results in a high intersystem interest in the object and associated performance penalty. This problem and its required action are discussed in “Notable exceptions to normal activity in ST10” on page 191.

Sorting on DB activity: calls

This is a total of all SQL calls that have been made to DB2, sorted according to tables. While this is somewhat similar to the analysis of Dynamic SQL Cache statistics, ST10 statistics effectively adds all different requests together, is not affected by DSC statement discarding, and is for an exact time frame.

If the system is well-tuned, the vast majority of the top used tables should not be SAP buffered. If any SAP buffered tables appear in the top 100 tables, there are a few reasons that could be involved.

Notable exceptions to normal activity in ST10

- SAP Buffered table with high update activity

  This usually is accompanied by a significant value in the Invalidation column. It may indicate that the table in question should be adjusted to remove SAP buffering. As a rule, SAP buffered tables should not have any significant update activity against them, as the cost of keeping application servers in sync then outweighs the benefit of buffering with increased DB2 activity. In this case, it should be verified by looking at other ST10 statistics for other periods to verify that this activity is normally present in your system. If so, the next step is to find the activity causing the updates, probably with SAP tracing as outlined in 12.6, “ST05 SQL trace” on page 183. If the update activity cannot be removed, the table may have to be unbuffered using transaction SE11, bearing in mind that this is an SAP repair that should be undertaken with due caution and involvement of the SAP technical support team.

- SAP Buffered table with high DB activity and no updates or invalidations

  This is a somewhat harder problem to deal with and can have several causes:

  – Use of ABAP code explicitly using BYPASS BUFFER syntax.

    This requires investigation into why this programming technique was used.

  – Table in question is involved in a join with other tables.

    The solution may involve changing the code to perform a manual join, in which the ABAP code performs simple SQL requests in the tables involved, resulting in an overall performance gain. In the case where the other tables are not SAP buffered, you probably will have no choice but to accept the activity.

  – ORDER BY has been specified.

    It may be beneficial to remove the ordering clause, enabling the request to be satisfied from the buffer, and subsequently sort the rows in the program. In this case you should verify that a small number of rows are involved, which should be the case with a buffered table.

A

Checking SAP data sharing setup

This appendix illustrates a procedure for checking the setup and customization of SAP described in Part 1.

Check ICLI setup

- 2 ICLIs per SAP dialog instance
- 1 primary ICLI & 1 standby ICLI

Check for buffer pool tuning

- ZZBPOOLT or DB2B for 4.6x

  Ensure that some buffer pool tuning has been done (for example, Tablespaces/Indexspaces in BP5 through BP11).

VB update tables setup

- Ensure that transport KDOK000668 has been installed:

  STMS → IMPORT OVERVIEW → DISPLAY

- Ensure that VBxxx tables are in separate 4K tablespaces:

  SE14 → VBxxx → EDIT → Storage Parameters → Tablespace

  – Tablespace name = table name
  – Buffer pool should not be BP2 (we recommend BP4)

- Ensure that only primary key index is defined:

  SE14 → VBxxx → EDIT → Storage parameters → Index → Index

  – Name suffix = 0
  – UniqueRule = P
  – Buffer pool should not be BP3 (we recommend BP4)

- Ensure that VBxxx tables are partitioned properly:

  SELECT NAME, PARTITIONS, LOCKPART FROM SYSIBM.SYSTABLESPACE WHERE NAME IN (‘VBHDR’, ‘VBMOD’, ‘VBDATA’)

  – PARTITIONS must be greater than 0
  – LOCKPART must = ‘Y’

- Ensure that SAP R/3 Profile is set up:

  RZ10

  – if SAP R/3 < 4.6 then check:

    rdisp/vb_key_comp = HOST/SYNR/WPNR/DATE/TIME/STMP

  – If SAP R/3 => 4.6 also need

dynp/trans_id_format = 2

- Ensure that DB2 catlog updates have been done:

  SELECT NAME, CARDF, NPAGES FROM SYSIBM.SYSTABLES WHERE CREATOR = 'SAPR3' AND NAME IN ('VBHDR', 'VBMOD', 'VBDATA')

  – CARDF = 50 & NPAGES = 10

  SELECT TSNAME, CARD, NPAGES FROM SYSIBM.SYSTABSTATS WHERE OWNER = 'SAPR3' AND TSNAME IN ('VBHDR', 'VBMOD', 'VBDATA')

  – CARDF = 50 & NPAGES = 10
  – If no records are found the table is not partitioned.

  SELECT NAME, NACTIVE FROM SYSIBM.SYSTABLESPACE WHERE CREATOR = 'SAPR3' AND NAME IN ('VBHDR', 'VBMOD', 'VBDATA')

  – NACTIVE = 10

  SELECT TBNAME, CLUSTERRATIO, FIRSTKEYCARDF, FULLKEYCARDF, NLEAF, NLEVELS

  FROM SYSIBM.SYSINDEXESWHERE TBCREATOR = 'SAPR3' AND TBNAME IN ('VBHDR', 'VBMOD', 'VBDATA')

  – CLUSTERRATION = 0, FIRSTKEYCARDF=-1, FULLKEYCARDF=-1, NLEAF=-1, & NLEVELS=-1

  SELECT TBNAME, COLCARDF, HEX(HIGH2KEY), HEX(LOW2KEY) FROM SYSIBM.SYSCOLUMNS WHERE TBCREATOR = 'SAPR3' AND TBNAME IN ('VBHDR', 'VBMOD', 'VBDATA')

  – COLCARDF=-1, HIGH2KEY='4040...', & LOW2KEY='4040...'

  SELECT COUNT(*)FROM SYSIBM.SYSCOLDISTWHERE TBOWNER = 'SAPR3' AND TBNAME IN ('VBHDR', 'VBMOD', 'VBDATA')

  – COUNT = 0

Check that logon load balancing has been implemented

- SMLG

Check batch scheduling groups

- SM61

Check that VB dispatching has been implemented

- Ensure that SAP R/3 Profile is set up:

  RZ10

  – rdisp/vb_dispatching = 1
  – rdisp/vb_included_server = null or not defined OR (acceptable but not preferred)
  – rdisp/vb_dispatching = 0
  – rdisp/vbname = $(rdisp/myname)

- Ensure that tables ASGRP & APSRV are initialized:

  SE16 Table Browser

  Do update groups align with logon load balancing groups?

DBM1 virtual storage

- Workprocess utilization:

  SM50 - cpu button: quick check on idle work processes

  ST03 analysis to estimate work process reduction potential

- MAXKEEPD
- Global Dynamic Statement Cache in data space

Sysplex failover setup

- Ensure that SAP R/3 Profile is set up:

  RZ10

  – dbs/db2/ssid
  – dbs/db2/ssid_standby
  – rsdb/db2_host_standby
  – rsdb/db2_port_standby
  – rsdb/reco_symmetric = ON
  – rsdb/reco_trials = 1
  – rsdb/reco_ping_cmd = <empty>
  – rsdb/reco_sleep_time = 1
  – rsdb/reco_sync_all_server = OFF
  – rsdb/reco_tcp_service = <empty>
  – dbs/db2/hosttcp_standby (=>SAP R/3 4.6x)
  – dbs/db2/planname_standby (=> SAP R/3 4.6x)

B

Systems managed duplexing

This appendix describes an example of system managed duplexing for LOCK structure and SCA.

Activating a policy including system duplexing

These are the steps the system performs when we activate a policy in which duplexing for LOCK structure and SCA is requested:

Example 12-1 Duplexing for LOCK structure and SCA

```
IXC536I DUPLEXING REBUILD OF STRUCTURE DBSG_LOCK1 571
  INITIATED.
  REASON: CHANGE IN CFRM ACTIVE POLICY DUPLEX OPTION
  IXC570I SYSTEM-MANAGED DUPLEXING REBUILD STARTED FOR STRUCTURE 572
  DBSG_LOCK1 IN COUPLING FACILITY ICF3
   PHYSICAL STRUCTURE VERSION: B93F2901 1266D406
   LOGICAL STRUCTURE VERSION: B93F2901 1266D406
   START REASON:  POLICY-INITIATED
   AUTO VERSION: B93F2AB7 EA859A63
  IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_GBP32K 573
  WAS NOT INITIATED BY MVS. REASON:
  CONNECTORS DO NOT SUPPORT SYSTEM-MANAGED PROCESSES
  IXC536I DUPLEXING REBUILD OF STRUCTURE DBSG_SCA 574
  INITIATED.
  REASON: CHANGE IN CFRM ACTIVE POLICY DUPLEX OPTION
  IXC570I SYSTEM-MANAGED DUPLEXING REBUILD STARTED FOR STRUCTURE 575
  DBSG_SCA IN COUPLING FACILITY ICF3
   PHYSICAL STRUCTURE VERSION: B93F28FF 4305CDA4
   LOGICAL STRUCTURE VERSION: B93F28FF 4305CDA4
   START REASON:  POLICY-INITIATED
   AUTO VERSION: B93F2AB8 0854C545
  IXC538I DUPLEXING REBUILD OF STRUCTURE MQSGCSQ_ADMIN 576
  WAS NOT INITIATED BY MVS. REASON:
  STRUCTURE WITH NO CONNECTORS HAS NEVER BEEN SYSTEM-MANAGED DUPLEXED
  IXC536I DUPLEXING REBUILD OF STRUCTURE MQSGTEST 577
  INITIATED.
  REASON: CHANGE IN CFRM ACTIVE POLICY DUPLEX OPTION
  IXC570I SYSTEM-MANAGED DUPLEXING REBUILD STARTED FOR STRUCTURE 578
  MQSGTEST IN COUPLING FACILITY ICF3
   PHYSICAL STRUCTURE VERSION: B93ED74C F7A29726
   LOGICAL STRUCTURE VERSION: B93ED74C F7A29726
   START REASON:  POLICY-INITIATED
   AUTO VERSION: B93F2AB8 2BF990A5
  IXC538I DUPLEXING REBUILD OF STRUCTURE STRIRSD 579
  WAS NOT INITIATED BY MVS. REASON:
  DUPLEXING REBUILD NOT ALLOWED FOR THE STRUCTURE
  IXC578I SYSTEM-MANAGED DUPLEXING REBUILD SUCCESSFULLY ALLOCATED 580
  STRUCTURE MQSGTEST.
  OLD COUPLING FACILITY: ICF3
  OLD PHYSICAL STRUCTURE VERSION: B93ED74C F7A29726
  NEW COUPLING FACILITY: ICF4
  NEW PHYSICAL STRUCTURE VERSION: B93F2AB8 B57BC345
```

```
LOGICAL STRUCTURE VERSION: B93ED74C F7A29726
 AUTO VERSION: B93F2AB8 2BF990A5
IXC574I ALLOCATION INFORMATION FOR SYSTEM-MANAGED DUPLEXING REBUILD
581
OF STRUCTURE MQSGTEST
 AUTO VERSION: B93F2AB8 2BF990A5
 CFNAME     STATUS/FAILURE REASON
 --------   ---------------------
 ICF3       RESTRICTED BY REBUILD OTHER
 ICF4       STRUCTURE ALLOCATED
                                   INFO110: 00000002 CE000800 00000000
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 582
MQSGTEST HAS COMPLETED THE ALLOCATION PHASE
AND IS ENTERING THE COPY PHASE.
 TIME: 04/09/2003 18:40:13.553408
 AUTO VERSION: B93F2AB8 2BF990A5
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 583
ISTGENERIC HAS COMPLETED THE STARTUP PHASE
AND IS ENTERING THE QUIESCE PHASE.
 TIME: 04/09/2003 18:40:13.700810
 AUTO VERSION: B93F2AB7 90CAC701
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 584
DBSG_LOCK1 HAS COMPLETED THE STARTUP PHASE
AND IS ENTERING THE QUIESCE PHASE.
 TIME: 04/09/2003 18:40:13.742211
 AUTO VERSION: B93F2AB7 EA859A63
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 585
DBSG_SCA HAS COMPLETED THE STARTUP PHASE
AND IS ENTERING THE QUIESCE PHASE.
 TIME: 04/09/2003 18:40:13.798027
 AUTO VERSION: B93F2AB8 0854C545
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 586
ISTGENERIC HAS COMPLETED THE QUIESCE PHASE
AND IS ENTERING THE ALLOCATION PHASE.
 TIME: 04/09/2003 18:40:14.473308
 AUTO VERSION: B93F2AB7 90CAC701
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 587
DBSG_LOCK1 HAS COMPLETED THE QUIESCE PHASE
AND IS ENTERING THE ALLOCATION PHASE.
 TIME: 04/09/2003 18:40:14.522272
 AUTO VERSION: B93F2AB7 EA859A63
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 588
DBSG_SCA HAS COMPLETED THE QUIESCE PHASE
AND IS ENTERING THE ALLOCATION PHASE.
 TIME: 04/09/2003 18:40:14.607838
 AUTO VERSION: B93F2AB8 0854C545
IXC578I SYSTEM-MANAGED DUPLEXING REBUILD SUCCESSFULLY ALLOCATED 589
STRUCTURE ISTGENERIC.
OLD COUPLING FACILITY: ICF4
```

```
OLD PHYSICAL STRUCTURE VERSION: B93F28FE 2245E24C
NEW COUPLING FACILITY: ICF3
NEW PHYSICAL STRUCTURE VERSION: B93F2ABA 2CFD86A6
LOGICAL STRUCTURE VERSION: B93F28FE 2245E24C
 AUTO VERSION: B93F2AB7 90CAC701
IXC574I ALLOCATION INFORMATION FOR SYSTEM-MANAGED DUPLEXING REBUILD
590
OF STRUCTURE ISTGENERIC
 AUTO VERSION: B93F2AB7 90CAC701
 CFNAME     STATUS/FAILURE REASON
 --------   ---------------------
 ICF4       RESTRICTED BY REBUILD OTHER
 ICF3       STRUCTURE ALLOCATED
                                   INFO110: 00000002 CE001800 00000000
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 591
ISTGENERIC HAS COMPLETED THE ALLOCATION PHASE
AND IS ENTERING THE ATTACH PHASE.
 TIME: 04/09/2003 18:40:15.040593
 AUTO VERSION: B93F2AB7 90CAC701
IXC578I SYSTEM-MANAGED DUPLEXING REBUILD SUCCESSFULLY ALLOCATED 592
STRUCTURE DBSG_LOCK1.
OLD COUPLING FACILITY: ICF3
OLD PHYSICAL STRUCTURE VERSION: B93F2901 1266D406
NEW COUPLING FACILITY: ICF4
NEW PHYSICAL STRUCTURE VERSION: B93F2ABA C5007CC7
LOGICAL STRUCTURE VERSION: B93F2901 1266D406
 AUTO VERSION: B93F2AB7 EA859A63
IXC574I ALLOCATION INFORMATION FOR SYSTEM-MANAGED DUPLEXING REBUILD
593
OF STRUCTURE DBSG_LOCK1
 AUTO VERSION: B93F2AB7 EA859A63
 CFNAME     STATUS/FAILURE REASON
 --------   ---------------------
 ICF3       RESTRICTED BY REBUILD OTHER
 ICF4       STRUCTURE ALLOCATED
                                   INFO110: 00000001 CE007800 00000000
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 594
DBSG_LOCK1 HAS COMPLETED THE ALLOCATION PHASE
AND IS ENTERING THE ATTACH PHASE.
 TIME: 04/09/2003 18:40:15.646698
 AUTO VERSION: B93F2AB7 EA859A63
IXC578I SYSTEM-MANAGED DUPLEXING REBUILD SUCCESSFULLY ALLOCATED 595
STRUCTURE DBSG_SCA.
OLD COUPLING FACILITY: ICF3
OLD PHYSICAL STRUCTURE VERSION: B93F28FF 4305CDA4
NEW COUPLING FACILITY: ICF4
NEW PHYSICAL STRUCTURE VERSION: B93F2ABB 2C281E64
LOGICAL STRUCTURE VERSION: B93F28FF 4305CDA4
 AUTO VERSION: B93F2AB8 0854C545
```

```
IXC574I ALLOCATION INFORMATION FOR SYSTEM-MANAGED DUPLEXING REBUILD
596
OF STRUCTURE DBSG_SCA
 AUTO VERSION: B93F2AB8 0854C545
 CFNAME     STATUS/FAILURE REASON
 --------   ---------------------
 ICF3       RESTRICTED BY REBUILD OTHER
 ICF4       STRUCTURE ALLOCATED
                                   INFO110: 00000001 CE007800 00000000
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 597
DBSG_SCA HAS COMPLETED THE ALLOCATION PHASE
AND IS ENTERING THE ATTACH PHASE.
 TIME: 04/09/2003 18:40:16.004431
 AUTO VERSION: B93F2AB8 0854C545
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 598
ISTGENERIC HAS COMPLETED THE ATTACH PHASE
AND IS ENTERING THE COPY PHASE.
 TIME: 04/09/2003 18:40:16.151053
 AUTO VERSION: B93F2AB7 90CAC701
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 599
DBSG_LOCK1 HAS COMPLETED THE ATTACH PHASE
AND IS ENTERING THE COPY PHASE.
 TIME: 04/09/2003 18:40:16.333892
 AUTO VERSION: B93F2AB7 EA859A63
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 600
DBSG_SCA HAS COMPLETED THE ATTACH PHASE
AND IS ENTERING THE COPY PHASE.
 TIME: 04/09/2003 18:40:16.622587
 AUTO VERSION: B93F2AB8 0854C545
IXC572I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 601
DBSG_LOCK1 HAS COMPLETED THE INITIALIZATION
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
ATTACH SUBPHASE.
 TIME: 04/09/2003 18:40:17.257238
 AUTO VERSION: B93F2AB7 EA859A63
IXC572I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 602
DBSG_LOCK1 HAS COMPLETED THE ATTACH
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
LIST SUBPHASE.
 TIME: 04/09/2003 18:40:17.280701
 AUTO VERSION: B93F2AB7 EA859A63
IXC572I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 603
DBSG_SCA HAS COMPLETED THE INITIALIZATION
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
ATTACH SUBPHASE.
 TIME: 04/09/2003 18:40:17.833504
 AUTO VERSION: B93F2AB8 0854C545
IXC572I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 604
DBSG_SCA HAS COMPLETED THE ATTACH
```

```
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
LIST SUBPHASE.
 TIME: 04/09/2003 18:40:17.934603
 AUTO VERSION: B93F2AB8 0854C545
IXC572I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 605
DBSG_SCA HAS COMPLETED THE LIST
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
EXIT SUBPHASE.
 TIME: 04/09/2003 18:40:22.875742
 AUTO VERSION: B93F2AB8 0854C545
IXC572I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 606
DBSG_LOCK1 HAS COMPLETED THE LIST
SUBPHASE OF THE COPY PHASE AND IS ENTERING THE
EXIT SUBPHASE.
 TIME: 04/09/2003 18:40:44.817045
 AUTO VERSION: B93F2AB7 EA859A63
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 607
DBSG_LOCK1 HAS COMPLETED THE COPY PHASE
AND IS ENTERING THE DUPLEX ESTABLISHED PHASE.
 TIME: 04/09/2003 18:40:45.687861
 AUTO VERSION: B93F2AB7 EA859A63
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 608
REACHED THE DUPLEX ESTABLISHED PHASE FOR STRUCTURE DBSG_LOCK1
STRUCTURE IS DUPLEXED
 OLD PHYSICAL STRUCTURE VERSION: B93F2901 1266D406
 NEW PHYSICAL STRUCTURE VERSION: B93F2ABA C5007CC7
 LOGICAL STRUCTURE VERSION: B93F2901 1266D406
 AUTO VERSION: B93F2AB7 EA859A63
```

ICF failure

We describe here the reaction of the system when we deactivate the ICF holding the primary LOCK structure.

As expected, the secondary structures take control and the workload on the rest of the member can continue without disruption.

Following are the syslog messages recorded during the ICF failure. For consistency we do not exclude the rest of system messages.

Example 12-2 Failure of ICF

```
12933  DATA SET     101  SYSTEM ID SYSM DATE 04/22/2003 2003.112 LINE  38,957
PAGE       1
 *IXL158I PATH DA IS NOW NOT-OPERATIONAL TO CUID: FFF1 119
                    COUPLING FACILITY 002064.IBM.83.00000001055A
                                      PARTITION: 5  CPCID: 00
```

```
*IXL158I PATH DB IS NOW NOT-OPERATIONAL TO CUID: FFF1 120
                   COUPLING FACILITY 002064.IBM.83.00000001055A
                                     PARTITION: 5  CPCID: 00
 IXC467I RESTARTING PATHOUT STRUCTURE IXCFT01 LIST 8 121
         USED TO COMMUNICATE WITH SYSTEM SYSN
         RSN: I/O ERROR.  DIAG038: 37
         DIAG073: 09210225 0000000C 0C1C0C06 00000000 00000000
 IXC467I REBUILDING PATH STRUCTURE IXCFT01 122
         RSN: LOST CONNECTIVITY TO FACILITY
         DIAG073: 08750001 0822C000 0100C000 00000000 00000000
 IXC467I STOPPING PATHOUT STRUCTURE IXCFT01 LIST 8 123
         USED TO COMMUNICATE WITH SYSTEM SYSN
         RSN: HALT I/O FAILED
         DIAG073: 08220004 0000000C 08750C02 0000000C 0C1C0C06
 IXC518I SYSTEM SYSM NOT USING 124
         COUPLING FACILITY  002064.IBM.83.00000001055A
                            PARTITION: 5    CPCID: 00
         NAMED ICF4
        REASON: CONNECTIVITY LOST.
        REASON FLAG: 13300002.
 IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 125
 ISTGENERIC IS BEING STOPPED
 TO SWITCH TO THE NEW STRUCTURE DUE TO
 LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
 IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 126
 ISTGENERIC HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
 AND IS ENTERING THE SWITCH PHASE.
  TIME: 04/22/2003 16:07:16.934350
  AUTO VERSION: B93F2AB7 90CAC701
 IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 127
 LOG_DFHLOG_001 IS BEING STOPPED
 TO SWITCH TO THE NEW STRUCTURE DUE TO
 LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
 IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 128
 LOG_DFHLOG_001 HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
 AND IS ENTERING THE SWITCH PHASE.
  TIME: 04/22/2003 16:07:16.940433
  AUTO VERSION: B94F554E EF3C1E43
 IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 129
 LOG_DFHSHUNT_001 IS BEING STOPPED
 TO SWITCH TO THE NEW STRUCTURE DUE TO
 LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
 IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 130
 LOG_DFHSHUNT_001 HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
 AND IS ENTERING THE SWITCH PHASE.
  TIME: 04/22/2003 16:07:16.946298
  AUTO VERSION: B94F55D6 02C21F66
 IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 131
 RRS_RMD IS BEING STOPPED
```

```
  TO SWITCH TO THE NEW STRUCTURE DUE TO
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,010  PAGE      2
  LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
  IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 132
  RRS_RMD HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
  AND IS ENTERING THE SWITCH PHASE.
   TIME: 04/22/2003 16:07:16.952550
   AUTO VERSION: B94F39B7 61551600
  IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 133
  RRS_RESTART IS BEING STOPPED
  TO SWITCH TO THE NEW STRUCTURE DUE TO
  LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
  IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 134
  RRS_RESTART HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
  AND IS ENTERING THE SWITCH PHASE.
   TIME: 04/22/2003 16:07:16.958401
   AUTO VERSION: B94F39BA CAA7E409
  IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 135
  RRS_ARCHIVE IS BEING STOPPED
  TO SWITCH TO THE NEW STRUCTURE DUE TO
  LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
  IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 136
  RRS_ARCHIVE HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
  AND IS ENTERING THE SWITCH PHASE.
   TIME: 04/22/2003 16:07:16.964396
   AUTO VERSION: B94F39BB DDD5A069
  IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 137
  DBSG_LOCK1 IS BEING STOPPED
  TO SWITCH TO THE NEW STRUCTURE DUE TO
  LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
  IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 138
  DBSG_LOCK1 HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
  AND IS ENTERING THE SWITCH PHASE.
   TIME: 04/22/2003 16:07:16.970405
   AUTO VERSION: B94F5423 53643787
  IXC522I REBUILD FOR STRUCTURE 139
  DBSG_GBP0 IS BEING STOPPED
  TO SWITCH TO THE NEW STRUCTURE DUE TO
  LOSS OF CONNECTIVITY TO THE STRUCTURE
  IXC522I REBUILD FOR STRUCTURE 140
  DBSG_GBP2 IS BEING STOPPED
  TO SWITCH TO THE NEW STRUCTURE DUE TO
  LOSS OF CONNECTIVITY TO THE STRUCTURE
  IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 141
  DBSG_SCA IS BEING STOPPED
  TO SWITCH TO THE NEW STRUCTURE DUE TO
  LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
  IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 142
  DBSG_SCA HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
```

```
AND IS ENTERING THE SWITCH PHASE.
 TIME: 04/22/2003 16:07:16.987048
 AUTO VERSION: B94F54A5 A746EB41
IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 143
MQSGCSQ_ADMIN IS BEING STOPPED
TO SWITCH TO THE NEW STRUCTURE DUE TO
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,063  PAGE       3
LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 144
MQSGCSQ_ADMIN HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
AND IS ENTERING THE SWITCH PHASE.
 TIME: 04/22/2003 16:07:16.992969
 AUTO VERSION: B94F5635 B0526882
IXC522I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 145
MQSGTEST IS BEING STOPPED
TO SWITCH TO THE NEW STRUCTURE DUE TO
LOSS OF CONNECTIVITY TO THE OLD STRUCTURE
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 146
MQSGTEST HAS COMPLETED THE DUPLEX ESTABLISHED PHASE
AND IS ENTERING THE SWITCH PHASE.
 TIME: 04/22/2003 16:07:16.999074
 AUTO VERSION: B94F56B4 0F9FDFAD
IXC521I REBUILD FOR STRUCTURE IXCFT01 147
HAS BEEN STARTED
IXC467I STARTED REBUILD FOR PATH STRUCTURE IXCFT01 148
        RSN: LOST CONNECTIVITY TO FACILITY
        DIAG073: 08750001 0822C000 0100C000 00000000 00000000
IXG219I SYSTEM LOGGER PROCESSED TRANSITION TO SIMPLEX MODE 149
FOR STRUCTURE LOG_DFHSHUNT_001
IXG219I SYSTEM LOGGER PROCESSED TRANSITION TO SIMPLEX MODE 150
FOR STRUCTURE RRS_RESTART
DSNB303E  *DBS1 DSNB1REE A LOSS OF CONNECTIVITY WAS 151
DETECTED
           TO GROUP BUFFER POOL GBP0
DSNB744I  *DBS1 DSNB1GBR DUPLEXING IS BEING STOPPED 152
FOR GROUP BUFFER POOL GBP0
           SWITCHING TO SECONDARY
           REASON = LOSSCONNPRI
           DB2 REASON CODE = 00000000
IXG219I SYSTEM LOGGER PROCESSED TRANSITION TO SIMPLEX MODE 153
FOR STRUCTURE RRS_ARCHIVE
DSNB303E  *DBS1 DSNB1REE A LOSS OF CONNECTIVITY WAS 154
DETECTED
           TO GROUP BUFFER POOL GBP2
DSNB744I  *DBS1 DSNB1GBR DUPLEXING IS BEING STOPPED 155
FOR GROUP BUFFER POOL GBP2
           SWITCHING TO SECONDARY
           REASON = LOSSCONNPRI
           DB2 REASON CODE = 00000000
```

```
  IXC521I REBUILD FOR STRUCTURE STRIRSD 156
  HAS BEEN STARTED
  IXC521I REBUILD FOR STRUCTURE STRVSAMS 157
  HAS BEEN STARTED
  DXR143I IRSD001 REBUILDING LOCK STRUCTURE BECAUSE IT HAS FAILED OR AN
  IRLM LOST CONNECTION TO IT
  D XCF
  IXC334I  16.07.52  DISPLAY XCF 161
     SYSPLEX BPEPLEXT:   SYSM              SYSN
  D XCF,CPL,TYPE=SYSPLEX
  IXC358I  16.08.17  DISPLAY XCF 164
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,116  PAGE       4
  SYSPLEX COUPLE DATA SETS
  PRIMARY    DSN: SYS1.XCF.GRS.TEST.CDS01
             VOLSER: SYS01M     DEVN: 0AB1
             FORMAT TOD         MAXSYSTEM MAXGROUP(PEAK) MAXMEMBER(PEAK)
             07/31/2002 11:11:48        8      50   (39)      123   (43)
             ADDITIONAL INFORMATION:
              ALL TYPES OF COUPLE DATA SETS SUPPORTED
              GRS STAR MODE IS SUPPORTED
  ALTERNATE  DSN: SYS1.XCF.GRS.TEST.CDS02
             VOLSER: SYS02M     DEVN: 09BE
             FORMAT TOD         MAXSYSTEM MAXGROUP       MAXMEMBER
             07/31/2002 11:11:51        8       50             123
             ADDITIONAL INFORMATION:
              ALL TYPES OF COUPLE DATA SETS SUPPORTED
              GRS STAR MODE IS SUPPORTED
 *DXR167E IRS1001 IRLM HAS DETECTED A DELAY IN COMPLETION OF ASID01FB
  PROCESS
  D R,R
  IEE112I 16.08.41 PENDING REQUESTS 168
  RM=1    IM=0     CEM=0     EM=0     RU=0    IR=0    NOAMRF
  ID:R/K     T SYSNAME  JOB ID   MESSAGE TEXT
        0037 R SYSM     STC13037 *0037 DSI802A CNM03    REPLY WITH VALID
                                  NCCF SYSTEM OPERATOR COMMAND
 *0048 IXC402D SYSN LAST OPERATIVE AT 16:07:07. REPLY DOWN AFTER SYSTEM
  RESET, OR INTERVAL=SSSSS TO SET A REPROMPT TIME.
  R 48,DOWN
  IEE600I REPLY TO 0048 IS;DOWN
  IXC101I SYSPLEX PARTITIONING IN PROGRESS FOR SYSN REQUESTED BY XCFAS.
  REASON: SYSTEM STATUS UPDATE MISSING
  IEA257I CONSOLE PARTITION CLEANUP IN PROGRESS FOR SYSTEM SYSN.
  IEE501I CONSOLE MAESN    FAILED, REASON=SFAIL .  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBEN0000 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBEN0003 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBENMQS1 FAILED, REASON=ABTERM.  ALL ALTERNATES
```

```
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE TSSTALA  FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBEN0001 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBEN0004 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBENMQS2 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE TSSTRIA  FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBCNC100 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBEN0002 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBEN0005 FAILED, REASON=ABTERM.  ALL ALTERNATES
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,169  PAGE      5
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEE501I CONSOLE BBENMQS3 FAILED, REASON=ABTERM.  ALL ALTERNATES
  UNAVAILABLE, CONSOLE IS NOT SWITCHED
  IEA258I CONSOLE PARTITION CLEANUP COMPLETE FOR SYSTEM SYSN.
  BBMZS152I A CAS has left the XCF group; SYSTEM=SYSN,JOB=BBENCAS,
  SSID=BBCN,MEMBER=SYSN
  BBMXCL62I Event Based Communications with     SYSN has terminated.
  IXC307I STOP PATHOUT REQUEST FOR STRUCTURE IXCFT01 194
          LIST 8 TO COMMUNICATE WITH SYSTEM SYSN COMPLETED
          SUCCESSFULLY: HALT I/O FAILED
  IXC467I STOPPING PATHOUT STRUCTURE IXCFT02 LIST 8 191
          USED TO COMMUNICATE WITH SYSTEM SYSN
  IXC307I STOP PATHIN REQUEST FOR STRUCTURE IXCFT01 195
          LIST 9 TO COMMUNICATE WITH SYSTEM SYSN COMPLETED
          SUCCESSFULLY: SYSPLEX PARTITIONING OF REMOTE SYSTEM
          RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

  IXC467I STOPPING PATHIN STRUCTURE IXCFT01 LIST 9 192
          USED TO COMMUNICATE WITH SYSTEM SYSN
          RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

  IXC467I STOPPING PATHIN STRUCTURE IXCFT02 LIST 9 193
  IXC105I SYSPLEX PARTITIONING HAS COMPLETED FOR SYSN 196
  - PRIMARY REASON: SYSTEM STATUS UPDATE MISSING
  - REASON FLAGS: 000008
          USED TO COMMUNICATE WITH SYSTEM SYSN
          RSN: SYSPLEX PARTITIONING OF REMOTE SYSTEM

  IXC307I STOP PATHIN REQUEST FOR STRUCTURE IXCFT02 197
          LIST 9 TO COMMUNICATE WITH SYSTEM SYSN COMPLETED
          SUCCESSFULLY: SYSPLEX PARTITIONING OF REMOTE SYSTEM
  IXC307I STOP PATHOUT REQUEST FOR STRUCTURE IXCFT02 198
```

```
        LIST 8 TO COMMUNICATE WITH SYSTEM SYSN COMPLETED
        SUCCESSFULLY: SYSPLEX PARTITIONING OF REMOTE SYSTEM
IXC808I ELEMENTS FROM TERMINATED SYSTEM SYSN WERE NOT PROCESSED BY 199
THIS SYSTEM.  ARM COUPLE DATA SET IS NOT AVAILABLE TO THIS SYSTEM.
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 200
DBSG_LOCK1 HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.160906
 AUTO VERSION: B94F5423 53643787
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 201
ISTGENERIC HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.284360
 AUTO VERSION: B93F2AB7 90CAC701
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 202
LOG_DFHLOG_001 HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.412613
 AUTO VERSION: B94F554E EF3C1E43
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 203
LOG_DFHSHUNT_001 HAS COMPLETED THE SWITCH PHASE
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,222  PAGE       6
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.540267
 AUTO VERSION: B94F55D6 02C21F66
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 204
RRS_RMD HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.667880
 AUTO VERSION: B94F39B7 61551600
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 205
RRS_RESTART HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.795174
 AUTO VERSION: B94F39BA CAA7E409
ISG345I CONTENTION NOTIFICATION LOST FOR RESOURCE 206
S=SYSTEMS,IOASYS  ,CTM.RES
GQSCAN RC=000C,RSN=000C
ISG345I CONTENTION NOTIFICATION LOST FOR RESOURCE 207
S=SYSTEMS,IOASYS  ,CTM.RES
GQSCAN RC=000C,RSN=000C
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 208
RRS_ARCHIVE HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
 TIME: 04/22/2003 16:08:58.922391
 AUTO VERSION: B94F39BB DDD5A069
IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 209
DBSG_SCA HAS COMPLETED THE SWITCH PHASE
AND IS ENTERING THE CLEANUP PHASE.
```

```
  TIME: 04/22/2003 16:08:59.045414
  AUTO VERSION: B94F54A5 A746EB41
 IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 210
 MQSGCSQ_ADMIN HAS COMPLETED THE SWITCH PHASE
 AND IS ENTERING THE CLEANUP PHASE.
  TIME: 04/22/2003 16:08:59.175027
  AUTO VERSION: B94F5635 B0526882
 IXC571I SYSTEM-MANAGED DUPLEXING REBUILD FOR STRUCTURE 211
 MQSGTEST HAS COMPLETED THE SWITCH PHASE
 AND IS ENTERING THE CLEANUP PHASE.
  TIME: 04/22/2003 16:08:59.311031
  AUTO VERSION: B94F56B4 0F9FDFAD
 IXC302I STOP PATHOUT REQUEST FOR STRUCTURE IXCFT02 212
         LIST 0 TO COMMUNICATE WITH SYSTEM SYSN REJECTED:
         UNKNOWN PATH
         DIAG037=18 DIAG074=08710000 RC,RSN=00000008 081A0004
 IXC302I STOP PATHIN REQUEST FOR STRUCTURE IXCFT02 213
         LIST 0 TO COMMUNICATE WITH SYSTEM SYSN REJECTED:
         UNKNOWN PATH
         DIAG037=18 DIAG074=08710000 RC,RSN=00000008 081A0004
 IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE ISGLOCK, 214
 CONNECTOR ISGLOCK#SYSM:
       0001002B
       00000000 00000000 00000007 00000000
       00000000 00000000 00000000 00000026
       00000000 00000000 00000000 00000000
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,275  PAGE       7
       00000001 00000000 00000002 00000000
       00000000 00000000 00000000 0000237A
       00000000 00000000 00000000 00000000
       00000002 00000000 00000003 00000000
       00000000 00000000 00000000 0000142C
       00000000 00000000 00000000 00000000
 IXL020I CLEANUP FOR LOCK STRUCTURE ISGLOCK, 215
 CONNECTION ID 02, STARTED BY CONNECTOR ISGLOCK#SYSM
 INFO: 0001 0002000F 00000020
 IXL021I GLOBAL CLEANUP FOR LOCK STRUCTURE ISGLOCK, 216
 CONNECTION ID 02, BY CONNECTOR ISGLOCK#SYSM
 HAS COMPLETED.
 INFO: 00000000 00000001 00000001
 IXL022I LOCAL CLEANUP FOR LOCK STRUCTURE ISGLOCK, 217
 CONNECTION ID 02, BY CONNECTOR ISGLOCK#SYSM
 HAS COMPLETED.
 INFO: 00000000 00000002 00000000
 IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 218
 BEEN COMPLETED FOR STRUCTURE DBSG_LOCK1
 STRUCTURE NOW IN COUPLING FACILITY ICF3
  PHYSICAL STRUCTURE VERSION: B94F5423 E919D185
  LOGICAL STRUCTURE VERSION: B94F5421 E4A34885
```

```
 AUTO VERSION: B94F5423 53643787
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 219
BEEN COMPLETED FOR STRUCTURE ISTGENERIC
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B93F2ABA 2CFD86A6
 LOGICAL STRUCTURE VERSION: B93F28FE 2245E24C
 AUTO VERSION: B93F2AB7 90CAC701
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 220
BEEN COMPLETED FOR STRUCTURE LOG_DFHLOG_001
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F5550 A5282920
 LOGICAL STRUCTURE VERSION: B94F5546 2B861066
 AUTO VERSION: B94F554E EF3C1E43
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 221
BEEN COMPLETED FOR STRUCTURE LOG_DFHSHUNT_001
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F55D6 E9DD6543
 LOGICAL STRUCTURE VERSION: B94F55CC 56E5972C
 AUTO VERSION: B94F55D6 02C21F66
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 222
BEEN COMPLETED FOR STRUCTURE RRS_RMD
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F39B8 946C6F66
 LOGICAL STRUCTURE VERSION: B94F39B7 21058280
 AUTO VERSION: B94F39B7 61551600
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 223
BEEN COMPLETED FOR STRUCTURE RRS_RESTART
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F39BC 5A860285
 LOGICAL STRUCTURE VERSION: B94F39BA 6FA67582
 AUTO VERSION: B94F39BA CAA7E409
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,328  PAGE      8
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 224
BEEN COMPLETED FOR STRUCTURE RRS_ARCHIVE
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F39BD E9073084
 LOGICAL STRUCTURE VERSION: B94F39BB 9E065369
 AUTO VERSION: B94F39BB DDD5A069
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 225
BEEN COMPLETED FOR STRUCTURE DBSG_SCA
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F54A6 3B7BBC01
 LOGICAL STRUCTURE VERSION: B94F54A4 844313A3
 AUTO VERSION: B94F54A5 A746EB41
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 226
BEEN COMPLETED FOR STRUCTURE MQSGCSQ_ADMIN
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F5636 49679D42
 LOGICAL STRUCTURE VERSION: B93ED74B 7EA69845
```

```
 AUTO VERSION: B94F5635 B0526882
IXC577I SYSTEM-MANAGED DUPLEXING REBUILD HAS 227
BEEN COMPLETED FOR STRUCTURE MQSGTEST
STRUCTURE NOW IN COUPLING FACILITY ICF3
 PHYSICAL STRUCTURE VERSION: B94F56B4 A35CFF2A
 LOGICAL STRUCTURE VERSION: B93ED74C F7A29726
 AUTO VERSION: B94F56B4 0F9FDFAD
IXC522I REBUILD FOR STRUCTURE 228
STRIRSD IS BEING STOPPED
TO FALL BACK TO THE OLD STRUCTURE DUE TO
CONNECTOR SPECIFIC REASON
 USER CODE: 00000067
IXL023I CLEANUP FOR LOCK STRUCTURE ISGLOCK, 229
CONNECTION ID 02, BY CONNECTOR ISGLOCK#SYSM
HAS COMPLETED.
INFO: 00000000 000001AD 00000000 00000000 00000000 00000000
      00100000 0000024D 00000000 00000000 00000000 0000024C
      00000000 00000000 00000000 00000000 00000000
DXR137I IRS1001 GROUP STATUS CHANGED. IRS2 002 HAS BEEN DISCONNECTED
FROM THE DATA SHARING GROUP
IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE DBSG_LOCK1, 231
CONNECTOR DBSGIRLM$IRS1001:
      00010018
      00000000 00000000 00000002 00000000
      00000000 00000000 00000000 000004A3
      00000000 00000000 00000000 00000000
      00000001 00000000 00000006 00000000
      00000000 00000000 00000000 000000AE
      00000000 00000000 00000000 00000000
      00000002 00000000 00000002 00000000
      00000000 00000000 00000000 0000020F
      00000000 00000000 00000000 00000000
IXL020I CLEANUP FOR LOCK STRUCTURE DBSG_LOCK1, 232
CONNECTION ID 02, STARTED BY CONNECTOR DBSGIRLM$IRS1001
INFO: 0001 00020012 00000040
IXL021I GLOBAL CLEANUP FOR LOCK STRUCTURE DBSG_LOCK1, 233
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,381  PAGE      9
CONNECTION ID 02, BY CONNECTOR DBSGIRLM$IRS1001
HAS COMPLETED.
INFO: 00000000 00000009 00000006
IXL022I LOCAL CLEANUP FOR LOCK STRUCTURE DBSG_LOCK1, 234
CONNECTION ID 02, BY CONNECTOR DBSGIRLM$IRS1001
HAS COMPLETED.
INFO: 00000008 00000002 00000000
IST1494I  PATH SWITCH STARTED   FOR RTP CNR0000E
IST1818I  PATH SWITCH REASON: SHORT REQUEST RETRY LIMIT EXHAUSTED
IST314I  END
IXG219I SYSTEM LOGGER PROCESSED TRANSITION TO SIMPLEX MODE 238
FOR STRUCTURE LOG_DFHLOG_001
```

```
IST1494I  PATH SWITCH FAILED    FOR RTP CNR0000E
IST1495I  NO ALTERNATE ROUTE AVAILABLE
IST314I  END
IXG219I SYSTEM LOGGER PROCESSED TRANSITION TO SIMPLEX MODE 242
FOR STRUCTURE RRS_RMD
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_LOCK1 243
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE ISTGENERIC 244
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE LOG_DFHLOG_001 245
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE LOG_DFHSHUNT_001 246
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_RMD 247
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_RESTART 248
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_ARCHIVE 249
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_SCA 250
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE MQSGCSQ_ADMIN 251
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE MQSGTEST 252
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
ATR222I LOG TAKEOVER FOR SYSTEM SYSN HAS COMPLETED SUCCESSFULLY.
IXC526I STRUCTURE IXCFT01 IS REBUILDING FROM 254
COUPLING FACILITY ICF4 TO COUPLING FACILITY ICF3.
REBUILD START REASON: CONNECTIVITY LOST TO STRUCTURE
INFO108: 00000001 00000000.
DXR168I IRS1001 DELAYED PROCESSES NO LONGER DETECTED
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,434  PAGE      10
IXL014I IXLCONN REBUILD REQUEST FOR STRUCTURE IXCFT01 256
WAS SUCCESSFUL.  JOBNAME: XCFAS ASID: 0006
CONNECTOR NAME: SIGPATH_010000DA CFNAME: ICF3
IXL015I REBUILD NEW STRUCTURE ALLOCATION INFORMATION FOR 257
STRUCTURE IXCFT01, CONNECTOR NAME SIGPATH_010000DA
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   ---------------------------------
```

```
 ICF4       INSUFFICIENT CONNECTIVITY 00000000
 ICF3       STRUCTURE ALLOCATED
IXC457I REBUILT STRUCTURE IXCFT01 ALLOCATED 258
WITH 64 LISTS WHICH SUPPORTS FULL SIGNALLING CONNECTIVITY
AMONG 8 SYSTEMS AND UP TO 1431 SIGNALS
IXC526I STRUCTURE STRVSAMS IS REBUILDING FROM 259
COUPLING FACILITY ICF4 TO COUPLING FACILITY ICF3.
REBUILD START REASON: CONNECTIVITY LOST TO STRUCTURE
INFO108: 00000001 00000000.
IXL014I IXLCONN REBUILD REQUEST FOR STRUCTURE STRVSAMS 260
WAS SUCCESSFUL.  JOBNAME: DLISASS1 ASID: 0083
CONNECTOR NAME: IXCLO0160001 CFNAME: ICF3
IXL015I REBUILD NEW STRUCTURE ALLOCATION INFORMATION FOR 261
STRUCTURE STRVSAMS, CONNECTOR NAME IXCLO0160001
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   ---------------------------------
 ICF4       INSUFFICIENT CONNECTIVITY 00000000
 ICF3       STRUCTURE ALLOCATED
IXC521I REBUILD FOR STRUCTURE DBSG_GBP0 262
HAS BEEN COMPLETED
IXC521I REBUILD FOR STRUCTURE DBSG_GBP2 263
HAS BEEN COMPLETED
IXC521I REBUILD FOR STRUCTURE STRIRSD 264
HAS BEEN STOPPED
DSNB745I  *DBS1 DSNB1GBR THE TRANSITION BACK TO 265
SIMPLEX MODE HAS COMPLETED FOR
           GROUP BUFFER POOL GBP0
DSNB338I  *DBS1 DSNB1GBR THE REBUILD FOR GROUP BUFFER 266
POOL GBP2 HAS COMPLETED SUCCESSFULLY
IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE STRIRSD, 267
CONNECTOR XCFIRSD$$IRSD001:
      00010009
      00000000 00000000 00000002 00000000
      00000000 00000000 00000000 000519A0
      00000000 00000000 00000000 00000000
      00000001 00000000 00000000 00000000
      00000000 00000000 00000000 00000000
      00000000 00000000 00000000 00000000
      00000002 00000000 00000002 00000000
      00000000 00000000 00000000 00051274
      00000000 00000000 00000000 00000000
IXL031I CONNECTOR CLEANUP FOR LOCK STRUCTURE STRIRSD, 268
CONNECTOR XCFIRSD$$IRSD001, HAS COMPLETED.
INFO: 00010009 00000000 00000000 00000000 00000000 00000004
DXR149I IRSD001 REBUILD OF LOCK STRUCTURE WAS STOPPED DUE TO FAILURE
OF A REQUIRED IRLM IN THE GROUP
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,487  PAGE      11
CSQE008I ?S1 Recovery event from MQS2 received for structure TEST
CSQE011I ?S1 Recovery phase 1 started for structure 271
```

```
TEST connection name CSQEMQSGMQS204
CSQE013I ?S1 Recovery phase 1 completed for structure 272
TEST connection name CSQEMQSGMQS204
IXC538I DUPLEXING REBUILD OF STRUCTURE ISTGENERIC 273
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_GBP0 274
WAS NOT INITIATED BY MVS. REASON:
NO COUPLING FACILITY PROVIDED BETTER OR EQUIVALENT CONNECTIVITY
IST1504I  XCF CONNECTION WITH ESPOP000.AONM     IS INOPERATIVE
IST1501I  XCF TOKEN = 0200001000140002
IST1578I  DEVICE INOP DETECTED FOR ISTTOMON BY ISTTSCXT CODE = 100
IST314I  END
IST1578I  SOFT   INOP DETECTED FOR ISTTOMON BY ISTTSC8X CODE = 005
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_GBP2 280
WAS NOT INITIATED BY MVS. REASON:
NO COUPLING FACILITY PROVIDED BETTER OR EQUIVALENT CONNECTIVITY
IST1097I  CP-CP SESSION WITH ESPOP000.AONM     TERMINATED
IST1280I  SESSION TYPE = CONWINNER - SENSE = 80020000
IST314I  END
IST1494I  PATH SWITCH STARTED   FOR RTP CNR0000F
IST1819I  PATH SWITCH REASON: TG INOP
IST314I  END
IST1494I  PATH SWITCH STARTED   FOR RTP CNR00011
IST1819I  PATH SWITCH REASON: TG INOP
IST314I  END
IST1196I  APPN CONNECTION FOR ESPOP000.AONM     INACTIVE - TGN =  21
IXC538I DUPLEXING REBUILD OF STRUCTURE STRIRSD 291
WAS NOT INITIATED BY MVS. REASON:
DUPLEXING REBUILD NOT ALLOWED FOR THE STRUCTURE
IST1097I  CP-CP SESSION WITH ESPOP000.AONM     TERMINATED
IST1280I  SESSION TYPE = CONLOSER  - SENSE = 08420001
IST314I  END
IST1488I  INACTIVATION OF RTP CNR0000E AS ACTIVE  TO ESPOP000.AONM
IST619I  ID = CNR0000E FAILED - RECOVERY IN PROGRESS
IST129I  UNRECOVERABLE OR FORCED ERROR ON NODE CNR0000E - VARY INACT
SCHED
IST259I  INOP RECEIVED FOR ISTPOMON CODE = 01
IST493I  VARY INACT FOR ID = ISTPOMON OVERRIDDEN BY HARD INOP
IST619I  ID = ISTPOMON FAILED - RECOVERY IN PROGRESS
IST129I  UNRECOVERABLE OR FORCED ERROR ON NODE ISTPOMON - VARY INACT
SCHED
IST1494I  PATH SWITCH FAILED    FOR RTP CNR0000F
IST1495I  NO ALTERNATE ROUTE AVAILABLE
IST314I  END
IST1488I  INACTIVATION OF RTP CNR00010 AS PASSIVE TO ESPOP000.AONM
IST619I  ID = CNR00010 FAILED - RECOVERY IN PROGRESS
IST129I  UNRECOVERABLE OR FORCED ERROR ON NODE CNR00010 - VARY INACT
SCHED
```

```
  IST105I  CNR00010 NODE NOW INACTIVE
  IXC521I REBUILD FOR STRUCTURE STRVSAMS 309
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,540  PAGE      12
  HAS BEEN COMPLETED
  .HASP100 MQPUTA   ON INTRDR      DB2                   FROM TSU13857
  TSSTDEM
  IRR010I  USERID TSSTDEM  IS ASSIGNED TO THIS JOB.
  DXR136I IRSD001 HAS DISCONNECTED FROM THE DATA SHARING GROUP
  .HASP101 MQPUTA   HELD
  IXG303I DIRECTED OFFLOAD FOR LOGSTREAM ATR.BPEPLEXT.RM.DATA STARTED.
  DFS2011I IRLM FAILURE - IMS QUIESCING DCS1
  IST871I  RESOURCE CNR00010 DELETED
  CSQE008I ?S1 Recovery event from MQS2 received for structure TEST
  CSQE012I ?S1 Recovery phase 2 started for structure 318
  TEST connection name CSQEMQSGMQS204
  IXG304I DIRECTED OFFLOAD FOR LOGSTREAM ATR.BPEPLEXT.RM.DATA IS
  COMPLETE.
  IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_ARCHIVE 320
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_RESTART 321
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXG209I RECOVERY FOR LOGSTREAM ATR.BPEPLEXT.RM.DATA 322
  IN STRUCTURE RRS_RMD COMPLETED SUCCESSFULLY.
  IST1488I  INACTIVATION OF RTP CNR0000F AS PASSIVE TO ESPOP000.A0NM
  IST619I  ID = CNR0000F FAILED - RECOVERY IN PROGRESS
  IST129I  UNRECOVERABLE OR FORCED ERROR ON NODE CNR0000F - VARY INACT
  SCHED
  IST105I  ISTPOMON NODE NOW INACTIVE
  IST105I  CNR0000E NODE NOW INACTIVE
  IST871I  RESOURCE CNR0000E DELETED
  IST105I  CNR0000F NODE NOW INACTIVE
  IST871I  RESOURCE CNR0000F DELETED
  LOGON
  IXC521I REBUILD FOR STRUCTURE IXCFT01 332
  HAS BEEN COMPLETED
  IST663I  INIT OTHER REQUEST               FAILED  , SENSE=80140001
  IST664I  REAL  OLU=ESPOP000.A03M       REAL  DLU=ESPOP000.A0NM
  IST889I  SID = E093E35E93163E70
  IST314I  END
  IST1110I  ACTIVATION OF CP-CP SESSION WITH ESPOP000.A0NM     FAILED
  IST1280I  SESSION TYPE = CONWINNER - SENSE = 80140001
  IST1002I  RCPRI=0004 RCSEC=0000
  IST314I  END
  IXC465I REBUILD REQUEST FOR STRUCTURE IXCFT01 WAS SUCCESSFUL 341
  WHY REBUILT:
          LOST CONNECTIVITY TO FACILITY
```

```
  IXC302I STOP PATHOUT REQUEST FOR STRUCTURE IXCFT01 342
          LIST 0 TO COMMUNICATE WITH SYSTEM SYSN REJECTED:
          UNKNOWN PATH
          DIAG037=18 DIAG074=08710080 RC,RSN=00000008 081A0004
  IXC302I STOP PATHIN REQUEST FOR STRUCTURE IXCFT01 343
          LIST 0 TO COMMUNICATE WITH SYSTEM SYSN REJECTED:
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,593  PAGE      13
          UNKNOWN PATH
          DIAG037=18 DIAG074=08710080 RC,RSN=00000008 081A0004
  IXC538I DUPLEXING REBUILD OF STRUCTURE LOG_DFHSHUNT_001 344
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE LOG_DFHLOG_001 345
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_RMD 346
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  .AJ(13874)
  .HASP890 JOB(MQPUTA) 348
  .HASP890 JOB(MQPUTA)    STATUS=(AWAITING EXECUTION),CLASS=G,
  .HASP890                PRIORITY=9,SYSAFF=(SYSM),HOLD=(NONE)
  ICH70001I TSSTDEM  LAST ACCESS AT 16:07:02 ON TUESDAY, APRIL 22, 2003
  .HASP373 MQPUTA   STARTED - INIT 04   - CLASS G - SYS SYSM
  IEF403I MQPUTA - STARTED - TIME=16.09.19
  .HASP100 TSSTRIA  ON TSOINRDR
  .HASP373 TSSTRIA  STARTED
  IEF125I TSSTRIA - LOGGED ON - TIME=16.09.20
  IXL023I CLEANUP FOR LOCK STRUCTURE DBSG_LOCK1, 355
  CONNECTION ID 02, BY CONNECTOR DBSGIRLM$IRS1001
  HAS COMPLETED.
  INFO: 00000000 00000802 00000000 00000000 00000000 00000000
        00400000 00000905 00000000 00000000 00000000 00000904
        00000000 00000000 00000000 00000000 00000000
 *DSNB325A  *DBS1 DSNB5STS THERE IS A CRITICAL SHORTAGE 356
  OF SPACE IN GROUP BUFFER POOL GBP0
  IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_SCA 357
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  CSQJ002I ?S1 END OF ACTIVE LOG DATA SET 358
  DSNAME=CSQT.MQS1.LOGCOPY1.DS03, STARTRBA=00000237E000
  ENDRBA=000002A85FFF
  CSQJ001I ?S1 CURRENT COPY 1 ACTIVE LOG DATA SET IS 359
  DSNAME=CSQT.MQS1.LOGCOPY1.DS04, STARTRBA=000002A86000
  ENDRBA=00000318DFFF
  CSQJ002I ?S1 END OF ACTIVE LOG DATA SET 360
  DSNAME=CSQT.MQS1.LOGCOPY2.DS03, STARTRBA=00000237E000
  ENDRBA=000002A85FFF
```

```
 CSQP018I ?S1 CSQPBCKW CHECKPOINT STARTED FOR ALL BUFFER POOLS
 CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 362
 POOL 1, 0 PAGES WRITTEN
 CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 363
 POOL 0, 0 PAGES WRITTEN
 CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 364
 POOL 2, 0 PAGES WRITTEN
 CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 365
 POOL 3, 0 PAGES WRITTEN
 ?S1 DISPLAY THREAD(*) TYPE(INDOUBT)
 CSQV401I ?S1 DISPLAY THREAD REPORT FOLLOWS -
 CSQV420I ?S1 NO INDOUBT THREADS FOUND
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,646  PAGE      14
 CSQ9022I ?S1 CSQVDT ' DISPLAY THREAD' NORMAL COMPLETION
 CSQP021I ?S1 Page set 0 new media recovery 370
 RBA=0000025BCCB4, checkpoint RBA=0000025BCCB4
 CSQP021I ?S1 Page set 1 new media recovery 371
 RBA=0000025BD713, checkpoint RBA=0000025BD713
 CSQP021I ?S1 Page set 2 new media recovery 372
 RBA=000002AD64A4, checkpoint RBA=000002AD64A4
 CSQP021I ?S1 Page set 3 new media recovery 373
 RBA=0000025BD619, checkpoint RBA=0000025BD619
 CSQP021I ?S1 Page set 4 new media recovery 374
 RBA=0000025C1522, checkpoint RBA=0000025C1522
 CSQJ001I ?S1 CURRENT COPY 2 ACTIVE LOG DATA SET IS 375
 DSNAME=CSQT.MQS1.LOGCOPY2.DS04, STARTRBA=000002A86000
 ENDRBA=00000318DFFF
 CSQJ072E ?S1 ARCHIVE LOG DATA SET 376
 'CSQTLOG1.MQS1.E03112.T1609259.B0000007' HAS BEEN ALLOCATED TO
 NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
 CSQJ072E ?S1 ARCHIVE LOG DATA SET 377
 'CSQTLOG1.MQS1.E03112.T1609259.A0000007' HAS BEEN ALLOCATED TO
 NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
 CSQJ072E ?S1 ARCHIVE LOG DATA SET 378
 'CSQTLOG2.MQS1.E03112.T1609259.B0000007' HAS BEEN ALLOCATED TO
 NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
 CSQJ072E ?S1 ARCHIVE LOG DATA SET 379
 'CSQTLOG2.MQS1.E03112.T1609259.A0000007' HAS BEEN ALLOCATED TO
 NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
 CSQJ033I ?S1 FULL ARCHIVE LOG VOLUME 380
 DSNAME=CSQTLOG1.MQS1.E03112.T1609259.A0000007, STARTRBA=00000237E000
 ENDRBA=000002A85FFF, STARTLRSN=B94F367B7370 ENDLRSN=B94F61432563,
 UNIT=SYSDA, COPY1VOL=SMST10, VOLSPAN=NO CATLG=YES
 CSQJ033I ?S1 FULL ARCHIVE LOG VOLUME 381
 DSNAME=CSQTLOG2.MQS1.E03112.T1609259.A0000007, STARTRBA=00000237E000
 ENDRBA=000002A85FFF, STARTLRSN=B94F367B7370 ENDLRSN=B94F61432563,
 UNIT=SYSDA, COPY2VOL=SMST12, VOLSPAN=NO CATLG=YES
 CSQJ139I ?S1 LOG OFFLOAD TASK ENDED
 +CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
```

```
CSQE014I ?S1 Recovery phase 2 completed for structure 384
TEST connection name CSQEMQSGMQS204
IXC538I DUPLEXING REBUILD OF STRUCTURE MQSGTEST 385
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
D A,L
IEE114I 16.09.29 2003.112 ACTIVITY 387
 JOBS     M/S    TS USERS    SYSAS    INITS   ACTIVE/MAX VTAM     OAS
00006    00041    00011      00030    00078    00011/00020       00017
 LLA      LLA      LLA      NSW  S  VLF      VLF      VLF      NSW  S
 APPC     APPC     APPC     NSW  S  ASCH     ASCH     ASCH     NSW  S
 RMF      RMF      IEFPROC  NSW  S  SDSF     SDSF     SDSF     NSW  S
 ISGECMON ISGECMON ISGECMON OWT  S  TSO      TSO      STEP1    OWT  S
 NET      NET      NET      NSW  S  OSASF    OSASF    OSASF1   IN   S
 BBEMSRM  BBEMSRM  SVOS     NSW  SO CTMCMEM  CTMCMEM  MONITOR  NSW  S
 CSFM     CSFM              NSW  S  JES2     JES2     IEFPROC  NSW  S
 RACF     RACF     RACF     NSW  S  OAM      OAM      IEFPROC  NSW  S
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,699  PAGE      15
 INETD4   STEP1    OMVSKERN OWT  AO TCPIP    TCPIP    TCPIP    NSW  SO
 OMPROUTE OMPROUTE *OMVSEX  IN   SO FTPD1    STEP1    UTCPIP   OWT  AO
 FTP2D1   STEP1    UTCPIP   OWT  AO DBIRLMS1 DBIRLMS1          NSW  S
 DBCTLS1  DBCTLS1  IEFPROC  NSW  S  DLISASS1 DLISASS1 IEFPROC  NSW  S
 DBRCS1   DBRCS1   IEFPROC  NSW  S  CNMPSSIM CNMPSSIM NETVIEW  NSW  S
 CNMP03   CNMP03   NETVIEWM NSW  SO CONTROLM CONTROLM CTM      NSW  S
 BBEM     BBEM     BBEM     NSW  SO BBEMCAS  BBEMCAS  CAS      NSW  SO
 BBEMPAS  BBEMPAS  PAS      NSW  SO HSMM     HSMM     DFHSM    NSW  S
 IMWEBSRB IMWEBSRB WEBSRVB  IN   SO CONTROLD CONTROLD MONITOR  NSW  S
 CTDPRINT CTDPRINT CTDPRINT NSW  S  RRS      RRS      RRS      NSW  S
 CICSTEST CICSTEST CICS     NSW  SO MQS1MSTR MQS1MSTR PROCSTEP NSW  S
 DBS1MSTR DBS1MSTR IEFPROC  NSW  S  DBS1IRLM DBS1IRLM          NSW  S
 DBS1DBM1 DBS1DBM1 IEFPROC  NSW  S  DBS1DIST DBS1DIST IEFPROC  NSW  SO
 DBS1SPAS DBS1SPAS IEFPROC  NSW  S  MQS1CHIN MQS1CHIN PROCSTEP IN   SO
 TSST026P BDP0132           IN   J  MQPUT    PUTMSGS           IN   J
 MQPUTA   PUTMSGS           IN   J
TSSTFRF OWT  O   TSSTIBB OWT      TSSTPAA IN       TSSTIBA OWT
TSSTFDC OWT      TSSTEIL OWT      TSSTDEM OWT      TSSTVGG OWT
TSST100 OWT      TSST090 OWT      TSSTRIA OWT
.HASP100 MQPUTA   ON INTRDR      DB2                  FROM TSU13857
TSSTDEM
IRR010I  USERID TSSTDEM  IS ASSIGNED TO THIS JOB.
*IXC585E STRUCTURE MQSGTEST IN COUPLING FACILITY ICF3, 390
PHYSICAL STRUCTURE VERSION B94F56B4 A35CFF2A,
IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF  80%.
ENTRIES:  IN-USE:       5342 TOTAL:       7229,  73% FULL
ELEMENTS: IN-USE:      42556 TOTAL:      42558,  99% FULL
EMCS:     IN-USE:         11 TOTAL:       3940,   0% FULL
-                                                --TIMINGS (MINS.)--
          ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
```

```
 SERV  PG  PAGE  SWAP   VIO SWAPS
-MQPUTA   PUTMSGS          $ 2192@    98     70    .00    .00     .2
 8431   0     0     0     0     0
IEF404I MQPUTA - ENDED - TIME=16.09.34
-MQPUTA   ENDED.  NAME-DB2                  TOTAL TCB CPU TIME=   .00
 TOTAL ELAPSED TIME=    .2
.HASP395 MQPUTA   ENDED
.HASP309 INIT 04   INACTIVE ******** C=VG
SE '16.09.34 JOB13874 $HASP165 MQPUTA   ENDED AT BPET  MAXCC=2192',
LOGON,USER=(TSSTDEM)
-                                                    --TIMINGS (MINS.)--
          ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
 SERV  PG  PAGE  SWAP   VIO SWAPS
-MQPUT    PUTMSGS          $ 2192@    97     65    .01    .00    2.5
15391   0     0     0     0     0
IEF404I MQPUT - ENDED - TIME=16.09.34
-MQPUT    ENDED.  NAME-DB2                  TOTAL TCB CPU TIME=   .01
 TOTAL ELAPSED TIME=   2.5
.HASP395 MQPUT    ENDED
.HASP309 INIT 03   INACTIVE ******** C=HG
SE '16.09.34 JOB13860 $HASP165 MQPUT    ENDED AT BPET  MAXCC=2192',
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,752  PAGE      16
LOGON,USER=(TSSTDEM)
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
DSNB327I  *DBS1 DSNB5STS GROUP BUFFER POOL GBP0 HAS ADEQUATE FREE
SPACE
.HASP100 MQPUTB   ON INTRDR      DB2                 FROM TSU13857
TSSTDEM
IRR010I  USERID TSSTDEM  IS ASSIGNED TO THIS JOB.
.CJ(13876),P
.CJ(13877),P
.HASP890 JOB(MQPUTA) 414
.HASP890 JOB(MQPUTA)    STATUS=(AWAITING OUTPUT),CLASS=G,
.HASP890                PRIORITY=9,SYSAFF=(ANY),HOLD=(NONE),
.HASP890                PURGE=YES,CANCEL=YES
.HASP890 JOB(MQPUTB) 415
.HASP890 JOB(MQPUTB)    STATUS=(AWAITING OUTPUT),CLASS=G,
.HASP890                PRIORITY=9,SYSAFF=(ANY),HOLD=(NONE),
.HASP890                PURGE=YES,CANCEL=YES
SE '16.10.19 JOB13876 $HASP165 MQPUTA   ENDED AT BPET - CANCELLED',
LOGON,USER=(TSSTDEM)
SE '16.10.19 JOB13877 $HASP165 MQPUTB   ENDED AT BPET - CANCELLED',
LOGON,USER=(TSSTDEM)
.HASP250 MQPUTA PURGED -- (JOB KEY WAS B94F614A)
.HASP250 MQPUTB PURGED -- (JOB KEY WAS B94F6150)
+CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
```

```
CSQJ002I ?S1 END OF ACTIVE LOG DATA SET 421
DSNAME=CSQT.MQS1.LOGCOPY1.DS04, STARTRBA=000002A86000
ENDRBA=00000318DFFF
CSQJ001I ?S1 CURRENT COPY 1 ACTIVE LOG DATA SET IS 422
DSNAME=CSQT.MQS1.LOGCOPY1.DS01, STARTRBA=00000318E000
ENDRBA=000003895FFF
CSQJ002I ?S1 END OF ACTIVE LOG DATA SET 423
DSNAME=CSQT.MQS1.LOGCOPY2.DS04, STARTRBA=000002A86000
ENDRBA=00000318DFFF
CSQJ001I ?S1 CURRENT COPY 2 ACTIVE LOG DATA SET IS 424
DSNAME=CSQT.MQS1.LOGCOPY2.DS01, STARTRBA=00000318E000
ENDRBA=000003895FFF
CSQP018I ?S1 CSQPBCKW CHECKPOINT STARTED FOR ALL BUFFER POOLS
CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 426
POOL 1, 0 PAGES WRITTEN
?S1 DISPLAY THREAD(*) TYPE(INDOUBT)
CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 428
POOL 2, 1 PAGES WRITTEN
CSQV401I ?S1 DISPLAY THREAD REPORT FOLLOWS -
CSQV420I ?S1 NO INDOUBT THREADS FOUND
CSQ9022I ?S1 CSQVDT ' DISPLAY THREAD' NORMAL COMPLETION
CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 432
POOL 0, 13 PAGES WRITTEN
CSQP021I ?S1 Page set 0 new media recovery 433
RBA=0000025BD5EB, checkpoint RBA=0000025BD5EB
CSQP019I ?S1 CSQP1DWP CHECKPOINT COMPLETED FOR BUFFER 434
POOL 3, 5 PAGES WRITTEN
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,805  PAGE      17
CSQP021I ?S1 Page set 1 new media recovery 435
RBA=000003193F56, checkpoint RBA=000003193F56
CSQP021I ?S1 Page set 2 new media recovery 436
RBA=000003193F56, checkpoint RBA=000003193F56
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
CSQP021I ?S1 Page set 3 new media recovery 439
RBA=000003179C26, checkpoint RBA=000003179C26
CSQP021I ?S1 Page set 4 new media recovery 440
RBA=000003193F56, checkpoint RBA=000003193F56
CSQJ072E ?S1 ARCHIVE LOG DATA SET 441
'CSQTLOG1.MQS1.E03112.T1610318.B0000008' HAS BEEN ALLOCATED TO
NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
CSQJ072E ?S1 ARCHIVE LOG DATA SET 442
'CSQTLOG1.MQS1.E03112.T1610318.A0000008' HAS BEEN ALLOCATED TO
NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
CSQJ072E ?S1 ARCHIVE LOG DATA SET 443
'CSQTLOG2.MQS1.E03112.T1610318.B0000008' HAS BEEN ALLOCATED TO
NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
CSQJ072E ?S1 ARCHIVE LOG DATA SET 444
```

```
 'CSQTLOG2.MQS1.E03112.T1610318.A0000008' HAS BEEN ALLOCATED TO
 NON-TAPE DEVICE AND CATALOGUED, OVERRIDING CATALOG PARAMETER
 CSQJ033I ?S1 FULL ARCHIVE LOG VOLUME 445
 DSNAME=CSQTLOG1.MQS1.E03112.T1610318.A0000008, STARTRBA=000002A86000
 ENDRBA=00000318DFFF, STARTLRSN=B94F61432564 ENDLRSN=B94F61824443,
 UNIT=SYSDA, COPY1VOL=SMST12, VOLSPAN=NO CATLG=YES
 CSQJ033I ?S1 FULL ARCHIVE LOG VOLUME 446
 DSNAME=CSQTLOG2.MQS1.E03112.T1610318.A0000008, STARTRBA=000002A86000
 ENDRBA=00000318DFFF, STARTLRSN=B94F61432564 ENDLRSN=B94F61824443,
 UNIT=SYSDA, COPY2VOL=SMST12, VOLSPAN=NO CATLG=YES
 CSQJ139I ?S1 LOG OFFLOAD TASK ENDED
 IEA630I  OPERATOR TSSTRIA  NOW ACTIVE,   SYSTEM=SYSM    , LU=LINP3019
 /F DBIRLMS1,STATUS,ALLD
 F BBEMSRM, F DBIRLMS1,STATUS,ALLD
 SV00501W UNKNOWN COMMAND F
 /D XCF,STR,STRNAME=STRIRSD
 F BBEMSRM, D XCF,STR,STRNAME=STRIRSD
 SV00501W UNKNOWN COMMAND D
 IST1494I  PATH SWITCH FAILED    FOR RTP CNR00011
 IST1495I  NO ALTERNATE ROUTE AVAILABLE
 IST314I  END
 IST1488I  INACTIVATION OF RTP CNR00011 AS ACTIVE  TO ESPOPOOO.AONM
 IST619I  ID = CNR00011 FAILED - RECOVERY IN PROGRESS
 IST129I  UNRECOVERABLE OR FORCED ERROR ON NODE CNR00011 - VARY INACT
 SCHED
 IST105I  CNR00011 NODE NOW INACTIVE
 XS6511E  SESSION TERMINATED, ACBNAME=$BBEN   , SSNAME=BBEN  BBEM
 IST871I  RESOURCE CNR00011 DELETED
*DSNB325A  *DBS1 DSNB5STS THERE IS A CRITICAL SHORTAGE 464
 OF SPACE IN GROUP BUFFER POOL GBP0
 DSNB327I  *DBS1 DSNB5STS GROUP BUFFER POOL GBP0 HAS ADEQUATE FREE
 SPACE
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,858  PAGE      18
 +CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
 +CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
 CQ, MQCC=2 MQRC=2192
 +CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
 /D XCF,STR,STRNM=STRIRSD
 F BBEMSRM, D XCF,STR,STRNM=STRIRSD
 SV00501W UNKNOWN COMMAND D
 +CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
 +CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
 CQ, MQCC=2 MQRC=2192
 +CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
 D XCF,STR,STRNM=STRIRSD
 IXC360I  16.12.40  DISPLAY XCF 476
 STRNAME: STRIRSD
  STATUS: ALLOCATED
         STRUCTURE CLEANUP IN PROGRESS
```

```
 POLICY INFORMATION:
  POLICY SIZE    : 1536 K
  POLICY INITSIZE: 768 K
  POLICY MINSIZE : 0 K
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : ENABLED
  PREFERENCE LIST: ICF4     ICF3
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 04/09/2003 18:30:47
  CFNAME         : ICF4      NO SYSTEMS CONNECTED TO COUPLING FACILITY
  COUPLING FACILITY: 002064.IBM.83.00000001055A
                     PARTITION: 5   CPCID: 00
  ACTUAL SIZE    : N/A
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: B93F289D 44485160
  LOGICAL  VERSION: B93F289D 44485160
  SYSTEM-MANAGED PROCESS LEVEL: NOT AVAILABLE
  XCF GRPNAME    : IXCLO015
  DISPOSITION    : KEEP
  ACCESS TIME    : 0
  NUMBER OF RECORD DATA LISTS PER CONNECTION: 16
  MAX CONNECTIONS: 7
  # CONNECTIONS  : 2

 & AMPERSAND DENOTES CONNECTOR WHO LOST CONNECTIVITY TO STRUCTURE
 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
 ---------------- -- -------- -------- -------- ---- ----------------
 XCFIRSD$$IRSD001 01 00010009 SYSM     DBIRLMS1 007C FAILED-PERSISTENT
 XCFIRSD$$IRSD002 02 00020007 SYSN     DBIRLMS2 006F FAILED-PERSISTENT
+CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
SETXCF FORCE,CON,STRNM=STRIRSD,CONNM=ALL
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,911  PAGE      19
IXC363I THE SETXCF FORCE FOR ALL CONNECTIONS FOR STRUCTURE 479
STRIRSD WAS ACCEPTED:
REQUEST WILL BE PROCESSED ASYNCHRONOUSLY
IXC538I DUPLEXING REBUILD OF STRUCTURE STRIRSD 480
WAS NOT INITIATED BY MVS. REASON:
STRUCTURE DOES NOT SUPPORT SYSTEM-MANAGED PROCESSES
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
D XCF,STR,STRNM=STRIRSD
IXC360I  16.13.54  DISPLAY XCF 484
```

```
 STRNAME: STRIRSD
  STATUS: ALLOCATED
         STRUCTURE CLEANUP IN PROGRESS
  POLICY INFORMATION:
   POLICY SIZE    : 1536 K
   POLICY INITSIZE: 768 K
   POLICY MINSIZE : 0 K
   FULLTHRESHOLD  : 80
   ALLOWAUTOALT   : NO
   REBUILD PERCENT: 1
   DUPLEX         : ENABLED
   PREFERENCE LIST: ICF4     ICF3
   ENFORCEORDER   : NO
   EXCLUSION LIST IS EMPTY

  ACTIVE STRUCTURE
  ----------------
   ALLOCATION TIME: 04/09/2003 18:30:47
   CFNAME         : ICF4      NO SYSTEMS CONNECTED TO COUPLING FACILITY
   COUPLING FACILITY: 002064.IBM.83.00000001055A
                      PARTITION: 5   CPCID: 00
   ACTUAL SIZE    : N/A
   STORAGE INCREMENT SIZE: 256 K
   PHYSICAL VERSION: B93F289D 44485160
   LOGICAL  VERSION: B93F289D 44485160
   SYSTEM-MANAGED PROCESS LEVEL: NOT AVAILABLE
   DISPOSITION    : KEEP
   ACCESS TIME    : 0
   NUMBER OF RECORD DATA LISTS PER CONNECTION: 16
   MAX CONNECTIONS: 7
   # CONNECTIONS  : 0
 F DBIRLMS1,ABEND
 DXR124E IRSD001 ABENDED VIA MODIFY COMMAND
 IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=DBIRLMS1, ASID=007C.
 IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=DBIRLMS1, ASID=007C.
 IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=DBIRLMS1, ASID=007C.
 DXR121I IRSD001 END-OF-TASK CLEANUP SUCCESSFUL - HI-CSA   1282K  -
 HI-ACCT-CSA    132K
 IEF450I DBIRLMS1 DBIRLMS1 - ABEND=S000 U2020 REASON=00000000 491
         TIME=16.14.11
 IEF404I DBIRLMS1 - ENDED - TIME=16.14.11
 IEF352I ADDRESS SPACE UNAVAILABLE
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  39,964  PAGE      20
 .HASP395 DBIRLMS1 ENDED
 IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=007C.
 S DBIRLMS1
 IRR812I PROFILE ** (G) IN THE STARTED CLASS WAS USED 497
         TO START DBIRLMS1 WITH JOBNAME DBIRLMS1.
 .HASP100 DBIRLMS1 ON STCINRDR
```

```
IEF695I START DBIRLMS1 WITH JOBNAME DBIRLMS1 IS ASSIGNED TO USER STC
   , GROUP EXPUSTC
.HASP373 DBIRLMS1 STARTED
IEF403I DBIRLMS1 - STARTED - TIME=16.14.30
DXR180I IRSD001 AUTOMATIC RESTART MANAGER IS NOT ENABLED
DXR117I IRSD001 INITIALIZATION COMPLETE
+CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
F DBCTLS1,RECONNECT
IXL013I IXLCONN REQUEST FOR STRUCTURE STRIRSD FAILED. 508
JOBNAME: DBIRLMS1 ASID: 008F CONNECTOR NAME: XCFIRSD$$IRSD001
IXLCONN RETURN CODE: 0000000C,  REASON CODE: 02010C09
        CONADIAG0:   00000002
        CONADIAG1:   00000008
        CONADIAG2:   00010C09
        CONADIAG10:  00400000
DXR135E IRSD001 CONNECT TO LOCK TABLE STRIRSD          FAILED FOR
GLOBAL INITIALIZATION, RETURN=0C, REASON=0C09
DFS039I IRLM IDENTIFY REQUEST FAILED, RC=16-01  DCS1
DFS625I IRLM RECONNECT COMMAND FAILED, CODE - 8  DCS1
S DBIRLMS1,STATUS
IRR812I PROFILE ** (G) IN THE STARTED CLASS WAS USED 513
        TO START DBIRLMS1 WITH JOBNAME DBIRLMS1.
.HASP100 DBIRLMS1 ON STCINRDR
IEFC452I DBIRLMS1 - JOB NOT RUN - JCL ERROR 515
.HASP396 DBIRLMS1 TERMINATED
IEE122I START COMMAND JCL ERROR
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=009E.
F DBIRLMS1,STATUS
DXR101I IRSD001 STATUS SCOPE=DISCON 520
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS    UNITS         HELD    WAITING    RET_LKS
        NO INFORMATION AVAILABLE
+CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
D XCF,STR,STRNM=STRIRSD
IXC360I  16.16.09  DISPLAY XCF 525
STRNAME: STRIRSD
 STATUS: ALLOCATED
        STRUCTURE CLEANUP IN PROGRESS
 POLICY INFORMATION:
  POLICY SIZE    : 1536 K
  POLICY INITSIZE: 768 K
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,017  PAGE      21
  POLICY MINSIZE : 0 K
```

```
  FULLTHRESHOLD  : 80
  ALLOWAUTOALT   : NO
  REBUILD PERCENT: 1
  DUPLEX         : ENABLED
  PREFERENCE LIST: ICF4     ICF3
  ENFORCEORDER   : NO
  EXCLUSION LIST IS EMPTY

 ACTIVE STRUCTURE
 ----------------
  ALLOCATION TIME: 04/09/2003 18:30:47
  CFNAME         : ICF4      NO SYSTEMS CONNECTED TO COUPLING FACILITY
  COUPLING FACILITY: 002064.IBM.83.00000001055A
                   PARTITION: 5   CPCID: 00
  ACTUAL SIZE    : N/A
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: B93F289D 44485160
  LOGICAL  VERSION: B93F289D 44485160
  SYSTEM-MANAGED PROCESS LEVEL: NOT AVAILABLE
  DISPOSITION    : KEEP
  ACCESS TIME    : 0
  NUMBER OF RECORD DATA LISTS PER CONNECTION: 16
  MAX CONNECTIONS: 7
  # CONNECTIONS  : 0
SETXCF FORCE,STR,STRNM=STRIRSD
IXC353I THE SETXCF FORCE REQUEST FOR STRUCTURE 527
STRIRSD WAS ACCEPTED:
REQUEST WILL BE PROCESSED ASYNCHRONOUSLY
IXC538I DUPLEXING REBUILD OF STRUCTURE STRIRSD 528
WAS NOT INITIATED BY MVS. REASON:
DUPLEXING REBUILD NOT ALLOWED FOR THE STRUCTURE
DXR141I IRSD001 THE LOCK TABLE STRIRSD          WAS ALLOCATED IN A
VOLATILE     FACILITY
IXL014I IXLCONN REQUEST FOR STRUCTURE STRIRSD 529
WAS SUCCESSFUL.  JOBNAME: DBIRLMS1 ASID: 008F
CONNECTOR NAME: XCFIRSD$$IRSD001 CFNAME: ICF3
IXL015I STRUCTURE ALLOCATION INFORMATION FOR 530
STRUCTURE STRIRSD, CONNECTOR NAME XCFIRSD$$IRSD001
 CFNAME     ALLOCATION STATUS/FAILURE REASON
 --------   ---------------------------------
 ICF3       STRUCTURE ALLOCATED
 ICF4       NO CONNECTIVITY
DXR132I IRSD001 SUCCESSFULLY JOINED THE DATA SHARING GROUP. GLOBAL
INITIALIZATION IS COMPLETE
DFS626I - IRLM RECONNECT COMMAND SUCCESSFUL. DCS1
+CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
```

```
  F DBIRLMS1,ABEND
  DXR124E IRSD001 ABENDED VIA MODIFY COMMAND
  DFS2011I IRLM FAILURE - IMS QUIESCING DCS1
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,070  PAGE      22
  IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=DBIRLMS1, ASID=008F.
  IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=DBIRLMS1, ASID=008F.
  IXL030I CONNECTOR STATISTICS FOR LOCK STRUCTURE STRIRSD, 542
  CONNECTOR XCFIRSD$$IRSD001:
        0001000A
        00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
        00000001 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
        00000002 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
        00000000 00000000 00000000 00000000
  IXL031I CONNECTOR CLEANUP FOR LOCK STRUCTURE STRIRSD, 543
  CONNECTOR XCFIRSD$$IRSD001, HAS COMPLETED.
  INFO: 0001000A 00000000 00000000 00000000 00000000 00000004
  DFS039I IRLM IDENTIFY REQUEST FAILED, RC=08-40  DCS1
 *0049 DFS039A  IRSD  NOT ACTIVE. REPLY  RETRY, CANCEL, OR DUMP. DCS1
  IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=DBIRLMS1, ASID=008F.
  DXR121I IRSD001 END-OF-TASK CLEANUP SUCCESSFUL - HI-CSA    664K  -
  HI-ACCT-CSA    132K
  IEF450I DBIRLMS1 DBIRLMS1 - ABEND=S000 U2020 REASON=00000000 548
          TIME=16.16.57
  IEF404I DBIRLMS1 - ENDED - TIME=16.16.57
  IEF352I ADDRESS SPACE UNAVAILABLE
  .HASP395 DBIRLMS1 ENDED
  IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=008F.
  S DBIRLMS1
  IRR812I PROFILE ** (G) IN THE STARTED CLASS WAS USED 554
          TO START DBIRLMS1 WITH JOBNAME DBIRLMS1.
  .HASP100 DBIRLMS1 ON STCINRDR
  IEF695I START DBIRLMS1 WITH JOBNAME DBIRLMS1 IS ASSIGNED TO USER STC
     , GROUP EXPUSTC
  .HASP373 DBIRLMS1 STARTED
  IEF403I DBIRLMS1 - STARTED - TIME=16.17.21
  DXR180I IRSD001 AUTOMATIC RESTART MANAGER IS NOT ENABLED
  DXR117I IRSD001 INITIALIZATION COMPLETE
  F DBCTLS1,RECONNECT
  DFS625I IRLM RECONNECT COMMAND FAILED, CODE - 4.  DCS1
  +CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
  +CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
  CQ, MQCC=2 MQRC=2192
  +CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
  F DBCTLS1,RECONNECT
```

```
 DFS625I IRLM RECONNECT COMMAND FAILED, CODE - 4.  DCS1
 R 49,RETRY
 IEE600I REPLY TO 0049 IS;RETRY
 IXC538I DUPLEXING REBUILD OF STRUCTURE STRIRSD 570
 WAS NOT INITIATED BY MVS. REASON:
 DUPLEXING REBUILD NOT ALLOWED FOR THE STRUCTURE
 DXR141I IRSD001 THE LOCK TABLE STRIRSD          WAS ALLOCATED IN A
 VOLATILE     FACILITY
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,123  PAGE      23
 IXL014I IXLCONN REQUEST FOR STRUCTURE STRIRSD 571
 WAS SUCCESSFUL.  JOBNAME: DBIRLMS1 ASID: 0090
 CONNECTOR NAME: XCFIRSD$$IRSD001 CFNAME: ICF3
 DXR132I IRSD001 SUCCESSFULLY JOINED THE DATA SHARING GROUP. GLOBAL
 INITIALIZATION IS COMPLETE
 DFS626I - IRLM RECONNECT COMMAND SUCCESSFUL. DCS1
*IXC585E STRUCTURE DBSG_GBP0 IN COUPLING FACILITY ICF3, 575
 PHYSICAL STRUCTURE VERSION B94F3A9E 10CDE2C1,
 IS AT OR ABOVE STRUCTURE FULL MONITORING THRESHOLD OF  80%.
 ENTRIES:  IN-USE:        132 TOTAL:        748,  17% FULL
 ELEMENTS: IN-USE:        132 TOTAL:        147,  89% FULL
 +CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
 +CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
 CQ, MQCC=2 MQRC=2192
 +CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
 IXC586I STRUCTURE DBSG_GBP0 IN COUPLING FACILITY ICF3, 579
 PHYSICAL STRUCTURE VERSION B94F3A9E 10CDE2C1,
 IS NOW BELOW STRUCTURE FULL MONITORING THRESHOLD.
 D XCF,STR,STRNM=STRIRSD
 IXC360I  16.19.14  DISPLAY XCF 581
 STRNAME: STRIRSD
  STATUS: ALLOCATED
  POLICY INFORMATION:
   POLICY SIZE    : 1536 K
   POLICY INITSIZE: 768 K
   POLICY MINSIZE : 0 K
   FULLTHRESHOLD  : 80
   ALLOWAUTOALT   : NO
   REBUILD PERCENT: 1
   DUPLEX         : ENABLED
   PREFERENCE LIST: ICF4     ICF3
   ENFORCEORDER   : NO
   EXCLUSION LIST IS EMPTY

  ACTIVE STRUCTURE
  ----------------
   ALLOCATION TIME: 04/22/2003 16:16:36
   CFNAME         : ICF3
   COUPLING FACILITY: 002064.IBM.51.000000068371
                      PARTITION: 6   CPCID: 00
```

```
  ACTUAL SIZE    : 768 K
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: B94F62DE 92663487
  LOGICAL  VERSION: B94F62DE 92663487
  SYSTEM-MANAGED PROCESS LEVEL: 8
  XCF GRPNAME    : IXCLO015
  DISPOSITION    : KEEP
  ACCESS TIME    : 0
  NUMBER OF RECORD DATA LISTS PER CONNECTION: 16
  MAX CONNECTIONS: 7
  # CONNECTIONS  : 1

 CONNECTION NAME  ID VERSION  SYSNAME  JOBNAME  ASID STATE
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,176  PAGE      24
 ---------------- -- -------- -------- -------- ---- ----------------
 XCFIRSD$$IRSD001 01 0001000B SYSM     DBIRLMS1 0090 ACTIVE

 STRUCTURE IN TRANSITION
 -----------------------
  REASON IN TRANSITION: CONNECT OR DISCONNECT IN PROGRESS
  ALLOCATION TIME: 04/09/2003 18:30:47
  CFNAME         : ICF4      NO SYSTEMS CONNECTED TO COUPLING FACILITY
  COUPLING FACILITY: 002064.IBM.83.00000001055A
                     PARTITION: 5   CPCID: 00
  ACTUAL SIZE    : N/A
  STORAGE INCREMENT SIZE: 256 K
  PHYSICAL VERSION: B93F289D 44485160
  SYSTEM-MANAGED PROCESS LEVEL: NOT AVAILABLE
  XCF GRPNAME    : IXCLO015
S DBCTLS2
.HASP100 DBCTLS2  ON STCINRDR
IEF695I START DBCTLS2  WITH JOBNAME DBCTLS2  IS ASSIGNED TO USER
DBCUSR  , GROUP EXPUSTC
.HASP373 DBCTLS2  STARTED
IEF403I DBCTLS2 - STARTED - TIME=16.19.59
DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DFSPBTS2  DCS2
DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DFSVSM00  DCS2
+CSQX500I ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE started
+CSQX038E ?S1 CSQXRCTL Unable to put message to SYSTEM.QSG.CHANNEL.SYN
CQ, MQCC=2 MQRC=2192
+CSQX599E ?S1 CSQXRCTL Channel CH.SYSM.SYSB.BPE ended abnormally
DFS0403W IMS REGISTER  CALL TO MVS ARM FAILED - RETURN CODE= 0C,
REASON CODE= 0160. DCS2
DFS3410I DATASETS USED ARE IMSACBA FORMATA MODBLKSA DCS2
DFS0579W - FIND FAILED FOR DDNAME PROCLIB  MEMBER = DLISASS2 RETURN
CODE=X'00000004'  DCS2
START DLISASS2,PARM=(DLS,DCS2)
 DCS2
START DLISASS2,PARM=(DLS,DCS2)
```

```
 DFS0579W - FIND FAILED FOR DDNAME PROCLIB  MEMBER = DBRCS2   RETURN
 CODE=X'00000004'  DCS2
 START DBRCS2,PARM=(DRC,DCS2)
  DCS2
 START DBRCS2,PARM=(DRC,DCS2)
 .HASP100 DLISASS2 ON STCINRDR
 .HASP100 DBRCS2   ON STCINRDR
 DFS3613I - STM TCB INITIALIZATION COMPLETE  DCS2
 DFS3613I - MOD TCB INITIALIZATION COMPLETE  DCS2
 IEF695I START DBRCS2   WITH JOBNAME DBRCS2   IS ASSIGNED TO USER
 DBCUSR  , GROUP EXPUSTC
 .HASP373 DBRCS2   STARTED
 IEF403I DBRCS2 - STARTED - TIME=16.20.02
 IEF695I START DLISASS2 WITH JOBNAME DLISASS2 IS ASSIGNED TO USER
 DBCUSR  , GROUP EXPUSTC
 .HASP373 DLISASS2 STARTED
 IEF403I DLISASS2 - STARTED - TIME=16.20.02
 DFS3613I - STC TCB INITIALIZATION COMPLETE  DCS2
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,229  PAGE      25
 DFS3613I - CTX TCB INITIALIZATION COMPLETE  DCS2
 DFS3613I - DYC TCB INITIALIZATION COMPLETE  DCS2
 DFS3613I - ESS TCB INITIALIZATION COMPLETE  DCS2
 DFS3613I - TRA TCB INITIALIZATION COMPLETE  DCS2
 DFS3613I - RDS TCB INITIALIZATION COMPLETE  DCS2
 DFS3613I - DYC RECALL TCB INITIALIZATION COMPLETE  DCS2
 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DCS2DBS2  DCS2
 DFS3613I - RST TCB INITIALIZATION COMPLETE  DCS2
 DFS0578I - READ SUCCESSFUL FOR DDNAME PROCLIB  MEMBER = DFSVSM00  DCS2
 DFS2500I DATASET  DFSOLP00 SUCCESSFULLY ALLOCATED DCS2
 DFS2500I DATASET  DFSOLS00 SUCCESSFULLY ALLOCATED DCS2
 DFS2500I DATASET  DFSOLP01 SUCCESSFULLY ALLOCATED DCS2
 DFS2500I DATASET  DFSOLS01 SUCCESSFULLY ALLOCATED DCS2
*DFS227A - CTL REGION WAITING FOR DLS REGION (DLISASS2) INIT - DCS2
 DFS228I - DLS REGION STORAGE COMPRESSION INITIALIZED  DCS2
 DFS228I - DLS REGION DYNAMIC ALLOCATION INITIALIZED  DCS2
 DFS228I - DLS RECALL TCB INITIALIZATION COMPLETE  DCS2
 DFS563I PSB BPBSUCA0 REQUIRES UNKNOWN DMB BPFSUCE , PSB STOPPED.  DCS2
 DFS563I PSB BPBSUCC0 REQUIRES UNKNOWN DMB BPFSUCE , PSB STOPPED.  DCS2
 DFS563I PSB BPBSUCL0 REQUIRES UNKNOWN DMB BPFSUCE , PSB STOPPED.  DCS2
 DFS2208I DUAL   LOGGING IN EFFECT ON IMS LOG DATA SET    DCS2
 DFS2208I DUAL   LOGGING IN EFFECT ON WRITE AHEAD DATA SET DCS2
 DFS2207I IMS  LOG(S)  BLOCKSIZE= 26624, BUFNO=005 DCS2
 DFS3613I - DLG TCB INITIALIZATION COMPLETE DCS2
 DFS3613I - DRC TCB INITIALIZATION COMPLETE  DCS2
 DFS3382I DL/I CF INITIALIZATION COMPLETE   DCS2
 IXL014I IXLCONN REQUEST FOR STRUCTURE STRVSAMS 636
 WAS SUCCESSFUL.  JOBNAME: DLISASS2 ASID: 00A2
 CONNECTOR NAME: IXCL00160002 CFNAME: ICF3
 DFS228I - DLS REGION INITIALIZATION COMPLETE  DCS2
```

```
DFS3613I - CTL TCB INITIALIZATION COMPLETE  DCS2
*DFS989I IMS (DBCTL) READY (CRC=() - DCS2
DFS2500I DATASET  DFSOLP02 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLS02 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLP03 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLS03 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLP04 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLS04 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLP05 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSOLS05 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSWADS0 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSWADS1 SUCCESSFULLY ALLOCATED DCS2
DFS2500I DATASET  DFSWADS2 SUCCESSFULLY ALLOCATED DCS2
F DBIRLMS1,STATUS,ALLD
DXR102I IRSD001 STATUS 653
        SUBSYSTEMS IDENTIFIED
        NAME     STATUS    RET_LKS   IRLMID  IRLM_NAME
        DCS1     UP              0    001      IRSD
(ERE
DFS4445I CMD FROM MCS/E-MCS CONSOLE USERID=TSSTRIA: ERE DCS1
DFS4445I CMD FROM MCS/E-MCS CONSOLE USERID=TSSTRIA: ERE DCS2
DFS101 MULTIPLE RESTART COMMANDS ARE INVALID DCS1
DFS058I ERESTART COMMAND IN PROGRESS DCS2
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,282  PAGE      26
DFS0618A A RESTART OF A NON-ABNORMALLY TERMINATED SYSTEM MUST SPECIFY
EMERGENCY BACKUP OR OVERRIDE DCS2
DFS0618A A RESTART OF A NON-ABNORMALLY TERMINATED SYSTEM... DCS2
DFS000I....MUST SPECIFY EMERGENCY BACKUP OR OVERRIDE. DCS2
DFS3626I RESTART HAS BEEN ABORTED DCS2
DCS2ERE OVERRIDE
DFS4445I CMD FROM MCS/E-MCS CONSOLE USERID=TSSTRIA: ERE OVERRIDE DCS2
DFS058I ERESTART COMMAND IN PROGRESS DCS2
DFS680I USING CHKPT 03105/123930 DCS2
DFS3257I ONLINE LOG CLOSED ON DFSOLP02 DCS2
DFS3257I ONLINE LOG CLOSED ON DFSOLS02 DCS2
DFS3257I ONLINE LOG NOW OPENED ON DFSOLP03 DCS2
DFS3257I ONLINE LOG NOW OPENED ON DFSOLS03 DCS2
DFS3261I WRITE AHEAD DATA SET NOW ON DFSWADS0 DCS2
DFS3261I WRITE AHEAD DATA SET NOW ON DFSWADS1 DCS2
DFS682I  BATCH-MSG PROGRAM BPBSUCAD JOB SUCOD20  MAY BE RESTARTED FROM
673
CHKPT ID DCS2
DFS994I *CHKPT 03112/162217**SIMPLE* DCS2
DFS3499I ACTIVE DDNAMES: MODBLKSA IMSACBA          MODSTAT ID:    0
DCS2
DFS3804I LATEST RESTART CHKPT: 03112/162217 DCS2
.HASP100 $DBCARC  ON INTRDR      SOPORTE              FROM STC13883
DBRCS2
IRR010I  USERID DBCUSR   IS ASSIGNED TO THIS JOB.
```

```
  DFS2484I JOBNAME=$DBCARC  GENERATED BY LOG AUTOMATIC ARCHIVING DCS2
  DFS994I EMERGENCY START COMPLETED. DCS2
  DFS0653I PROTECTED CONVERSATION PROCESSING WITH RRS/MVS ENABLED  DCS2
  DFS4444I DISPLAY FROM ID=DCS2 683
   DSNM003I  IMS/VS DCS2 FAILED TO CONNECT TO SUBSYSTEM DBS2 RC=00
 *DFS0800I AWAITING NOTIFICATION FROM SUBSYS DBS2      DCS2
  DFS0800I AWAITING NOTIFICATION FROM SUBSYS DBS2 DCS2
  IXC538I DUPLEXING REBUILD OF STRUCTURE ISTGENERIC 685
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE LOG_DFHLOG_001 686
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE LOG_DFHSHUNT_001 687
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_RMD 688
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_RESTART 689
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE RRS_ARCHIVE 690
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_LOCK1 691
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,335  PAGE      27
  IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_GBP0 692
  WAS NOT INITIATED BY MVS. REASON:
  NO COUPLING FACILITY PROVIDED BETTER OR EQUIVALENT CONNECTIVITY
  IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_GBP2 693
  WAS NOT INITIATED BY MVS. REASON:
  NO COUPLING FACILITY PROVIDED BETTER OR EQUIVALENT CONNECTIVITY
  IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_SCA 694
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE MQSGCSQ_ADMIN 695
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE MQSGTEST 696
  WAS NOT INITIATED BY MVS. REASON:
  NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
  IXC538I DUPLEXING REBUILD OF STRUCTURE STRIRSD 697
  WAS NOT INITIATED BY MVS. REASON:
  DUPLEXING REBUILD NOT ALLOWED FOR THE STRUCTURE
  DCS2CHE FREEZE
  DFS4445I CMD FROM MCS/E-MCS CONSOLE USERID=TSSTRIA: CHE FREEZE DCS2
```

```
DFS058I CHECKPOINT COMMAND IN PROGRESS DCS2
DFS994I *CHKPT 03112/162332**FREEZE* DCS2
DFS3499I ACTIVE DDNAMES: MODBLKSA IMSACBA          MODSTAT ID:    0
DCS2
DFS3804I LATEST RESTART CHKPT: 03112/162332 DCS2
DFS603I IMS DLS CLEANUP ( EOT ) COMPLETE FOR JS DLISASS2 DLISASS2
    ,RC=00
IEF404I DLISASS2 - ENDED - TIME=16.23.32
IEF352I ADDRESS SPACE UNAVAILABLE
.HASP395 DLISASS2 ENDED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=00A2.
DFS3257I ONLINE LOG CLOSED ON DFSOLP03 DCS2
DFS3257I ONLINE LOG CLOSED ON DFSOLS03 DCS2
.HASP100 $DBCARC  ON INTRDR      SOPORTE               FROM STC13883
DBRCS2
IRR010I  USERID DBCUSR   IS ASSIGNED TO THIS JOB.
DFS2484I JOBNAME=$DBCARC  GENERATED BY LOG AUTOMATIC ARCHIVING DCS2
DFS092I IMS LOG TERMINATED DCS2
DFS2091I IMS TIMER SERVICE SHUTDOWN COMPLETED DCS2
DFS0617I RDS BUFFERS HAVE BEEN SUCCESSFULLY PURGED DCS2
ATR169I RRS HAS UNSET EXITS FOR RESOURCE MANAGER 717
IMS.DCS2____V061.STL.SANJOSE.IBM REASON: UNREGISTERED
DFS994I IMS (DBCTL) SHUTDOWN COMPLETED  DCS2
DFS627I IMS RTM CLEANUP ( EOT ) COMPLETE FOR JS DBCTLS2 .DBCTLS2
.IEFPROC ,RC=00
IEF404I DBRCS2 - ENDED - TIME=16.23.33
.HASP395 DBRCS2   ENDED
IEF404I DBCTLS2 - ENDED - TIME=16.23.33
IEF352I ADDRESS SPACE UNAVAILABLE
.HASP395 DBCTLS2  ENDED
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=00A3.
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=009E.
.HASP100 DEFVRACF ON INTRDR      DEL OLMO              FROM TSU13869
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,388  PAGE      28
TSST090
IRR010I  USERID TSST090  IS ASSIGNED TO THIS JOB.
ICH70001I TSST090  LAST ACCESS AT 15:58:48 ON TUESDAY, APRIL 22, 2003
.HASP373 DEFVRACF STARTED - INIT 03   - CLASS G - SYS SYSM
IEF403I DEFVRACF - STARTED - TIME=16.24.10
-                                                    --TIMINGS (MINS.)--
          ----PAGING COUNTS---
-JOBNAME  STEPNAME PROCSTEP    RC   EXCP   CONN    TCB    SRB  CLOCK
 SERV  PG  PAGE  SWAP   VIO SWAPS
-DEFVRACF DEFRECA1             00    107     68    .00    .00     .0
  976   0     0     0     0     0
IEF404I DEFVRACF - ENDED - TIME=16.24.11
-DEFVRACF ENDED.  NAME-DEL OLMO            TOTAL TCB CPU TIME=   .00
 TOTAL ELAPSED TIME=    .0
.HASP395 DEFVRACF ENDED
```

```
.HASP309 INIT 03   INACTIVE ******** C=HG
SE '16.24.11 JOB13886 $HASP165 DEFVRACF ENDED AT BPET  MAXCC=0',LOGON,
USER=(TSST090)
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=TSSTEIL , ASID=0088.
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=TSSTEIL , ASID=0088.
IEF450I TSSTEIL $TSSTDBS $TSSTDBS - ABEND=S522 U0000 REASON=00000000
742
        TIME=16.24.20
.HASP395 TSSTEIL  ENDED
.HASP250 TSSTEIL PURGED -- (JOB KEY WAS B94F58E3)
IEA989I SLIP TRAP ID=X13E MATCHED.  JOBNAME=*UNAVAIL, ASID=0088.
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=0088.
IEA631I  OPERATOR TSSTRIA  NOW INACTIVE, SYSTEM=SYSM    , LU=LINP3019
IEF126I TSSTRIA - LOGGED OFF - TIME=16.25.35
.HASP395 TSSTRIA  ENDED
.HASP250 TSSTRIA PURGED -- (JOB KEY WAS B94F613D)
IEA989I SLIP TRAP ID=X33E MATCHED.  JOBNAME=*UNAVAIL, ASID=009C.
IEA630I  OPERATOR TSSTVGG  NOW ACTIVE,   SYSTEM=SYSM    , LU=LINP3014
*DBS2 START DB2
S DBS2MSTR
.HASP100 DBS2MSTR ON STCINRDR
IEF695I START DBS2MSTR WITH JOBNAME DBS2MSTR IS ASSIGNED TO USER
DBS1SYS , GROUP EXPUSTC
.HASP373 DBS2MSTR STARTED
IEF403I DBS2MSTR - STARTED - TIME=16.26.04
DSNZ002I *DBS2 DSNZINIT SUBSYSTEM DBS2 SYSTEM PARAMETERS LOAD MODULE
NAME IS DSNZDBS2
IXL014I IXLCONN REQUEST FOR STRUCTURE DBSG_SCA 760
WAS SUCCESSFUL.  JOBNAME: DBS2MSTR ASID: 009C
CONNECTOR NAME: DB2_DBS2 CFNAME: ICF3
DSN7507I  *DBS2 DSN7LSTK 761
SCA STRUCTURE DBSG_SCA IS ALLOCATED IN A VOLATILE STRUCTURE.
DSN7407I  *DBS2 DSN7GAR1 762
AUTOMATIC RESTART MANAGER REGISTER FAILED.
  MVS IXCARM RETURN CODE = 0000000C,
  MVS IXCARM REASON CODE = 00000160.
S DBS2IRLM
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_SCA 764
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,441  PAGE      29
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
.HASP100 DBS2IRLM ON STCINRDR
IEF695I START DBS2IRLM WITH JOBNAME DBS2IRLM IS ASSIGNED TO USER
DBS1SYS , GROUP EXPUSTC
.HASP373 DBS2IRLM STARTED
IEF403I DBS2IRLM - STARTED - TIME=16.26.06
DXR180I IRS2002 AUTOMATIC RESTART MANAGER IS NOT ENABLED
DXR117I IRS2002 INITIALIZATION COMPLETE
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_LOCK1 771
```

```
WAS NOT INITIATED BY MVS. REASON:
NO SUITABLE COUPLING FACILITY IN PREFERENCE LIST
DXR141I IRS2002 THE LOCK TABLE DBSG_LOCK1       WAS ALLOCATED IN A
VOLATILE     FACILITY
IXL014I IXLCONN REQUEST FOR STRUCTURE DBSG_LOCK1 772
WAS SUCCESSFUL.  JOBNAME: DBS2IRLM ASID: 0088
CONNECTOR NAME: DBSGIRLM$IRS2002 CFNAME: ICF3
DXR132I IRS2002 SUCCESSFULLY JOINED THE DATA SHARING GROUP WITH    4M
LOCK TABLE ENTRIES AND    25576 RECORD LIST ENTRIES
DSNY001I  *DBS2 SUBSYSTEM STARTING
IEC161I 056-084,DBS2MSTR,DBS2MSTR,BSDS1,,,DBS2S.BSDS01,
IEC161I DBS2S.BSDS01.DATA,CATALOG.DB2USER.ICF
IEC161I 056-084,DBS2MSTR,DBS2MSTR,BSDS1,,,DBS2S.BSDS01,
IEC161I DBS2S.BSDS01.INDEX,CATALOG.DB2USER.ICF
IEC161I 062-086,DBS2MSTR,DBS2MSTR,BSDS1,,,DBS2S.BSDS01,
IEC161I DBS2S.BSDS01.DATA,CATALOG.DB2USER.ICF
IEC161I 056-084,DBS2MSTR,DBS2MSTR,BSDS2,,,DBS2S.BSDS02,
IEC161I DBS2S.BSDS02.DATA,CATALOG.DB2USER.ICF
IEC161I 056-084,DBS2MSTR,DBS2MSTR,BSDS2,,,DBS2S.BSDS02,
IEC161I DBS2S.BSDS02.INDEX,CATALOG.DB2USER.ICF
IEC161I 062-086,DBS2MSTR,DBS2MSTR,BSDS2,,,DBS2S.BSDS02,
IEC161I DBS2S.BSDS02.DATA,CATALOG.DB2USER.ICF
DSNJ127I  *DBS2 SYSTEM TIMESTAMP FOR BSDS= 03.112 13:15:58.36
DSNJ001I  *DBS2 DSNJW007 CURRENT COPY 1 ACTIVE LOG 789
DATA SET IS DSNAME=DBS2S.LOGCOPY1.DS03,
STARTRBA=000010E00000,ENDRBA=000012FBFFFF
DSNJ001I  *DBS2 DSNJW007 CURRENT COPY 2 ACTIVE LOG 790
DATA SET IS DSNAME=DBS2S.LOGCOPY2.DS03,
STARTRBA=000010E00000,ENDRBA=000012FBFFFF
DSNJ099I  *DBS2 LOG RECORDING TO COMMENCE WITH 791
STARTRBA=00001116B000
S DBS2DBM1
.HASP100 DBS2DBM1 ON STCINRDR
IEF695I START DBS2DBM1 WITH JOBNAME DBS2DBM1 IS ASSIGNED TO USER
DBS1SYS , GROUP EXPUSTC
.HASP373 DBS2DBM1 STARTED
IEF403I DBS2DBM1 - STARTED - TIME=16.26.12
D GRS
ISG343I 16.26.14 GRS STATUS 798
SYSTEM    STATE               SYSTEM    STATE
SYSM      CONNECTED
GRS STAR MODE INFORMATION
  LOCK STRUCTURE (ISGLOCK) CONTAINS 1048576 LOCKS.
STEM ID SYSM DATE 04/22/2003 2003.112 LINE  40,494  PAGE      30
  THE CONTENTION NOTIFYING SYSTEM IS SYSM
  SYNCHRES:       NO
S DBS2DIST
.HASP100 DBS2DIST ON STCINRDR
IEF695I START DBS2DIST WITH JOBNAME DBS2DIST IS ASSIGNED TO USER
```

```
DBS1SYS , GROUP EXPUSTC
.HASP373 DBS2DIST STARTED
IEF403I DBS2DIST - STARTED - TIME=16.26.18
IEF196I IEF237I E250 ALLOCATED TO SYS00161
DSNR001I  *DBS2 RESTART INITIATED
DSNR003I  *DBS2 RESTART...PRIOR CHECKPOINT RBA=00001114E090
IEF196I IEF285I   DB2710.SDSNLOAD                              KEPT
IEF196I IEF285I   VOL SER NOS= IP0007.
DSNR004I  *DBS2 RESTART...UR STATUS COUNTS 809
IN COMMIT=0, INDOUBT=0, INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
DSNR005I  *DBS2 RESTART...COUNTS AFTER FORWARD 810
RECOVERY
IN COMMIT=0, INDOUBT=0
DSNR006I  *DBS2 RESTART...COUNTS AFTER BACKWARD 811
RECOVERY
INFLIGHT=0, IN ABORT=0, POSTPONED ABORT=0
IXC538I DUPLEXING REBUILD OF STRUCTURE DBSG_GBP0 812
WAS NOT INITIATED BY MVS. REASON:
NO COUPLING FACILITY PROVIDED BETTER OR EQUIVALENT CONNECTIVITY
IXL014I IXLCONN REQUEST FOR STRUCTURE DBSG_GBP0 813
WAS SUCCESSFUL.  JOBNAME: DBS2DBM1 ASID: 00A3
CONNECTOR NAME: DB2_DBS2 CFNAME: ICF3
DSNB302I  *DBS2 DSNB1GC1 GROUP BUFFER POOL GBP0 IS 814
           ALLOCATED IN A VOLATILE STRUCTURE
DSNB315I  *DBS2 DSNB1GC1 GROUP BUFFER POOL GBP0 IS 815
ALLOCATED AS GBPCACHE YES
DSNR002I  *DBS2 RESTART COMPLETED
*DBS2RECOVER POSTPONED
S DBS2SPAS
DSNV434I  *DBS2 DSNVRP NO POSTPONED ABORT THREADS FOUND
DSN9022I  *DBS2 DSNVRP 'RECOVER POSTPONED' NORMAL COMPLETION
DSNL003I  *DBS2 DDF IS STARTING
.HASP100 DBS2SPAS ON STCINRDR
IEF695I START DBS2SPAS WITH JOBNAME DBS2SPAS IS ASSIGNED TO USER
DBS1SYS , GROUP EXPUSTC
.HASP373 DBS2SPAS STARTED
DSNL515I  *DBS2 DSNLILNR TCP/IP BIND FAILED FOR PORT 825
446 WITH
           RETURN CODE=1115 AND REASON CODE=744C7247
IEF403I DBS2SPAS - STARTED - TIME=16.26.30
DSNL004I  *DBS2 DDF START COMPLETE 827
           LOCATION  DBS1
           LU        ESPOP000.DBS2LU
           GENERICLU ESPOP000.DBSGLU
           DOMAIN    -NONE
           TCPPORT   446
           RESPORT   5020
DSN9022I  *DBS2 DSNYASCP 'START DB2' NORMAL COMPLETION
```

Related publications

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

IBM Redbooks

For information about ordering these publications, see “How to get IBM Redbooks” on page 239. Note that some of the documents referenced here may be available only in softcopy.

- *DB2 on the MVS Platform: Data Sharing Recovery,* SG24-2218
- *SAP on DB2 for z/OS and OS/390: DB2 System Cloning*, SG24-6287
- *SAP on DB2 UDB for OS/390 and z/OS: High Availability Solution Using System Automation*, SG24-6836
- *SAP on DB2 Universal Database for OS/390 and z/OS: Multiple Components in One Database (MCOD)*, SG24-6914
- *SAP R/3 on DB2 for OS/390: Database Availability Considerations*, SG24-5690
- *z/OS Version 1 Release 3 and 4 Implementation*, SG24-6581

Other IBM publications

These publications are also relevant as further information sources:

- *DB2 UDB for OS/390 and z/OS V7 Data Sharing: Planning and Administration*, SC26-9935
- *SAP on DB2 UDB for z/OS: Planning Guide (SAP Web Application Server 6.20)*, SC33-7959
- *SAP R/3 on DB2 UDB for OS/390 and z/OS: Connectivity Guide,* SC33-7965

SAP publications

These publications are available on http://www.sap.com (login required), the SAP Notes Web site listed below, or SAP product CDs:

- *SAP 6.2.0 Database Administration Guide, Virtual Storage Considerations*
- *SAP High Availability Guide*
- *SAP on DB2 UDB for OS/390 and z/OS: Database Administration Guide, SAP Web Application Server, Release 6.20*
- *DB2/390: Backup and Recovery Options*, SAP Note 83000

Online resources

These Web sites are also relevant as further information sources:

- CF Level and the Coupling Facility

  http://www-1.ibm.com/servers/eserver/zseries/pso/cftable.html
- DB2 UDB for OS/390 and z/OS V7 PDF files

  http://www-1.ibm.com/servers/s390/os390/bkserv/swpdf/db2udbv7.html
- IBM z/OS RMF Tools and Downloads

  http://www-1.ibm.com/servers/eserver/zseries/zos/rmf/rmfhtmls/rmftools.htm
- SAP Notes (login required)

  https://websmp203.sap-ag.de/notes
- SAP OSS Notes

  http://sappoint.com/oss/
- SAP R/3 on DB2 for OS/390 and z/OS PDF files

  http://www-1.ibm.com/servers/s390/os390/bkserv/swpdf/sapr3.html
- SAP Web site

  http://www.sap.com/
- SAP Service Marketplace (may require password)

  http://www.service.sap.com
- Technology for disk storage systems

  http://www.storage.ibm.com/disk/ess/technology.htm
- Washington System Center Flash 10011

  http://www-1.ibm.com/support/docview.wss?uid=tss1flash10011

- Washington System Center Flash 10159

  `http://www-1.ibm.com/support/docview.wss?uid=tss1flash10159`
- Whitepaper: *SAP BW Query Performance with DB2 on ZSeries*

  `http://www-1.ibm.com/support/docview.wss?uid=tss1wp100322`
- Whitepaper: *Tuning SAP / DB2 / zSeries*

  `http://www-1.ibm.com/support/docview.wss?uid=tss1wp100287`
- z/OS V1R4.0 migration and installation

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

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

Index

Numerics

SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing

(0.5” spine)
0.475”<->0.875”
250 <-> 459 pages

SAP on DB2 for z/OS and OS/390: High Availability and Performance Monitoring with Data Sharing

Architecture and design choices affecting availability and performance

Workload monitoring and system healthcheck for high availability

Sysplex, DB2 data sharing and SAP tuning

This IBM Redbook addresses the challenges posed by monitoring high availability, scalability, and performance in an SAP sysplex data sharing environment.

It introduces the motivations for utilizing a design based on DB2 data sharing. It includes the principal SAP-DB2 data sharing architecture options and trade-offs used in the industry today and issues that play a role in both high availability and scalability, such as failover design, database connectivity design, workload splitting and load balancing, MCOD, and coupling facility design.

The book discusses single point of failure, important failover scenarios and outage avoidance, automation of high availability constructs, and backup and recovery considerations in data sharing environments.

Performance issues are detailed in the order you would approach them at planning and implementation time. First, it discusses tuning the sysplex, which is the base for a well-performing DB2 data sharing system, then tuning the DB2 data sharing system, which is the base for a well-performing SAP system, and finally, tuning the SAP system.

The book focuses on initial planning for performance and monitoring it afterward, and explains the key points to look for to health-check your system and maintain high performance.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

SG24-6950-00 ISBN 0738453072