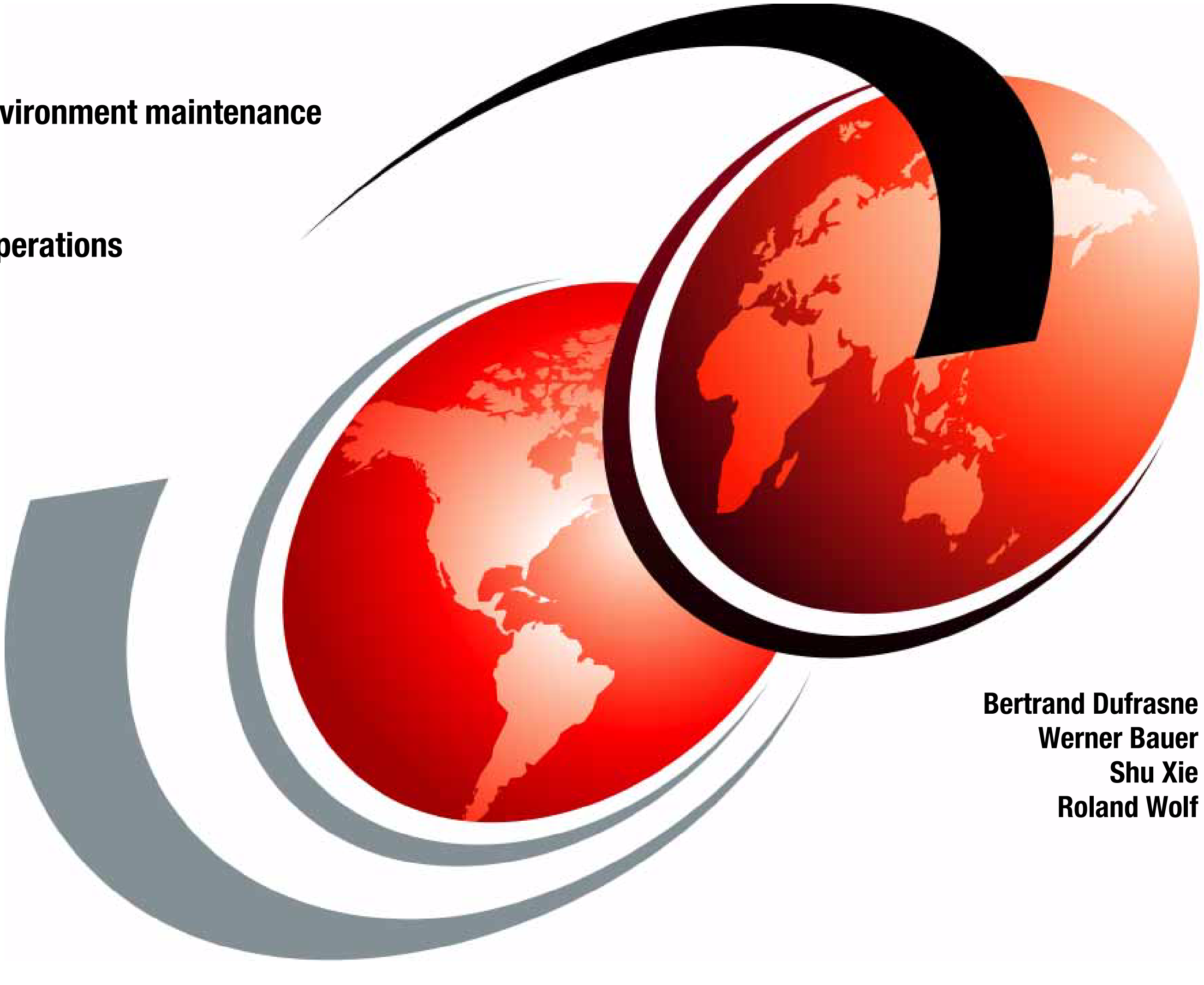
IBM z/OS Global Mirror Planning, Operations, and Best Practices

Planning considerations

Healthy environment maintenance

Ongoing operations



Bertrand Dufrasne
Werner Bauer
Shu Xie
Roland Wolf

ibm.com/redbooks

Redpaper

International Technical Support Organization

IBM z/OS Global Mirror Planning, Operations, and Best Practices

October 2013

REDP-4878-00

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

First Edition (October 2013)

This edition applies to z/OS Global Mirror, also known as Extended Remote Copy as available with IBM z/OS V1.13.

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 grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

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

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

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

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

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 illustrate 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.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at `http://www.ibm.com/legal/copytrade.shtml`

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

CICS®
DB2®
DS8000®
Easy Tier®
Enterprise Storage Server®
FICON®
FlashCopy®
GDPS®
Geographically Dispersed Parallel Sysplex™
HyperSwap®
IBM®
IMS™
MVS™
NetView®
Parallel Sysplex®
PR/SM™
RACF®
Redbooks®
Redpaper™
Redbooks (logo) ®
Resource Measurement Facility™
RMF™
S/390®
System Storage®
System z®
System z10®
TDMF®
Tivoli®
XIV®
z/OS®
z/VM®
z10™
zSeries®

The following terms are trademarks of other companies:

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

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

Preface

IBM® z/OS® Global Mirror (zGM), also known as Extended Remote Copy (XRC), is a combined hardware and software solution that offers the highest levels of continuous data availability in a disaster recovery (DR) and workload movement environment. Available for the IBM DS8000® Storage System, zGM provides an asynchronous remote copy solution.

This IBM Redpaper™ publication takes you through best practices for planning, tuning, operating, and monitoring a zGM installation.

This publication is intended for clients and storage administrators who need to understand and maintain a zGM environment.

Authors

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

Bertrand Dufrasne is an IBM Certified IT Specialist and Project Leader for IBM System Storage® disk products at the ITSO, San Jose Center. He has worked at IBM in various IT areas. He has written many IBM Redbooks® publications, and has developed and taught technical workshops. Before joining the ITSO, he worked for IBM Global Services as an Application Architect. He holds a Master of Electrical Engineering.

Werner Bauer is a Certified Consulting IT Specialist in Germany. He has more than 30 years of experience in storage software and hardware, and with IBM S/390® and IBM z/OS. His areas of expertise include DR solutions based on IBM enterprise disk storage systems. Werner is a frequent speaker at storage conferences and GUIDE SHARE Europe (GSE) meetings.

He has also written extensively in various IBM Redbooks publications about the DS8000. He holds a degree in Economics from the University of Heidelberg, and in Mechanical Engineering from FH Heilbronn. He currently works with System Vertrieb Alexander (SVA), an IBM Premier Business Partner.

Shu Xie joined IBM China in 2010, and has been working with the ATS mainframe team for about 3 years. ATS is an IBM Premier Business Partner. She focuses on mainframe DR, and participates in many client DR tests and implementation projects. She also works as a System Programmer for mainframe systems at the IBM China System Center, supporting z/OS-related client requests.

Roland Wolf is a Certified IT Specialist in Germany. He has worked at IBM for 26 years, and has extensive experience with high-end disk-storage hardware in IBM System z® and Open Systems. He works in Technical Sales Support. His areas of expertise include performance analysis and DR solutions in enterprises using the unique capabilities and features of the IBM disk storage systems, the DS8000, and IBM XIV®.

He has contributed to various IBM Redbooks publications about IBM Enterprise Storage Server® (ESS), the DS8000 Architecture, and the DS8000 Copy Services. He holds a Ph.D. in Theoretical Physics.

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

Alan McClure, Eduard Diel, Dash Miller, Hank Sautter, David Shackelford, Herbert Yee, Eric Seftel, and Megan Gilge
IBM

Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author, all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

- Use the online **Contact us** review Redbooks form found at the following website:

  ibm.com/redbooks

- Send your comments in an email to the following email address:

  redbooks@us.ibm.com

- Mail your comments to the following address:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with Really Simple Syndication (RSS) feeds:

  http://www.redbooks.ibm.com/rss.html

1

IBM z/OS Global Mirror overview

This book focuses on best practices for IBM z/OS Global Mirror (zGM), previously known as Extended Remote Copy (XRC). We assume the reader is already familiar to some degree with zGM concepts and functionality.

The goal of this overview chapter is to ensure a basic, common understanding of zGM functions and terminology. For more information about zGM characteristics, see *z/OS DFSMS Advanced Copy Services*, SC35-0428, and the IBM Redbooks publication, *IBM System Storage DS8000 Copy Services for IBM System z*, SG24-6787.

1.1 An introduction to z/OS Global Mirror

Mirroring typically means copying data from storage systems at one site where the application servers are located (*primary site*) to a remote location with storage systems and eventually also application servers (*auxiliary site*, also referred to as a *secondary site*).

There are several mirroring options for z/OS:

- Synchronous mirroring with IBM System Storage DS8000 Metro Mirror. *Synchronous mirroring* has the advantage that the data at the disaster recovery (DR) site, also known as an *auxiliary site*, is current in case of a disaster. Synchronous mirroring, however, is limited regarding distance.

  DS8000 Metro Mirror is supported over distances of up to 300 kilometers (km), but with increasing distance synchronous mirroring slows down write input/output (I/O) response time, which can adversely affect production performance. Practically, this can limit the distance between the application site (primary site) and the DR site (auxiliary site) to less than 300 km.

  DS8000 Metro Mirror is managed on a volume-pair basis. This is not easy to manage, and in case of a rolling disaster, some automation is needed to ensure data consistency across the many volumes involved. IBM Geographically Dispersed Parallel Sysplex™ (GDPS®) and IBM Tivoli® Storage Productivity Center for Replication provide these functions.

- Asynchronous mirroring with DS8000 Global Mirror. As the name implies, with *asynchronous mirroring*, data at the DR site (auxiliary site) might not be current in case of a disaster. Asynchronous solutions have a *session* concept. You do not have to deal with individual volume pairs, but with sessions, which are a *collection of volume pairs*. This approach makes it easier to handle, and data consistency is ensured in case of a disaster. With DS8000 Global Mirror, data can be mirrored around the globe.

  Global Mirror has the least influence on production performance. However, what might sound good also has some disadvantages. DS8000 Global Mirror has no function to control the Recovery Point Objective (RPO), which is the time that the auxiliary site lags behind the primary site.

  During high write activity at the primary site, data at the auxiliary site can fall behind, perhaps for seconds but even for minutes or hours (if, for example, the bandwidth between the two sites is not sufficient to handle the load). Companies might have service-level agreements that specify how far data can fall behind at the DR site.

- Using zGM, which is a z/OS function that is part of DFSMS data facility product (DFSMSdfp). However, because zGM uses a special communication method with the storage system, the storage system must support this particular communication method. It is enabled for the DS8000 by applying the optional zGM license key.

  For zGM, the System Data Mover (SDM) DFSMSdfp component is copying the data from the storage systems at the primary site to the storage systems at the auxiliary site. Data is copied asynchronously. zGM also has a session concept. This ensures that all volumes at the auxiliary site are consistent up to the last consistency time.

  A function called *pacing* throttles write I/Os at the primary site, to avoid overloading the copy process. Therefore the RPO cannot grow uncontrolled. Write pacing provides a compromise. The RPO cannot grow infinitely, and application performance impact is mitigated.

  You can implement zGM over unlimited distances. It is a DR solution, but it can also be used for data migration.

1.2 Terms and process definitions for z/OS Global Mirror

The zGM product is a software and hardware cooperative remote copy implementation. SDM mirrors write activities from a local storage system at the primary site to a remote storage system at the auxiliary site. Although the main zGM implementation consists of host-resident software, special zGM support is required in the DS8000 (or equivalent storage system) to attach the zGM primary volumes.

The storage systems at the primary site need to have the zGM license key applied. Although the storage systems at the auxiliary site do not need a zGM license during normal operation, they will need such a license in case you ever want to copy data back from the auxiliary site to the primary site with zGM after a disaster. Therefore, it is a good practice to have the license also applied to the storage systems at the auxiliary site. Figure 1-1 illustrates a basic zGM environment.

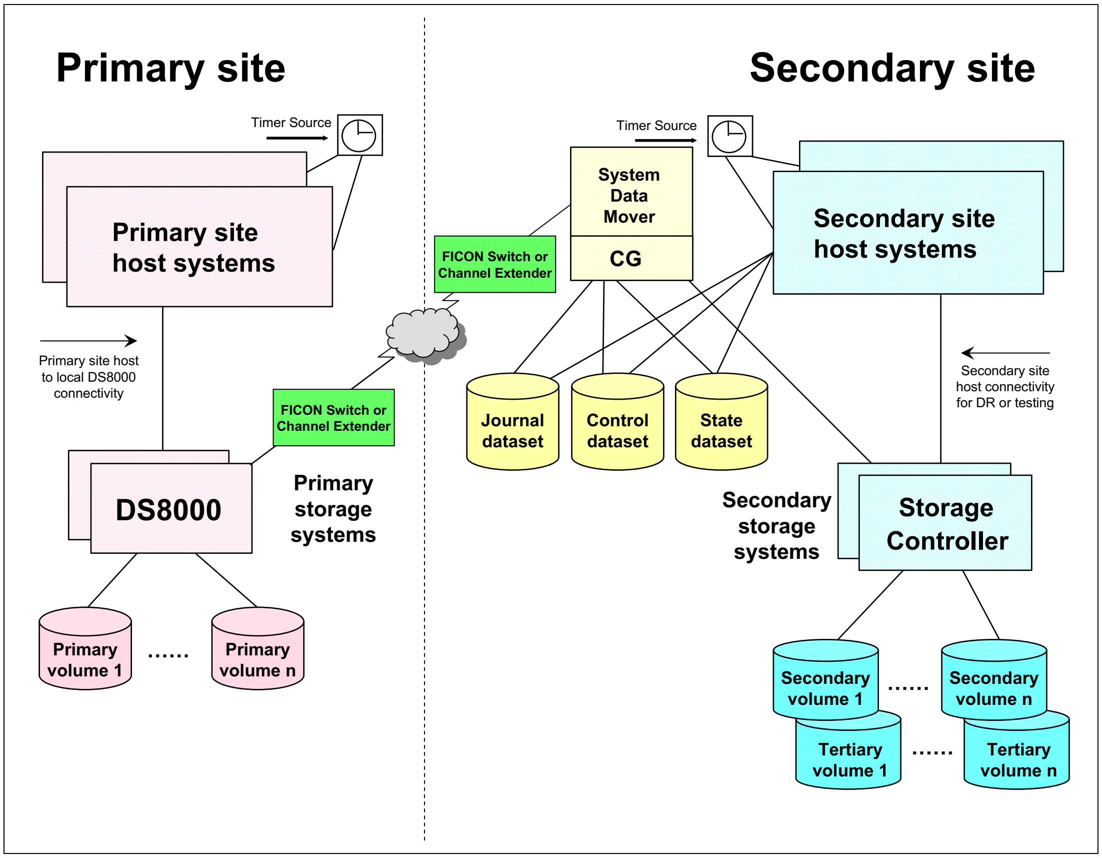


Figure 1-1 Basic zGM environment

1.2.1 Components in zGM

This section describes some of the main components of a zGM configuration, as shown in Figure 1-1.

Primary systems

Primary systems, also called *application systems*, *region 1 systems*, or *local systems*, are the host systems where the applications run. SDM can run on a system at the primary site, but this is *not* the preferred configuration.

If there are multiple systems at the primary site, they must have a common clock reference to maintain data consistency across multiple primary storage systems.

Common time

It is important to zGM configurations that the application systems have the same time, in terms of coordination of the system clocks (systems do *not* need to have the same coordinated universal time (UTC) time offset). The Server Time Protocol (STP) feature is designed to provide the capability for multiple servers and *coupling facilities* to maintain time synchronization with each other. See 1.2.3, “How zGM maintains data consistency” on page 11 for more information.

Primary storage systems

The primary storage systems are the physical storage units where a collection of application and systems volumes that are designated to be mirrored to an auxiliary site are located. This collection of volumes could be all of the volumes at the primary site, or a subset of them. Note that zGM can scale to manage a virtually unlimited number of storage systems and volumes.

Note: The storage systems at the primary site must be enabled for zGM. For the DS8000, the zGM license key must have been applied for all primary storage systems.

The *sidefile* is a logical concept that represents the use of a cache in the primary storage system, where the linked list of updates (*record sets*) is located before being read by the SDM reader tasks. A record set is a 60 kilobyte (KB) buffer, which can be filled or only partly filled.

All updates to the primary volumes can be managed by one storage control session, called *single reader mode*. Alternatively, the primary volumes that are managed by one zGM session under one logical control unit (LCU), called *multiple reader mode*, are linked and grouped into one sidefile, and reserved in the cache until it is successfully read by SDM. The number of record sets for one primary device that were not yet read by SDM is referred to as the *residual count* of that primary device.

The hardware bitmaps that are provided by the primary storage system are used to track data that has not yet been sent to the auxiliary system. This data is used for resynchronization if the replication suspends. There are two hardware bitmaps that are maintained in the primary storage system for each volume. Only one bitmap is active at any time.

Periodically, SDM attempts to toggle the bitmaps to reduce the amount of data that is copied during resynchronization after suspension. Installation can control the frequency of bitmap toggling by tuning the bitmap **`ChangedTracks`** and **`Delaytime`** zGM parmlib parameters.

Every storage system used in a z/OS environment should have the parallel access volume (PAV) and HyperPAV features, and HyperPAV should be enabled in z/OS.

Auxiliary systems

Auxiliary systems, also called *secondary systems*, *recovery systems*, *region 2 systems*, or *remote systems*, are used to take over the role of the primary systems to run applications when the primary systems are no longer available for use. Data from the primary systems must be available for use on the auxiliary systems. This availability is accomplished through zGM replication and other data backup and recovery techniques.

Tip: It is a best practice to run SDM in a system at the auxiliary site.

Auxiliary storage systems

The auxiliary storage systems at the DR site are the physical storage units where a collection of zGM-mirrored volumes are located. This collection of volumes can be mirrors of all of the volumes of the primary site, or a subset of them. The auxiliary storage systems contain the devices that are used to rebuild and run the production environment on the data recovery center in the event of a loss of production processing capabilities.

> **Note:** Although storage systems at the auxiliary site do not need to be enabled for zGM (they do not need the zGM license), it is a best practice to also enable these systems for zGM. This enables you to use zGM to copy data from the auxiliary site back to the primary site when necessary.

For all systems, the IBM System Storage Easy Tier® function should be enabled (a license key is required). You should use a few extent pools, and allocate volumes with storage pool striping enabled.

The storage subsystem at the auxiliary site should also have the PAV and HyperPAV features, and HyperPAV should be enabled in the SDM z/OS system.

You should plan for DR tests. This can easily be achieved if the IBM FlashCopy® feature is available at the auxiliary storage subsystems, with enough capacity to flash the volumes at the auxiliary site to create a test system. You should also use FlashCopy to create a consistent set of volumes whenever you have to resynchronize your volume pairs.

The zGM devices

This section provides information about the different devices referenced and used by zGM.

Primary device

A zGM primary device is a production host-defined device that is used for normal daily read and write activity, which zGM copies to a remote location. This device must be part of a primary storage system that is zGM-capable, such as the DS8000 system with the optional zGM feature enabled. The primary devices are an IBM System z FICON® environment attached to one or more host application systems, which are running on one or more z/OS, IBM zSeries®/Virtual Machine (z/VM®), or Linux for System z guest system images.

Auxiliary device

The zGM auxiliary devices are duplicates of the zGM primary devices, and maintain that status by asynchronously receiving via SDM all updated write activity that occurs on the primary devices. These devices can be part of any storage subsystem that is supported by the auxiliary host systems, and by the z/OS system where SDM is running. Each zGM primary device has a corresponding auxiliary device.

When zGM is started, both primary and auxiliary devices must be online to the SDM systems. Therefore, the primary and auxiliary devices must have different volume serial numbers (VOLSERs).

However, it is a best practice to have some naming conventions in place that enable you to identify by VOLSER which volumes form a pair. The suggestion is to actually use a VOLSER prefix that is not used for a production disk (for example, XR) and then follow it with the auxiliary device number on the SDM system. This practice enables use of SDM's `SecondaryVolserPattern` parameter, which is an easy way to ensure that the data mover does not mistakenly overwrite a production primary volume.

Tertiary device

Tertiary devices are those devices that are typically used in a recovery process to take over the primary site's workload. They can also be a backup of the auxiliary devices to ensure that there is a consistent copy of all recoverable devices in place at the same time as zGM auxiliary devices are resynchronized.

These devices are point-in-time copies of the zGM auxiliary devices that are created by the `FlashCopy` function. They are stored on the same auxiliary storage system as the zGM auxiliary devices. Point-in-time copies to these devices can be made, to enable DR testing without interrupting the zGM process. Do *not* use space efficient FlashCopy. As mentioned previously, some meaningful naming convention should be established.

Utility devices

When SDM issues channel commands to read updates from the primary storage control unit cache, it must specify a device address, even though the data in the cache belongs to several different devices. This device address is the *utility device*, which is used by SDM to offload updates from the primary subsystem cache.

Depending on the workload in an LCU, more than one utility device might be required to offload the update write activity. Readers can work either independently or in parallel to offload updates for a given logical subsystem (LSS).

Ensure that the utility device is not heavily used by the primary system, because heavy use might prevent SDM from getting access to the device. If the device is busy, the offload process is slower. For best performance, use dedicated devices for the utility devices. Dedicated devices are required for Enhanced Readers (see 1.2.6, "The zGM enhanced multiple reader (enhanced reader) function" on page 14).

It is important that HyperPAV alias addresses can be used for the utility devices.

FlashCopy on auxiliary storage systems

When the DS8000 systems are used at the recovery site, you can combine FlashCopy functions with zGM functions.

For example, you can use this combination to create data for DR testing, or for consistent data for point-in-time backup purposes. The Zero Suspend FlashCopy function, provided by GDPS and XRC, provides the ability to use the FlashCopy function and XRC for auxiliary devices without interrupting DR protection.

> **Note:** During a DR, zGM verifies that the auxiliary VOLSERs and device numbers match those used in the last active zGM session, and will fail the recovery if they do not match. This check is done to ensure that zGM is recovering the correct information. When the FlashCopy function is used to create another set of auxiliary volumes for DR testing, all data, including the VOLSER, is copied to a new device number.
>
> At this point, the VOLSER and device number do not match. The **`ONLINE`** parameter indicates that recovery is using tertiary direct access storage devices (DASDs), and zGM is to verify only the VOLSER, enabling any device online with a VOLSER that matches a valid auxiliary VOLSER to be used for recovery.

SDM

SDM is a DFSMSdfp component. The system or systems with the data movers must have connectivity to the primary volumes *and* to the auxiliary volumes. When application systems write to the primary volumes, SDM manages the process of mirroring those updates to the auxiliary volumes, and ensures that updates to the remote volumes are made in the same order in which they were made to the application volumes, maintaining sequence consistency.

SDM also performs the recovery processes, including applying remaining updates to form a consistency group to auxiliary devices, relabeling auxiliary volumes and making them ready for use at the recovery site.

SDM for zGM operates in specific system address spaces, which have the prefix “ANT”. `ANTAS000` handles zGM Time Sharing Option (TSO) commands and application programming interface (API) requests. There is one `ANTAS000` address space for each system image. The `ANTAS000` address space is automatically initialized during initial program load (IPL), and automatically restarts when this address space is canceled.

`ANTAS`*`nnn`* (where *`nnn`* `= 001 - 020`) address space manages the zGM session, one for each zGM session in a single system image, and starts when the **`XSTART`** command is issued for its corresponding zGM session. The `ANTCL`*`nnn`* address space manages zGM sessions in a system image. These sessions are coupled to a zGM master session as a single cluster session, which is automatically started during IPL.

If an `ANTCL`*`nnn`* address space is canceled, the system suspends all volumes for any `ANTAS`*`nnn`* address space that is coupled to a master session through the cluster session.

FICON switch and channel extender configurations for zGM

All connections in a zGM configuration are Fibre Channel connections (FICON). Because it is a best practice to run SDM at the auxiliary site, there usually is the requirement to somehow bridge the distance between the primary application storage systems and SDM. The Extended Distance FICON function of the DS8000 provides optimized channel programs for long-distance reads as they occur when the SDM reads data from the application storage systems.

If the distance between the primary site and the auxiliary site is *less than 100 km*, FICON channel extenders are not required. Most FICON switches have embedded functions that provide zGM transmission optimization. Often, Fibre Channel over Internet Protocol (FCIP) routers are used. More detailed information about connectivity options is provided in 2.8, “Connectivity” on page 39.

A zGM session

A zGM session is a single instance of an SDM that manages replication and consistency of a particular set of zGM primary volumes. Each SDM has one zGM session that is responsible for a group of volumes. The SDM maintains the consistency for the volumes that are participating in the zGM session, across LSSs in the DS8000 system, and across the DS8000 systems (in addition to other primary storage systems that support zGM).

Multiple zGM sessions can be in effect per z/OS system (up to 20). Multiple instances of the zGM sessions on separate z/OS images are also possible.

A zGM session can operate as one of the following session types:

XRC In this type, zGM is operating a recovery session, using *state*, *journal*, and *control data sets*. Auxiliary volume consistency is ensured. The XRC session mode should be used for zGM sessions that are *intended for use for DR purposes*.

Migration In this type, zGM ensures data consistency in the session when the migration is complete, but it does *not* use the journal or control data sets, so it cannot be used for DR. This type of session is intended for *data migration* or *workload movement purposes*.

The zGM configuration provides a highly scalable architecture, with the ability to manage thousands of volumes with coupled extended remote copy (CXRC) and clustering. CXRC permits up to 14 zGM sessions or zGM cluster sessions to be coupled together in one *zGM master session* to ensure that all volumes are consistent to the same time across all coupled zGM sessions.

Clustering provides the ability to link up to 13 zGM coupled sessions to one *zGM cluster session* in a single logical partition, permitting up to 182 (14 x 13) zGM sessions to be coupled to a single master session.

LCU

A z/OS LCU, also called LSS in DS8000 terms, plays an important role for zGM. With DS8000, the LCU is completely virtualized. Therefore, it does not matter how many LCUs you define, or in which LCUs you define large or small volumes, but you must use odd-numbered and even-numbered LCUs. For zGM, however, LCUs are important, and you should ensure that the I/O workload is evenly distributed across the LCUs.

Storage control session

When you start a zGM session (using the `XSTART` TSO command), and then establish the first volume pair (using the `XADDPAIR` TSO command), a zGM *storage control session* is created in the DS8000 LSS (or LCU) to which the primary volume belongs. This zGM storage control session in the LCU has a counterpart (*reader task*) in SDM. It is possible to have multiple zGM storage control sessions in an LCU, where a specific set of primary volumes is defined for each storage control session that is created for that LCU.

SDM uses a separate utility device and runs a separate reader task for each zGM storage control session, making it possible to transfer data in parallel between one LCU and one SDM. Note that when using enhanced readers with alias auxiliary sessions, a single utility device can actually be associated with multiple reader tasks and storage control sessions.

> **Note:** Storage control sessions are often referred to as readers. There is indeed a one to one correspondence between a storage control session and a reader. However, keep in mind that storage control sessions effectively refer to the structures within the storage controller, while readers refer to the tasks within the data mover that service them.

The zGM infrastructure data sets

The *journal data set*, *control data set, state data set, cluster data set, cluster state data set*, and *master data set* are used by SDM to harden time-consistent groups of updated records that are received from the primary volumes on disk. SDM also controls the process of applying them to the auxiliary volumes to maintain consistency. SDM creates consistency groups and writes them to the journal data sets before applying them to auxiliary devices.

State data set

The state data set acts as the table of contents for the session, and contains the status of the XRC session and of associated volumes that XRC is managing. Session performance information is also recorded in this data set.

Control data set

The control data set contains consistent group information about the auxiliary volumes and the journal data sets. It contains information that is necessary for recovery operations.

Journal data set

The journal data set contains the temporary user data that was changed at the primary volumes and read by the SDM. The SDM writes the data that forms a consistency group to the journal to harden the data before the SDM applies the data to the auxiliary volumes.

Because all changes made on the primary volumes are written to the journal data sets, these journal data sets must be capable of sustaining a high write workload:

- Use dedicated volumes for the journal data sets.
- Stripe the journal data set across several volumes.
- Use HyperPAV for these volumes.
- Allocate these volumes in DS8000 extents pools with storage pool striping.

Master data set

The master data set records information that is used to ensure recoverable consistency among all XRC subsystems that are contained within the Coupled XRC system (CXRC). It is only required for a coupled zGM (CXRC) environment.

Cluster data set

The cluster data set is used to restart a cluster session.

Cluster state data set

The cluster state data set contains performance information that is consolidated from all coupled sessions in the cluster.

1.2.2 Data flow

This section describes the data flow for zGM.

A zGM configuration

Figure 1-2 shows a zGM configuration.

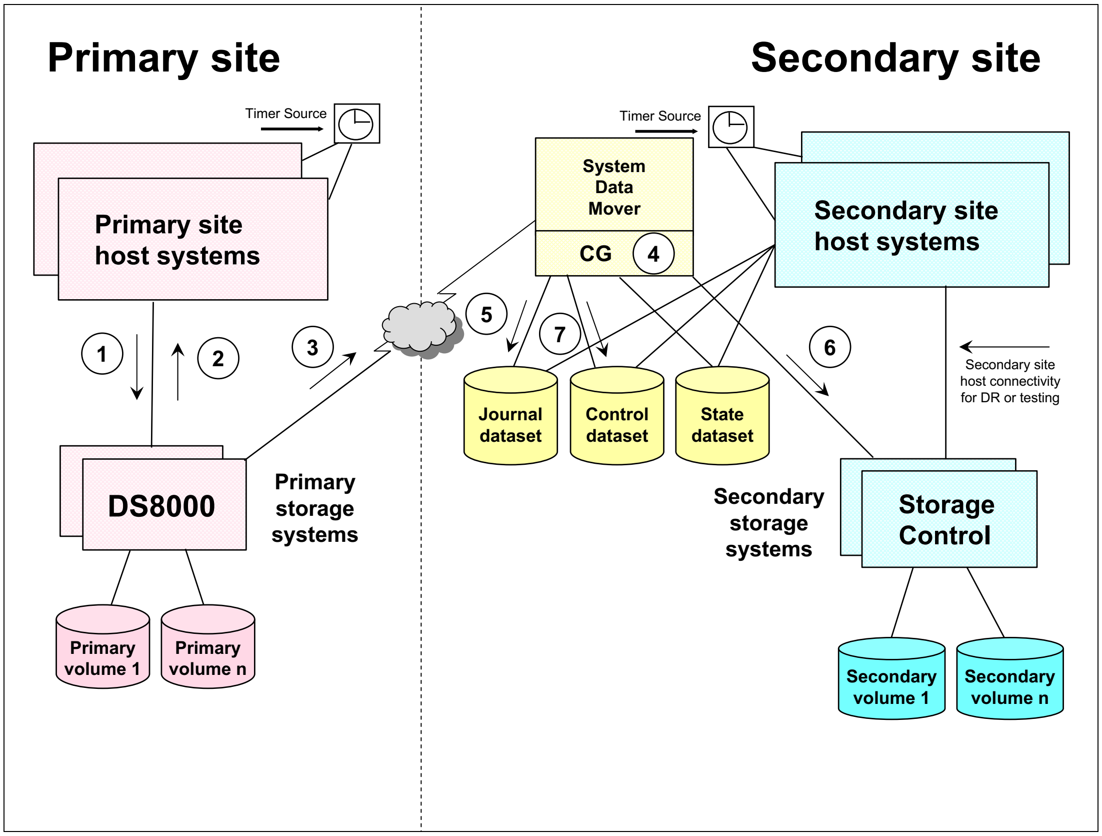


Figure 1-2 Data flow for z/OS Global Mirror

Figure 1-2 illustrates a simplified view of the zGM components, and the data flow logic. The following steps form the logic for the zGM data flow:

1. The primary system writes to the primary volumes.
2. The application I/O operation completion is signaled when the data is written to the primary DS8000 cache and non-volatile storage (NVS). This is when channel end and device end are returned to the primary system. Therefore, the application write I/O operation has completed, and now the updated data is mirrored asynchronously, as described in the remaining steps.
3. The DS8000 system groups the updates into record sets in the cache. These groups are asynchronously offloaded from the cache to the SDM system. As zGM uses this asynchronous copy technique, there is no performance effect on the I/O operations of the primary applications *if there is no write pacing in place*.

   However, write data might stay longer in the cache than without zGM active. Therefore, use larger cache sizes when using zGM. For detailed information about write pacing influencing the performance of the primary applications' I/O operations, see "Write Pacing" on page 60.
4. The record sets from one or more primary storage systems are processed into consistency groups (CGs) by the SDM. The CG contains records that have their order of update preserved across multiple LCUs in a DS8000 system, across multiple DS8000 systems, and across other storage subsystems that are participating in the same zGM session.

This preservation of order is absolutely vital for dependent write I/O operations, such as databases and their logs. The creation of CGs ensures that zGM mirrors data to the auxiliary site with point-in-time, cross-volume-consistent integrity.

5. When a CG is formed, it is written from SDM real storage buffers to the journal data sets.
6. Immediately after the CG has been hardened on the journal data sets, the records are written to their corresponding auxiliary volumes. Those records are also written from SDM's real storage buffers. Because of the data in transit between the primary and auxiliary sites, the data on the auxiliary volumes is slightly less current than the data at the primary site.
7. The control data set is updated to reflect that the records in the CG have been written to the auxiliary volumes, and to reflect the consistency time of the volume group.

1.2.3 How zGM maintains data consistency

This section provides a detailed description of the characteristics of time-stamping, and of the consistency groups.

Time-stamping process

Maintaining the update sequence for applications whose data is being copied in real time is a critical requirement for applications that run dependent write I/O operations. If data is copied out of sequence, serious integrity exposures can prevent recovery procedures from working. The zGM system uses special algorithms to ensure consistency for all data.

The starting point for maintaining update sequence integrity is when a record is written by an application system to a primary volume of a zGM managed pair. When a record is written, a time stamp is attached to the I/O. The storage subsystem keeps the record together with the time stamp in cache, and the SDM reads the records from the cache, including the time-stamp information. Time-stamped writes enable zGM to ensure data consistency.

The time-stamping code is an extension of the IBM z/OS Input/Output Supervisor (IOS) Start Subchannel (SSCH) code, so it applies to all data. Deferred writes that are held in the main storage buffers for database applications are processed in SSCH order. This ordering ensures that the time-stamping process delivers update sequence integrity support accurately and efficiently.

Consistency group

The SDM holds the update writes, read from perhaps multiple primary storage systems in memory buffers, and orders these writes into consistency groups (CGs). The CGs contains records that have their order of update preserved (based on time stamps) across (potentially many) storage systems that are participating in the same zGM session.

This preservation of order is vital for dependent write I/O operations that are used, such as databases and logs. The creation of CGs ensures that zGM applies the updates to the auxiliary volumes with consistency for any type of data.

Common time reference

A common time reference is required across the different systems.

IBM z/OS

When a zGM pair is established, this is signaled to the primary system, and the host system DFSMSdfp (SDM) software starts to time-stamp all write I/Os to the primary volumes. This stamping is necessary to provide consistency across multiple LCUs.

If those primary volumes are shared by systems that are running on different System z processors, the sysplex timer or STP feature is required to provide a common time reference. If all of the primary systems are running in different system logical partitions (LPARs) on the *same* System z processor, you can use the system time-of-day clock.

IBM z/VM

In z/VM V5.4 or later, with small programming enhancement (SPE) authorized program analysis reports (APARs), z/VM can use STP to time-stamp guest and system disk write I/O operations. This enables the data for z/VM and its guests to be copied by zGM, and consistency can be provided across z/VM (and guest) data, and z/OS (or other z/VM) data. The following requirements must be met before this support can be used:

- All LPARs of this type for which you want to mirror the data with zGM as a single consistency group must be running on processor complexes with the same Coordinated Timing Network (CTN) to have a common time source.
- The CTN configuration must meet the requirements that are specified by the virtual machine (VM) operating system.
- The STP time-stamping feature in VM must be enabled in the `VM SYSTEM CONFIG` file.

For more information about z/VM time-stamping support, see one of the following IBM publications:

- *z/VM V5R4.0 CP Planning and Administration*, SC24-6083
- *z/VM V6R2 CP Planning and Administration*, SC24-6178

Linux for System z

The IBM zSeries direct access storage device (DASD) driver for Linux supports time-stamping of writes for FICON-attached DASDs, and also contains support for device blocking. With this support, GDPS can manage the copy of Linux data by zGM. If there is a primary site disaster (or for a planned site switch), GDPS can automate recovery of Linux data, and can restart Linux systems in the recovery site by starting them with the recovered zGM copy volumes.

Ask your Linux distributor whether the required changes for zGM (also known as XRC) are included in the DASD drivers for your distribution.

When Linux is running under VM, in an STP environment where z/VM time-stamping for guests is enabled, the Linux guest does not need to support time stamping, because z/VM can perform this function on behalf of the Linux guest.

For more information about consistency group formation see *IBM System Storage DS8000 Copy Services for IBM System z*, SG24-6787.

1.2.4 Time-consistent recovery

One of the most valuable features of zGM is its ability to perform a time-consistent recovery. After a disaster that affects the primary site, the auxiliary site must be made ready for takeover. This involves getting the auxiliary volumes to a state where their contents are usable in anticipation of application restart. The **`XRECOVER`** command provides a single-step process to recover individual zGM sessions. With automation such as GDPS, a push-button approach to recovering large-scale zGM environments is available.

The **`XRECOVER`** command commits the last available consistency groups to the auxiliary volumes for each zGM session, reports the consistency time for the auxiliary volumes, and then attaches the volume serial number of each auxiliary device to that of its matching primary device.

You can also issue the **XRECOVER** command to recover a group of interlocked coupled sessions to the same consistency time.

For more detailed information about interlocked sessions, see *z/OS DFSMS Advanced Copy Services*, SC35-0428.

The zGM recovery process

The requirements for normal operation include procedures for restarting online applications if an outage occurs (for example, a power failure or processor failure).

For zGM DR, use the same procedure at the auxiliary site. The same validations for ensuring data consistency and restarting your applications must be done at your auxiliary site, just as they are done today at your application site.

To accomplish this, zGM provides data currency across all volumes in the zGM session. If a disaster occurs at the primary site, in-progress updates are lost. Nevertheless, recovery can be achieved to an identified consistent point in time. The recovery technique does not require a database restore followed by forward recovery.

Recovery can follow the procedure that is used at the primary site for a system outage, consistent to the point in time that is given. Any transaction that has been completed after this point in time might require investigation, and can be re-created against the auxiliary disks after recovery (by using the **XRECOVER** function), or accounted for in your business process.

1.2.5 Write pacing and device blocking

As a client, you probably want the best solution for your environment, and have objectives for a DR solution:

- Supports an unlimited distance
- Has low-bandwidth requirements
- Has minimum effect on the copy process on your production

All of these objectives cannot be achieved at the same time.

You must consider what is important for your environment. For long-distance mirroring, usually not much bandwidth is available (or it becomes quite expensive). Having the application systems accomplish as many writes as they can might be good for your application performance.

However, with limited bandwidth between the sites and long distances, the time (RPO) that the DR site lags behind might increase substantially, and even reach hours. It might result in suspends of pairs, or even the whole session, and you would have to perform a resynchronization, which increases the RPO time even more.

Usually, you must compromise. You must keep the RPO at a reasonable value in accordance with your service level agreements, by slowing down the application write I/O operations when the zGM copy process cannot catch up for some reason. For example, this can occur when the network links to the remote site are saturated.

In the past, zGM used *device blocking* to handle overload situations. The device blocking process is still supported, but is no longer the best practice. Device blocking enables blocking of application writes to occur for brief intervals, and only on the specific volumes that have accumulated a large number of updates in the cache. The zGM parmlib is used to set the level of updates that trigger device blocking.

The best practice solution is called *write pacing*. Write pacing works by injecting a small delay on writes. This occurs as each zGM record set is created in the primary storage subsystem cache for a given volume. As the device residual count increases, so does the magnitude of the pacing, eventually reaching a maximum value at a target residual count.

For complicated I/Os with several channel command words (CCWs), or transport control words (TCWs) in IBM System z High Performance FICON (zHPF), each CCW or TCW is delayed. Both this maximum injected delay value, and the target residual count at which it is effective, can be specified for each volume through zGM commands.

Doing so provides a greater level of flexibility than device blocking, where the threshold can be specified only at a session level. The device also remains ready to process I/O requests, enabling application read activity to continue *unpaced* while the device writes are being *paced*.

The amount of delay added to each record set varies as a function of the pacing level, and the pacing level is determined by the associated write pacing parameter settings, and the existing residual count for the volume of interest.

For more information about using write pacing values or device blocking, see the latest version of *z/OS DFSMS Advanced Copy Services*, SC35-0428. Some more detailed information about write pacing is provided in “Write Pacing” on page 60.

1.2.6 The zGM enhanced multiple reader (enhanced reader) function

Over time, zGM has undergone several improvements. A *fixed primary volume-to-single reader* relationship was available in the first versions of zGM. This can lead to periods when the production workload demand, and write updates to a specific primary volume, occur at a rate greater than a single zGM storage control session (reader task) can sustain.

Later, zGM was enhanced to enable multiple readers in a single storage control session (`SCSESSION`). In this model, there is a fixed relationship between a volume (or set of volumes) and a specific reader (or utility device that read the updates for that volume or set of volumes).

This functionality expanded the capability of zGM to read more updates from a single LCU. This support can provide improvement over the single-reader approach, but, for some applications, the volume or volumes that are associated with a specific reader can saturate that reader.

DS8000 Release 3 in 2009 included a high-performance function called *zGM enhanced multiple reader (enhanced reader)*.This function provides greater performance than the standard multiple readers or multiple utility volumes approach in earlier releases of zGM (previously XRC).

The enhanced multiple reader function removed the fixed association of primary volume to one reader task, enabling all SDM readers in the LCU to equally contribute to the data transfer of the modified data to the specified SDM.

The zGM enhanced multiple reader function improves copy bandwidth for high-write activity volumes. It does so by having multiple SDM readers process updates to a single volume if necessary.

Tip: The zGM enhanced multiple reader function is the preferred zGM configuration.

The enhanced reader approach defines a pool of readers in an LCU under one zGM session, to share equally in the reading and draining of the updates to the zGM volumes in the specified LCU. If a single primary volume has a high burst of update activity that exceeds the single reader capacity, the enhanced reader functionality spreads these updates across the LCU's multiple readers to achieve a much higher drain rate.

The key difference between the single reader and the enhanced multiple reader approaches is shown in Figure 1-3 on page 16. The single reader approach defines one or more readers in an LCU, where each reader is associated with a fixed set of primary volumes.

This implies that each of these readers in the LCU can only process updates to the volumes associated with that specific reader. A single reader does have a maximum data transfer rate, and a single primary volume or a few primary volumes can sustain a write update rate that exceeds the single reader bandwidth (capacity).

In addition to the increase in the single volume update rate that can be supported with zGM enhanced multiple reader, the equal sharing of the updates across the LCU readers also implies a notable improvement in the associated distance latency costs.

During high-demand periods, fewer SDM read cycles are necessary to transfer the same capacity. This results in fewer SDM read cycles for the same capacity transferred, and less time is wasted or used in distance latency.

The zGM enhanced multiple reader function treats all defined readers for an LCU as a pool of readers (a primary and one or more auxiliary sessions). All updates to the total set of mirrored volumes in that LCU are balanced across this pool of readers. If a single volume would have exceeded the throughput of a single reader, the enhanced support now spreads that single volume update demand across the primary and auxiliary sessions.

The number of control unit sessions that are used for a given LSS can be controlled explicitly, by manually adding or removing utility volumes using the same value for `SCSESSION` on each utility volume. On XQuery reports, `SCSESSION` is also called Storage Networking Industry Association (SNIA) Certified Storage Networking (SCSN).

Utility volumes are specified using volume pairs with a unique primary VOLSER, and an auxiliary VOLSER of XRC utility volume (XRCUTL). Utility volumes are added with the **XADDPAIR** command, and are removed with the **XDELPAIR** command.

When using the explicit management model (XRCUTL volumes are specified, and the value of `NumberReaderSessions` is not set), I/O operations for auxiliary sessions are performed using the specified utility volumes.

If the `NumberReaderSessions` setting specifies more storage control sessions for this LSS than there are utility volumes, therefore implicitly managing the number of auxiliary sessions, the aliases of the utility volumes are used (if present), or other available volumes somewhere in the same LSS are used (if there are no aliases). Always define sufficient XRCUTL volumes plus aliases to meet the defined `NumberReaderSessions` setting.

The zGM enhanced multiple reader (enhanced reader) concept is illustrated in Figure 1-3 on page 16. This diagram shows a scenario where the write update demand is sufficient to fill four readers to capacity, with leftover demand for the next read cycle. The write updates are balanced across the readers, even when the demand does not fill any single reader to capacity.

The *sidefile* is a logical concept that represents the use of cache in the primary storage subsystem, where updates are located before being read by the SDM reader tasks. When the sidefile is empty, no cache is used.

The storage subsystem cache is only assigned for subsystem use as the sidefile fills. For example, if there are 40 updates to be transferred to the SDM, and four readers to transfer those updates, each reader sidefile is expected to have 10 updates. An example zGM enhanced reader setup is shown in Figure 1-3.

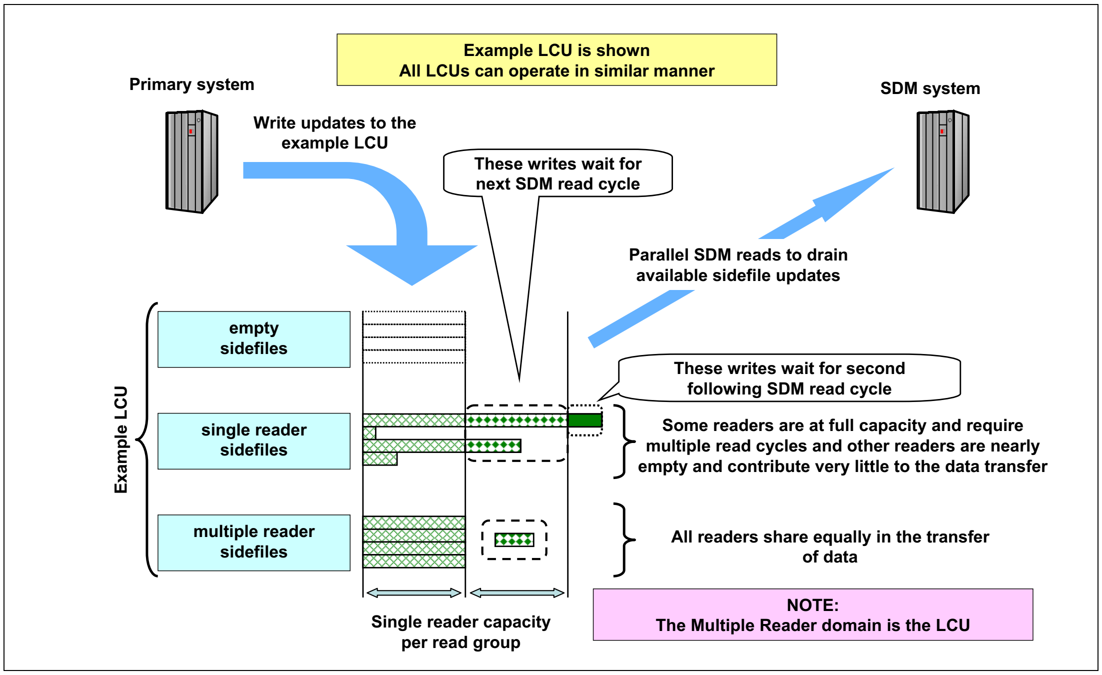

Example LCU is shown
All LCUs can operate in similar manner
Primary system
SDM system
Write updates to the example LCU
These writes wait for next SDM read cycle
Parallel SDM reads to drain available sidefile updates
Example LCU
empty sidefiles
single reader sidefiles
multiple reader sidefiles
These writes wait for second following SDM read cycle
Some readers are at full capacity and require multiple read cycles and other readers are nearly empty and contribute very little to the data transfer
All readers share equally in the transfer of data
Single reader capacity per read group
NOTE:
The Multiple Reader domain is the LCU

Figure 1-3 The zGM enhanced multiple reader concept

1.2.7 Management with GDPS in zGM

GDPS, an industry-leading availability solution that is offered through IBM Global Services, is a multi-site solution that is designed to provide the capability to perform the following tasks, thereby improving application availability:

- Monitor systems, disks, and tape subsystems.
- Automate Parallel Sysplex operational tasks.
- Perform planned and unplanned activities on system or site levels from a single point of control.

GDPS is independent of the transaction or database applications being used. It builds on the runtime capability of automation products, such as IBM Tivoli System Automation. These products provide the first level of automation, such as bringing up applications with no manual intervention during IPL. GDPS automation can coexist with an existing automation product. Also, GDPS relies on storage subsystems that support copy service technologies:

- The Peer-to-Peer Remote Copy (PPRC) architecture
- The XRC architecture
- The zGM architecture

To give clients flexibility, IBM offers the GDPS Qualification Program for other vendors to validate that their copy service structure meets the GDPS requirement.

GDPS/XRC

GDPS/XRC helps you manage replication, and it automates the process of recovering the production environment with limited manual intervention. This includes invoking the Capacity BackUp (`CBU`) command, consequently providing significant value, reducing the duration of the recovery window and requiring less operator interaction.

GDPS/XRC is specifically an automated DR solution. GDPS/XRC controls the remote mirroring, and automates the recovery of production data and workloads in the recovery site. The systems running GDPS/XRC are typically in the recovery site, remote from the production systems, and are not members of the sysplex at the primary site. After a disaster, the production systems are restored by GDPS/XRC at the recovery site.

GDPS/XRC makes it easier to control the resources and zGM during normal operations, planned changes, and after a disaster. GDPS/XRC supports the management of the SDM LPARs (shutdown, IPL, and automated recovery), and provides support for switching your production data and systems to the recovery site. User-customizable scripts can be used that control how GDPS/XRC reacts to specified error situations, and that can also be used for planned events.

GDPS/XRC also comes with a set of tools that help to monitor the health of a zGM environment.

Remote Copy Management Facility

One subset of the GDPS solution is the Remote Copy Management Facility (RCMF) offering. RCMF is an automated disk subsystem and remote copy management facility with a high-level user interface. This interface is implemented in the form of Interactive System Productivity Facility (ISPF)-like displays, and virtually eliminates the tedious and time-consuming work of using TSO commands.

The RCMF offerings are now generally replaced by more full-featured GDPS offering peers.

1.2.8 Metro zGM configuration

Some clients need a solution that provides an even higher level of data protection. You can combine zGM with the DS8000 Metro Mirror synchronous copy function. This is called a Metro zGM configuration, or MzGM.

Figure 1-4 shows the MzGM configuration.

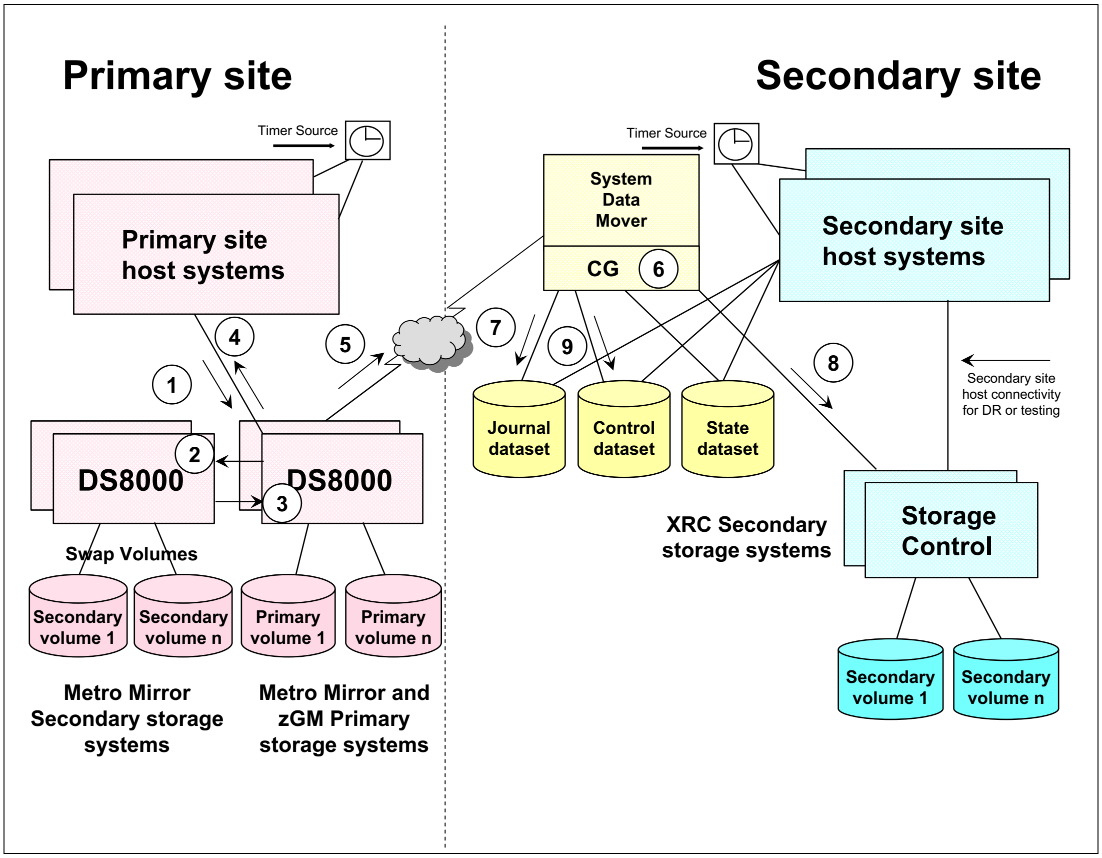


Figure 1-4 MzGM data flow

Figure 1-4 illustrates a simplified view of the MzGM components and the data flow logic. In this case, the primary storage systems perform a dual function, acting as the primary system for both the DS8000 Metro Mirror (synchronous) replication and the zGM (asynchronous) replication. This configuration requires the addition of the DS8000 Metro Mirror auxiliary volumes, called *swap volumes* in this example.

Swap volumes

Swap volumes are used in a three-site environment, which simultaneously includes DS8000 Metro Mirror for synchronous data mirroring and zGM for asynchronous data mirroring. This is called a Metro zGM (MzGM) environment. The swap volumes are the DS8000 Metro Mirror copies, or secondaries, that are not currently being used as zGM primary volumes. These volumes are located on separate storage systems from the zGM primary or auxiliary volumes, and can become zGM primary volumes at any time, if there is a need to swap the production workload to these storage systems.

The following steps form the logic for the MzGM data flow:

1. The primary system writes the update to the primary DS8000 system cache and NVS.
2. The primary storage system forwards the update to the auxiliary storage system cache and NVS.
3. The auxiliary storage system acknowledges successful receipt of the write updates to the primary storage system.

4. The application I/O operation completion is signaled. This is when channel end and device end are returned to the primary system. Therefore, the application write I/O operation has completed. The updated data is now mirrored asynchronously, as described in the following steps.
5. The DS8000 system groups the updates into record sets, which are asynchronously offloaded from the cache to the SDM system. As zGM uses this asynchronous copy technique, there is no performance effect on the primary applications' I/O operations, if there is no write pacing in place. For detailed information about how write pacing influences the performance of primary applications' I/O operations, see “Write Pacing” on page 60.
6. The record sets, perhaps from multiple primary storage subsystems, are processed into CGs by the SDM. The CG contains records that have their order of update preserved across multiple LCUs in a DS8000 system, across multiple DS8000 systems, and across other storage subsystems that are participating in the same zGM session.

   This preservation of order is vital for dependent write I/O operations, such as databases and their logs. The creation of CGs ensures that zGM mirrors data to the auxiliary site with point-in-time, cross-volume-consistent integrity.
7. When a CG is formed, it is written from the SDM real storage buffers to the journal data sets.
8. Immediately after the CG has been hardened on the journal data sets, the records are written to their corresponding auxiliary volumes. Those records are also written from SDM's real storage buffers. Because of the data in transit between the primary and auxiliary sites, the data on the auxiliary volumes is slightly less current than the data at the primary site.
9. The control data set is updated to reflect that the records in the CG have been written up to the consistency group time of the auxiliary volumes.

1.2.9 The MzGM Incremental Resync function

In case the synchronously mirrored volumes need to be swapped, the SDM would try to read from the DS8000 Metro Mirror auxiliary volumes. The SDM must be instructed to read from the new DS8000 Metro Mirror primary volumes instead. This is possible without a full initial copy from the swap volumes. However, it requires GDPS/MzGM. Only with GDPS/MzGM is an incremental resynchronization from the swap volumes possible.

Important: For the MzGM Incremental Resync feature (incremental resynchronization), the primary volumes and auxiliary volumes must be in the same DS8000 LSS.

Figure 1-5 on page 20 shows the configuration for MzGM Incremental Resync.

In a three-site mirroring configuration (also known as MzGM), planned or unplanned failover (IBM HyperSwap®) from the primary storage systems A to the DS8000 Metro Mirror auxiliary storage systems B is still permitted. When the HyperSwap occurs, and when the primary systems are running on the B volumes (swap volumes), the updates to these volumes are not replicated to the zGM auxiliary storage system C, because there is no zGM relationship between storage system B and storage system C.

To establish the zGM configuration between B and C without an incremental resynchronization, a full zGM initial copy is required. This full copy requires a long period of time if there is not a recent copy on C.

Figure 1-5 shows an MzGM Incremental Resync configuration.

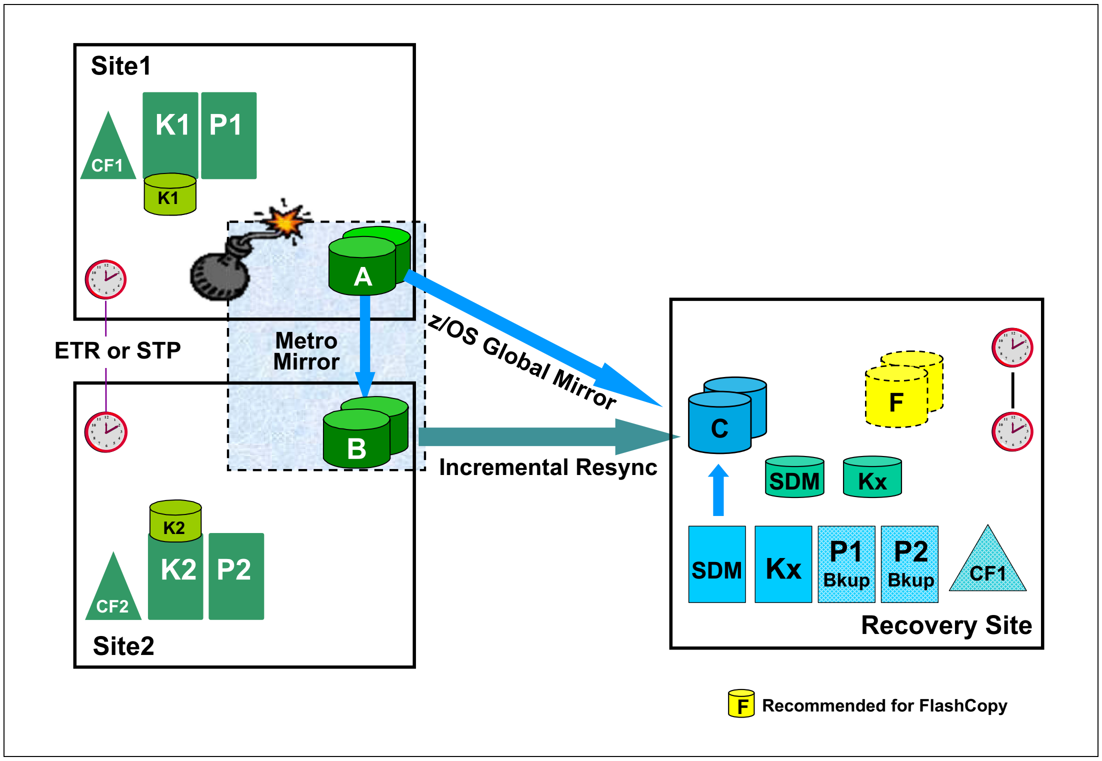


Figure 1-5 Sample configuration for MzGM Incremental Resync

With the GDPS/MzGM Incremental Resync feature, installation can avoid a full zGM initial copy to build up the zGM relationship between B and C. The MzGM Incremental Resync feature on primary storage system A synchronizes the zGM hardware bitmaps to the DS8000 Metro Mirror auxiliary storage system B. After HyperSwap, the Metro Mirror auxiliary storage system keeps track of the updates to the zGM hardware bitmaps on it. Therefore, it is possible to transfer only the updated data from B to C, instead of a full initial copy by zGM.

MzGM Incremental Resync can only be supported in conjunction with GDPS/MzGM, and is then called GDPS/MzGM Incremental Resync. It permits automatic failover/failback management between Ds8000 Metro Mirror primary storage systems and auxiliary storage systems, without affecting the primary systems' I/O. It is capable of a restart at the zGM recovery site when primary sites completely fail.

In addition, after a GDPS HyperSwap, it reduces the zGM resynchronization time from hours to minutes, by eliminating the full zGM initial copy. For detailed information about the GDPS/MzGM Incremental Resync function, contact your IBM services representative.

Note: MzGM Incremental Resync is a separately licensed feature of the DS8000 storage system. It is supported exclusively through GDPS.

2

Considerations about z/OS Global Mirror planning

The planning information presented in this chapter applies to most z/OS Global Mirror (zGM) installations:

- A two-site disaster recovery (DR) solution based on zGM and its basic interface, previously known as Extended Remote Copy (XRC)
- A two-site DR solution based on Geographically Dispersed Parallel Sysplex (GDPS)/zGM (also known as GDPS/XRC)
- A three-site solution based on GDPS/MM and GDPS/zGM with Incremental Resync, requiring some extra considerations

If zGM is in migrate mode, that is an exception that is addressed separately. For example, it requires less effort to use zGM to move volumes, or entire direct access storage device (DASD) configurations, in the course of site relocations or when consolidating data centers.

2.1 What to expect from z/OS Global Mirror

Just copying data on disk to another location is not enough to successfully resume IT operations at an auxiliary site, especially after a complete and prolonged primary site outage. Besides understanding the need to plan for other resources, it is also crucial to understand the implications to the overall IT environment when implementing zGM.

Statements, such as "zGM does not affect applications...", are not always completely true, and depend on how well zGM is configured and optimized:

- Correct configuration
- Sufficient resources
- Effective monitoring
- Informative reporting

The zGM product requires additional emphasis on measuring performance before and after zGM is implemented, to understand early on whether zGM performs as expected, or whether more optimization and tuning are required.

In the course of daily business, zGM also requires configuration adjustments before substantial redistribution of data and workload occurs. Adjustments are required, either within an existing zGM configuration, or when adding additional workload to an existing zGM configuration. In this context, it might be feasible to review the disk storage pool concept, and to position the installation to minimize the efforts needed to adjust a zGM configuration when the workload or capacity substantially changes.

2.2 Requirements for z/OS Global Mirror

Some planning is required before implementing zGM. There are the actual hardware and software requirements for zGM itself. In addition, it is important to understand that all involved application systems (in this example, z/OS images and z/OS logical partitions, or LPARs) need to have a common time reference. For z/GM, it is a Parallel Sysplex using the Server Time Protocol (STP). When only a single system is used, the time-of-day clock (TOD clock) is sufficient. This applies also to more than one LPAR if they run on the same system.

All write input/outputs (I/Os) to zGM primary volumes are time-stamped. These time-stamped write I/Os are processed by the SDM, to maintain the I/O sequence at the auxiliary site when replicating the write I/O to the auxiliary volumes.

The following sections subdivide the requirements in hardware and software categories, and also distinguish between server requirements and storage system requirements.

2.2.1 Hardware requirements

Two components in zGM interface with each other. There is a software component running under z/OS control, and a corresponding software component in the primary storage system firmware.

System z-related requirements

Represented by the SDM software component, zGM runs only under z/OS, which in turn needs a suitable System z server. Because SDM is widespread, and so are System z servers, usually a potential zGM client has the appropriate server environment already installed, and there is no need for more details about the System z server hardware required to run z/OS.

Primary disk storage system requirements

There is a slightly different perspective regarding the relevant disk storage systems, and the requirements for the primary disk storage system.

Primary storage system firmware requirements

The counterpart of the SDM is the firmware in the primary storage system. This primary storage system has to have the zGM feature activated. To verify that the feature is activated, use the **lskey** DSCLI command, as shown in Example 2-1.

Example 2-1 Display active features within a DS8000 zGM primary system

```
dscli> lskey IBM.2107-75TAKE5
Date/Time: April 8, 2013 10:56:58 AM MST IBM DSCLI Version: 7.7.0.566 DS: IBM.2107-75TAKE5
Activation Key                              Authorization Level (TB) Scope
==========================================================================
Encryption Authorization                    on                       All
Global mirror (GM)                          130.8                    All
High Performance FICON for System z (zHPF)  on                       CKD
I/O Priority Manager                        130.8                    All
IBM FlashCopy SE                            130.8                    All
IBM HyperPAV                                on                       CKD
IBM System Storage DS8000 Thin Provisioning on                       All
IBM System Storage Easy Tier                on                       All
IBM database protection                     on                       FB
IBM z/OS Distributed Data Backup            on                       FB
Metro/Global mirror (MGM)                   130.8                    All
Metro mirror (MM)                           130.8                    All
Operating environment (OEL)                 130.8                    All
Parallel access volumes (PAV)               130.8                    CKD
Point in time copy (PTC)                    130.8                    All
RMZ Resync                                  130.8                    CKD
Remote mirror for z/OS (RMZ)                130.8                    CKD
dscli>
```

The DS8000 in Example 2-1 has a pretty good list of activated features. The last line of this **lskey** output shows that it has zGM, labeled as *Remote mirror for z/OS (RMZ)*, and indicated as actively covering 130.8 terabytes (TB) of primary storage capacity.

You can also see the RMZ Resync listed, which indicates the incremental resynchronization support for zGM in MzGM configurations. Note that MzGM with RMZ Resync support needs GDPS management to be used.

> **Tip:** Upgrade the disk storage systems to the latest firmware level before you start to implement zGM.

Primary storage system cache requirements

Another crucial resource with the primary storage system is cache size, because of the intermediate buffering needs of the write data in the primary storage system cache.

> **Tip:** Plan for sufficient cache in the primary storage system.

Although the current cache size might be sufficient to service decent response time with good read hit ratio to the application I/O, it can become a limited resource as soon as zGM competes with cache storage as well.

Experience shows that cache size needs to be increased before implementing zGM. Of course, this is only possible when the maximum cache size is not already installed.

Note that SDM reads the data directly out of primary storage system cache, through standard Fibre Channel connection (FICON) I/O, but the data is highly optimized and concatenated. After the data has arrived safely at the auxiliary site, and is secured in journal data sets, the corresponding cache segments in the primary storage system cache are released.

Auxiliary disk storage system requirements

The auxiliary storage system hosts all of the auxiliary volumes that correspond to their respective primary volumes. Each individual primary volume corresponds to an individual auxiliary volume of the same size and *device geometry*. Note that device geometry in this context is *track capacity and number of tracks per cylinder*. Again, a good practice is for the primary and auxiliary volumes to be the same size.

The auxiliary disk storage system receives almost all I/O resulting from a write I/O to zGM primary volumes. SDM performs this write I/O to the auxiliary disk storage system through conventional FICON I/O, again highly concatenated. No particular capabilities and license requirements are needed for the auxiliary disk storage system, *if* it either *serves only as a zGM auxiliary device*, or *is natively connected to z/OS as a simplex volume*.

Pay attention to the resource use within the auxiliary storage system when it serves more than SDM write I/Os to the auxiliary volumes. For instance, when FlashCopy is used to create tertiary zGM volumes (FlashCopy target volumes), the server part in an auxiliary DS8000, the device adapters (DAs), and the back-end ranks might experience additional workload, potentially overloading these resources.

For example, suppose that you start to use FlashCopy to copy all of the auxiliary volumes in a suspended zGM session before you start to re-establish the zGM session. At the same time, application systems continue to update the suspended primary volumes, which are now in a `PENDING` state. This has the potential to put extra stress on the auxiliary storage, which must carry all of the following activities at the same time:

- Resynchronize activities to all auxiliary volumes.
- Update application activities that also arrive to auxiliary volumes.
- Service all of the FlashCopy relationships for all auxiliary volumes, using system resources, such as DA ports, Fibre Channel loops, and hard disk drives (HDDs), also referred to as *back-end resources*.

These activities can slow down SDM replication performance. Therefore, consider configuring the auxiliary storage systems at least as the primary storage systems. Also consider adding additional resources to the auxiliary storage systems when regular FlashCopy activities are planned on top of SDM activities.

Auxiliary storage system firmware requirements

As soon as the zGM auxiliary storage system becomes the zGM primary storage system, in the course of a failover/failback process, the required zGM firmware feature also needs to be active and available at the auxiliary storage system. In this case, the same requirements apply as for the primary storage system, as described in “Primary storage system firmware requirements” on page 23.

Auxiliary storage system cache requirements

Pay attention when planning for different technology at the primary and auxiliary sites. For example, when you add new disk storage systems into the configuration with new and improved technology, you might consider placing the new disk storage system as the zGM auxiliary storage system (so at the auxiliary site) to get the best zGM system performance and recoverability.

Basically, changing an existing zGM configuration that was already well-balanced requires you to look into all of the other components in this configuration. Replacing the system that hosts the SDMs with faster processors, faster channels, and faster access to main storage might reveal shortcomings in other components, such as reaching the bandwidth limit between both sites.

Auxiliary storage system configuration best practices

A good practice is to *configure the auxiliary disk storage system in a similar manner as the primary disk storage system*. This includes cache size, back-end configuration, and disk storage technology.

Furthermore, consider the simplest method of mapping the primary volumes to the auxiliary volumes, through a one-to-one configuration, In such a configuration, every primary disk storage system, and every logical control unit (LCU) or subsystem identifier (SSID), has a corresponding identical auxiliary disk storage system and LCU (or SSID).

Ideally, all LCUs must have the same structure and layout. Such a configuration approach will ensure that, if the primary disk storage systems are balanced, the auxiliary disk storage systems will also be balanced. However, the following factors can potentially inhibit the balance of this type of configuration:

- Different technology is used for the primary disk storage system and the auxiliary disk storage system.
- Different HDDs are used for the primary and auxiliary disk storage systems, so there is no ability to double the HDD capacity for the auxiliary disk storage subsystem to accommodate tertiary copies.
- Tertiary copy activity in the auxiliary disk storage system (FlashCopy of auxiliary volumes) over-commits resources due to regular FlashCopy activities.
- There is a large primary disk storage subsystem, and its corresponding auxiliary disk storage subsystem cannot be doubled in size to accommodate tertiary copies.

2.2.2 Software requirements

In this example, software requirements apply only to z/OS.

Although zGM is an integral part of z/OS, and SDM is a component of DFSMS data facility product (DFSMSdfp), it is still a good practice to apply the latest maintenance to the z/OS systems hosting the SDM address spaces. The best practice in this context also calls for the most recent z/OS version and release.

2.2.3 Considerations for zSeries/Virtual Machine

When zGM is also going to manage volumes that are attached to zSeries/Virtual Machine (z/VM) systems, these volumes are either defined to z/VM as unsupported disks, or as full volume minidisks.

For a disk device that is defined as unsupported under z/VM, the **SET PATH GROUP** command is unable to reestablish the path group after a path has been removed.

The SDM can run either on a separate z/OS LPAR, or on a z/OS guest that is running on the same or a different z/VM system. If a z/OS guest system is used for SDM, the volumes must be defined as unsupported to the host z/VM system.

If applications run on an older local z/VM system, writes are not time-stamped.

In z/VM V6.1 and later, time-stamping is supported. See the following link for the announcement and software maintenance details:

http://www.vm.ibm.com/zvm610/

z/OS and Linux operating systems perform time-stamp writes even when running as guests under z/VM.

2.3 Initial z/OS Global Mirror planning considerations

Before providing information about planning considerations, this section briefly reviews the key purpose of zGM, and provides information about its potential configuration challenges, as depicted in Figure 2-1.

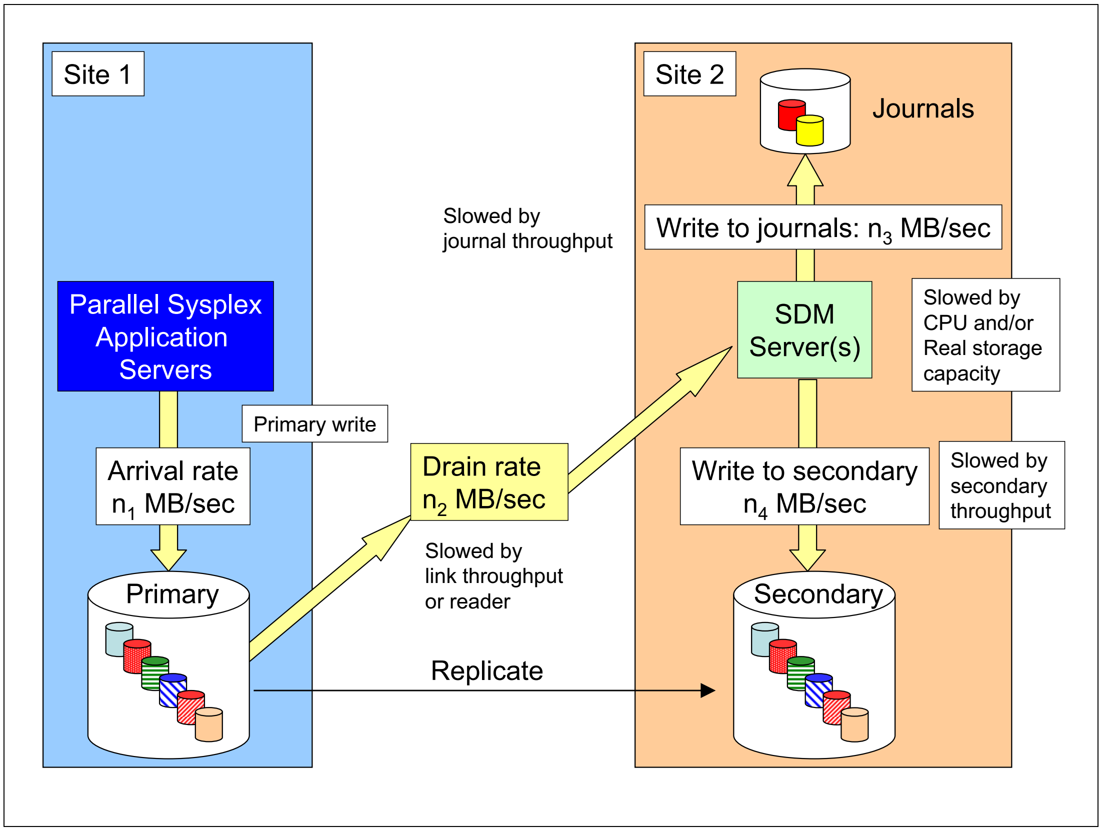

Site 1
Site 2
Journals
Slowed by journal throughput
Write to journals: n_3 MB/sec
Parallel Sysplex Application Servers
SDM Server(s)
Slowed by CPU and/or Real storage capacity
Primary write
Arrival rate n_1 MB/sec
Drain rate n_2 MB/sec
Write to secondary n_4 MB/sec
Slowed by secondary throughput
Slowed by link throughput or reader
Primary
Secondary
Replicate

Figure 2-1 The zGM primary write data flow and date rates

Assuming average data rates as indicated in Figure 2-1 (and assuming average data rates are feasible in the context of asynchronous data replication), we observe a certain amount of data arriving in the primary storage systems. Basically, SDM's role is to replicate the data with as little recovery point objective (RPO) offset as possible.

In this case, to *replicate data* implies reading the data from the primary disk storage systems, pulling the data over the network infrastructure between both sites, and applying the data twice at the auxiliary site (Site 2):

- Once to the journal data sets
- A second time when applying the data to the corresponding auxiliary volumes

The goals are to contain RPO between Site 1 and Site 2, and not to overrun the temporarily used cache storage in the primary storage systems. To accomplish these goals, it is necessary to establish an infrastructure that satisfies the following inequations regarding the infrastructure between sites, SDM resources, and the connectivity between SDM and its associated storage subsystems.

The first example ($n_1 < n_2$) is common. For completeness, the other conditions are briefly reviewed here, although they are not usual in ordinary circumstances:

$n_1 < n_2$

The amount of write data that arrives at the primary storage systems is *on average* (the mathematical average, and in this context it might be the average over 24 hours in the most busy workload period) smaller than what can be pulled by SDM out of the primary storage systems and transferred through the infrastructure between Site 1 and Site 2.

The difference between n_1 and n_2 determines the RPO during different workload phases during the day-end, week-end, month-end, quarter-end, and year-end processing. The bigger the difference ($n_1 << n_2$), the smaller the maximum RPO will be.

During a brief peak period, the arrival data rate n_1 might be bigger than n_2, and is temporarily cached in DS8000 cache sidefiles. Again, this condition ought not to last for more than a few minutes, showing the RPO value and suspending the sessions.

$n_1 = n_2$

This is the unlikely case that the arrival rate of write data to be pulled over to Site 2 is exactly the same as the maximum that can be handled by SDM.

In this circumstance, you most likely will never reach DR readiness during the first attempt to establish the XRC session. After the session reaches full `DUPLEX` under these conditions, it would just survive the current configuration and workload, and this would be insufficient, and too short on infrastructure and other resources, for SDM to complete the task.

$n_1 > n_2$

Due to insufficient bandwidth and other shortcomings on the SDM site, the arrival rate of application write data at Site 1 immediately exceeds the bandwidth capabilities.

In this case, you will never reach DR readiness, but eventually overrun the primary storage system cache and reach suspend conditions.

In addition to these requirements, you also need to accommodate SDM with resources in Site 2, in terms of central processing unit millions of instructions per second (CPU MIPS), sufficient real storage for SDM, and channel paths connectivity. This ensures certain conditions ($n_2 < n_3$? n_4) regarding data throughput in Site 2:

- The difference between n_2 and n_3 has to be observed with tools, such as those mentioned in Chapter 6, "Scenarios for ongoing operations" on page 79.
- This is particularly of interest in case of changing workloads, redistribution of workloads, and especially when the n_1 megabytes per second (MBps) numbers change.
- It is a good practice to start with about $n_3 = 1.5 \times n_2$ as a general guideline.

- Usually, the infrastructure for n_4 offers more bandwidth than n_3, because it is configured to accommodate the entire workload in case of a site switch to Site 2.
- So the conditions are $n_1 < n_2 < n_3 ? n_4$, or even better, $n_1 < n_2 < n_3 < n_4$.

The key number is n_2. To identify this crucial number, you need to plan for collecting meaningful data during peak workload periods, using IBM System Management Facility (SMF) and IBM Resource Measurement Facility™ (IBM RMF™). Then, either calculate the number through available tools in the installation, or ask IBM support for a bandwidth analysis based on the SMF and RMF data.

This concludes the basic considerations for planning. The following sections contain information about more detailed planning considerations.

2.3.1 Determining what data should be copied

A simple approach might be to replicate all volumes from Site 1 to Site 2. Basically, this is a valid approach, and obviously the most simple solution to identify the volumes that need to be included. However, this also leads to the biggest possible n_2 number, with all configuration implications on the infrastructure and required resources (see Figure 2-1 on page 26).

When bandwidth and SDM resources are under technical and economical pressure, you will need to identify what data or volumes can be exempted from being continually replicated, because their content is not necessarily required for recovery at Site 2.

Volumes that do not have to be mirrored (copy-once volumes)

It might be possible to exclude some volumes from active mirroring, although they are still required at Site 2. These volumes include those that hold permanently allocated data sets, but their contents are not required.

An example of such volumes is paging data sets. To enable a recovery at Site 2, these volumes must be available in Site 2. SDM still might copy those volumes to Site 2. The corresponding volume pair can be deleted as soon as they reached full `DUPLEX` state.

Another option might be to establish a separate XRC session with the session type `MIGRATE`, include all volumes that only need to be copied, and end this migration session when all volumes have reached full `DUPLEX` state. A variation of this approach might be to suspend the migration session. Repeat this `copy` process on a periodic basis, or as required (for instance, after allocating a new data set).

To further limit write data, some installations have set up separate storage management subsystem (SMS) storage groups for work volumes, to store temporary data sets and other data irrelevant to potential recovery actions. Therefore, these volumes contain volatile data and do not have to be constantly replicated. These volumes might also be put in the category just to be copied, to ensure that the work environment is available in Site 2. Remember that Site 2 might be used for a planned failover.

When there is a strong need to further cut down on the volume of write data, you can also consider not replicating volumes that can be easily re-created, such as database volumes used in data mining applications.

Lastly, you might identify a set of volumes or storage groups that contain data with recovery point objectives (RPOs) and recovery time objectives (RTOs) not as stringent as those volumes in a production environment. This enables you to use other DR methods to recover the data. Some test data might fall in this category.

If still exercised in an installation, stand-alone dump volumes might also be excluded from SDM replication management. They might fall in the category just to be copied, to ensure that the space is available at Site 2.

Coupling facility considerations

Persistent structures in the coupling facility (CF) will usually not be available at Site 2. The use of zGM is usually related to long-distance data replication, and therefore is out-of-reach for CF duplexing, which in turn would require additional configuration considerations at Site 2. Even when using zGM within sysplex distances, these structures would be unusable, because they would not be consistent with the recovered disk.

When system logger is used, the log stream should be backed by disk staging data sets. Otherwise, the data that is stored only in the CF structures and data spaces will not be available during recovery at Site 2. This applies to any of the system logger exploiters, such as IBM CICS® log domain and IBM IMS™ shared message queues.

Another IMS consideration applies to multiple area data sets (MADS). All MADS must be included in the SDM sessions. Otherwise, it cannot be ensured that the copy of the MADS will be consistent at Site 2 during recovery.

Last but not least concerning IMS: the write-ahead data set (WADS) is usually also duplexed, as with most logging volumes. In this case, it is not necessary to have both WADS included in the SDM session if the spare WADS are included in the SDM session. If dynamic allocation of the WADS is used, they can become active at Site 2.

2.4 System Data Mover considerations

SDM manages the zGM activity for all volume pairs within a zGM session. This section describes some planning considerations for SDM.

For optimal performance, it is a good practice to run SDM or coupled SDMs within one or more dedicated z/OS LPARs. Because the SDM address space runs at master priority, this avoids the SDM from interfering with other workloads running on the same z/OS LPAR. Note that SDM address spaces are not subject to system resources manager (SRM) controls.

Clustering SDMs in a dedicated z/OS LPAR ensures that the appropriate resources, such as CPU, real storage, and I/O, are available to each SDM instance in that LPAR. This also includes bandwidth requirements for primary and auxiliary disk storage systems, and bandwidth for journal data sets. Details are provided in the following sections.

2.4.1 SDM location

As explained in *z/OS DFSMS Advanced Copy Services*, SC35-0428 (available at `http://publibz.boulder.ibm.com/epubs/pdf/antgr190.pdf`), the SDM can run on Site 1 (the primary site), Site 2 (the auxiliary site), or on an intermediate site. Most zGM clients implemented the SDMs at Site 2 for the following reasons:

- Depending on the distance, the cost of the infrastructure between sites is a substantial part of the overall solution. In particular, the cost will go up when the required bandwidth increases.

  Bandwidth requirements will be the lowest when the SDMs run on Site 2, because bandwidth capacity is needed only to retrieve updates from the primary disk storage subsystem.

- If the SDMs run in Site 1, and the journal data sets are in Site 2, twice the bandwidth is required:
  - The data must be written remotely to the journal data sets.
  - After that, the data must be written to the auxiliary disk storage system.

  To ensure the recovery of the auxiliary systems, the SDM must have access to the auxiliary disk systems, the journal data sets, the SDM state, and the control data sets. These data sets must be in the same location as the auxiliary disk systems, as shown in Figure 2-2.

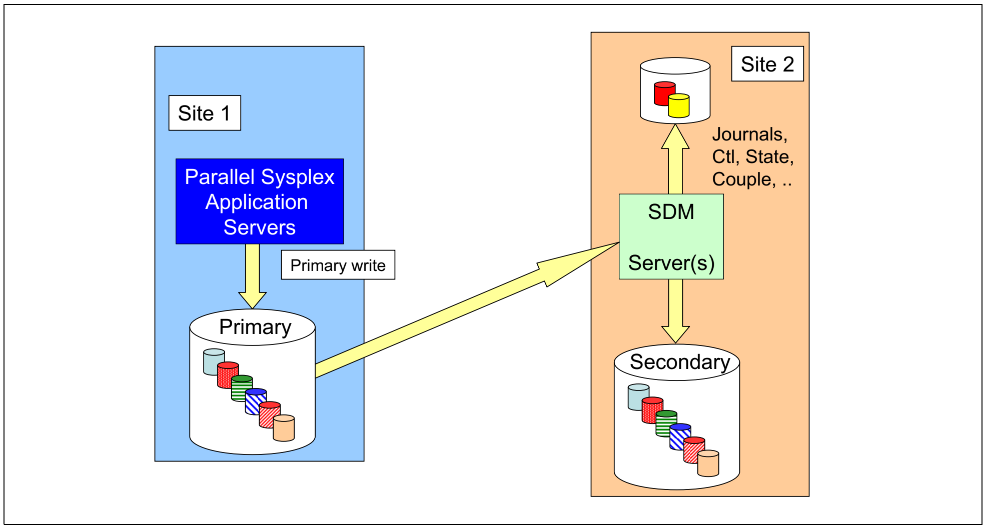

Site 1
Parallel Sysplex Application Servers
Primary write
Primary
Site 2
Journals, Ctl, State, Couple, ..
SDM
Server(s)
Secondary

Figure 2-2 Suggested SDM location on Site 2

Enough processing power must be available at Site 2 not just for the SDM systems, but also for running all business-critical applications after a failover to Site 2. SDMs will probably no longer be needed after a failover to Site 2, so its processing power can be used with other processing power at Site 2 to run all production applications.

2.4.2 SDM resources

Each SDM instance runs as a system address space with high dispatching priority. It is non-swappable, and uses resources such as CPU, storage, and I/O. SDM address spaces follow a strict naming convention, and start with “`ANTAS`” followed by a 3-digit number (starting with `001` and increasing in increments of one). So, the first SDM address space is `ANTAS001`, the second SDM address space is `ANTAS002`, and so on up to the maximum number of 20 SDM instances within the same z/OS LPAR.

When clustering more than one SDM in the same z/OS LPAR, a separate address space is required to coordinate the SDMs within this particular LPAR. Its role is to provide dependent write consistency across all SDMs and their sessions, for all auxiliary volumes at Site 2, at any time.

There are two more system address spaces available, with a name also starting with “ANT” they will always automatically be started during initial program load (IPL). The functions that are provided by these address spaces are briefly described in Table 2-1.

Table 2-1 Address space descriptions

Address space	Description
ANTMAIN	This address space handles all concurrent copy activities, is always active, and is created during system IPL. Canceling this address space automatically restarts this address space.
ANTAS000	This address space handles Time Sharing Option (TSO) commands and application programming interface (API) requests that control zGM, is always active, and is created during system IPL. It automatically restarts after an address space cancel request.
ANTCL*nnn*	This address space manages zGM sessions in a z/OS LPAR. The sessions are grouped or coupled together to a zGM master session, and logically appear as a single replication instance.
ANTAS001 through ANTAS020	Each **XSTART** command creates an SDM instance and its concerned address space, with an incremental number from 001 - 020 in the address space name.

As previously stated, you can run up to 20 zGM sessions with address spaces of ANTAS001 through ANTAS020 in a single logical partition with the following limitations:

- If you enable clustering for a logical partition, you can run up to 19 zGM sessions, but no more than 13 zGM sessions can be coupled in that logical partition.
- If you disable clustering, you can run up to 20 zGM sessions.

Other factors, such as real storage capacity, auxiliary storage capacity, and available processor capacity, might limit the effective number of SDM sessions in a z/OS LPAR to fewer sessions than the architectural limits make available.

2.4.3 Storage requirements

In general, a System z processor, running under the control of a recent z/OS version and release, provides sufficient virtual and real storage to potential SDM address spaces. The following sections provide information about some storage requirements to support SDM.

Virtual storage considerations

A good practice is not to restrict virtual storage for any SDM address spaces. If you consider restricting virtual storage of an ANTAS*nnn* address space, use 2 GB as a useful minimum to avoid fragmentation issues, and the related increased virtual storage management resource usage.

Almost all of the required storage is in the private area of the SDM address space.

Real storage considerations

SDM does heavily exercise I/Os, and therefore uses many and large buffers for its I/O activities. When these buffers are used for I/O operations, they have to be backed up by real storage. To save CPU cycles, and to ensure best data throughput, you keep the related real storage pages fixed to avoid page fix and page release resources.

There are several types of buffers that go into the equation when evaluating the required buffer space:

- Buffers used to host channel programs
- Buffers used to synchronize volume pairs

- Buffers used by access methods to read and write journals and control data sets
- Buffers used to read from primary storage subsystems, and to write to the auxiliary storage subsystems (this is the largest space for required buffers)

The amount of real storage required depends on several factors, and can be calculated following the details outlined in *z/OS DFSMS Advanced Copy Services*, SC35-0428.

A good practice is to control the real storage requirements for SDMs through the following parmlib parameters, shown in Example 2-2, and apply the indicated numbers to control the SDM storage usage.

Example 2-2 Set storage parameter for SDM in parmlib

```
/* ******************************************************************** */
/* ANTXINxx                                                             */
/*          SDM related parmlib member to set SDM parameters            */
/*          Details in z/OS DFSMS Advanced Copy Services, SC35-0428     */
/* ******************************************************************** */
/* Good practice to set storage parameters for SDMs are the :  */
/* ******************************************************************** */
STORAGE                                                                -
          BuffersPerStorageControl (25000)                             -
          TotalBuffers             (25000)                             -
          IODataAreas              (2048)                              -
          PermanentFixedPages      (1500)                              -
          ReleaseFixedPages        (NO)
/* ******************************************************************** */
/* In this context you would also define to utilze zIIP if available:   */
/* ******************************************************************** */
STARTUP                                                                -
          zIIPEnable               (FULL)
```

Total buffers per Example 2-2 are about 1.5 GB, considering that a record set buffer is 60 KB. The amount of 1,500 fixed pages is just another 6 MB, and IO data areas for the channel programs are negligible in this context. Therefore, you should plan for approximately 1.7 GB real storage for an SDM instance.

Experience shows that real storage shortages contribute most to increasing average session delay times, and to increasing CPU consumption. They have the potential to reduce the SDM overall throughput, which in turn might end in session suspends due to primary storage system cache being exhausted.

Paging space considerations for SDM LPARs

Some paging space is required by zGM to support the trace data space and the data movement address spaces. When you use more than a single SDM in its dedicated LPAR, multiply the numbers shown in Table 2-2 by the number of SDMs running in the same LPAR. Usually, page data sets are sufficiently sized to be large enough to hold more data than shown in Table 2-2.

Table 2-2 Page space requirements per address space

Address space	Data space requirements	Address space requirements
ANTCL*nnn*	15 MB	1 MB
ANTAS*nnn*	15 MB	Up to 2 GB per address space

2.4.4 Processor requirement considerations for SDM

SDM is running mainly I/O operations and is, therefore, less CPU-bound and more I/O-bound. Alternatively, SDM is working highly in parallel, and takes advantage when there is more than one processor available for SDM workload, which leads to fewer resources being dispatched.

SDM processing and code execution became eligible on System z Integrated Information Processor (zIIP). This possibility helps to free up central processors (CP), and lower overall total cost of computing for selected data and workloads. Because SDM is eligible for the zIIP, it enables you to offload the zGM mirroring workload from CPs, optimizing overall resource use.

A good practice is not to limit SDM's LPAR to too-small CPU resources, because it then also has the potential to slow SDM down. Record sets keep growing in the primary storage subsystems cache sidefile, until they reach suspend conditions and display defined RPO limits.

General guidelines used in the past for numbers, such as 5 MIPS for 100 writes, are not helpful in this case, and might vary depending on the kind of write I/Os and the average transfer size. Nowadays with single-CP performance and MIPS in the range of 250+ MIPS, plan for peak workload and some additional resources for the LPAR that hosts the SDMs.

When running SDM in an LPAR with shared processors, ensure that the following conditions are met:

- The SDM has the highest logical CP share on the processor, to ensure that it has preemption priority for I/O interrupts if preemption occurs. Therefore, the scheduler ensures that the CP processes the highest-priority task of all those tasks that are currently ready to run, with SDM tasks running at high dispatching priorities.
- The SDM has a high IBM System z Processor Resource/System Manager (IBM PR/SM™) weight, to ensure that the logical CP share is not shortened too much if it begins to manage processor use.

The following items summarize the good practices shown in this section:

- Run the SDM or coupled SDMs in one or more dedicated LPARs.
- Assign sufficient real storage to each SDM LPAR.
- Do not constrain or limit CPU resources for SDM LPARs.
- Provide enough bandwidth between Site 1 and Site 2, and to auxiliary disk storage systems and journal data sets.

2.4.5 Initializing volume pairs

In general, the order in which primary devices are added does not matter. However, there are a few specific circumstances where order might be important.

Synchronization order

Most clients use the SDM to automate and control the volume synchronization process, by using the **`SYNCH`** and **`SCSYNCH`** parameters. In some cases, it might be preferable to synchronize certain volumes before others. These cases occur when either of the following conditions exist:

- Limited bandwidth and high activity volumes exist. If there is a limited amount of bandwidth between Site 1 and Site 2, having the most active primary devices in `DUPLEX` mode for the majority of the synchronization process, might unnecessarily prolong the synchronization time. You can include these volumes at the end of the list of **`XADDPAIR`** commands to reduce this affect.

- Primary Redundant Array of Independent Disks (RAID) arrays are overloaded. If a physical RAID array on the primary disk system is operating close to its performance limit (for example, exceptionally high read activity), starting several synchronization tasks on the devices in that array at the same time might result in unacceptable response time degradation for the primary applications.

  In this case, spread the volumes specified with the **`XADDPAIR`** command so it is unlikely that these volumes will synchronize at the same time.

  Note: Due to the use of extent pool striping, good-sized extent pools (with a decent number of ranks in an extent pool), combined with Easy Tiering, the chance of overloading individual ranks is less likely than before.

The XRC synchronization process is self-pacing, and should be able to be performed at any time during the primary application workload cycle.

Remember that the objective of the synchronization process is to get all volume pairs in `DUPLEX` mode as quickly as possible, while not causing problems to the primary applications. The synchronization speed of a single volume is not critical if the overall throughput achieved is satisfactory. If volume synchronization affects the primary application I/O, the SDM is designed to dynamically suspend the initialization and resynchronization process.

2.4.6 SDM session considerations

Up to 20 zGM sessions are permitted in each z/OS system image managed by an SDM control session. The zGM product is extremely scalable, and can manage more than 20,000 volume pairs, by spreading the SDM sessions across up to 14 z/OS images. This expands the SDM management from a single z/OS image clustered session to a coupled extended remote copy (CXRC) configuration across 14 z/OS images. Such a zGM master session can hold up to 182 SDM sessions.

A more crucial consideration is how big an SDM session can be. There is not particular limit, and the number of volumes in an SDM session is determined by the maximum number of storage control sessions that can be handled by a single SDM session. In theory, this is approximately 40 x 256 primary volumes, and the same number of auxiliary volumes (10,240 XRC volume pairs).

However, this does not take into consideration all of the other resources needed, and especially bandwidth between sites. Generally, do not exceed 32 primary storage control sessions. More realistically and from experience, other criteria come into account:

- Number of MBps
- Number of input/output operations per second (IOPS) issued from the application servers

Usually, the MBps are the key factor, and to some extent also the volume size. When the number of MBps and the number of IOPS is unknown, you need to use some bandwidth sizing tool for a zGM configuration, and research the application write workload.

With a general guideline of 1,500 primary volumes maximum, it still depends whether these 1,500 primary volumes are 3390-3 volumes or 3390-54 volumes. With 3390-54 volumes, it is necessary to understand the I/O load from the application servers, and most likely you need to split these volumes into two or more additional SDM sessions. To maintain write sequence consistency between SDM sessions, couple these SDM sessions (as previously mentioned).

If possible, it is useful to group the zGM volume pairs by application type, and then spread different applications and their zGM primary volumes in individual SDM sessions. With large or exceptionally busy applications, you might have to create more than one SDM session. There are installations with small capacity configurations but exceptionally high IO rates, and a 1:1 read/write ratio (or even more writes than reads). In these installations, an SDM session might span only a few hundred SDM volume pairs.

2.5 Hardware definition considerations

There are a number of considerations for how to define the primary and auxiliary disk systems in the input/output definition file (IODF). In a typical zGM scenario, the SDM system and recovery system are at a remote site, with only SDM connectivity between the two sites. Another possible scenario is that the two sites are within sysplex distances, and a more general cross-site connectivity exists.

2.5.1 Primary site definitions

In a typical zGM configuration, the primary systems generally do not have access to the auxiliary disk systems. If dedicated XRC utility devices (UD), also called *fixed UD*, are being used, they can be defined at IPL as offline to the primary systems. This method prevents I/O to these volumes, and reduces the chance of interaction between the SDM and the primary systems.

2.5.2 Recovery site definitions

In a typical zGM configuration, there is a separate IODF for both the primary and recovery sites. The auxiliary IODF should be transferred to the primary site whenever it is updated, to ensure that it is available to the primary systems when a disaster occurs.

Note: Generally, the scope of a client's IODF is single site. A client with several sites might not necessarily have a single IODF to cover all sites. Therefore, it is likely that there is a single IODF for both environments.

You can have a number of operating system configurations (OSconfigs) at the recovery site:

- SDM system OSconfig. The SDM system requires access to both the primary and auxiliary disk systems. If FlashCopy is used, this system also requires access to the FlashCopy disks (unless some solution, such as GDPS controlling system, is used to perform the FlashCopy operation). For the SDM systems, the primary, auxiliary, and tertiary devices can be defined with the unit control blocks (UCBs) above the line, because SDM supports UCBs above the line.

  The SDM primary, auxiliary, and tertiary devices can all potentially be defined as offline at IPL. In that case, the devices must be brought online before use. The following factors are advantages of defining the primary devices as offline:

  - Only those devices that are copied can be varied online.
  - You can avoid potential problems at IPL when there are connectivity issues.
  - You can avoid potential duplicate device problems at IPL.

    An advantage of defining the auxiliary devices offline is that you can avoid potential duplicate device problems at IPL. If a FlashCopy operation is done with offline secondaries, the tertiary devices do not need to be brought online to the SDM systems, assuming that the FlashCopy operation is performed from the SDM systems.

- OSconfig for recovery on auxiliary or tertiary devices. This OSconfig must have the auxiliary or tertiary devices (but no others) defined to the operating system. It must also have any other production-only devices defined that might not be defined to the SDM OSconfig.

2.5.3 Recovery site in a production sysplex

If the recovery site is within the production sysplex, or there is standard FICON distance connectivity between primary systems and auxiliary disks and vice versa, the hardware configuration definition (HCD) considerations are different than those in a typical zGM environment. There will normally be a single IODF for both sites.

It is possible that the processor in the primary site has connectivity to the auxiliary disk system. In this case, it would be essential to have two OSconfigs for the primary systems, with either the primary or auxiliary devices offline at IPL.

To ensure that the correct version of the volumes is used, do not define alternative devices to the OSconfigs. However, if alternative devices are not defined, the SDM must have a separate OSconfig.

2.5.4 HyperPAV

Enabling parallel write I/O for zGM auxiliary devices, and also to the fixed utility device in the corresponding primary system, can improve overall data mover throughput, and reduce the average session delay. It is a good practice to enable parallel writes to the auxiliary logical subsystems, which are configured in the same fashion as its corresponding primary storage subsystem. Define each concerned LCU with its 3390B and 3390A volumes correspondingly in either site.

Do not consider obsolete static alias management practices, but only Hyper parallel access volume (HyperPAV). See Example 2-1 on page 23.

To activate HyperPAV, update the parmlib member **IECIOSxx**, and add the parameter, as shown in Example 2-3.

Example 2-3 Activate HyperPAV in parmlib member IECIOSxx to survive IPLs

```
HYPERPAV=YES
```

To check whether HyperPAV is active, issue the IBM MVS™ system command shown in Example 2-4 from the MVS console.

Example 2-4 Verify whether HyperPAV is active

```
D IOS,HYPERPAD HYPERPAV
IOS098I 11.12.47 HYPERPAV DATA 213
HYPERPAV MODE IS SET TO YES
```

For detailed HyperPAV information on a certain MVS device, use the DEVSERV command shown in Example 2-5.

Example 2-5 Query HyperPAV information on certain MVS device address

```
DS QP,8100,HPAV
IEE459I 11.06.27 DEVSERV QPAVS 203
HOST SUBSYSTEM
CONFIGURATION CONFIGURATION
------------- --------------------
```

```
UNIT UNIT UA
NUM. UA TYPE STATUS SSID ADDR. TYPE
---- -- ---- ------ ---- ---- ------------
8100 00 BASE-H 8100 00 BASE
**** UNLISTED DEVICE(S) AND REASON CODES :
8180(0E) 8181(0E) 8182(0E) 8183(0E) 8184(0E) 8185(0E) 8186(0E)
. . . . . . . .
*** (0E) - DEVICE IS A HYPERPAV ALIAS
*** DEVICE(S) IN HYPERPAV ALIAS POOL
```

In order to dynamically activate HyperPAV, you can use the command shown in Example 2-6.

Example 2-6 Dynamically activate HpyerPAV

```
SETIOS HYPERPAV=YES
IOS189I HYPERPAV MODE CHANGE INITIATED - CONTROL UNIT CONVERSION WILL
COMPLETE ASYNCHRONOUSLY
```

Note that after the next IPL, this setting is lost. The better way is to put the HyperPAV activation step into parmlib.

Note that when you issue another **DISPLAY** command immediately after the **SETIOS** command, you might get a REQUEST REJECTED message, as shown in Example 2-7.

Example 2-7 HyperPAV activation in progress

```
D IOS,HYPERPAV
IOS085I REQUEST REJECTED. CHANGE/DISPLAY ACTIVE
```

As SETIOS in Example 2-6 shows, this process is happening asynchronously, and obviously is not finished yet at the time of this **DISPLAY** command.

2.6 Planning to protect critical data and applications

All primary volumes are treated equally by SDM when replicating data from these primary volumes onto their corresponding auxiliary volumes. Usually, the most crucial resource with the biggest potential to become insufficient is the line bandwidth between primary and auxiliary sites. In that case, it might be helpful to understand which volumes are more important than other volumes to be replicated, as mentioned in 2.3.1, “Determining what data should be copied” on page 28.

Another consideration is the write I/O skew between volumes, and the fact that some volumes contain less business critical data than other volumes. When there is a potential period where the amount of write data to arrive at the primary storage system is temporarily bigger than what the bandwidth to the auxiliary site can replicate, consider write pacing to protect critical data and the SDM session.

Figure 2-3 illustrates write pacing.

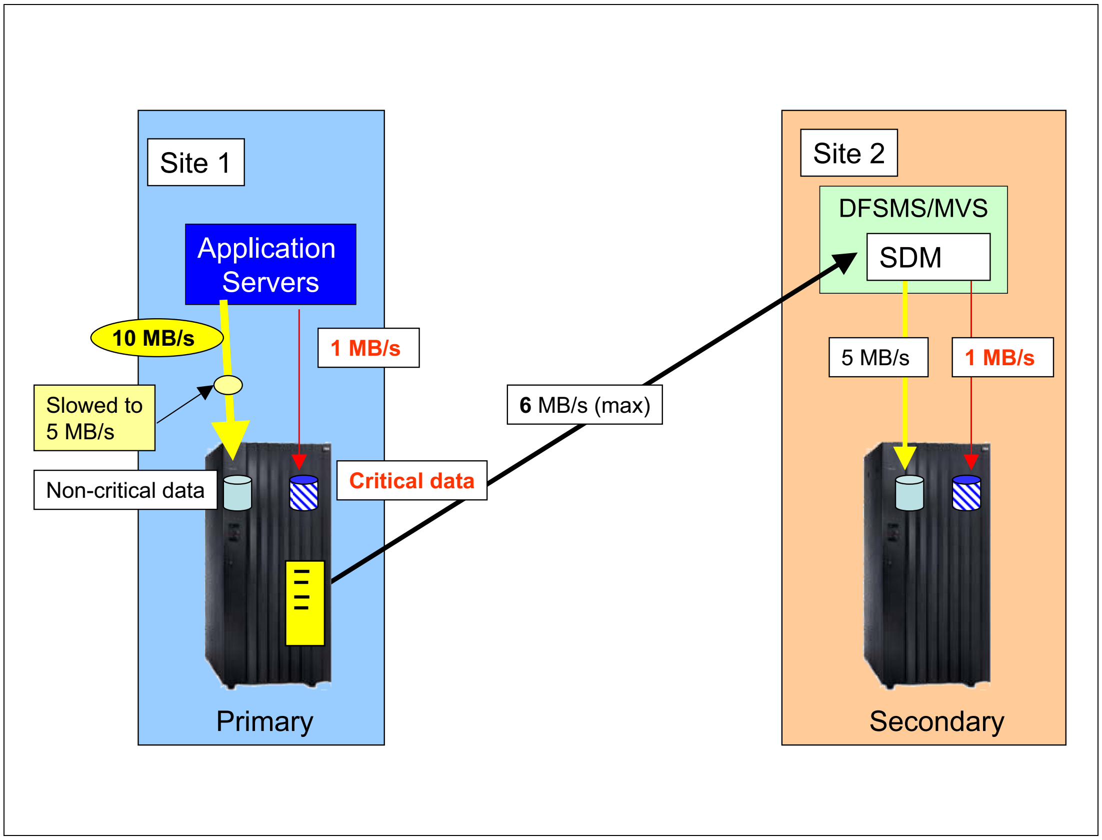


Figure 2-3 Workload pacing effect

Figure 2-3 shows how to protect critical activities and workload, in addition to not endangering the SDM session at the expense of non-critical write workloads. Consider identifying critical data and volumes that might not be subject to workload pacing, and subject all other remaining volumes to workload pacing. Again, this has to be considered to avoid SDM suspend conditions due to cache overruns, as mentioned in 2.3, “Initial z/OS Global Mirror planning considerations” on page 26.

2.7 Planning for configuration and workload changes

With its reader tasks, SDM is geared toward an LCU scope, or parts of an LCU through the storage control session concept. Therefore, the reader task has to address all write activities to all volumes within the LCU in question, or to a subset of volumes within an LCU when more than one storage control session is created.

This situation can have some consequences for the SDM reader task, when the workload and number of primary volumes increases within an LCU. For example, a client plans to increase the number of IBM DB2® logging volumes by 16 (using 3390-27 volumes), because of a significant workload growth within their DB2 subsystems.

The storage administrator has 16 spare volumes available, but all 16 volumes are in the same LCU. As soon as these 16 DB2 logging volumes are used by these busy DB2 subsystems, the SDM reader task is probably by far over-committed, and cache over-runs caused by this larger LCU can result in suspend conditions.

With a different pooling philosophy, and by pooling all volumes (especially horizontally) across many or all of the LCUs in the configuration, the workload increase can be spread across more than just one SDM reader task. If the spare pool with its 16 volumes is spread out across eight LCUs with two spares each, that avoids the skew increasing as badly, because now the increased SDM workload is spread across eight reader tasks instead of one.

This consideration applies to all pools or storage groups that are subject to be replicated by a zGM configuration. If you have not taken prior actions to address the additional workload through SDM, as soon as the workload increases in its relevant LCU there is always the potential for session suspends, and exceeding the RPO value between Site 1 and Site 2.

2.8 Connectivity

Connectivity planning is extremely important for every DR implementation. For a zGM implementation, the connectivity between the primary storage systems and the SDMs is also important. This scenario assumes that the SDM is running at the auxiliary (or DR) site, which is a best practice. Although synchronous DS8000 Metro Mirror uses Fibre Channel protocol for mirroring, all connections used in zGM are FICON connections. Normally, when using an asynchronous replication method, long distances must be spanned.

Bandwidth

The most important thing to consider is planning for the required bandwidth. For your connectivity planning, you need a solid base. You need to know how much traffic will be going across the network. You are only interested in the *write workload*.

You need at least some data about your write workload, normally at an interval of about 15 minutes, for at least one week long. If you have times with much higher write activity, for example at the end of a month, you might want to measure and analyze such a week. You need data from all of the systems with volumes that you want to mirror.

You can use any tool that can provide this kind of data. For example, the RMF *Channel Path Activity* report provides the write throughput for any FICON channel. You have to sum the numbers to get the total write throughput.

If you have Tivoli Productivity Center, you can use its performance monitor to get the write throughput and the write response time. When you have implemented zGM, you might want to compare the write response times.

When you have the total write throughput of your environment for a week, you can draw a chart and examine the average and the peaks. For synchronous mirroring, you have to make your bandwidth planning to cope with the peaks. With asynchronous mirroring, as with zGM, you can use a lower bandwidth, but *this will increase your RPO time*.

In your write throughput chart, you can draw a line at the throughput bandwidth that you are planning for. If it goes through some of the peaks, your zGM environment can only copy that much data up to your bandwidth line, and the amount of data that is above the line must be copied later. This will increase your RPO time. If RPO time is too large, you can use write pacing, which slows down your write activity. IBM has tools to plan for zGM bandwidth. Contact IBM to help you plan your bandwidth requirements.

FICON considerations

To bridge longer distances (more than 10 kilometers, or km), connectivity devices are needed that support the required distance. Depending on the distance, this can be dense wavelength division multiplexer (DWDM), FICON switches, FICON channel extenders, networking connectivity, and Internet Protocol (IP) routers. Some devices offer additional functions, such as compression or encryption.

Some devices have been tested to work well in a zGM environment. A list of tested devices can be found at the following website:

http://www.ibm.com/support/docview.wss?uid=ssg1S7003277

The vendor will determine the maximum distance supported. The vendor should be contacted for its distance capability, line quality requirements, and storage area network (SAN) and wide area network (WAN) attachment capabilities. The vendors should also be consulted regarding hardware and software prerequisites when using their products in a DS8000 zGM configuration. IBM is not responsible for third-party products.

Two options exist to improve performance of IBM zGM over Fibre Channel over IP (FCIP) at distances greater than those provided by direct fiber connections:

- FICON emulation (also known as FICON acceleration)
- Extended Distance FICON (also known as enhanced information unit (IU) pacing).

All of the tested devices can support Extended Distance FICON when using the prerequisite server (IBM System z10® or later) and IBM DS8000 storage systems running R3.1 licensed code or later.

FCIP routers

When using FCIP routers, follow these guidelines:

- For each primary controller LCU, there should only be one channel-path identifier (CHPID) from the SDM, bridging the extended distance to the primary LCU for each FCIP tunnel in the configuration.
- Each tunnel in a multiple-FCIP tunnel configuration should have at least one extended CHPID to each primary LCU in the configuration.
- The FCIP FICON emulation (XRC)-enabled tunnels should be defined with a 1-second keep-alive timeout.
- If you use HyperPAV for the SDM, include the PAV addresses, but if you do not use it for SDM, the primary controller CHPID definitions should only include the required primary base addresses (no aliases).
- WAN extension equipment creates a FICON Device Control Block (FDCB) for each CHPID port to each device port, for each LPAR to each LCU, for all defined devices on that LCU. This uses memory in the extension equipment, and care should be taken to understand the number of FDCB images that would need to be supported. PAV alias addresses also need FDCBs.

  For example, the Brocade 7800 can support about 100,000 FDCBs (per 7800 pair), but the FX8-24 FCIP complex can support about 160,000 FDCBs. There are two FCIP complexes on the FX8-24. Therefore, in a 10 gigabits per second (Gbps) configuration, the FX8-24 can support two FICON-emulating FCIP tunnels, and provide support for up to 160,000 FDCB images on each FCIP complex.

FICON switches

With FICON switches (or directors), you can order small form-factor pluggables (SFPs) that support more than 10 km, but usually the distances used for zGM configurations go beyond the capabilities that are supported by FICON switches.

2.9 Concurrent copy sessions

Concurrent copy (CC) also uses a data mover when creating a point-in-time copy. This data mover runs in an address space, `ANTMAIN`, which is automatically started at IPL. In this chapter, SDM refers to the SDM in zGM.

The concurrent copy data mover communicates with a concurrent copy storage control session in the DS8000 LCU. There are two types of storage control sessions in the DS8000 LCU:

- The zGM storage control sessions
- The concurrent copy storage control sessions

In this chapter, storage control sessions are *zGM storage control sessions*.

The use of concurrent copy can affect zGM, because an aggregate of cache sidefiles for both is taken into account when determining if limits are exceeded. Each logical storage subsystem (also known as LCU) in a DS8000 storage system can support up to 64 sessions. This session limit is a *combination of zGM and concurrent copy sessions*.

Therefore, if you use concurrent copy, make sure that your SDM configuration does not use the maximum number of possible sessions.

2.10 Mapping

When setting up a zGM environment, volume pairs are added to zGM sessions. The volume pairs, and the primary and auxiliary volumes, are added by volume serial number (VOLSER) pairs. It is a best practice to follow some rules.

Naming conventions

Establish some naming conventions that help you identify the volume pairs by VOLSER. For example, your primary (production) volumes could all start with a “P” as the first letter of the VOLSER, and your auxiliary (secondary) devices could start with an “S” while using the same five letter terms for the rest of each VOLSER.

Device number

When you select a volume to be added to a zGM session, and you have to select a target volume, you should have a look at the device address also. It is a best practice to use the same storage system internal address (or address within an LCU) for the primary and auxiliary volume.

For example, your primary volume has `VOLSER: PDB217`, and it is device address `0017` in an LCU with `LCUADD 00` at address `17`. Your auxiliary device on your auxiliary storage subsystem should also be in an LCU with `LCUADD 00` at address `17`, and it should have, for example (according to our naming convention), `VOLSER: SDB217`.

To obtain the DS8000 internal address for a z/OS device address *nnnn*, you can use the **CQUERY DEV(*nnnn*)** command. The command will show you the DS8000 logical subsystem (LSS), which corresponds to LCUADD, and the channel connection address (CCA).

Logical control units

We already mentioned that zGM works on an LCU basis, and SDM readers are attached to LCUs. You need to balance the workload across the LCU. The more LCUs you have, the more readers you have, but also the more logical paths to the storage systems you have, and you might reach the logical path limit. These are all things that must be considered, and you have to find the optimum configuration for your environment.

When adding volumes to your zGM session from one LCU, select the target volumes on the auxiliary system from the same LCU.

Different technology

Usually, the primary application storage systems are carefully planned regarding capacity, cache size, and disk performance. Sometimes, not so much attention is paid to the auxiliary storage systems, and slower, higher-capacity drives are used with smaller cache sizes. However, it is important that the auxiliary storage systems can sustain the same write workload as the primary storage systems.

Although the cache of the auxiliary storage systems is not so important, write cache (non-volatile storage, or NVS) *is* important. And because write cache scales with cache size, your auxiliary storage system should have the same cache size as the primary storage system. In case of a disaster, your auxiliary storage system should be capable of running the same workload as the primary storage system.

After recovering from a disaster, you also want to copy your data back to the primary site. In this case, your auxiliary storage system would also need the zGM license.

So, in general, it is a best practice to have the auxiliary storage system as equally equipped as the primary storage system.

FlashCopy volumes at the auxiliary site

A DR solution is of no value if you do not practice the DR. From time to time, you must plan for a DR test. This can easily be achieved if you can use FlashCopy at the auxiliary storage systems to create a test system.

You need twice the storage capacity at the auxiliary site, and a FlashCopy license. If you decided to use FlashCopy, you could use higher-capacity drives at the auxiliary storage system:

- Use large extent pools in the DS8000.
- Use extent pool striping.
- Allocate the FlashCopy target volumes within the same extent pool as your production volumes.

In this scenario, each physical disk drive at the auxiliary storage system will hold about the same capacity of production data as the primary physical drives. In addition to that, it will hold the FlashCopy target capacity. The I/O density per physical drive is the same for the primary drives and the twice-the-capacity auxiliary disk drives, because during normal operation, no I/O goes to the FlashCopy target volumes.

For example, if you have 450 GB disk drives at the primary site, you could have 900 GB disk drives at the auxiliary site for the mirrored data and the FlashCopy targets.

When you plan a DR test, you can suspend the zGM session, take the FlashCopies, and restart the session. Now you can run your tests by using the FlashCopy target volumes. Many clients also plan to use the FlashCopy target volumes after a disaster. By doing this, you still have another copy in case something went wrong when you started your host systems at the recovery site.

It is also a best practice to first take a FlashCopy of the zGM auxiliary volumes whenever you have to resync a zGM session. This is because, during the resync process, data consistency cannot be guaranteed for the auxiliary volumes.

2.11 The data set for zGM

This section provides information about planning for the various zGM data sets.

2.11.1 Storage management subsystem

If the journal data sets are striped (best practice), they must be on storage management subsystem (SMS)-managed volumes. The state data set is a partitioned data set extended (PDSE). You can find more information about journal, control, and state data set allocation in *z/OS DFSMS Advanced Copy Services*, SC35-0428.

2.11.2 Infrastructure data sets placement considerations for zGM

The zGM data sets (the journal, control, and state data sets) are critical to the performance of the zGM environment. The journals require the most careful consideration. Consider how to allocate them for best performance. You have the following options:

- Dedicated approach. Dedicate selected resources to the journal volumes.

  Dedicate one or more ranks to the journal data sets:

  - With the virtualization of the logical storage on the DS8000, this does require exceptionally careful management of the placement of the logical volumes.
  - If the journal data set rank or ranks share higher-level resources with only some of the auxiliary target volumes, a performance imbalance might exist for the auxiliary updates.
  - You might even consider solid-state drives (SSDs) for the journal volumes. SSDs are extremely fast for random reads, and not much faster for sequential writes, but SSDs used in the DS8870 can nevertheless provide about 50% more write throughput than normal disk drives.

- Shared approach. Share resources between the journal volumes, the auxiliary target volumes, and, if you have FlashCopy, also with the FlashCopy volumes:

  - Use large extent pools, and allocate the volumes with extent pool striping.
  - Alternate journal volumes between even-numbered and odd-numbered LCUs, to use both internal servers of a DS8000.
  - Ensure that the journal data sets, auxiliary target volumes, and tertiary FlashCopy volumes are equally placed, to share all the common resources.

  This is the most common approach currently used by IBM zGM clients.

Experience has shown that placing the journal data sets over many ranks, and sharing those ranks with auxiliary targets, has worked well. Placing the control and state data sets with the auxiliary targets has also worked well. Monitor the performance of all volumes to ensure that the health of the environment remains high.

2.12 Dynamic workload pacing

One of the advantages of zGM is that you can balance the needs of DR with the needs of application performance. With zGM, you can use a high-performance workload balancing algorithm called *write pacing*. In situations where the SDM offload rate falls behind the primary system's write update rate, record sets start to accumulate in cache.

This accumulation is dynamically detected by the primary DS8000 microcode, and the microcode responds by slowly but progressively reducing the application updates, to find the correct pacing rate to balance the application updates to the SDM offload rate. In most cases, this pacing is accomplished without pausing the application updates.

Write pacing is a good way to optimize RPO time, and to reach RPO service level agreements.

In the early versions of zGM, when it was still called XRC, a dynamic workload balancing algorithm, called *device blocking*, was used. The device blocking mechanism is used to balance the write activity from primary systems with the SDM's capability to offload cache during write peaks, or during a temporary lack of resources to SDM, and with minimal influence on the primary systems.

Device blocking is no longer a recommended mechanism or process, as stated in 1.2.5, "Write pacing and device blocking" on page 13. It can also unnecessarily affect reads.

If you have applications that need the highest level of performance, you can exclude them from write pacing (for example, by specifying the **`DONOTBLOCK`** parameter for the **`XADDPAIR`** command). But be careful, do not use this option for too many volumes, unless you do not care about RPO and you want RPO to grow.

For more information about write pacing and device blocking, see *z/OS DFSMS Advanced Copy Services*, SC35-0428, and "Write Pacing" on page 60.

3

Scenarios for installation deployment and upgrade

This chapter provides information about important aspects to consider when installing z/OS Global Mirror (zGM) for the first time, or when upgrading some of the components.

Although some information is briefly repeated here, we assume that you have completed your planning for zGM as described in Chapter 2, “Considerations about z/OS Global Mirror planning” on page 21.

3.1 Preparing for initial deployment

Previously know as Extended Remote Copy (XRC), zGM has been available in z/OS for many years. It is a part of DFSMS data facility product (DFSMSdfp), and no additional z/OS software is required for its deployment.

The following list includes important actions to take, or settings to verify, before initial deployment:

- As usual, when you want to use a new function for the first time, it is a best practice to check for any recent authorized program analysis reports (APARs) and apply them, if necessary.

  Check the `IBM.Function.ExtendedRemoteCopy` Preventive Service Planning (PSP) bucket. PSP buckets can be searched at the following Uniform Resource Locator (URL):

  `http://www.software.ibm.com/webapp/set2/psearch/search?domain=psp`

  You might have to apply program temporary fixes (PTFs) to your application logical partitions (LPARs) and to your System Data Mover (SDM) LPARs.

- For the storage system, it is also a best practice to have a current Licensed Code level, not older than one year. Make sure that the zGM license is applied to at least your primary DS8000, or to another supported vendor’s storage systems. To verify the applied licenses for the DS8000, you can use the Data Studio GUI, or you can use the **`lskey`** DS command-line interface (CLI) command.

  Also, confirm that the parallel access volume (PAV) and HyperPAV license is active. Your DS8000 should also have the z High Performance Fibre Channel connections (FICON) feature (zHPF) license. For details, see Example 2-1 on page 23.

- Make sure that your primary host systems use a common clock. See 2.2, “Requirements for z/OS Global Mirror” on page 22.

- If you have not yet configured the auxiliary storage systems, configure them, if possible, identically to the primary storage systems. One exception can be for volumes that you do not want to mirror, or the volumes for the zGM data sets used at the auxiliary site. See “Volumes that do not have to be mirrored (copy-once volumes)” on page 28.

- Prepare and set up your SDM LPARs. The hardware configuration definition (HCD) definitions must include the primary volumes, the auxiliary volumes, and the volumes needed by zGM:

  – Journal data sets
  – Control data sets
  – State data sets
  – Other data sets

- Enable the PAV and HyperPAV feature in your DS8000, and define the PAV addresses in HCD.

- Enable zHPF, if your storage system supports it, by specifying `ZHPF=YES` in the **`IECIOSxx`** parmlib member, and `SAM_USE_HPF(YES)` in the **`IGDSMSxx`** parmlib member. This is normally the default, starting with z/OS R13, but you should confirm these settings. The SDM can take advantage of zHPF when writing to the journal data sets.

- If you are using FlashCopy, define the FlashCopy target volumes. Define them in extra logical control units (LCUs), but in the same DS8000 extent pools as the auxiliary volumes.

  If you are going to use FlashCopy target volumes, you might also want to define them in the SDM system, but FlashCopies can also be done to volumes that are not defined in HCD by using the **`TARGET(...)`** and **`TGTUCB(NO)`** parameters of the **`TSO FCESTABL`** command.

  Prepare jobs to run the FlashCopies. For a FlashCopy source volume at the auxiliary site in an even LCU, choose a FlashCopy target volume also in an even LCU. In the same way, choose targets for volumes in odd LCUs. Try to keep a one-to-one relationship regarding the device addresses.

- Initialize the auxiliary volumes. Choose some naming convention that enables you to identify a volume pair, primary to auxiliary, from the volume serial number (VOLSER). See "Naming conventions" on page 41 for a suggested approach.

- Set up the long-distance FICON connection between the primary site storage systems and the SDM host system. This will involve the configuration of dense wavelength division multiplexer (DWDM), channel extenders, or Fibre Channel over Internet Protocol (IP), called FCIP, routers. Contact the appropriate vendor for best practices to use their devices. It is beyond the scope of this book to cover those devices.

- Test that your SDM system can access the primary volumes.

  During the planning part, you should have defined your SDM configuration:

  – The number of sessions
  – The session names
  – The determination of whether coupled sessions are required

  You can now allocate the zGM data sets as storage management subsystem (SMS)-managed data sets:

  – Journal data sets
  – State data sets
  – Control data sets
  – Master data sets
  – Cluster state data sets
  – Cluster data sets

  You might have to set up or modify your SMS environment first, to make sure that the data sets are allocated to the volumes that you want.

- For the zGM data sets, you have to follow a naming convention. The data set names always contain the session name (see 2.4.2, "SDM resources" on page 30 for more details). For more information about how to allocate the zGM data sets see *z/OS DFSMS Advanced Copy Services*, SC35-0428.

- The SDM address spaces `ANTAS`*`nnnn`* must be IBM RACF®-authorized to access the zGM data sets.

  Authorize zGM or XRC Time Sharing Option (TSO) commands by adding the command names to the **`AUTHCMD PARM`** parameter of the **`IKJTSOxx`** member of `SYS1.PARMLIB`.

- Review the zGM parmlib entries. Not all default entries are good. You should modify some of them, as detailed in Chapter 5, "Tuning z/OS Global Mirror" on page 65.

You can find additional aspects of the environment that must be validated in Appendix A, "Checklists for z/OS Global Mirror" on page 93.

Now you can issue the **XSTART** TSO command on the system that contains the SDM. Specify the session ID that is associated with the journal data set names. Select the appropriate level of recovery for your environment, normally XRC.

Vary all primary and auxiliary volumes online to the SDM. Issue the **XADDPAIR** TSO commands on the SDM to perform the copy of primary to auxiliary volumes. You should prepare jobs to issue these TSO commands, because the first pair specifies the utility volume for each LCU.

You might want to use the **XSET SYNCH(0)** command first, specifying that zero initial copies should be started, and then run the job with the **XADDPAIR** command. When you now set **SYNCH** to the maximum value, which is 45, the SDM starts the initial copy process, but it will optimize the sequence in which the volumes will be copied. If you do not follow this method, the volumes will be copied in the order that you issued the **XADDPAIR** commands.

Issue **XQUERY** commands to verify when zGM has reached the state that all volumes are copied, consistency groups are formed, and the zGM environment is operating as expected.

For more details about how to set up a zGM environment, see *z/OS DFSMS Advanced Copy Services*, SC35-0428.

3.2 Testing z/OS Global Mirror

After you have set up your first zGM environment, start by experimenting with and testing zGM functions.

Perform the following tests:

- Performance test. If possible, run some performance tests to check that your environment can copy data from the primary to the auxiliary volumes as fast as expected. Some workload generator can be useful. If your environment does not work as expected have a look at Chapter 5, “Tuning z/OS Global Mirror” on page 65.
- Disaster recovery (DR) test. Test the DR process. This test is a must.

Decide which LPARs will act as recovery systems. You have to think about the input/output (I/O) definition file (IODF). If the recovery system is identical to the primary system, you can use the same IODF.

You have the following choices to run a test:

- FlashCopies can be taken from the auxiliary volumes. You can just pause zGM, run the **XADVANCE** command to apply the latest updates from the journal and form a consistent set of volumes. The **XADVANCE** command does not change the VOLSER of the auxiliary volumes. Now you can make FlashCopies of the auxiliary volumes.

  Then, you can resume zGM mirroring. The **XRECOVER** with the **TERTIARY** parameter can now be used to attach the VOLSERs of all FlashCopy target volumes. Now you can use the FlashCopy target volumes to run your test.

  Many zGM installations use the FlashCopy target volumes (tertiary volumes) as the actual DR volumes. They keep the auxiliary volumes untouched in case something goes wrong when the recovery systems are started.

- FlashCopy volumes are not available. If FlashCopy target volumes are not available, you have to end zGM mirroring to be able to run a disaster recovery test. End the session (**XEND** command) and use the **RECOVER** command to apply the latest updates from the journal data set and to attach the VOLSERs of the auxiliary volumes to the VOLSERs of the corresponding primary volumes.

  Now you can run your tests. After testing has completed, you need to perform the following tasks:

  – Re-initialize the auxiliary volumes with the VOLSER that they had before.
  – Set the volumes online to SDM.
  – Start a session again, performing a full initial copy.

  This takes some time. Therefore, use an environment with FlashCopy volumes.

3.3 Monitoring z/OS Global Mirror

When zGM is running, you need to monitor it. An SDM can generate detailed statistics that reflect its operation, such as delay time, buffer availability, residuals, and write pacing. You have to watch for abnormal conditions, such as SUSPEND messages, and you must regularly check the consistency group time.

If the consistency time lags behind the current time, the SDM cannot cope with the workload, your recovery point objective (RPO) increases, and you might have to react. SDM statistics can be processed with the XRC Performance Monitor (XPM).

SDM is performing many I/Os, reading and writing. Therefore, all I/O-related IBM Resource Measurement Facility (RMF) reports are important, and should be used to monitor the SDM health.

Monitoring is also mentioned in 6.1, “Performance monitoring and analysis” on page 80. Geographically Dispersed Parallel Sysplex (GDPS)/XRC provides some tools that make it easier for you to monitor a zGM environment.

Monitoring also encompasses the following tasks:

- Monitoring the network. Most routers or switches have the feature (sometimes an optional, priced feature) to monitor the network traffic.
- Monitor the primary storage systems. With RMF, you can monitor the I/O response times, the write rate, and the use of the channels used for zGM mirroring.
- Monitor the SDM. With RMF, monitor the SDM central processing unit (CPU) use, I/O rates, throughput, and response times.
- Monitor the auxiliary storage systems. You can have RMF running in the SDM LPAR, and thus monitor the write I/O rate, response time, and throughput of the auxiliary volumes and journal volumes. Tivoli Productivity Center will provide you some more insight into your auxiliary storage systems.
- Use **XQUERY** commands to check the health of your zGM environment.
- Use the XPM.
- Use GDPS/XRC for more information about your zGM environment.

Have some short-term monitoring active to see what is going on at this point in time, but also have long-term monitoring to observe how things change over time. Tivoli Productivity Center is ideal for long-term monitoring of storage systems, because it keeps history data that can easily be viewed.

Performance observations

When looking at the SDM I/O statistics for the primary storage systems, you usually only see read I/Os, probably with large connect times (in the range of 50 - 80 milliseconds, or ms). This is due to the long I/O chains that are used by the SDM, and there will be a significant amount of I/O concurrency on the FICON ports, resulting in FICON elongation.

Pending time can also be high (around 50 ms), due to the long distance between the SDM and the primary storage systems. Pending time is primarily command response (CMR) time, and is dependent on the distance (about 1 ms for each 100 kilometers, or km). However, it is even more dependent on the number of devices in the network path between the SDM and the primary storage systems. Disconnect time should be small.

The I/O to the auxiliary systems is a pure write workload. There should hardly be any disconnect time, assuming that the auxiliary storage systems, arrays, internal paths, and non-volatile storage (NVS) can cope with the write workload.

3.4 Adding storage capacity

When you have a zGM environment already in place and being used, and you need to add more capacity, this new capacity must be mirrored with zGM. You have to go through the planning steps again, and ask the following questions:

- Is the bandwidth between the SDM and the primary storage system still sufficient?
- Does the SDM LPAR have enough resources to handle the additional load?
- Is it necessary to rebalance the workloads?
- Do the switches and routers have enough ports?

The jobs to manage zGM need to be modified to include the additional volume pairs as well.

An important consideration is when to add the mirroring pairs to an SDM session. When adding volumes to a session and during the first copy process, the SDM will not use the utility devices to perform the I/Os, but instead use the device addresses of the volumes. The newly added volumes do not compromise the consistency group time directly. The added volume pairs are not really part of the consistency group until the initial copy process has finished.

However, as the initial copy process competes for the networking resources with the normal mirroring I/O, the initial copy *can have a significant effect* on the consistency time due to overloaded networks or other resources. Therefore, it is a best practice to add new zGM pairs only during times of low I/O activity.

Use the **QUICKCOPY** parameter for newly initialized empty volumes.

3.5 Upgrading the server

Whenever you have to perform some maintenance on the SDM LPAR, or upgrade the SDM server, you can suspend the SDM sessions. This is accomplished using the **XSUSPEND TIMEOUT** command. This command is issued whenever you want to stop the SDM for a planned activity, such as a maintenance update, or moving the SDM to a different site or a different LPAR.

The **XSUSPEND TIMEOUT** command will end the ANTAS*nnn* address space, and inform the involved LCUs that the zGM session is suspended. The DS8000 will then use the hardware bitmap to record changed tracks, and will free the write updates from cache.

When the zGM session is restarted with the **XSTART** command, and volumes are added back to the session with the **XADDPAIR** command, the hardware bitmap maintained by the DS8000 while the session was suspended will be used to resynchronize the volumes, so full volume resynchronization is avoided.

You can use the **XADDPAIR *session_id* SUSPENDED** command to add all of the volume pairs back into the session and start resynchronization. Again, do not resynchronize all volumes during peak production workload. You can use the **XSET** command to first set the **SYNCH(0)** parameter to zero before using the **XADDPAIR** command, and then change **SYNCH(45)** to the maximum, to enable SDM to optimize the sequence of resynchronization.

3.6 Upgrading the storage system

Replacing a storage system with a new one is always a challenging task, because the data somehow needs to be transferred from the old system to the new one.

Some kind of mirroring usually provides a migration option with only a short disruption, or even no interruption, of I/O services.

There are several options that you can use to replace a storage system:

- The zGM system is essentially a mirroring technique, and it can also be used to port data from one storage system to another. However, because one volume can only be in one zGM relationship, you would need to end zGM to the remote location, then set up a new zGM environment to establish mirroring between the old storage system and the new one.

  Assume that a storage system at the production site must be replaced.

  Normally, DR capability is not required during a data migration. Therefore, the session can be started with the **XSTART *session_id* ERRORLEVEL(VOLUME) SESSIONTYPE(MIGRATE)** command. This session type does not require journal or control data sets. You just need a state data set. When all of the data is copied, the production systems need to be stopped, and the primary storage systems should be disconnected or switched off.

  The **XRECOVER** command attaches the target volumes of the new storage system to the original VOLSER. After re-cabling the new storage system to the production hosts, the systems can be restarted.

  The disadvantage of this method is that a full resynchronization is required when zGM is set up between the new storage system and the auxiliary site storage systems.

- The zGM system can be combined with DS8000 Metro Mirror. This is called a Metro zGM configuration, or MzGM. DS8000 Metro Mirror can be used to port data from one DS8000 storage system to another DS8000. However, this would require a DS8000 Metro Mirror license in both storage systems. In addition, after the migration, a full zGM resynchronization is also needed.

  However, if GDPS/MzGM is used, an *incremental* resynchronization is possible, *if* the DS8000 systems also have the zGM Resynchronization license.

- The use of IBM Softek Transparent Data Migration Facility (IBM TDMF®) z/OS is another data migration option. When both a primary and an auxiliary storage system are replaced, which is often the case, you can add the new volume pairs with empty primary volumes to the session. You can then use TDMF to port the data from the old primary storage system to the new one.

  At the same time, data is mirrored with zGM to the remote site. After TDMF has switched from the old volumes to the new ones, you can remove the old pairs from the session.

  This data migration with TDMF is nondisruptive for the production workload.

4

Keeping your z/OS Global Mirror environment working correctly

For z/OS Global Mirror (zGM), working correctly means two things: *Maximizing data currency* and *avoiding unacceptable production application effect*.

A stable zGM environment is strongly related to a balanced use of resources:

- System Data Movers (SDMs)
- Disk storage systems
- Logical subsystems
- Readers
- Devices

If there is insufficient balance, the SDM data offload process might start to fall behind on the write activity of the production systems. This might in turn have a minor (or in some cases, severe) effect on those production systems.

To be able to assess the health of a zGM environment, you must have established some ongoing monitoring. Information about monitoring a zGM environment is provided in Chapter 6, “Scenarios for ongoing operations” on page 79.

4.1 Data currency

The term *data currency* describes the time difference since the last data was written at the primary site, and when the same data was written to the auxiliary site. It determines the amount of data loss at the remote site after a disaster. A business might have specific requirements (*average*, *maximum*, or *percentile*) for this time difference. These are known as the recovery point objectives, or RPO.

Only synchronous copy solutions such as DS8000 Metro Mirror can achieve RPO = 0. All asynchronous copy solutions have some data currency time difference, which can range from a few seconds to over an hour or more, depending on circumstances.

For zGM, *data currency time* is also known as *data exposure time*, which provides an approximation of the time difference between data that is written to the primary volumes and data that is secured on the journal data set with Extended Remote Copy (XRC) mode. To understand the data currency of a zGM environment, monitor the data exposure time. The data currency time of zGM is typically expected to be less than two minutes.

Note: The data exposure time is different than delay time. Delay time indicates the time difference between the most recent record that is written to the primary control units and the consistency group time that has been written to the auxiliary volumes. The data exposure time is less than the delay time.

Note: The data currency time is meaningless for a zGM migration mode session. This section only provides information about an XRC mode session.

In a correctly functioning zGM environment, data currency is maintained in a stable pattern, with tolerance of a short-term increase during peak application workload periods. A steady increase of data currency time indicates problems in the zGM environment.

Record set creation rate versus zGM offload rate

When zGM is used, primary disk write activity is stored in the primary storage system cache as *record sets* until the SDM can offload them. During periods when the record set creation rate exceeds the SDM offloading rate, the data currency time difference increases.

The record set creation rate is related to the intensity of the application write workload, and is not expected to be controlled. Therefore, the software and hardware components that affect the SDM offloading rate should be investigated to identify the bottleneck for zGM data currency:

- Bandwidth between the primary storage system and SDM.

  The zGM SDM uses the connection between the primary storage system and SDM to offload the record sets. Shortage of bandwidth between primary storage systems and SDM affects the SDM *read record sets* rate.

  The bandwidth capacity of each component on the connection route between the primary storage system and the SDM should be determined considering the following factors:

  – The primary storage control cache, non-volatile storage (NVS), and processor capacity

  – The capacity of the primary disk host adapters, and of the ports that are connected to the Fibre Channel connections (FICON) accessed by the SDM

  – FICON bandwidth accessed by the SDM to offload the record sets from primary storage systems

  - Channel extender and FICON switch capacity
  - Internet Protocol (IP) network bandwidth between the channel extenders and the FICON switches
- SDM session buffers.

  There are several types of buffers in the SDM address space. The major donator is the buffers that are used in the mainline data movement to read the record sets from the primary storage system. Record sets are temporarily stored and grouped into a consistency group in the data movement buffers by the SDM, and then deleted from the SDM session buffers after consistency groups are successfully written to the auxiliary disks.

  If the SDM session data movement buffer is full, SDM postpones the process of record set offloading from the primary storage system until the buffer is released. The number of data movement buffers that are used by each storage control session can be customized with the **`BuffersPerStorageControl`** parameter. The maximum number of buffers for one zGM session can be customized with the **`TotalBuffers`** parameter. A fully-configured zGM session can have a maximum of 25,000 buffers for data movement.

Some factors must be analyzed and considered to prevent the data movement buffers from filling:

- SDM LPAR processor capacity.

  SDM uses SDM logical partition (LPAR) processor millions of instructions per second (MIPS) to create consistency groups in data movement buffers, and to process input/output (I/O) operations for read record sets, write journal, and write auxiliary volumes. Shortage of SDM LPAR processor MIPS delays the speed of consistency group creation and consistency group writing to journal and auxiliary volumes.

  The basic guidelines for planning of the processor capacity is 5 - 6 MIPS per 100 primary write updates, with system usage of approximately 10 - 20 MIPS per SDM, and avoiding single processor configuration for SDM LPAR.

  The SDM can use System z Integrated Information Processor (zIIP) processors. However, I/O interrupts need a normal processor. It is a best practice to maintain a 1:1 relationship between zIIP and normal processors for the SDM.
- SDM LPAR real storage.

  The zGM system uses virtual storage for data movement buffers. When these buffers are used for I/O operations, they must be backed up by real storage. If real storage is not large enough to back up all the active buffers, storage paging activities are introduced to temporarily release the real storage shortage. Paging should be avoided. The SDM buffer release rate is influenced, because additional time is required to wait for paging activities. If your SDM LPAR uses paging, increase its memory.
- Consistency group journaling.

  After consistency groups are created in SDM buffers, they are journaled to zGM journal data sets first, and then written to auxiliary volumes. With slow journaling, consistency groups must be accumulated in the data movement buffers, and those buffers cannot be released for reuse.

  Monitor the I/O activity of the journal volumes. If delays occur in writing to the journal data sets, you might want to move the journal data sets to faster disks. You might even consider solid-state drives (SSDs) for the journals.

  The Easy Tier functionality should be enabled for the extent pools that contain the journal volumes, and the volumes should be defined with extent pool striping enabled. Note that Easy Tier is enabled by default for heterogeneous extent pools, but Easy Tier can also optimize homogeneous extent pools with just one disk type.

However, for homogeneous extent pools, Easy Tier must be explicitly enabled in the DS8000 (select **Manage ALL pools**).

- Writing to auxiliary volumes.

  Like consistency group journaling, data movement buffers that are occupied by consistency groups are not released until those consistency groups are committed on auxiliary volumes. Performance effects for writing to auxiliary volumes that are triggered by resource contention or restriction (processor MIPs, FICON channel bandwidth, auxiliary storage system capacity, parallel access volume (PAV) devices, and so on) cause an out-of-data movement buffers situation.

  Therefore, you should not use slower-spinning disk drives or higher-capacity disk drives at the auxiliary site, as compared to the primary site. Use HyperPAV and extent pool striping.

- Reader pacing.

  In an SDM session with multiple storage control sessions (readers), due to the limitation of the data movement buffer size, the leading readers must eventually wait for those lagging readers that are not keeping up with application updates. This procedure is called *reader pacing*.

  When reader pacing occurs for an extended period, the record-set offload rates for the lagging readers are continuously decreased. The data currency time of this session increases even when the data movement buffers are not in full condition. To avoid such conditions, you should have spread your workload (I/O load) as evenly as possible across the logical control units (LCUs).

- Session pacing in the coupled zGM environment.

  In the coupled zGM (CXRC) environment, session pacing means that if one zGM session encounters a delay situation, other sessions in this CXRC configuration must wait for that session. This wait helps ensure the creation of data consistency across the entire master session. The data currency time of the master session increases along with its worst zGM session, even though all of the other zGM sessions are still in correctly functioning status.

Unplanned zGM suspension

Unplanned zGM suspensions can happen, perhaps during a disaster, but also when the write I/O workload is too high. When the workload is too high, the SDMs cannot drain the record sets from the cache of the primary subsystems fast enough, so the cache fills up and the session must be suspended. Hardware bitmaps keep track of what data (tracks) have changed. You must have some monitoring to be alerted when a session is suspended. Geographically Dispersed Parallel Sysplex (GDPS)/XRC can make this much easier for you.

After a zGM planned or unplanned suspension, the data currency time difference increases along with wall clock time until the mirror is resynchronized. For planned zGM suspension, installation should understand and expect the increase of data currency time. Therefore, only the triggers for an unplanned zGM suspension are identified and mentioned here.

Primary storage system cache

The primary storage system cache is used to store record sets before they are offloaded by SDM. The primary system cache also temporarily stores the data of the production system read and write I/Os. To prevent it from affecting the application I/O performance, the primary storage system cache usage by zGM is limited by the DS8000 microcode.

There are two limitations. The first limitation is that a maximum of 60% of the cache size can be allocated by the zGM sidefile. The other limitation is that a maximum of 128,000 record sets can be in the primary storage system cache for each LCU that is managed by zGM. With a buffer size of 60,000 for a record set, this is about 7.5 GB of cache.

If either limitation is reached, a control unit has to address this condition (see “The zGM initiated suspend instead of long busy status”). The XRC Performance Monitor can show how much of the available cache for zGM is used.

The zGM initiated suspend instead of long busy status

Licensed Machine Code (LMC) level 5.2.0 for the DS8000 system provided an enhancement to minimize production effect during peak workload periods. For high workload peaks, which might temporarily overload the zGM configuration's bandwidth, the DS8000 system can initiate zGM `SUSPEND`, therefore preserving primary site application performance, an improvement over the previous `long busy` status.

This capability provides improved capacity and application performance during heavy bursts of write activities. It can also enable zGM to be configured to tolerate longer periods of communication loss with the primary storage subsystems, enabling zGM to stay active despite transient channel path recovery events.

This capability can provide fail-safe protection against application system effect that is related to unexpected data mover system events. How zGM should behave can be specified in the `ANTXIN00` parmlib member. In any case, if you plan for a zGM environment, implement the next-larger cache size compared to the cache size you would use in an environment without zGM.

Component outage

The zGM system automatically suspends zGM mirroring for a pair of volumes, or a whole session, if it detects any unplanned hardware or software error for zGM components:

- Primary storage system outage
- Primary storage control unit error
- Network failure
- Data transmission error
- SDM buffer allocation failure

In any case, if a suspend happens, you have to detect this, analyze the situation and the cause of the suspend, and eventually start the resynchronization process. If there was a disaster, you need to take actions to prepare the auxiliary system for production.

4.2 How can zGM affect production performance

One of the key benefits of zGM is the low performance penalty to production applications. However, when using zGM, it is important to ensure that systems are balanced and correctly configured. All effects fall into two categories: *resource contention between zGM activities and production workload activities* and *excessive injected write pacing by zGM*.

4.2.1 Resource contention

Some resources are shared by zGM activities and production workload activities. Most of them belong to primary storage systems according to the zGM configuration requirement. Excessive competition for those shared resources detracts from the performance of production systems:

- Primary storage system HA cards and ports, if they are configured as shared by SDM and production systems
- FICON connections between primary storage systems and local FICON switches, if they are configured as shared by SDM and production systems

- Primary storage system cache and NVS
- Primary storage arrays, ranks, and processors
- Primary storage control PAV devices, if they are defined to SDM as utility devices under a zGM-enhanced multiple reader configuration
- Real and pageable storage, processor, and channel capacity of primary systems, if zGM SDMs run on the primary systems
- Primary site FICON switches and ports
- Primary volumes assigned as SDM utility volumes

4.2.2 Excessive injected write pacing

As an alternative method to device blocking, write pacing is designed to protect zGM mirroring during the production peak workload window.

When the record set creation rate exceeds the zGM offloading rate for an extended period, the residual count for a specific primary volume increases continuously, and occasionally reaches defined threshold. When the threshold is reached, write pacing is injected to postpone new production updates.

To determine the root cause of write pacing injection, investigate the effects that influence the zGM offloading rate. See “Record set creation rate versus zGM offload rate” on page 54 for a detailed description of those effects.

> **Note:** Although a zGM session with migration mode does not require data consistency, write pacing injection still occurs if you set the write pacing level to the volume pairs in a migration session.

4.3 Maximizing data currency to maintain peak performance

To maximize data currency, achieve RPO service levels, and avoid unacceptable production effects, avoid prolonged situations where write workload exceeds SDM offload capability. This can be accomplished using a combination of the following methods:

- Provide sufficient zGM resources, and tune them correctly to meet peak offload demand at the LCU level.
- Balance the workload evenly across LCUs to maximize the use of available zGM resources.
- Distribute workload across time to minimize instantaneous intensity.
- Avoid mirroring any workload that is not needed for recovery.
- Enable zGM write pacing selectively (or temporarily) to regulate the workload in specific scenarios.
- Enable zGM write pacing globally and permanently to act as a shock absorber for unexpected workload spikes.

Peak offload demand

In general, the more intense and contained the peak workload at the LCU level is, the more zGM resources (storage area network (SAN), readers, SDMs, and so on) must be provided to meet peak demand. For the latest generation of IBM primary storage systems, the DS8000, throughput is independent on LCUs, and can range to several gigabytes per second (GBps).

The SDM for zGM, however, works on an LCU level, and zGM write rates at the LCU level can exceed 600 megabytes per second (MBps) and 25,000 record sets per second for a concentrated workload. In any given installation, it might be impractical to provision system resources that support the maximum architecturally possible workload on any LCU at any moment, so balancing workload across the LCUs is an essential part of a correctly working zGM system.

Balance across LCUs

The zGM system operates most efficiently when workload is balanced across all LCUs in all disk systems. For example, if you have four DS8870 systems with 40 LCUs each, it is ideal to have 1/4 of the write I/O during a peak period occur on each of the four DS8870 systems, and 1/160 of the write I/Os to occur on each of the 160 LCUs.

Although this level of accuracy is impractical, the closer you come to this goal, the more efficiently the data movers work. This approach minimizes the resources and management time that are required for the environment.

This might also require you to define more LCUs than required from an addressing point of view.

One strategy for accomplishing this balancing is to assign the first four volumes in a storage management subsystem (SMS) storage group to each of the four DS8870 systems. The next four volumes are again distributed across the four systems, but to different LCUs. If you started with even-numbered LCUs, you continue with odd-numbered LCUs to use all of the DS8000 resources.

In the DS8870, use large extent pools with many arrays, and use storage pool striping when defining the volumes. Continue the process until the storage group has one volume on each LCU. This strategy does not eliminate hot spots, but tends to minimize them. If a client is upgrading to a new SMS, this should be taken into consideration when creating the volume mapping for the TDMF moves.

Time distribution

Peak workload intensity can be reduced by scheduling the workload across an interval of time, instead of all at the same time. An example is for database reorganizations. Simultaneously submitting a massive number of jobs to reorganize all databases at the same time is not good for system performance in general, and can have an especially detrimental effect on zGM.

By spreading out initiation of those same jobs over several minutes, the work can be accomplished with a relatively small increase in elapsed time, while still preserving a good RPO time. Also consider workload intensity when scheduling other zGM resource-consuming events, such as a FlashCopy to produce a tertiary copy of data. To minimize the risk to large RPO times, wait until a relatively quiet period of the day to perform this task.

Avoiding unnecessary workload

Certain volumes are needed only for their allocations. Current data is not required. You can set up a zGM session just for these volumes. We call this session a *CopyOnce* session, which you might duplex occasionally, and keep suspended at all other times. To preserve good RPO time, it is important that these volumes are resynchronized only during periods of light overall workload, and that they are resuspended immediately upon reaching `DUPLEX` mode.

In addition to CopyOnce, there might be other situations where it is not necessary to mirror workload for an otherwise continuously mirrored volume. Such is the case during certain forms of volume initialization or full volume data copy, because the partially initialized or copied volume is of no use in a recovery situation. In this case, the volume can be suspended during the copy operation, and returned to DUPLEX mode before it is placed into production use.

Write Pacing

The zGM disaster recovery (DR) solution has a function called write pacing, which attempts to balance the needs of DR with the needs of application performance. It does this by throttling application writes to the level that enables the remote mirror to keep up.

The amount of pacing injected is limited to a level that is specified by the user, at a volume granularity. When the mirror is not keeping up, write pacing injects a variable amount of delay to each write I/O, depending on the settings for the volume, and on how much the mirror is behind. The delay is injected as disconnect time.

Write pacing can be used as a system-wide shock absorber to handle transient, unexpected imbalances between production workload and SDM offload capability. It can also be used to temporarily regulate specific planned, discretionary workload that might otherwise exceed the zGM system capabilities. However, prolonged pacing at high levels with broad scope might have unacceptable effects on production. It must be used carefully, after sufficient analysis and planning.

When enabling write pacing, a maximum permissible delay value is defined for each affected volume. To implement broad-scope, permanent pacing, it is essential that workload be assigned to storage pools based on performance requirements.

Discretionary workload (which might require a high level of pacing) cannot be located on the same volume as a critical online application database (which might require an extremely low level of pacing, or none at all). Permanent write pacing can be used selectively and temporarily to regulate workload in specific scenarios, even if it is not implemented globally.

Write pacing terms

Table 4-1 provides more detailed information about write pacing.

Table 4-1 Terms used in write pacing

Term	Definition
Device blocking	An older form of zGM delay that injects 1000 milliseconds (ms) of delay on every I/O to a volume when the residual count exceeds a threshold (*not* recommended because it also affects reads)
Maximum write pacing level	The limit of how much delay can be injected by write pacing, sometimes written as *write pacing level* (omitting “maximum”)
Residual count	The number of record sets in the controller cache, waiting to be read by zGM
Volume residual count	The residual count for a particular volume
Write pacing threshold	The volume residual count at which the maximum amount of delay is injected (called the `Write Pacing Residual Count` in parmlib)
DONOTBLOCK	A setting that turns off both device blocking and write pacing, used to ensure that zGM does not add any more delay to applications than the amount intrinsically added by having zGM active in the first place.

Write pacing is controlled via several settings in parmlib, and by parameters on zGM commands. For the **`XADDPAIR`** command, for example, the **`DVCBLOCK(WP`*`n`*`)`** option can be specified. **`WP`*`n`*** sets the maximum write pacing level to be used for the volume. There are 15 write pacing levels. Each level is specified as the characters “WP” and a hex digit 1-F. For example **`WP1`**, **`WP5`**, and **`WPD`** for levels 1, 5, and 13.

Write Pacing calculation

The calculation for write pacing is a stepwise function. In a stepwise function, the value of the function `f(`*x*`)` remains the same for a range of input values *x*, and then jumps when certain values of *x* are reached. The input value *x* for this function is the volume residual count at the time the I/O is received by the controller. The output value is the amount of disconnect time the controller injects, if any.

The number of steps used in calculating write pacing is determined by the write pacing level for the volume, and the height of all of the steps together is determined from the (global) write pacing threshold. To determine the height of each step, the threshold is divided by the write pacing number.

For example, write pacing level 1 (**`WP1`**) has one step, and write pacing level 4 (**`WP4`**) has 4 steps. Each step is (threshold / # steps) wide. So with a threshold of 1000, **`WP1`** would have one step of 1000, and **`WP4`** would have four steps of 250. There is also a “ground level” below the first step that injects no pacing. The number of the step determines how much delay is injected, and that relationship is shown in Table 4-2 on page 62.

As the volume residual count rises, after it passes the step boundary (the height of the step), the actual injected pacing is increased to the next pacing step. In our **`WP4`** example with a threshold of 1000, there is no pacing from volume residual 0-249, and at volume residual 250 the pacing increases to the first step. At volume residual 500 it increases to step 2, and so forth until volume residual 1000 is reached, at which time the peak pacing of step 4 is injected.

Table 4-2 shows the relationship between write pacing steps and time.

Table 4-2 Write pacing time

Write pacing step	Maximum write pacing time (ms)
1	0.020
2	0.040
3	0.100
4	0.200
5	0.500
6	1.000
7	2.000
8	5.000
9	10.000
A	25.000
B	50.000
C	100.000
D	200.000
E	500.000
F	1000.000

The following values result in various pacing maximums:

- **WP1** - **WP7** result in pacing maximums of 0.02, 0.04, 0.1, 0.5, 1, and 2 ms per record set. These levels are useful for volumes with high rates of small blocksize writes, such as database logs, where minimal response time effect is essential.
- **WP8** - **WPC** result in pacing maximums of 5, 10, 25, 50, and 100 ms per record set. These levels are useful for volumes with high MBps write rates.
- **WPD** - **WPF** result in pacing maximums of 200, 500, and 1000 milliseconds per record set. These levels should be used only in exceptional situations where an exceptionally high degree of pacing is required.

Figure 4-1 on page 63 shows what pacing step is injected at various volume residual counts, and the timings added to the write I/O for selected settings of the **DVCBLOCK** parameter, using a threshold of 1000. Note that the line for the **WPA** time delay is not included. It would dominate the Y axis of the chart because the injected delay is exponential.

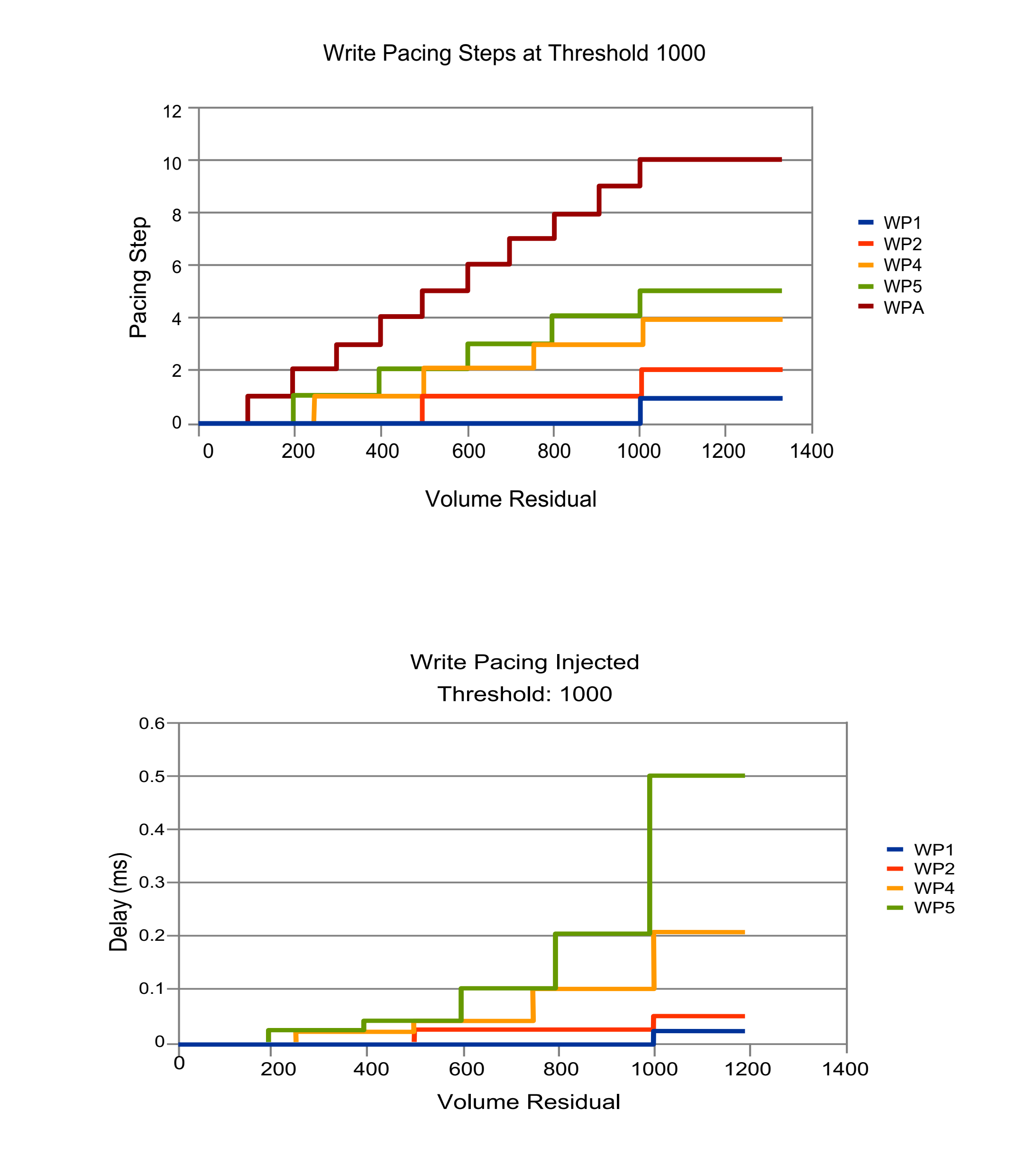


Figure 4-1 Write pacing steps and injected timings

Also see Chapter 6, “Scenarios for ongoing operations” on page 79 for more information about monitoring and operating a correctly working zGM environment.

5

Tuning z/OS Global Mirror

The z/OS Global Mirror (zGM) system works with cached storage controls to efficiently manage existing system resources for disaster recovery (DR). The System Data Mover (SDM) issues input/output (I/O) requests to drain record updates from the primary storage control cache that are designated to be copied by zGM.

Each logical zGM session can have one or more storage control sessions for each primary storage control. A logical zGM session manages all of the volumes that are associated with those storage control readers. To optimally and efficiently use all of the components and resources in a zGM configuration, a continuous tuning procedure is always required for each installation.

This chapter provides information about the following categories of zGM tuning concepts:

- Maximizing reader efficiency
- Maximizing write to auxiliary efficiency
- Maximizing write to journal data set efficiency
- Maximizing synchronization and resynchronization efficiency
- Reviewing parmlib parameters that do not have adequate default values

5.1 Maximizing reader efficiency

The zGM system offloads record sets from the primary storage control cache through storage control sessions (readers). An efficient reader configuration is essential to provide an offload rate to handle the peak workload and prevent negative effects on production.

Provide adequate resources on primary storage system

To maximize performance for the offloading process, ensure that the primary storage systems have adequate resources, including cache, non-volatile storage (NVS), and path connections, to handle the peak workload that is associated with remote copy sessions (and coexisting concurrent copy operations). Given adequate resources, the SDM functions smoothly even as the workload stress on it grows. Constrained resources become apparent because they create bottlenecks that impede the performance of the SDM.

You can help manage the peak write activity load for zGM-managed volumes by increasing the size of the storage control cache. Due to the large volume of data that is being moved, storage controls that support zGM might require significantly more cache memory than storage controls that do not support it. The peak load for the storage control dictates the amount of cache that is required.

Storage controls require more cache to prevent canceled zGM sessions as the peak load becomes heavier. Depending on the application write I/O rate, the primary storage subsystem might require additional cache. Sophisticated applications with high write content can require as much as 6 GB or more of subsystem cache. The requirements depend on the application, and on the capability of the storage subsystem. For example, in systems moving to an Extended Remote Copy (XRC) environment (zGM), cache size is suggested to be doubled.

Avoid resource contention

Many resources on the primary site, especially on the primary storage systems, are shared by the SDM and the production workload. Excessive resource contention hurts the performance of zGM offload processing from the primary storage systems, and the production workload performance.

Assign dedicated resources for zGM

Some of the shared resources can be separated:

- High availability (HA) cards and ports on the primary storage system
- Fibre Channel connections (FICON) bandwidth between primary storage systems and FICON switches
- The ports on the FICON switches

If possible, try to assign dedicated resources for zGM to prevent resource contention.

Provide enough capacity for shared resources

There are other resources that cannot be separated, such as the primary storage control cache, NVS, processors, and resources that must be shared. Ensure that the capacity of these resources can handle both the production workload activities and zGM activities during peak workload, to relieve the excessive resource competition situation.

Define dedicated volumes as utility volumes

To avoid conflicts with application programs that share a utility volume, use a dedicated reader volume. Sharing a utility volume can negatively affect the performance of both the SDM and the application program.

The zGM SDM software issues its I/O requests to the zGM utility volume that is associated with each primary storage control. This is part of the data transfer process.

You can assign one or more zGM utility volumes to each primary storage control by specifying the `SCSESSION` keyword on the **XADDPAIR** command. Each specified unique `SCSESSION` starts a separate reader. All volumes in a logical subsystem (LSS, also called a subsystem identifier, or SSID) with the same `SCSESSION` share the same reader. For more information about using zGM utility devices, see *z/OS DFSMS Advanced Copy Services*, SC35-0428.

Provide adequate host resources for the SDM

The SDM software requires adequate processor millions of instructions per second (MIPS), multiple processors, or IBM System z Integrated Information Processors (zIIPs), and storage in the storage system to accomplish its function in an efficient and timely fashion.

If the SDM does not have adequate host resources, it might not be able to drain the storage control cache rapidly enough under peak workload conditions. Based on the workload, if this continues for an extended time, the cache might become overcommitted, and ultimately this can affect the performance of the primary systems.

Ensure adequate processor capacity

Copy operations in zGM rely on host processing resources. Therefore, analyze the host resources that are necessary for your applications. Ensure that the SDM system has the processing capacity that is required to maintain optimal performance.

This includes defining processors with the appropriate MIPS capability (about 5 - 6 MIPS per 100 write updates) and multiprocessor capability (2 - 4 processors per SDM). The SDM supports a uniprocessor environment. Multiprocessors enable more parallelism, but if the MIPS are sufficient, the data mover runs efficiently on a single processor.

Assign storage in the storage system

When adequate storage (real storage) in the storage system is assigned for an SDM logical partition (LPAR), data movement and control buffers can be fixed in the storage system without requiring storage paging. That real storage allocation is controlled by the **PermanentFixedPages** parmlib setting.

> **Tip:** The **Permanent FixedPages** setting provides a method to keep the pages fixed for one SDM session. You can specify a value between 0 and 9999 megabytes (MB). A best practice value is 1500 MB.

XRC uses virtual storage to process client data, but ideally, we want all data to be in real storage, consequently the **PermamnentFixedPages** value of 1500 MB.

> **Tip:** The **BuffersPerStorageControl** setting specifies the number of buffers that could be assigned to one storage control session at the beginning, and it will grow to **TotalBuffers** as needed. A best practice value is 25,000.
>
> The **TotalBuffers** setting specifies the maximum number of buffers used for an XRC session. A best practice value is 25,000.
>
> The **IODataAreas** setting specifies the real storage allocation for XRC channel program and work areas that are associated with I/O operations. A best practice value is 2408.

Provide a stand-alone SDM system or LPAR for SDM

If possible, place the SDM on its own system with a large amount of storage in the storage system. This configuration enables the SDM to allocate many track-size buffers, and to optimize its I/O performance with minimal effect to the primary application I/O. A separate system also minimizes the effect to the system data mover, which prevents a potential bottleneck due to capacity problems during peak I/O update times.

Increase the SDM transfer performance

Increase the transfer performance by connecting the SDM system to the primary storage control with the highest bandwidth channels. If you use channel extenders or FICON switches, ensure that the compression feature is installed, because channel extenders typically double the amount of data per bandwidth.

> **Note:** The `MaxBytesTransferred` setting specifies the amount of data that could be transferred in a single channel program.

Control the SDM read process

The zGM system provides some controls to customize the SDM read process by installation. The `MaxTracksRead` parameter specifies the maximum number of record sets that can be read in a single read channel program by SDM. Increase the value of this parameter to decrease the number of read channel programs that are required to offload the same number of record sets. This adjustment provides a more efficient offloading process.

See Example 5-7 on page 77 for appropriate settings for `MaxTracksRead`, `ResidualLeftToRead`, and `ReadDelay`. Also see 5.4.1, “Parameters that optimize reader performance” on page 72.

Configure SDM balancing

zGM supports large configurations where many system data movers are required to handle the whole workload. The system data movers can exist in a coupled zGM master session. Any delays in one session might cause delays in other sessions. To prevent a persistent workload imbalance between data movers, use the following guidelines:

- Balance the workload (highly active volumes, readers, and LSS) across data movers.
- Run the maximum number of data movers that the available resources (processor and memory) on an individual LPAR permits.
- Enable clustering on LPARs and balance data movers across the clusters.
- Balance central processor complex (CPC) resources across the data mover LPARs.

Avoid excessive reader pacing in the zGM session

When reader pacing occurs, the ‘leading’ readers must stop the offloading process and wait for the ‘lagging’ reader. Even the ‘leading’ readers still have redundant capacity to offload the record sets.

To avoid the excessive reader pacing situation, first in priority, balance the primary system update workload on primary storage subsystems. For a configuration without ER, balance workload across the primary storage control sessions. For ER configuration, balance workload across primary LSSs.

If possible, always enable the enhanced multiple reader function to increase the parallelism under the primary LSS level. There are typically some ‘hot’ volumes in a production environment, such as volumes that contain DB2 active log data sets. When volumes are hot, it is more difficult to balance workload across storage control situations.

Excessive reader pacing situations are less likely with enhanced reader configurations. By implementing the excessive reader function, you can have multiple readers concurrently read record sets for those “hot” volumes and prevent the reader pacing situation.

There are two zGM parameters for controlling reader pacing: *ReaderPacingLimit* and *ReaderPacingWindow*. Increasing the value of these parameters increases the efficiency of data movement. However, these parameters should be used only in an environment where data movement buffers are plentiful. See Example 5-7 on page 77 for appropriate initial values.

5.2 Maximizing write to auxiliary efficiency

The number of SDM data movement buffers is limited. If the speed of writing to auxiliary volumes is not fast enough to offload consistency groups from the buffers, SDM must slow down the speed of the read primary updates. It is important to tune the zGM configuration for an efficient write-to-auxiliary process and maximum reader efficiency.

Balance the storage control configuration

Configure your system so that primary system activity does not exceed the capacity of the auxiliary storage subsystem.

zGM supports configurations where many primary site storage controls can funnel updated data to a single storage control on the recovery system. However, you must consider the number of copied primary volumes, and the rate of write activity to these volumes, in your overall work load evaluation for the configuration. An inadequate configuration can cause zGM to cancel the copy operation to one or more volumes, or even end the storage control session.

Distribute workload on auxiliary storage controls

Avoid directing all update activity to a small set of common volumes on a single auxiliary storage control. Typically, only a small number of devices receive most of the activity on a particular primary storage control.

Usually, 2 - 4 devices account for 50% or more of the total update activity, and 8 - 10 volumes account for 80% of the update activity during any stress time frame. If, for example, two primary storage controls channel their most active volumes into a single recovery site storage control, the auxiliary storage subsystem can rapidly become overcommitted with copy activity.

Optimize the recovery system setup

Allocate the zGM control, state, and journal data sets, and the auxiliary volumes behind storage controls that have large cache and NVS storages. This configuration, in addition to speeding up the zGM copy process, also provides a powerful platform for the recovery operation.

For the best configuration, configure the auxiliary site storage controls to have at least the same cache and NVS capacity as the primary site storage controls. In addition, avoid sharing the resources that are used by auxiliary volumes with other workloads, to prevent resource contention from influencing the performance of writing to auxiliary operations. This includes the following resources:

- Channel paths between SDM and auxiliary storage subsystems
- Auxiliary storage subsystem HA cards
- Ports
- Cache

- NVS
- Processors

Use parallel access volumes for auxiliary devices

Enabling parallel write I/O for zGM auxiliary devices can improve overall data mover throughput and reduce average session delays. To enable parallel writes, the auxiliary logical subsystems must be configured with parallel access volumes (PAVs). Static aliases can be used.

For best flexibility, use the HyperPAV feature of IBM system storage. The Dynamic Alias Management function of z/OS Workload Manager is not recommended for zGM auxiliary devices.

Control write to auxiliary operation by zGM parameters

The **ConsistencyGroupCombined** parameter specifies the number of consistency groups that is combined when they are written to auxiliary volumes by SDM. Higher values of **ConsistencyGroupCombined** increase the efficiency of writing to auxiliary operations, but use more data movement buffers. See 5.4.2, “Parameters that optimize consistency group writes to auxiliary” on page 73 for more information about the parameters’ values.

5.3 Maximizing write to journal data set efficiency

The zGM system journals all copy data in the journal data sets before the data is written to the auxiliary volumes. Maintain an efficient journaling process to avoid effects to zGM performance.

Note: For zGM sessions that are running in migration mode, there is no requirement for data consistency. No journaling process is required.

To maximize the performance of the journaling process, carefully determine where and how to allocate journal data sets. The following actions were allocation guidelines in the past:

- Separate journal data set volumes and auxiliary volumes into different LSSs.
- Define journal data sets as striped data sets.
- Spread as many disk volumes as possible to benefit the parallelism.

These guidelines are no longer strictly required in more recent installations that can benefit from storage pool striping and Easy Tier. Storage pool striping within the storage system relieves the need to worry about carefully spreading the stripes across arrays. However, data set striping is still relevant so that large consistency groups are written with many parallel streams.

Journal data set size is another consideration. If zGM fills all of the journal data sets, it must wait for auxiliary update processing to release journal space before continuing. If the full-journal condition is not relieved within a reasonable amount of time, zGM suspends the session, and then restarts it. Ensure that sufficient space is defined for the journal data set. Allocate the same amount of space for all journal data sets.

For more information about how to determine the correct journal data set size, and about optimizing performance, see “Specifying the journal data sets” in *z/OS DFSMS Advanced Copy Services*, SC35-0428.

5.4 Tuning parameters for zGM

In zGM, you have the flexibility of tailoring SDM operations to installation requirements. Key parameters can be modified through the use of XRC parameter data sets and parmlib members. Some parameters are fixed at session start, and some can be dynamically modified while a session is active.

This section describes the tuning parameters that are available for zGM, focusing on those that are most important, and for which you might need to specify a value different than the supplied default value. For more information about the topics that are mentioned in this section, see *z/OS DFSMS Advanced Copy Services,* SC35-0428.

Currently, there are over 80 documented SDM parameters and flags, in addition to some undocumented patches that IBM has provided for certain installations. With so many values that can be set, configuring and tuning a zGM environment can seem to be a daunting task at first. Fortunately, only a small subset of these parameters plays a critical role in optimizing session performance.

Figure 5-1 and the descriptions that follow logically group these parameters by the system function that they affect. Keeping this picture in mind will help you to understand the implications of parameter changes on both zGM and your production applications.

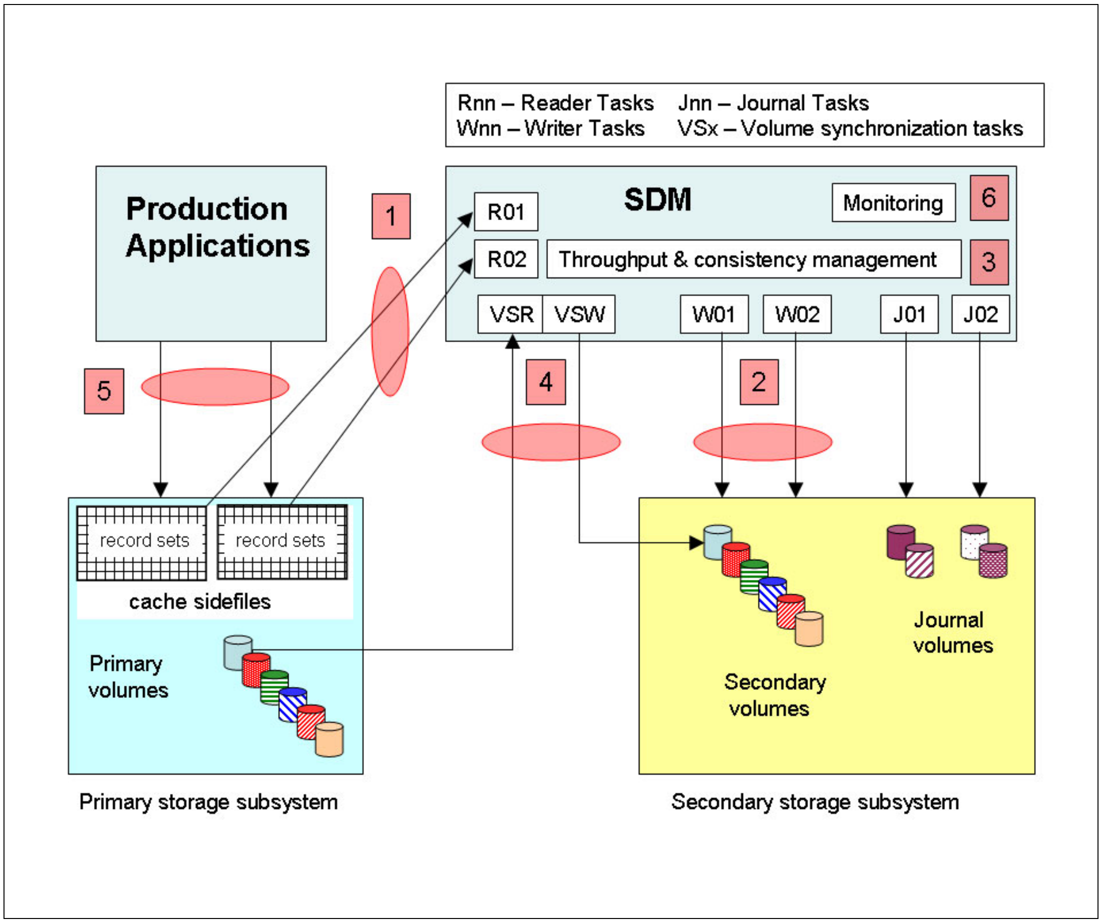


Figure 5-1 SDM tuning parameter categorization

Figure 5-1 on page 71 shows the following functional areas for grouping SDM parameters:

1. These are the parameters associated with the performance of SDM readers, which retrieve XRC record sets from the primary storage system cache, and store them in SDM's memory buffers.
2. These are the parameters associated with writes that occur when consistency groups are mirrored to zGM auxiliary volumes.
3. These are the parameters that optimize internal data mover throughput and the consistency group formation process.
4. These are the parameters that control volume copy activity that occurs during initial volume synchronization and resynchronization.
5. These are the parameters that control the pacing of application system write activity to help maintain recovery point objectives (RPOs) and improve SDM resiliency.
6. These are the parameters that control the frequency of session delay monitoring, and the creation of performance data that can be analyzed by various related XRC performance monitoring tools.

5.4.1 Parameters that optimize reader performance

The first group of parameters listed in Example 5-1 address SDM reader performance and the primary storage system from SDM perspective.

The goal of the parameters listed in Example 5-1 is to use sufficient parallelism to handle peak LSS record set creation rates, while avoiding systemic overload of available channel, storage area network (SAN), and disk host adapter resources. The values shown are a good starting point for a coupled data mover session servicing up to eight LSSs on a DS8700 primary storage subsystem.

Example 5-1 Parameters that address reader performance

```
SHADOW -
RequireUtility(YES) -
   UtilityDevice(FIX) -
   AllowEnhancedReader(YES) -
   NumberReaderTasks(*,4) -
   MaxTracksRead(64) -
   ResidualLeftToRead(64) -
   ReadDelay(250) -
   ReaderPacingLimit(50) -
   ReaderPacingWindow(2) -
PATCH NAME(XMTUN) OFFSET(X'53') -
   OLDDATA(X'02') NEWDATA(X'04')
```

Note the use of Enhanced Reader support, with **NumberReaderTasks** specifying four readers per LSS, which will work in parallel to offload recordsets from cache sidefiles. The utility device requirement can be satisfied with four base MVS devices (3390B) per LSS, or if HyperPAV is available, one base device can be specified and SDM will use 3390A devices for parallel reader activity.

The number of readers per LSS can be adjusted up or down depending on anticipated peak LSS workload, available channel paths, network bandwidth, and host adapter connections.

The combination of **MaxTracksRead** and the **XMTUN OFFSET(X'53')** patch combine to determine the size of the reader channel programs that will be used for an LSS, given its current workload. With the patch, readers associated with the LSS that has the oldest record sets remaining in cache will retrieve up to four times as many updates as the readers for the other LSS.

ReaderPacingLimit and **ReaderPacingWindow** further help the data mover prioritize its use of reader resources, by combining to identify those LSSs that can be omitted from a read cycle given their advanced time stamps.

ReadDelay controls the reader *polling frequency*, or how often the SDM checks for unread record sets in cache. The 250 ms value suggested here helps in coupled XRC environments where some data movers are busy when one has little or no work to do. Without the more frequent polling, the near-idle data mover can end up gating (delaying) the others.

5.4.2 Parameters that optimize consistency group writes to auxiliary

The goal is to apply all updates as quickly as possible to the auxiliary volumes to free up SDM memory buffers, enabling the next group of record sets to be retrieved from the cache sidefiles. Ideally, the throughput capabilities for writes to the auxiliary volumes will always be greater than or equal to the peak arrival rate through the replication infrastructure.

Example 5-2 shows the parameter settings that help to optimize throughput. Values shown are typical best practices for a DS8800 with HyperPAV enabled.

Example 5-2 Parmlib settings to address I/Os to the auxiliary volumes

```
SHADOW -
   PavVolumes(6) -
   PAVByteThreshold(2457600) -
   MaxBytesTransferred(2457600) -
   ConsistencyGroupCombined(5)
```

The **MaxBytesTransferred** value controls the maximum size of the individual channel programs that are used to write consistency group data to an auxiliary volume. We use a large value to optimize the *full stride write* capabilities of the latest technology disk storage systems.

The **PavVolumes** value enables SDM to use six multiple parallel I/O streams to write data for a consistency group to a single volume. This is especially important as volume sizes become larger and HyperPAV is used on application systems. However when a **PavVolumes** value greater than one is specified, the availability of a sufficient number of aliases (and preferably HyperPAV) on the auxiliary storage system is equally essential.

SDM will automatically use multiple parallel write streams on the 10 volumes that have the most data to write in a consistency group. Beyond that, the **PAVByteThreshold** is examined to determine if parallel streams are to be used. Because the data mover has a limited number of write tasks available, we increase the threshold value to favor parallelism across volumes.

In cases where a queue of consistency groups is waiting to be written to the auxiliary, the **ConsistencyGroupCombined** value enables them to be combined, with the potential for consolidation of multiple updates to the same records. Combining is an expensive operation in terms of CPU resources, however, so we use a moderate tuning value.

5.4.3 Parameters that optimize internal data mover throughput

SDM is a highly parallel execution software component, with many tasks concurrently active in its associated address space, ANTASO*nn*. Memory buffers are used to store record sets as they are read from the cache sidefile and move through the data mover pipeline. Therefore, the keys to optimizing throughput are to provide the highest speed processors to run large bursts of tasks, and to ensure that a sufficient number of buffers are available to enable fully concurrent operations for reading, journaling, and writing to auxiliary.

The following best practices help optimize throughput:

- Configure and fully use zIIP processors.
- Avoid central processing unit (CPU) capping.
- Configure SDM to use the maximum number of buffers permitted per SDM.
- Permanent page fix all of the storage, to avoid the system resource usage associated with repeated fixing and freeing.

Example 5-3 shows the related parameters.

Example 5-3 Parameters that provide optimal memory configuration for SDM

```
STARTUP -
   zIIPEnable(FULL)
STORAGE -
   TotalBuffers(25000) -
   BuffersPerStorageControl(25000) -
   PermanentFixedPages(1500) -
   ReleaseFixedPages(NO) -
   IODataAreas(2048)
```

The **ZIIPEnable(FULL)** value enables the SDM address spaces to fully use available zIIP processors, which run at full capacity, in the data movement pipeline. Beware of running this work on subcapacity processors. Even if total MIPS matches the calculated XRC workload requirements, the single stream instruction execute rate might be insufficient to sustain peak performance requirements.

5.4.4 Parameters that optimize volume copy activity

The objective of volume copy processing (synchronization) is to get all volume pairs in DUPLEX status as quickly as possible, while not causing problems to the primary applications. Related SDM parameters control the amount of parallelism used to copy the volumes, both at the LSS level and at the data mover session level. There are parameters that control the number of tracks copied in each channel program, in addition to parameters intended to prevent volume copy activity from creating excessive mirroring delays.

There are two different scenarios when volume copy is exercised: during initial volume synchronization, and during resynchronization after a planned or unplanned suspension event. More aggressive parameters are typically used during resynchronization, especially after a short-duration outage, because the number of tracks to be copied per volume is quite small relative to a full initial copy.

Example 5-4 on page 75 shows a good set of parameters to use for resynchronization of a data mover managing eight LSSs on a DS8800 storage system, where the session is running with **WritePacingResidualCnt(80)**.

Example 5-4 Parameters to optimize a resynchronization

```
BITMAP -
   DelayTime(00.02.00)
SHADOW
   HaltAnyInit(YES) -
   HaltThreshold(5120) -
VOLINIT -
   InitializationsPerPrimary(4) -
   InitializationsPerSecondary(4) -
   MaxNumberInitializations(32) -
   TracksPerRead(12) -
   TracksPerWrite(12)
```

The **InitializationPerPrimary** parameter determines the maximum number of volume copies per primary LSS that can run simultaneously. A matching value is always used for **InitializationPerSecondary**. Using moderate volumes for the number of copy tasks per LSS avoids overstressing the arrays in the primary and auxiliary storage systems.

MaxNumberInitializtions limits the total volume copy tasks in the data mover session, and is typically determined by multiplying the **InitiazationsPerPrimary** value by the number of LSSs managed by the data mover session.

The values for **TracksPerRead** and **TracksPerWrite** control the size of the channel programs used to perform volume copy. The value 12 shown in Example 5-4 on page 75 optimizes channel throughput, especially on long-distance track reads.

The **BitMapDelayTime** value controls the frequency that SDM tells the storage system about the updates that have been secured on the auxiliary. By decreasing the time interval of this communication (known as a *bitmap toggle*) to 2 minutes, the number of tracks to be copied after a suspend is minimized.

SDM has the ability to temporarily pause volume copy activity when a significant buildup of unread recordsets is accumulating in cache. The **HaltThreshold** and **HaltAnyInit** parameters shown in Example 5-4 on page 75 will result in such a pause for all volumes in a data mover session, when the residual count for any sidefile associated with the session exceeds 5120. This is the level at which maximum write pacing would be in effect for a single volume in a session that runs with **WritePacingResidualCnt(80)**.

5.4.5 Parameters that control pacing of application writes

XRC has three mechanisms for delaying or pausing write activity to primary volumes. One of these, known as *Extended Long Busy (ELB)*, can occur when an XRC sidefile reaches its record set count limit, or when the maximum cache utilization percentage is reached.

In such cases, it is preferable to suspend XRC, as opposed to incurring application I/O delays for an extended period of time. The **SuspendOnLongBusy** value shown in Example 5-5 on page 76 enables this suspension capability, and is strongly suggested.

The second mechanism for write delay is a former method known as *device blocking*, which causes a device to become not ready for 1 second when its residual count reaches a certain threshold. This method of delay is obsolete and is not recommended. However, it can be inadvertently enabled if the **DVCBLOCK** parameter is not specified on **XADDPAIR**. The **DfltWritePacingLvl** value shown in Example 5-5 on page 76 avoids this exposure by assigning the lowest level of write pacing in such cases.

The third mechanism for write delay is write pacing, which is a variable, write-delay injection that can be set based on device residual count. A best practice is to specify an appropriate pacing value for individual volumes via **XADDPAIR** or **XSET**. However, there is a key SDM parameter, **WrtPacingResidualCnt**, that controls the onset and incrementing of write pacing delay.

The value shown in Example 5-5 indicates that the maximum permitted write pacing for a volume is injecting when device residual count reaches 80 * 64 = 5120. This value can be adjusted higher or lower depending on how aggressive you want to be with maintaining a low RPO.

Example 5-5 Good practice ANTXINxx parameters for SDM sessions

```
SHADOW -
SuspendOnLongBusy(YES) -
StorageControlTimeout(00.02.00) -
DfltWritePacingLvl(1)-
WritePacingResidualCnt(80)
```

5.4.6 Parameters that control SDM monitoring

SDM has a monitor task that checks for delays and collects performance statistics. The interval of collection is specified in milliseconds by the **MonitorWakeUp** parameter. Also, the **MonitorOutput** parameter controls whether SDM save the statistics in the `STATE` data set. Because these statistics are often essential to the diagnosis of XRC performance issues, perform frequent collection, as shown in Example 5-6.

Example 5-6 Enabling collection of SDM monitor data

```
MONITOR -
MonitorOutput(ON)
MonitorWakeup(10000)
```

5.4.7 Creating an SDM parmlib data set

All SDM parameters can be specified in the `ANTXIN00` member of SYS1.Parmlib. However, for greater flexibility, create an `hlq.XCOPY.PARMLIB` data set. Information on how to accomplish this can be found in *z/OS DFSMS Advanced Copy Services*, SC35-0428.

Use the `ANTXIN00` member in `SYS1.PARMLIB` for parameters in the `NAMES` and `STARTUP` categories. Then use the **ALL** member in `hlq.XCOPY.PARMLIB` for parameters that are common to all sessions, and the specific session name member for any values unique to a specific session.

Parameters are read as part of command processing for **XSTART**, **XRECOVER**, and **XADVANCE** (if the data mover address space is not already active). They are also read when a cluster address space starts.

Many parameters, flags, and patches can also be dynamically updated by using the **XSET parmlib** command with **XACTION(APPLY)**, and parameter data set members can be checked at any time by using the **XSET parmlib** command with **XACTION(VERIFY)**.

5.4.8 Using XQuery to verify SDM parameter settings

Example 5-7 is an XQuery example listing all parameters, including the defaults, in alphabetical order (this is not an especially logical order, but makes it easy to find the parameter that you are looking for).

Example 5-7 XQuery-based ANTXINxx listing

```
XQUERY STARTED FOR SESSION(SESS1) ASNAME(ANTAS001) 433
XQUERY ENVIRONMENT_PARM REPORT - 001
NAME                    VALUE  NAME                    VALUE
-------------------------------------------------------------
zIIPEnable                YES  MiscLow                     2
AllowEnhancedReader       YES  MonitorOutput              ON
BuffersPerStorageCon    25000  MonitorWakeup           10000
ChangedTracks            7500  MHlq                  DE40581
ClusterMSession      ********  NoTimeStampCount         5000
ClusterName          ********  NumberReaderTasks         *,0
ConsistencyGroupComb        5  PacingReportThreshol       10
DatasetDelay               75  PavByteThreshold       512500
DeadSessionDelay           45  PavVolumes                  3
DefaultHlq               SDM1  PermanentFixedPages      1500
DefaultSessionId      DEFAULT  ReaderPacingLimit          33
DelayTime            00.02.00  ReaderPacingWindow          3
DeviceBlockingThresh       20  ReadDelay                 500
DfltWritePacingLvl          6  ReadRecordsPriority       252
EnableREFRESHS             NO  ReleaseFixedPages          NO
HaltAnyInit                NO  RequireUtility            YES
HaltThreshold            5120  ResidualLeftToRead         64
Hlq                      SDM1  ScheduleVerify             NO
InitializationsPerPr        4  SecondaryDeviceRange   (none)
InitializationsPerSe        4  SecondaryVolserPatte   (none)
InitializationMethod     FULL  SelectionAlgorithm       LOAD
InitializationReadWr      120  ShadowRead                 10
IODataAreas              2048  ShadowTimeoutPercent       40
JournalPriority           251  ShadowWrite                10
LowAttention              192  StorageControlTimeou 00.02.00
MaxBytesTransferred    512500  SuppressTimestamp          NO
MaxControlTasks           128  SuspendOnLongBusy         YES
MaxNumberInitializat       45  TotalBuffers            25000
MaxTotalReaderTasks        32  TracksPerRead              12
MaxTracksFormatted          0  TracksPerWrite             12
MaxTracksRead              64  UtilityDevice             FIX
MaxTracksUpdated            0  VerifyInterval             24
MinExtenderRead            55  WriteRecordsPriority      253
MinLocalRead                0  WrtPacingResidualCnt       80
MiscHigh                   15
-------------------------------------------------------------
XQUERY ENVIRONMENT_PARM REPORT COMPLETE FOR SESSION(SESS1)
```

6

Scenarios for ongoing operations

In this chapter, we describe some of the tasks that you might have to perform during normal operations.

It is important to have some performance monitoring active all the time, to be able to detect abnormal behavior and troubleshoot it.

Some times you will have to deal with suspended sessions. A session suspend can be unplanned, when something abnormal happens, or it can be initiated by command for a planned maintenance.

Some other tasks, such as workload redistribution and maintenance tasks, are also mentioned.

6.1 Performance monitoring and analysis

You have to establish some monitoring of your z/OS Global Mirror (zGM) environment. Of course, performance monitoring products like RMF or IBM Tivoli Storage Productivity Center for Replication (TSPC) can help quite a bit. But these products do not specifically monitor zGM. Of course, you could implement some zGM monitoring by regularly issuing all kinds of XQuery commands. But this is not really practical.

It is important to have the current performance numbers, but you also need some long-term performance monitoring:

- Real-time performance monitoring is important if one of your clients complains about bad performance. With real-time monitoring, you can see what is going on right now.
- Long-term performance monitoring enables you to detect abnormal conditions. If you do not know what *is normal*, you cannot judge if your zGM environment behaves well or not.

6.1.1 IBM TotalStorage XRC Performance Monitor

The IBM TotalStorage Extended Remote Copy (XRC) Performance Monitor is a licensed program that provides the ability to monitor and evaluate a running zGM configuration:

- Tuning
- System constraint determination
- System growth
- Capacity planning

The running configuration can be one that is operating for testing or benchmarking purposes, or an actual production environment. With XRC Performance Monitor, you can observe system performance in detail, from a really high level to a really low level, and in both real-time and historical modes.

It also produces several data sets that can be imported into spreadsheet programs to create performance reports and graphs. While it is monitoring performance, XRC Performance Monitor can automatically check key zGM parameters, and post notifications in the event that user-definable thresholds are exceeded.

The XRC Performance Monitor is an IBM licensed program. For more details, see *IBM TotalStorage XRC Performance Monitor Installation and User's Guide*, GC26-7479.

The XRC Performance Monitor uses the performance statistics generated by the System Data Mover (SDM). The statistical data generation process on each SDM must be activated by specifying **`MonitorOutput ON`** in the `hlq.XCOPY.PARMLIB` member for the session.

The XRC Performance Monitor provides the following statistics at preset intervals:

- Number of `DUPLEX`, `PENDING`, `SUSPENDED`, (and so on) volumes
- Information about device blocking
- Whether any long busy conditions occur
- The rate that the SDM was reading data from the primary control units in megabytes per second (MBps), and the MBps for each reader
- Number of fixed buffers available for the SDM
- Number of non-fixed buffers available for the SDM
- Delay time
- Delay reasons (non-zero queues)

- Maximum data loss exposure time
- Residual counts
- Cache in use by SDM

The XRC Performance Monitor consists of three separate modules, which are integrated under one user interface:

History Monitor
Summarizes how the z/GM (previously XRC) system was running at a prior point in time. At approximately 15-minute intervals, this report displays key performance information in an easy-to-read sorted table. Detailed information can be obtained about each reader and specific volumes.

This information, displayed graphically, can help identify volume contention that might have created bottlenecks. This report includes details about the telecommunications link, such as the amount of data transferred, and measures of the mirrored site, such as peak delay times and average delay times.

Real-Time Monitor
Provides a summary of how single or coupled system data movers are running. By displaying real-time information, the XRC Performance Monitor enables administrators to detect and resolve problems quickly.

Batch Exception Monitor
Checks the monitor information at user-defined intervals for predefined thresholds. This report generates console messages whenever thresholds are exceeded so that administrators can take action, therefore eliminating the need for someone to constantly check the monitor.

Automation products can intercept the write to operator (WTO) messages and take actions.

Optimize the telecommunication link

The bandwidth of the telecommunications link between the primary and auxiliary sites is one of the key manageable resources in an XRC system. Because of the cost of high-speed circuits, overbuying capacity can add expense without much benefit. Likewise, if the link is too slow, unacceptably long delays can be introduced when updating the mirrored system.

One of the key applications of the XRC Performance Monitor is to make sure that the link is sized properly. Typically, traffic on the telecommunications link varies throughout the day. For example, batch updates to an online system at night often produce much higher traffic on the XRC link than daytime operations. Unattended charting enables administrators to see latency as it varies throughout the day, so that they can size resources accordingly. Historical charts show trends, which can be useful for predicting future demands.

Monitor details

The following sections provide some more information about the long-term monitors.

History Monitor

The History Monitor performs the same function as the Real-Time Monitor (the same Interactive System Productivity Facility panel, or ISPF panel), but works with historical data that was recorded over time by the XRC Performance Monitor data collection utility. This is not the same as the Real-Time Monitor obtaining the data in real time by reading the `MONITOR1` member of `XRC STATE` data set.

The History Monitor uses HISTORY files that are generated by the XRC Performance Monitor data collection utility. The History Monitor summarizes data and displays statistics about the remote copy environment in a consistent and easy-to-read way, based on ISPF facilities. Each time the Enter button is pressed, the panel shows data from the skip-forward timeline.

History Summary

History Summary presents a picture of how a given system was running at a previous point in time. It summarizes the data, and displays key performance information in an easy-to-read and sortable table. Commands are provided that enable detailed information to be obtained about each reader, and about specific volumes.

It also produces output that provides a graphical summary of historic information. This includes details such as megabytes (MB) of data processed, peak delay times, and average delay times.

The History Summary uses the HISTORY files generated by the XRC Performance Monitor data collection utility. The summary records are each made up of 100 individual interval records. The length of time that is represented by this 100 records sample is dependent on the interval that is specified in the installation for the XRC Performance Monitor data collection utility.

Utility for XRC Performance Monitor data collection

The XRC Performance Monitor provides some utilities that provide the input for long-term monitors, or extent the functions to a personal computer (PC) for analyzing.

When the zGM SDM is set up to generate performance data, it creates, by default, a performance sample data every 10 seconds. The XRC Performance Monitor data collection utilities are designed to collect those data samples into a HISTORY file. Two utilities are available to create HISTORY files with a data.time naming convention, or a generation data group (GDG) naming convention.

The SDM updates the MONITOR1 member of the STATE data set with performance data approximately every 10 seconds. The data collection utility reads the MONITOR1 member and adds it to the end of the current active HISTORY data set.

The HISTORY data set is the input for History Monitor and History Summary.

Utility for Exporting data to Microsoft Excel

XRC Performance Monitor also provides the CSCXJEXL utility to export the data that was collected during the XRC Performance Monitor history summary operation to an hlq.EXPDATA data set. You can download the hlq.EXPDATA data set to a PC and import it into one of the Microsoft Excel templates that are provided in the hlq.SCSCDATA library.

You can also design your own Excel table in addition to the provided templates. The following data fields can be retrieved from the file:

- Peak delay for the interval
- Average delay for the interval
- Peak read rate for the interval
- Peak write rate for the interval
- Average read rate for the interval
- Average write rate for the interval
- Peak cache used for the interval
- Peak exposure for the interval
- Average exposure for the interval
- Fewest available buffers for the interval

- Average number of available buffers
- Total MB transferred during the interval

6.1.2 GDPS/XRC Performance Toolkit

Additional performance information can be obtained from the Geographically Dispersed Parallel Sysplex (GDPS)/XRC Performance Toolkit.

> **Note:** The GDPS/XRC Performance Toolkit can only be used in connection with GDPS services.

The following tools are part of the GDPS/XRC Performance Toolkit:

- History Formatter and Delay Reason Analyzer
- Write Pacing Monitor
- Write Pacing History Collector
- Hyper-Active Volume Checker

History Formatter and Delay Reason Analyzer

This tool consolidates and filters XRC monitor history data to help produce reports and graphs that assist in performance analysis and problem determination. It formats a group of XRC monitor history files, or the files generated by the `XRCXMHIS` function of this toolkit, and focuses on reader statistics and busiest volume information.

From this tool, you can determine the lagging session in a delay situation. It also reports best and worst performance intervals for each session, and an overall summary for the day's activity.

The toolkit provides two History Formatter programs: `XRCHFPDA` and `XRCHFPD2`, which provide different groups of statistics for the lagging coupled session in any given monitor interval. `XRCHFPDA` focuses on reader statistics and busiest volume information. `XRCHFPD2` focuses on time group statistics, buffer utilization, average service times, and maximum service times, for both readers and auxiliary volumes.

Write Pacing Monitors

Besides real-time reporting, Write Pacing Monitors also provide historical granular reporting of write pacing and record-set creation activity. They enable users to quickly and easily assess the following system information:

- Assess the XRC effect on production applications.
- Assess the overall health and capability of the XRC system.
- Identify the most active volumes, logical subsystems (LSSs), and logical control units (LCUs).
- Spot workload imbalances.

The program should be run via job control language (JCL) on the production system to cover primary storage. All LCUs that have write-pacing capable microcode should be covered by this program, and at least one volume in an LCU should be brought online for detection. The program can process a maximum of 512 LCUs.

Two load modules are provided for supporting standard readers and enhanced readers separately.

6.1.3 Primary systems performance monitoring

At the primary site, you are mainly interested in the performance of the primary storage subsystems. In case the performance of the production systems is no longer as expected, you should have some long-term records of I/O characteristics to be able to detect what has changed, including the following statistics:

- I/O rate
- Read/write ratio
- MBps written
- Response times:
  - Connect time
  - Disconnect time
  - Pending time
  - Input/output software queuing (IOSQ) time
- Cache hit percentage
- Rank statistics

Any tool that produces this data can be used, but it is important that *some* performance monitoring is in place. In general, you are not interested in performance data of individual volumes, but because zGM operates at the LCU level, you should also have performance numbers at the LCU level.

RMF and TSPC 5.1 are the preferred monitoring tools for I/O performance. From RMF, you are interested in the *direct access storage device (DASD) Activity Report*, the *Cache Subsystem Summary Report*, the *Channel Path Activity Report*, the *I/O Queuing Report*, the *Enterprise Storage Server (ESS) Disk Array*, and the *ESS Port* reports.

TSPC provides a consolidated view of all of the servers, because it is the view from the storage system. TSPC provides all relevant data, but has some more details, such as response times separate for read and write I/Os, the average transfer size for reads and writes, and whether I/Os were random or sequential.

SDM utility volumes

Of some special interest at the primary storage systems are the zGM utility volumes. They are used by the SDM to read all updates to the primary volumes:

- I/O rate
- Connect
- Disconnect
- IOSQ
- Pending time
- Number of parallel access volume (PAV) alias addresses used

Because large I/O chains are used, *connect* time will be quite high, but *disconnect* and *IOSQ* should be low. Pending time can also be quite high, but in any case you need to know what is normal for your environment to be able to recognize abnormal conditions.

6.1.4 SAN monitoring

Most storage area network (SAN) switches have at least some basic performance monitoring, and optional licensed advanced performance monitoring. Observe the port throughput for the Fibre Channel connections (FICON) ports used between the primary storage systems and the remote SDM.

Normally, a network line cannot transmit data at the nominal link speed across a long distance. However, if you observe the throughput curves, you will probably find that the throughput stays flat at a high throughput level. This is probably the maximum throughput that you can get on this connection. You should know this limit.

Another counter to look at are the relative port error counters. An increase in the error rate indicates a problem.

6.1.5 Auxiliary storage systems performance monitoring

Just like monitoring the primary production storage systems, there should also be performance monitoring for the auxiliary storage systems. If the auxiliary storage systems have the same performance capabilities as the primary systems, they should not have a performance problem. This is because the I/O load to these systems is only a fraction (just the write load) of the primary systems.

However, some installations use higher-capacity disks at the auxiliary site, and in this case the disk drives in the auxiliary storage systems could be a bottleneck.

You should also observe the performance of the volumes with the journal data sets. Because *all* writes to the auxiliary volumes go first to the journal data sets, the volumes with these data sets could be a bottleneck.

The same tools, such as RMF and TSPC, are normally used to monitor the auxiliary storage systems, just as for the primary systems.

6.1.6 Performance troubleshooting

Every performance problem is different, and we cannot provide a general approach for how to solve any given problem, but the following section provides some guidelines.

It is helpful to distinguish between performance problem categories:

- The production servers
- The production storage systems
- The network between the primary storage systems and the SDM
- The SDM
- The auxiliary storage system

Because the production servers are independent of the mirroring process, performance problems with servers are not mentioned here, but we need to address the production storage systems.

If someone complains about bad performance, this information cannot be expressed as a general feeling. It must be *documented*, for example, “last week a job ran for one hour and now it takes two hours.” With specific information, you can use your historic performance data to see if something in the overall workload profile has changed.

Because you are in a zGM environment, you are primarily interested in the write workload:

- Has the write workload increased?
- Has the application’s Disconnect time increased? If so, this could be an indication for write pacing.
- Are some LCUs much more busy than the others?

Next is an investigation of the network:

- Are all paths operational?
- Has the line throughput approached the link saturation?
- Are port error counts unusually high?
- Is the SDM performing as expected?
- Are there any memory shortages?
- What is the CPU consumption of the SDM?
- Do the auxiliary storage systems show any anomalies?
- Are there long response times in general?
- Are there long response times on the journal volumes?
- What are the FICON link response times?

TSPC provides these port response times, and long response times can indicate a problem.

Collecting data for problem analysis

If you cannot solve the performance problem and you need to open a problem record, you should have collected as much information as possible:

- A description of your environment
- Configuration data (XRC parmlib and so on)
- System logs
- Performance reports
- XRC Performance Monitor reports
- SAN performance reports
- If necessary, memory dumps or statesaves.

For the DS8000 system, you can trigger a statesave with the **F ANTAS00x,STATESAVE** operator command.

6.2 Planned outage support

Because the DS8000 system provides zGM *hardware bitmap* support, suspension of a zGM session has no primary system effect. This is due to the fact that the DS8000 maintains a *hardware bitmap* of updates, so it does not use cache resources while sessions are suspended.

6.2.1 Suspending a zGM session

The SDM supports two types of suspension.

The first is an **XSUSPEND VOLUME(ALL)** suspension. With this suspension, the SDM address space is not stopped. The ANTAS*nnn* address space continues to run. Also, with primary disk subsystems not supporting hardware bitmaps, the SDM continues to read updates. With the DS8000 system as the primary disk subsystem, the hardware bitmap is used to track updated tracks. No updates are applied to the journals or auxiliary volumes.

The second type of suspension is a *session suspension*. This is accomplished by using the **XSUSPEND TIMEOUT** command. This command is issued to stop the SDM for a planned activity, such as a maintenance update, or moving the SDM to a different site or a different logical partition (LPAR).

The **XSUSPEND TIMEOUT** command ends the ANTAS*nnn* address space, and informs the LCUs that the zGM session is suspended. The DS8000 system uses the hardware bitmap to record changed tracks, and frees the write updates from the cache.

When the zGM session is restarted with the **XSTART** command, and volumes are added back to the session with the **XADDPAIR** command, the hardware bitmaps that are maintained by the DS8000 system while the session was suspended are used to resynchronize the volumes, so full volume resynchronization is avoided.

6.2.2 Suspending volumes

The **XSUSPEND VOLUME** command accepts a list of volumes (up to 100 volume pairs can be specified) or ALL (meaning all volumes in the session). If ALL is specified, zGM suspends all volumes in the zGM session at the same consistency time. If a volume list is specified, zGM suspends all the volumes in the list at the same consistency time.

If it is necessary to suspend more than 100 pairs, but not all of the volumes in the session, multiple **XSUSPEND VOLUME(volser,volser,...) ATTIME()** commands with the same ATTIME will cause the volumes to suspend with the same consistency time. Be sure to pick an ATTIME far enough in the future to enable all of the commands to be processed.

When volumes are suspended with the **XSUSPEND VOLUME** command, the ANTAS*nnn* address space remains active. For the suspended volumes, the hardware bitmap that is maintained by the DS8000 system is used later during the resynchronization process.

6.3 Unplanned outage support

The DS8000 system, with its capability of maintaining hardware bitmaps, supports *unplanned outages*. This is the capability of avoiding full resynchronization of volumes after an unexpected failure on one or more of the components in the SDM data path.

When data starts to accumulate in the cache, there can be several reasons:

- A temporary write peak from the primary systems
- A temporary lack of resource for the SDM
- An outage in one or more of the components that are necessary for the SDM data moving process (for example, a failure in channel extender links, or failure of the SDM address space itself)

The zGM system has ways to deal with outage situations, such as the ones described in the first two items. See 2.12, “Dynamic workload pacing” on page 44, and “The zGM initiated suspend instead of long busy status” on page 57.

If this is an outage of a vital SDM component, the other mechanisms control the situation. When the data in the cache exceeds a predefined level, a time interval starts to decrease. This time interval value is set by the **TIMEOUT** parameter in the **XSET** command (not to be confused with the **TIMEOUT** parameter in the **XSUSPEND** command), and can be set individually for each LCU.

The TIMEOUT interval is reset every time the SDM reads from the storage control session. If there is no draining of the cache, the data will continue to accumulate, and will eventually reach a predefined high threshold.

A long busy condition is then presented to the primary systems, and I/Os are queued in the primary hosts. The duration of the long busy condition is the remaining part of the **TIMEOUT** interval. The long busy condition does not occur if the Suspend Instead of Long Busy function has been activated (see “The zGM initiated suspend instead of long busy status” on page 57).

When the interval expires, the storage control session is suspended. The hardware bitmap is used to reflect the changed tracks, and the cache resources are freed. The outage can be of any duration.

The primary systems continue to run without any effect from the zGM environment, and the DS8000 system keeps track of the write updates in the hardware bitmap. This bitmap is used at a later time, when the problem is fixed and the volume pairs are eventually added back to the zGM session with the **XADDPAIR** command.

7

Data migration

Often there is the need to move the content of one volume to another address, which could be on another storage subsystem. The z/OS Global Mirror (zGM) system is an easy-to-use tool to migrate whole volumes.

This chapter shows how easy it is to set up a zGM environment to move volumes.

7.1 Data migration

If the storage subsystem has a zGM license, you can also use zGM to move volumes to another location, and to another storage system. The migration can be between local devices or between remote sites. Moving data actually means copying data, and later switching to the new set of volumes.

When you move data with zGM, you are not interested in data consistency *during* the migration process. At any rate, zGM cannot guarantee it during the initial copy process. But zGM can ensure data consistency when the migration process finishes.

When most of the data is copied, updates to the source volumes are also mirrored to the target volumes. But you do not intend to run in this mode for a long time. When all of the volumes that you want to migrate are nearly synchronous, switch from the old volumes to the new ones.

Before you can start, you have to allocate a state data set as a storage management subsystem (SMS)-managed, partitioned data set extended (PDSE) data set. This data set has a predefined name of **hlq.XCOPY.session_id.STATE**. You can call the session (*session_id*) MIGRATE, for example.

In zGM, there is a special session type: SESSIONTYPE(MIGRATE). In this mode, the journal data sets and control data sets are not needed and are not used.

Because you do not run zGM in disaster protection mode, you do not want to stop the whole session if there is a problem with one volume pair. You need data consistency only at the end of the migration process, so you can specify ERRORLEVEL(VOLUME) for the session.

If there is a problem with one pair, it will be suspended, but zGM will continue to copy the other volume pairs. You can investigate why a pair failed and maybe restart the copy process with **XADDPAIR SUSPENDED**.

The **XSTART** command for a volume migration session could have the following options:

XSTART MIGRATE SESSIONTYPE(MIGRATE) ERRORLEVEL(VOLUME).

Note that coupled sessions are not supported for SESSIONTYPE(MIGRATE).

The target volumes must be initialized with a unique volume serial number (VOLSER), and source and target volumes must be online to the system where the data mover runs. You do not have to use a data mover in a remote system, because normally the data mover of the production system is used.

As with a normal zGM setup, you also have to define a UTILITY device for each logical control unit (LCU) with volumes that need to be migrated. The UTILITY device is specified with the first **XADDPAIR** command. The utility volume is not copied, and there should not be much activity on the utility device.

Now you add your volume pairs to the session with the **XADDPAIR** commands. If not too many volumes are involved, and the migration will not run for days, try to find a time for the migration when the systems are not so busy.

If you have to migrate a large number of volumes, it might be better to carry out several smaller migrations with only a subgroup of the volumes that need to be migrated.

You can control the number of initialization processes by using **XSET**:

- Change the **SYNCH** value to increase the total number of concurrent tasks that perform the migration copy.
- Change the **SCSYNCH** value to increase the maximum number of tasks that can be active concurrently on a single storage control.

Monitor the initial copy process by issuing **XQUERY** Time Sharing Option (TSO) commands. The **XQUERY** command returns the percentage of the initial copy that has finished.

Verify that the zGM volumes are in `DUPLEX` state with an **XQUERY** command. When the volume pair attains `DUPLEX` state, zGM has copied all of the primary volume contents and all subsequent updates to the XRC auxiliary volume. At the same time, zGM continues to update the auxiliary volume as changes occur to the primary.

Now, you can switch your applications from the primary volumes to the auxiliary volumes. To do so, you have to stop the applications that use the primary volumes.

To complete the migration at a known point in time, issue the **XEND** command with the `ATTIME` keyword or the `DRAIN` keyword.

You vary off the old source volumes, because the new volumes will get the VOLSERs of the old volumes.

Now you can issue the **XRECOVER** command to change the auxiliary VOLSER to be the same as the primary volume VOLSER. If the new volumes have more capacity than the old volumes, you need to use the IBM Device Support Facilities (ICKDSF) **REFORMAT REFVTOC** command to refresh the VTOC.

Verify that the new volumes are online. When they are, the applications can be restarted using the new volumes.

A

Checklists for z/OS Global Mirror

This appendix provides useful checklists to help you validate your z/OS Global Mirror (zGM) environment.

System Data Mover

Before setting up the System Data Mover (SDM), consider the following questions:

□ Did you implement the required I/O configurations for the SDM and the target recovery site systems?

□ Have you considered all of the implications of running the SDM, 24x7, in your recovery site, such as production data backups, remote management capability, and archiving performance and audit data?

□ Did you apply program temporary fixes (PTFs) to your application logical partitions (LPARs), and to your SDM LPARs?

□ Did you specify input/output (I/O) dispatching of `IOQ=PRTY` on the SDM, to ensure that it operates at the highest I/O priority? This is controlled by the `SYS1.PARMLIB` member `IEAIPS`*`xx`*. In addition, SDM data mover address spaces (`ANTAS001` - `ANTAS020`, and `ANTCL0nn`) should run in the `SYSTEM` service class. `ANTAS000` should run in `SYSSTC`.

□ Did you verify that there are no user exits (**`IEFUSI`**) that can inhibit the region size, performance, or priority of the SDM address space (`ANTAS00`*`n`*)? Also, ensure that extended common service area (ECSA) and extended system queue area (ESQA) sizes are kept at a minimum. Ideally, the total for both should not exceed 200 megabytes (MB).

□ Did you do a sizing of the SDM to primary systems bandwidth, SDM processor, processor resources, storage, and the SDM to auxiliary disk systems connectivity?

□ Did you ensure that adequate paging space exists in the SDM LPAR? For more information, see the *z/OS DFSMS Advanced Copy Services*, SC35-0428.

□ Did you authorize appropriate users to the Extended Remote Copy (XRC) Time Sharing Option (TSO) commands from the SDM?

□ Did you ensure that the SDM address spaces, `ANTAS000` and `ANTAS00`*`n`*, have `update` authority to the journal, control, state, or master data sets?

□ Did you enable Hyper parallel access volume (HyperPAV), if applicable?

□ Did you decide to use tertiary volumes and, if so, did you plan the auxiliary disk systems with enough storage capacity?

□ Did you allocate the XRC data sets: state, control, journal, and master (if using coupled data movers)? The procedures for allocating data sets are described in *z/OS DFSMS Advanced Copy Services*, SC35-0428. Ensure that these data sets are not subject to space release by the storage management subsystem (SMS).

□ Did you set up protection for the XRC state, control, journal, and master data sets from unauthorized updates?

□ Have you made the required alterations to XRC tuning parameters? For information, see Chapter 5, “Tuning z/OS Global Mirror” on page 65, and see the section on administering your XRC environment in *z/OS DFSMS Advanced Copy Services*, SC35-0428.

□ Did you install additional monitoring tools to monitor XRC and ensure that the Resource Management Facility (RMF) and System Management Facility (SMF) configuration is correct?

□ Have you established automated and manual procedures for stopping XRC in emergency situations?

□ Did you train operations staff as required?

□ Did you document the new configuration and procedures?

Primary systems

For primary systems, consider the following questions:

□ Did you review `IBM.Function.ExtendedRemoteCopy FIXCAT` for any maintenance that must be applied to application systems?

□ Have you confirmed that all systems that can access the XRC primary disk share a common time reference? Any systems that do not share a common time reference must not write to the disk, and must be configured with `SuppressTimestamp(YES)`.

□ Have you examined your production applications and procedures to determine if any changes are required due to implementing remote copy disaster recovery (DR) solutions? Examples might include reviewing the usage of tape during batch processing, and tape backups.

□ DId you complete a review of your workload to determine which volumes (such as temporary work data sets) do not need to be mirrored continuously with XRC, and would be candidates for CopyOnce?

□ Did you review the workload to determine if any significant imbalance exists across data storage subsystems and logical subsystems (LSSs)? In particular, have you done a review of job scheduling to identify potential sources of massive write workload bursts? If you identified such bursts, consider redistribution or rescheduling to provide better balance across storage resources and time.

Primary disk systems

For primary disk systems, consider the following questions:

□ Will data migration to an XRC-capable disk subsystem be required, and if so, have you planned for it?

□ Did you install the XRC support feature on all primary disk systems?

□ Did you ensure that you have sufficient host adapters with optimal port assignment and XRC workload isolation?

□ Did you verify that you will not exceed the logical path limits for primary LSSs?

□ Did you ensure that XRC primary volumes cannot be unintentionally relabeled while in XRC primary status?

□ Have you set aside sufficient base devices and (if used) associated PAV aliases for use as XRC utility volumes?

□ Did you specify `UTILITY(FIX)` and `REQUIREUTILITY(YES)` in the SDM parmlib, which ensures that volume pairs cannot be added unless covered by a utility?

Auxiliary disk systems

For auxiliary disk systems, consider the following questions:

- □ Did you plan XRC support on all auxiliary disk systems, to facilitate return from the recovery site to the primary site after a planned or unplanned failover?
- □ Did you develop a naming convention for VOLSERS of auxiliary volumes, to facilitate the relationship of the auxiliary volume to the primary volume?
- □ Did you ensure that you have sufficient host adapters with optimal port assignments?
- □ Did you initialize and label the auxiliary and tertiary volumes with `ICKDSF`?
- □ Did you ensure that auxiliary volumes are dedicated to XRC usage? This can be done by ensuring that the volumes are part of an SMS)-managed disabled storage group, or by varying the auxiliary volumes offline to all but the SDM.
- □ Did you validate that there is sufficient disk system cache and non-volatile storage (NVS) for all disk systems?
- □ If the auxiliary disk system will contain journal data sets, did you verify that there is enough bandwidth at the level of channel paths and disk arrays to support this configuration?
- □ Did you verify that there is enough SDM-channel connectivity to support the production workload after a failover?
- □ Did you plan a one-to-one relationship between primary and auxiliary disk systems and subsystem identifiers (SSIDs)? Although this is not a strict requirement, it reduces the effort needed to manage the XRC configuration, and minimizes the chance of performance problems occurring at the recovery site.
- □ If you are using a tertiary copy, what method will you use to create the copy? Is this feature (FlashCopy) installed on the auxiliary disk system?

Intersite connectivity

Evaluate the following intersite connectivity issues:

- □ Which technology will you use: Fibre Channel connections (FICON), with or without dense wavelength division multiplexer (DWDM), or channel extenders?
- □ Did you order DWDMs, channel extenders, and FICON director equipment with the correct type and number of ports?
- □ Did you complete bandwidth analysis, and translate this into a requirement for dark fibers or telecommunication links?
- □ Did you order the dark fiber or telecommunication connections? Are they available in time?
- □ Did you verify that your intersite connection meets requirements for both availability and performance?
- □ If you are using channel extenders:
  - □ Did you verify that the channel-extender configuration supports the required number of devices?
  - □ Did you evaluate the use of compression and verify that the compression features are installed and available?
  - □ Have you decided to use fixed utility devices, and have you allocated the devices?
- □ Have you optimized the channel extenders for XRC?

XRC testing

Design a test schedule that will enable you to perform the following tasks:

- □ Validate connectivity and bandwidth between SDM and the primary disk systems?
- □ Validate security controls for XRC commands and data set access? For information, see *z/OS DFSMS Advanced Copy Services*, SC35-0428.
- □ Perform functional testing (testing commands)?
- □ Develop and test XRC management procedures?
- □ Test automation procedures developed for XRC?
- □ Perform error injection tests?
- □ Create a phased plan for adding production volumes to an XRC session? For information, see *DFSMS Extended Remote Copy Reference Information for Advanced Users*, GC35-0482.
- □ Test a planned failover?
- □ Test an unplanned failover?

Evaluate your installation

After XRC is installed and operational, you should verify the following criteria for success:

- □ Is the operations staff trained and educated to manage XRC on a day-to-day basis?
- □ Do actual workload levels occur as planned, and can the XRC configuration manage them as planned?
- □ Are the procedures that are designed to identify and report status exception conditions working as designed?
- □ Are the procedures that are designed to identify and report performance exception conditions working as planned?
- □ Are capacity planning procedures in place to manage workload growth, and the growth of primary disk systems?
- □ Have you decided on a software maintenance strategy for the SDM? Is this the same as or different from your strategy for the primary systems?
- □ Have you set up service level agreements?

Related publications

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

IBM Redbooks publications

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only:

- *GDPS Family An Introduction to Concepts and Facilities*, SG24-6374
- *IBM System Storage DS8000 Copy Services for IBM System z*, SG24-6787

You can search for, view, download, or order these documents and other Redbooks, Redpapers, Web Docs, drafts, and additional materials, at the following website:

ibm.com/redbooks

Other publications

These publications are also relevant as further information sources:

- *z/VM V5R4.0 CP Planning and Administration*, GC24-6083
- *z/VM V6R2 CP Planning and Administration*, SC24-6178
- *z/OS DFSMS Advanced Copy Services*, SC35-0428
- *GDPS/MzGM Planning and Implementation Guide*, ZG24-1757
- *MVS Setting Up a Sysplex*, SA22-7625

Online resources

These websites are also relevant as further information sources:

- z/OS V1R12.0 information center

  http://publib.boulder.ibm.com/infocenter/zos/v1r12/index.jsp

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

IBM®

IBM z/OS Global Mirror Planning, Operations, and Best Practices

Redpaper™

Planning considerations

Healthy environment maintenance

Ongoing operations

IBM z/OS Global Mirror (zGM), also known as Extended Remote Copy (XRC), is a combined hardware and software solution that offers the highest levels of continuous data availability in a disaster recovery (DR) and workload movement environment. Available for the IBM DS8000 Storage System, zGM provides an asynchronous remote copy solution.

This IBM Redpaper publication takes you through best practices for planning, tuning, operating, and monitoring a zGM installation.

This publication is intended for clients and storage administrators who need to understand and maintain a zGM environment.

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

REDP-4878-00 ISBN 0738453501