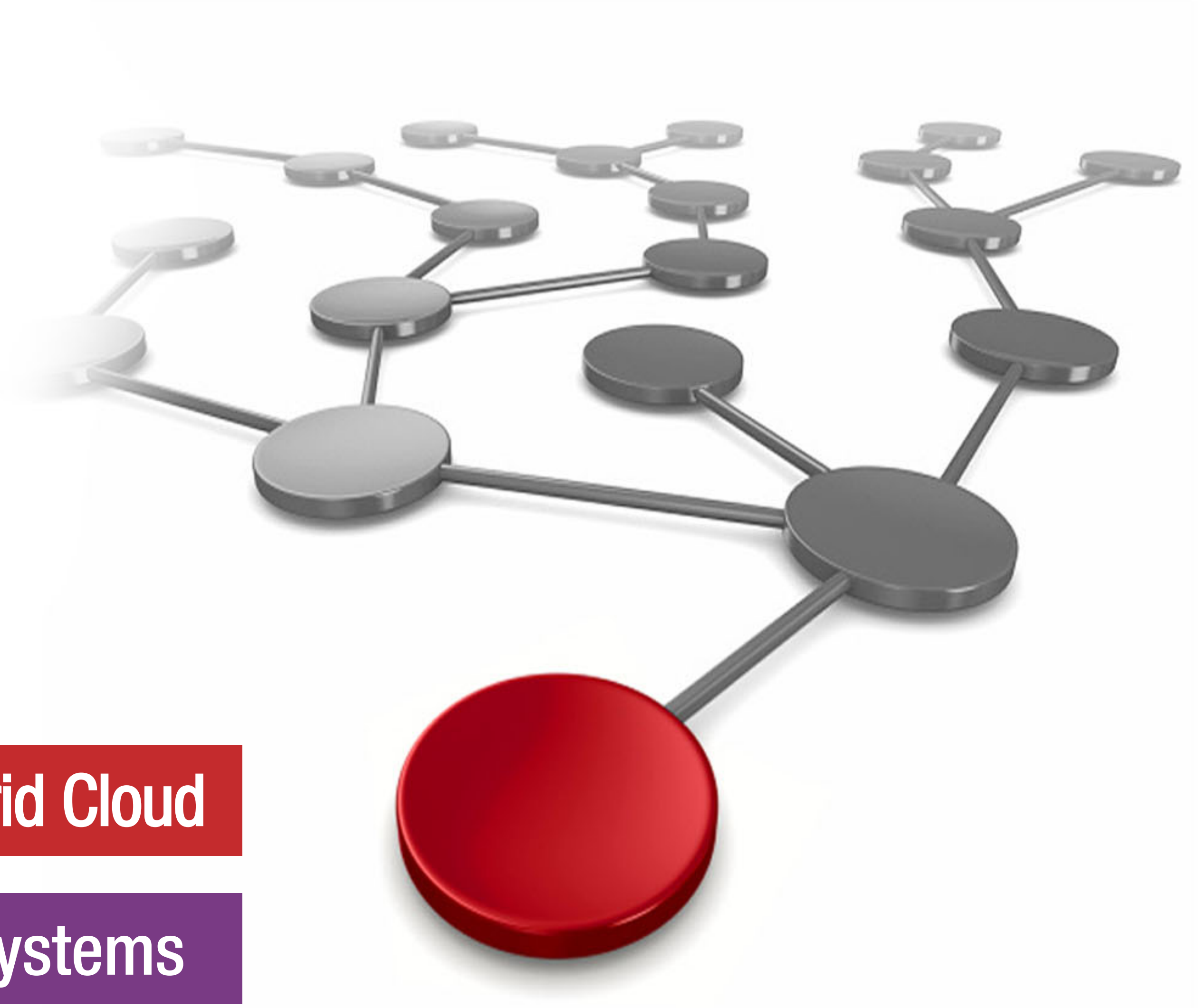
Redbooks
ibm.com/redbooks

Recommendations for Implementing Geographic Logical Volume Manager (GLVM) On-Premises and on the Cloud

Dino Quintero
Tim Simon
Antonio Bozzini
Carl Burnett
Vera Cruz
Anil Kalavakolanu
Jes Kiran
Gus Schlachter
Ravi A. Shankar
Tom Swart
Antony Steel



Hybrid Cloud

Power Systems

IBM

Redpaper

IBM Redbooks

Recommendations for Implementing GLVM On-Premises and on the Cloud

February 2024

REDP-5717-00

Note: Before using this information and the product it supports, read the information in "Notices" on page v.

First Edition (February 2024)

This edition applies to AIX Version 7.2.5 or later.

This document was created or updated on April 15, 2024.

Contents

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 products and/or the programs 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 provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

Preface

This IBM Redpaper introduces the AIX Geographic Logical Volume Manage and examines DR options for AIX, both on premises and in the Cloud. It also offers recommendations designed to ensure a smooth implementation and covers good practices around ongoing maintenance and monitoring.

At the time of writing this document Global Replication Service (GRS)[1], which provides disk mirroring capability within IBM Power Virtual Server (PowerVS) was only available between a limited number of Data Centers and its support by IBM PowerHA® SystemMirror® Enterprise Edition (PowerHA) had not been announced. Hence we are focusing only on IBM AIX® Geographic Logical Volume Manager (GLVM) which provides data mirroring capabilities between AIX systems. We will cover using GLVM as a tool to migrate from on-premises systems to PowerVS and as a DR solution. The DR solution could be solely in the cloud, or between on-premises and the cloud.

The main driver for writing this document was to address the varied experiences we have seen with the implementation of GLVM – both standalone and when managed by PowerHA. While many implementations go smoothly and the customer experiences very few issues in production – there are others that have been, shall we say, far from smooth. The aim is to document what has lead to a successful GLVM implementation and to provide warnings for where “there be dragons.”

Most of the recommendations will generally apply to any implementation of GLVM, but we will highlight the specific requirements dictated by PowerVS. It is also vital that the OS and applications teams understand that they will be working in a clustered environment, so practices and processes – particularly change control – must be modified to reflect this.

It is assumed that the reader is familiar with the IBM Redpaper by Ravi Shankar et al. *Asynchronous Geographic Logical Volume Mirroring Best Practices for Cloud Deployment*, REDP-5665[2].

Authors

This paper was produced by a team of specialists from around the world working at IBM Redbooks, Austin Center.

Dino Quintero is a Systems Technology Architect with IBM® Redbooks®. He has 28 years of experience with IBM Power technologies and solutions. Dino shares his technical computing passion and expertise by leading teams developing technical content in the areas of enterprise continuous availability, enterprise systems management, high-performance computing (HPC), cloud computing, artificial intelligence (AI) (including machine and deep learning), and cognitive solutions. He is a Certified Open Group Distinguished Technical Specialist. Dino is formerly from the province of Chiriqui in Panama. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Tim Simon is a Redbooks Project Leader in Tulsa, Oklahoma, USA. He has over 40 years of experience with IBM primarily in a technical sales role working with customers to help them

[1] https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-getting-started-GRS
[2] https://www.redbooks.ibm.com/abstracts/redp5665.html

create IBM solutions to solve their business problems. He holds a BS degree in Math from Towson University in Maryland. He has worked with many IBM products and has extensive experience creating customer solutions using IBM Power, IBM Storage, and IBM System z® throughout his career.

Antonio Bozzini is an IBM Technical Support Professional in Italy. He has 35 years experience in IBM Power and AIX, primarily working on AIX Support. His focus area is on performance analysis with 25 years of experience in supporting IBM customers in solving performance issues on Power/AIX environments. He has worked for AIX Development Support since 2016 and is an SME (subject matter expert) in that area. His area of expertise includes AIX, VIOS, PowerVM, extending to PHYP and other areas involved in performance.

Carl Burnett is a Distinguished Engineer within Power Systems Software Development. Over his thirty-plus year IBM career around Power and AIX, he has worked on multiple systems software projects, including AIX kernel features, distributed computing frameworks, operating system security, and scalable distributed filesystems. He has seen the platform grow from the uniprocessor IBM PC RT running AIX version 2 to its current state as a 240 core, 1920 HW thread, Enterprise system with AIX as one of the most reliable, secure, and scalable operating systems in the industry. Carl's current responsibilities include defining technical strategy and development plans for AIX and VIOS. Major focus areas include hybrid multi-cloud, automation, workload optimization, and enhanced Power virtualization for flexible infrastructure integration.

Vera Cruz is a consultant for IBM Power in IBM ASEAN Technology Lifecycle Services. She has 28 years of IT experience doing implementation, performance management, HA and risk assessment, and security assessment for IBM AIX and IBM Power across diverse industries, including banking, manufacturing, retail, and government institutions. She has been with IBM for 8 years. Before joining IBM, she worked for various IBM Business Partners in the Philippines and Singapore working as Tech Support Specialist and Systems Engineer for IBM AIX and IBM Power. She holds a degree in Computer Engineering from the Cebu Institute of Technology University in Cebu, Philippines.

Anil Kalavakolanu is a Master Inventor and Senior Technical Staff Member with AIX Support in Austin, Texas. He has 30 years experience as a Support Engineer with AIX.

Jes Kiran is a Software Architect for HA and DR Technologies on Power Systems. He has over twenty years of IT experience with expertise in High Availability, Disaster Recovery, PowerHA, VM Recovery Manager, Public Cloud, Hybrid Cloud, Storage replication and Container technologies. He is an IBM Master Inventor has authored many technical papers.

Gus Schlachter is a subject-matter expert in PowerHA/HACMP working with IBM Technlology Lifecycle Services in Austin, TX. He has broad experience with implementing and supporting PowerHA along with the associated skills in TCP/IP and LVM and extensive experience in ksh scripting and Regular Expressions.

Ravi A. Shankar is a Distinguished Engineer and is part of the Power Hybrid Cloud development team leading development of multi-cloud Power offerings. Ravi has extensive experience in Security, Business resiliency, and Cloud disciplines. He has been involved in enabling different Power Cloud offerings in IBM Cloud. Ravi also has deep understanding of HA & DR products including PowerHA SystemMirror for AIX and VM Recovery Manager.

Tom Swart is a Senior Software Engineer with IBM in the United States. He has 35 years of experience as Level 2 customer support with IBM, concentrating on the IBM products VM/ESA , PSSP, AIX, RSCT, and GLVM He holds a Bachelor of Arts and Science degree in Computer Science from Potsdam College in Potsdam, NY.

Antony Steel is a senior technical staff member working with IBM Australia. A research chemist by training, he brings a unique experience and perspective with over 30 years of experience in the IT industry as a programmer, customer, and IBM Business Partner. For over 20 years, he was with IBM Australia and Singapore as Senior Managing Consultant / Advanced Technical Support. Antony's customers include users, senior management, and other key stakeholders in a range of industries, including some of the largest financial and business institutions and government departments in Australia, New Zealand, and the Asia Pacific region. His areas of interest are IBM AIX, HADR, and clustering. He is an IBM Champion who has assisted with preparing HA and IBM AIX certification exams.

Now you can become a published author, too!

Here is an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an IBM Redbooks 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:

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:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, IBM Redbooks
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on LinkedIn:

  https://www.linkedin.com/groups/2130806

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

  https://www.redbooks.ibm.com/subscribe

- Stay current on recent Redbooks publications with RSS Feeds:

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

1

Introduction to AIX Geographic Logical Volume Manager

This chapter provides an introduction to AIX Geographic Logical Volume Manager (GLVM). GLVM provides a remote mirror solution between two locations using network connectivity. GLVM provides a business continuity solution for AIX based applications and it can also be utilized as a migration tool for migrating applications between locations where storage replication is not available – for example from your on-premises environment to a cloud environment. It can also provide data replication between a primary site and a recovery site in your business continuity or disaster recovery solution. It is assumed that the reader is familiar with the Redpaper *Asynchronous Geographic Logical Volume Mirroring Best Practices for Cloud Deployment*, REDP-5665[1].

It is important to note that while GLVM can operate standalone, or under the control of PowerHA, IBM strongly recommends using PowerHA to manage GLVM. In particular it provides further checking and therefore reduces the likelihood of data being corrupted. For further details see section 3.10, “Integration with Power HA SystemMirror Enterprise Edition” on page 40. If GLVM is just being used to migrate data from on-premises to the IBM cloud, then the overhead of PowerHA is not required.

This document covers:

- “Introduction to GLVM”
- “Recommended use cases for GLVM”

[1] https://www.redbooks.ibm.com/abstracts/redp5665.html

1.1 Introduction to GLVM

Geographic Logical Volume Manager has been a part of AIX for many years and is designed to mirror data at the AIX logical volume level between two different servers. The replication can be either synchronous or asynchronous.

The remote server – where the remote mirrored physical volumes reside – runs the Remote Physical Volume server (RPV server). There is one RPV server for each replicated physical volume and each RPV server is seen as a device – rpvserverN.

The Remote Physical Volume client (RPV client) is seen as device hdiskN running on the local Server. There is one RPV client for each remote mirrored physical volume. AIX LVM manages the mirroring between the local physical disks and the RPV clients, while the RPV client/server pair manage the transmission of the updates over the network and the manage the application of those updates to the disks on the remote server. This is shown in Figure 1-1.

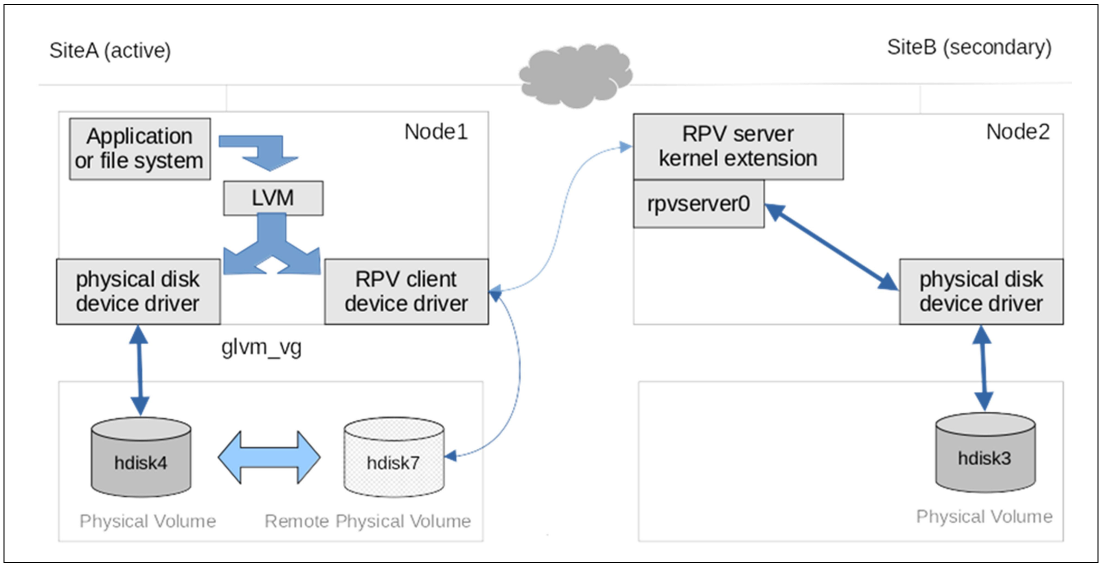


Figure 1-1 GLVM process path

Mirroring is configured using AIX Logical Volume Manager (LVM) mirror capabilities. Each Logical Volume has a copy on local physical volumes and a second copy is defined on the RPV Clients. A volume group that is configured to use geographic mirroring is called a Geographic Mirrored Volume Group (GMVG).

For situations where the Primary and DR instances are within 80 to 100 kilometers – this distance depends on the quality and its components and ultimately the latency introduced by the network – a synchronous solution would be suitable for most applications. However for greater distances, or if your application depends on very responsive I/O, asynchronous replication will be required.

1.1.1 Synchronous GLVM

As with any synchronous replication technology, when using synchronous GLVM the application does not receive acknowledgment that a write operation is complete until the data is written to the disk on the remote server. This means that the two copies of the mirror are always synchronized and no data will be lost if a failure occurs.

This also means that the I/O wait time for write operations in the application is extended by the length of time that it takes to copy the data to the remote site. This is why synchronous GLVM is restricted to situations where the two sites are within a distance of less than 100 km and the applications can tolerate the extended I/O wait for write operations.

Figure 1-2 shows the data flow for synchronous replication.

1. Application writes to LVM.
2. LVM writes to both the local hdisk and RPV Client.
3. The write returns from the local disk subsystem and the write is sent over the network to the RPV Server.
4. The remote disk is updated.
5. The I/O complete returns to the RPV Server.
6. The RPV Client is updated with the I/O done.
7. The LVM is updated that the second (mirror) I/O is complete.
8. I/O complete is returned to the application.

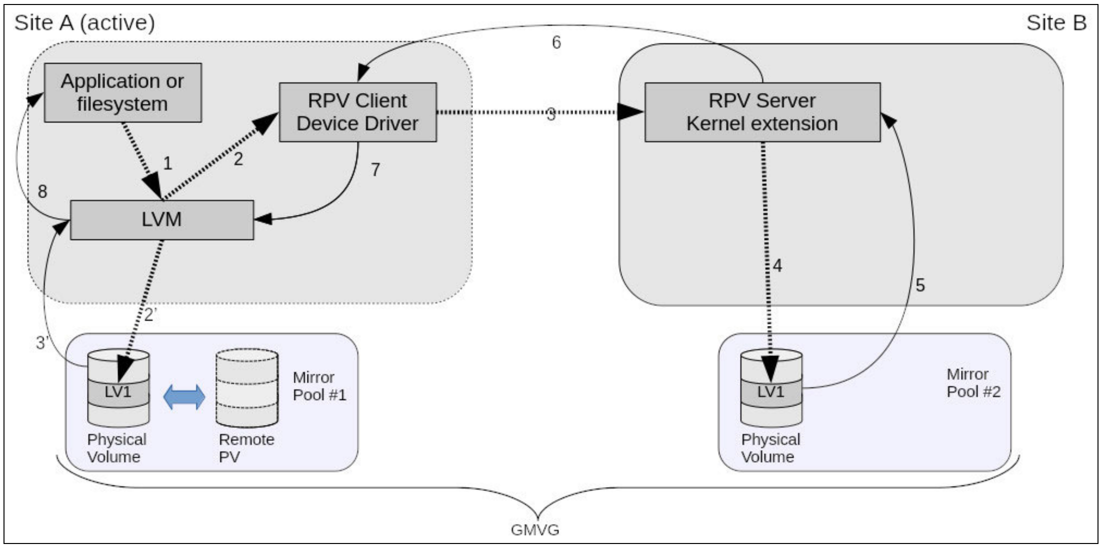


Figure 1-2 Data flow for synchronous GLVM

1.1.2 Asynchronous GLVM

Asynchronous GLVM makes use of a local cache (an additional AIX logical volume in the replicated volume group) to temporarily store the updates until they can be transmitted to the remote system and played against the physical volumes there. The nature of asynchronous replication means that there can be situations where the local server fails before all of the data has been replicated to the remote copy and some data can be lost. Your business recovery operations will need to account for the amount of data lost in order to restart the application.

In asynchronous GLVM, AIX LVM writes to both the local physical volumes and the cache. At some later time, the RPV client/server will play those cached writes against the remote physical volume in the order they were received. Once the remote write has been acknowledged, the entry in the cache will be cleared.

The cache needs to be carefully sized and should only be used to mask the network latency and perhaps handle a few spikes in I/O. The cache cannot be used to cover for insufficient network bandwidth. If the cache becomes full, all writes become synchronous until space is cleared as some of the cached I/Os are written to the remote site. After space has become available in the cache, asynchronous mode replication will resume. For this reason it is important to size the cache correctly. This will be an exercise in balancing application I/O, network bandwidth, and the amount of data that you can afford to lose in a disaster. See section 3.7.3, “Planning the cache” on page 37 for recommendations and guidance.

> **Note:** The cache represents the maximum amount of data that will be lost in a disaster. If the primary site fails, the cache contains all the data that has not yet been replicated to the DR site.

It is important to understand that only the LVM is aware of the nature of the RPV clients. The upper layers (application / file system) pass the I/O to the LVM and the LVM handles the mirroring and returns an IOdone to the application / file system once the response has been returned from the RPV client (synchronous mode) or aio cache (asynchronous mode).

Figure 1-3 shows the first stage in the data flow for asynchronous mode.

1. The application writes to the LVM.
2. LVM writes to both the local hdisk and RPV Client.
3. The write returns from the local disk subsystem.
   The write is stored in the cache.
4. The I/O complete returns from the cache disk subsystem.
5. The I/O complete returns to the LVM.
6. IOdone returns to the application.

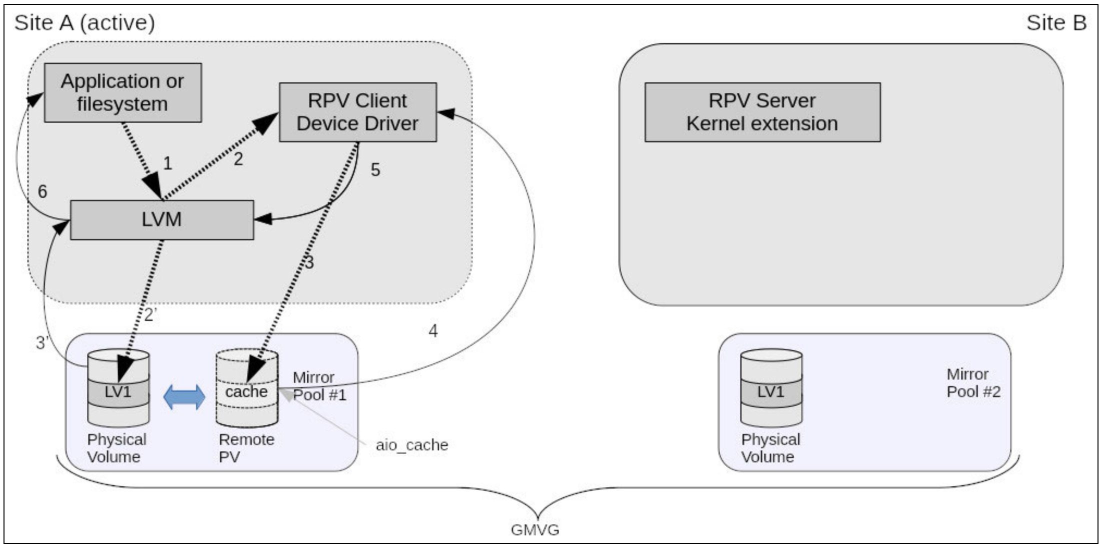


Figure 1-3 Initial stage of data flow in asynchronous GLVM

Once the data is in the GLVM cache, the systems will transmit it to the remote system where the writes are applied against the remote copy of the data in the same order they were received. This provides data consistency in the remote copy.

Figure 1-4 on page 5 shows the second stage in the data flow for asynchronous mode

1. The RPV client walks through the cache writes in the order in which they were stored.
2. The I/O is sent to the RPV Server.
3. The remote physical volume is updated.
4. The I/O complete returns from the disk subsystem.
5. The RPV Client is updated with the I/O complete.
6. The I/O is removed from the cache.

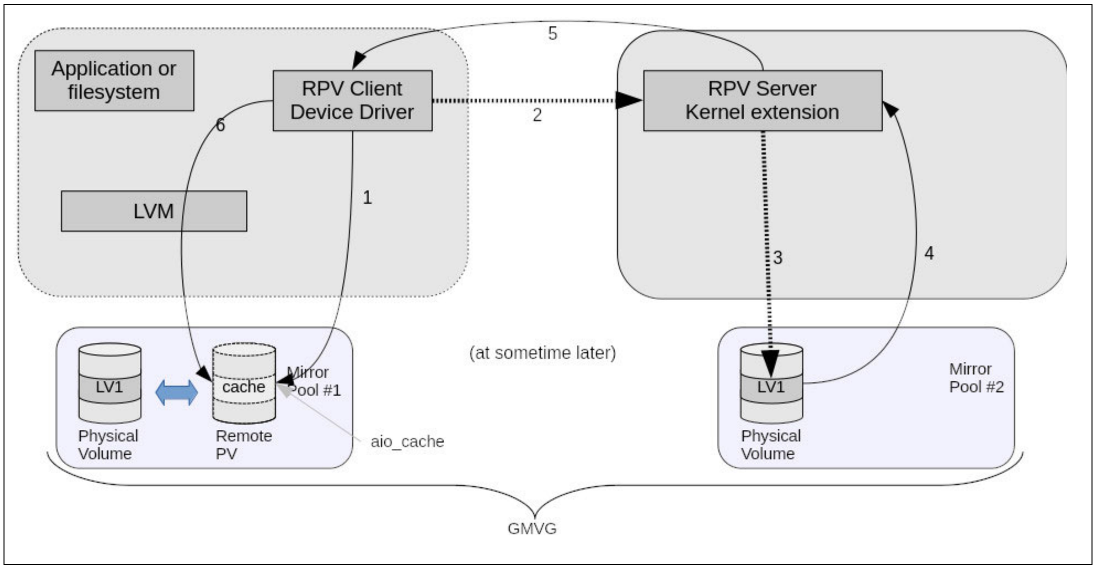


Figure 1-4 Second phase of asynchronous GLVM write operations

1.2 Recommended use cases for GLVM

As described earlier, GLVM operates in two modes – synchronous and asynchronous. These two modes have quite distinct requirements and use cases. This section discusses environments where GLVM might provide a workable solution and some rules of thumb for network bandwidth. Additional recommendations on planning are provided in Chapter 3, “Planning, sizing and tuning” on page 27.

GLVM is recommended in situations when:

- The application does not provide its own replication. Many databases will provide their own replication, which is typically more efficient and gives the administrator greater control – including the ability to roll transactions back and forward.
- There is the need to migrate data with only IP connectivity between data centers with a requirement for minimal downtime, and where tools such as rsync and Mass Data Migration devices are not suitable due to application I/O or data layout.
- The application I/O profile (write pattern) over time is well understood.
- A DR solution is required where there is no storage layer replication.
- A correctly sized network with redundancy is configured.

This paper includes recommendations about the OS and hardware that may also impact the suitability for GLVM as a solution in your environment, for example:

- It is recommended to use AIX 7.2.5 or above.
- Power hardware accelerated compression is recommended.
- The solution includes high performance storage and network with sufficient bandwidth to manage the application I/O.
- The network uses low latency components and has redundancy built in.
- PowerHA is recommended to assist in management.
- It is recommended that GLVM experienced resources are available to assist with the planning, implementation and training.

Note: It is a requirement of a successful GLVM implementation that the application's I/O profile is well understood and can be quantified. Experience has shown that the majority of post implementation issues are associated with network sizing and stability.

Initial problems with an implementation are typically due to poor planning, however in the longer term, failure to monitor the I/O as the applications grow, can become an issue.

The suitability of a synchronous solution will be dependent on the distance between sites and the quality of the networking hardware. This is seen as the round trip time for each packet as every application write I/O will have to wait for acknowledgment from the remote site. While different applications can handle different levels of I/O latency, the typical distance between data centers using synchronous replication would be less than 80 to 100 km.

For an example:

> If there is 100km between the data centers, then light would take about 0.7 milliseconds for the round trip and extra time is added by each switch and router in the path. Typically for a configuration with two data centers 100km apart with low latency networking equipment, 2-3 milliseconds is added to each I/O operation.

Analysis of the application I/O should focus on application writes, as application reads can always be handled by the local copy of the data – unless the volume group is in the process of recovering from a failure.

The application write throughput – particularly for any sustained peaks – should not exceed the ability of the network to transmit the updates to the DR site. Table 1-1 gives some examples of the absolute maximum I/O for the application at several common network speeds.

Note: Most I/O profile tools report I/O in B/s (bytes / second) while network speed is commonly reported in b/s (bits / second). As a rule of thumb, we use a factor of 10 rather than 8 in the conversion to allow for packet headers etc.

Table 1-1 Max I/O relative to common networks

Maximum I/O	Network (Gb/s)
<<100 MB/s	1
<< 500 MB/s	5
<< 1 GB/s	10
<< 4 GB/s	40
<< 10 GB/s	100

2

Configuring GLVM

This chapter demonstrates how to configure GLVM in your environment. We document the sample environment used in our lab and walk through the steps required to setup the GLVM cluster.

This example uses a three node cluster with two servers at the primary site and a third server at the remote location. This provides the ability to have asynchronous GLVM replication between the primary and secondary site and also allows the ability to recover from a server failure at the primary site.

The following topics are covered:

- “Steps involved to configure a simple cluster”
- “Setup instructions”
- “Useful lsglvm options”

2.1 Steps involved to configure a simple cluster

This section covers the steps to configure one of the most common configurations – a simple 3 node / 2 site GLVM cluster. At a high level, the steps required are:

- Configure sites.
- Create the RPV servers on the server in the secondary data center.
- Create the RPV clients on the first server in the primary data center.
- Create the GMVGs.
- Create the LVs and file systems.
- Stop activity on the first server in the primary site.
- Configure the RPV Client on the second server in the primary data center and import the VG.
- Stop activity on the second server in the primary site.
- Stop the RPV servers at the secondary data center.
- Create the RPV servers on the first server in the primary data center.
- Create the RPV clients on the server in the secondary data center.
- Import the GMVGs on the server in the secondary data center.
- Stop activity on the server at the server in the secondary data center.
- Stop the RPV servers on the first server in the primary data center and create the RPV servers on the second server in the primary data center.
- Start the RPV clients in the secondary data center and check the GMVGs.
- Stop the RPV clients and then the RPV servers.

The above steps will create two GMVGs with synchronous replication. To change the GMVGs to asynchronous mode, activate the aio_cache and change the mirror pools to “async” mode.

2.1.1 Lab layout

The lab environment consists of 2 sites, with 2 LPARs at the primary site and a single LPAR at the secondary site. The two LPARs at the primary site share the LUNs that we will be replicated to the secondary site, which means that the GLVM configuration steps must be done on both those LPARs. Figure 2-1 shows the overall site configuration. The logical volume layout for both GMVGs is shown in Figure 2-2 on page 9.

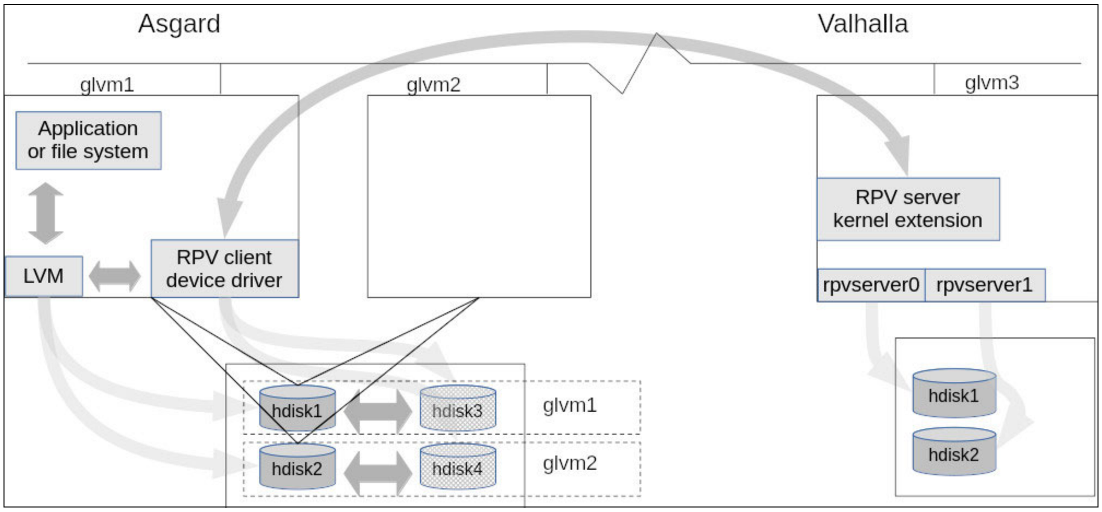


Figure 2-1 Lab setup

The details on the lab setup are shown in Table 2-1.

Table 2-1 Lab configuration

Setting	Site 1		Site 2
Site name	Asgard		Valhalla
Host name	glvm1	glvm2	glvm3
Address	192.169.200.125	192.169.200.254	192.169.200.196
Site alias	192.169.200.20		192.169.200.30
Mirror pool	glvm1		glvm2
PVIDs	00c9388038acc305		00c937e038ac7fe5
	00c9388038acab89		00c937e038ac8e46
Volume group	glvm1		
jfs2 log logical volume	ulv11		
jfs2 logical volume	ulv12		
aio cache	glvm1_val_ca		glvm1_asg_ca
file system	glvm1		
Volume group	glvm2		
jfs2 log logical volume	ulv21		
jfs2 logical volume	ulv22		
aio cache	glvm2_val_ca		glvm2_asg_ca
file system	glvm2		

The GMVG shows the logical volumes with 2 copies (one at each site / mirror pool) and the cache at each site to hold the outstanding writes for the remote site. Figure 2-2 Shows the physical view of the GMVGs, the logical view would have the physical disks and the RPV clients at the active site.

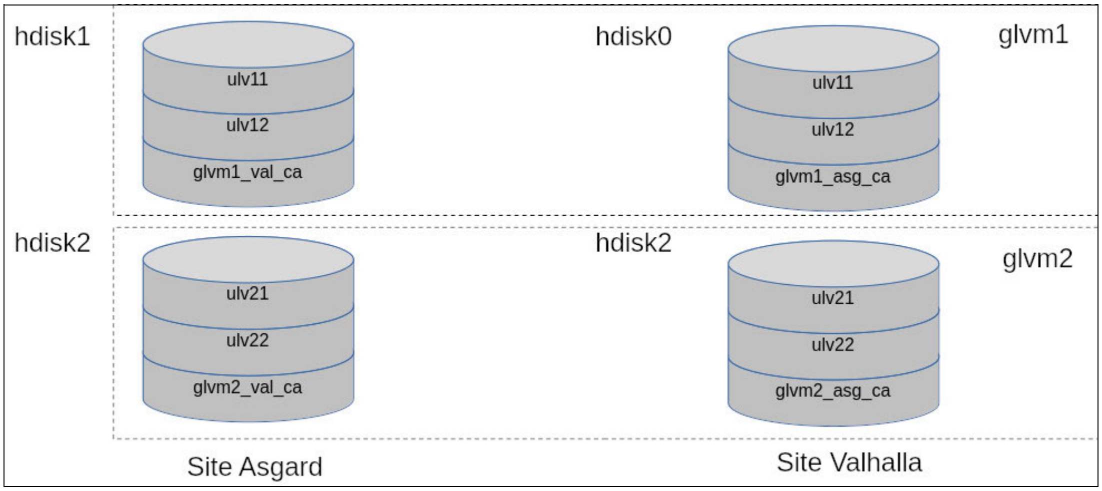


Figure 2-2 logical volume layout for the GMVGs

2.1.2 LPAR disk and network configuration

Each site, as shown in Table 2-1 on page 9 has an alias IP addresses that is used for GLVM communication. This is used to move the GLVM configuration between the two nodes at the Asgard site and is also required if PowerHA is used to manage the replication. If PowerHA is used, then the GLVM is built on PowerHA managed persistent alias addresses.

Example 2-1 shows the storage and network configuration for LPAR glvm1

Example 2-1 LPAR glvm1 server configuration (site Asgard)

```
# lspv
hdisk0          00fa00d66c59c9d7                    rootvg          active
hdisk1          00c9388038acab89                    None
hdisk2          00c9388038acc305                    None

# ifconfig en1
en1:
flags=1e084863,81cc0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD(ACTIVE),LARGESEND,CHAIN>
        inet 192.169.200.125 netmask 0xffffff00 broadcast 192.169.200.255
        inet 192.169.200.20 netmask 0xffffff00 broadcast 192.169.200.255
         tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
```

Example 2-2 shows the storage and network configuration for LPAR glvm2

Example 2-2 LPAR glvm2 server configuration (site Asgard)

```
# lspv
hdisk0          00fa00d66c59c9d7                    rootvg          active
hdisk1          00c9388038acc305                    None
hdisk2          00c9388038acab89                    None

# ifconfig en1
en1:
flags=1e084863,81cc0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD(ACTIVE),LARGESEND,CHAIN>
   inet 192.169.200.254 netmask 0xffffff00 broadcast 192.169.200.255
    tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
```

Example 2-3 shows the storage and network configuration for LPAR glvm3

Example 2-3 LPAR glvm3 server configuration (site Valhalla)

```
# lspv
hdisk0          00c937e038ac7fe5                    None
hdisk1          00fa00d66c59c9d7                    rootvg          active
hdisk2          00c937e038ac8e46                    None

# ifconfig en1
en1:
flags=1e084863,81cc0<UP,BROADCAST,NOTRAILERS,RUNNING,SIMPLEX,MULTICAST,GROUPRT,64B
IT,CHECKSUM_OFFLOAD(ACTIVE),LARGESEND,CHAIN>
   inet 192.169.200.254 netmask 0xffffff00 broadcast 192.169.200.255
    tcp_sendspace 262144 tcp_recvspace 262144 rfc1323 1
```

2.2 Setup instructions

This section provides detailed instructions for each of the steps involved in creating the GLVM cluster.

> **Note:** The following steps outline configuring mirror pools and super strict mode. While this is only required if using asynchronous GLVM, it is recommended also for synchronous GLVM and will do no harm.

2.2.1 Configure sites

If using PowerHA, the site names must match the PowerHA site name. Sites can be configured as follows:

```
/usr/sbin/rpvsitename -a [sitename]
```

This can also be done using SMIT:

```
smitty glvm_utils → Remote Physical Volume Servers.
 → Remote Physical Volume Server Site Name Configuration.
 → Define / Change / Show Remote Physical Volume Server Site Name.
```

Enter site name as shown in Example 2-4.

Example 2-4 Setting site name by way of SMIT

```
          Define / Change / Show Remote Physical Volume Server Site Name

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                      [Entry Fields]
* Remote Physical Volume Server Site Name           [Asgard]




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Repeat on the node in the other site.

2.2.2 Create RPV servers on the server in the secondary data center

This can be performed using the command line and both RPV servers can be created with one command as seen in Example 2-5.

Example 2-5 Creating RPV Server

```
# mkdev -c rpvserver -s rpvserver -t rpvstype -a \
rpvs_pvid=00c937e038ac7fe5 00c937e038ac8e46 -a client_addr='192.168.200.20' \
-a auto_online='n'
rpvserver0 Available
rpvserver1 Available
```

This can also be done using SMIT:

smit glvm_utils → Remote Physical Volume Servers.
→ Add Remote Physical Volume Servers.

Select:

- The local physical volume from the name and PVID listed.

Set:

- “Configure Automatically at System Restart” to no.
- “Start New Devices Immediately” to yes.

See Example 2-6.

Example 2-6 Creating RPV Servers via SMIT

```
                             Add Remote Physical Volume Servers

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Physical Volume Identifiers                         00c937e038ac7fe5 00c937e038ac8e46
* Remote Physical Volume Client Internet Address     [192.168.200.20]          +
  Configure Automatically at System Restart?         [no]                       +
  Start New Devices Immediately?                     [yes]                      +




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The attributes for the RPV servers can be seen in Example 2-7

Example 2-7 Displaying attributes of the RPV servers

```
# lsattr -El rpvserver0
auto_online n                                  Configure at System Boot   True
client_addr 192.169.200.30                     Client IP Address          True
rpvs_pvid   00c9388038acab890000000000000000 Physical Volume Identifier True
# lsattr -El rpvserver1
auto_online n                                  Configure at System Boot   True
client_addr 192.169.200.30                     Client IP Address          True
rpvs_pvid   00c9388038acc3050000000000000000 Physical Volume Identifier True
```

In Example 2-8 we use the **lsrpvserver** command to list the RPV servers.

Example 2-8 Using lsrpvserver command

```
# lsrpvserver -H
# RPV Server     Physical Volume Identifier       Physical Volume
# -----------------------------------------------------------
  rpvserver0     00c9388038acab89                 hdisk1
  rpvserver1     00c9388038acc305                 hdisk2
```

2.2.3 Create the RPV clients on the first server in the primary data center

Again this can be performed using the command line as shown in Example 2-9, creating both RPV clients in one step.

Example 2-9 Creating RPV Client

```
# mkdev -c disk -s remote_disk -t rpvclient -a pvid=’00c937e038ac7fe5000000
00c937e038ac8e460000000, \
-a server_addr='192.168.200.30' -a local_addr='192.168.200.20' \
-a io_timeout='180'
hdisk3 Available
hdisk4 Available
```

This can also be done using SMIT:

`SMIT glvm_utils` → Remote Physical Volume Clients.

Select: IPv6 if required.

Enter: the RPV Server IP address.

Select:

- The local network address.
- The hdisk on the server that this client will point to.
- I/O timeout Interval (10).
- Start New Devices Immediately (yes).

See Example 2-10.

Example 2-10 Adding RPV Client by way of SMIT

```
                             Add Remote Physical Volume Clients

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                        [Entry Fields]
  Remote Physical Volume Server Internet Address      192.168.200.30
  Remote Physical Volume Local Internet Address       192.168.200.20
  Physical Volume Identifiers                         00c937e038ac7fe50000000000000000
00c937e038ac8e4600>
  I/O Timeout Interval (Seconds)                     [10]                    #
  Start New Devices Immediately?                     [yes]                  +




F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

The attributes for these RPV clients can be seen in Example 2-11 on page 14.

Example 2-11 RPV client attributes

```
# lsattr -El hdisk3
io_timeout   180                              I/O Timeout Interval          True
local_addr   192.169.200.20                   Local IP Address (Network 1)  True
local_addr2  none                             Local IP Address (Network 2)  True
local_addr3  none                             Local IP Address (Network 3)  True
local_addr4  none                             Local IP Address (Network 4)  True
pvid         00c937e038ac7fe50000000000000000 Physical Volume Identifier    True
server_addr  192.169.200.30                   Server IP Address (Network 1) True
server_addr2 none                             Server IP Address (Network 2) True
server_addr3 none                             Server IP Address (Network 3) True
server_addr4 none                             Server IP Address (Network 4) True
# lsattr -El hdisk4
io_timeout   180                              I/O Timeout Interval          True
local_addr   192.169.200.20                   Local IP Address (Network 1)  True
local_addr2  none                             Local IP Address (Network 2)  True
local_addr3  none                             Local IP Address (Network 3)  True
local_addr4  none                             Local IP Address (Network 4)  True
pvid         00c937e038ac8e460000000000000000 Physical Volume Identifier    True
server_addr  192.169.200.30                   Server IP Address (Network 1) True
server_addr2 none                             Server IP Address (Network 2) True
server_addr3 none                             Server IP Address (Network 3) True
server_addr4 none                             Server IP Address (Network 4) True
```

Example 2-12 shows the RPV clients using **lsrpvclient** command

Example 2-12 Using lsrpvclient command

```
# lsrpvclient -H
# RPV Client     Physical Volume Identifier       Remote Site
# -----------------------------------------------------------
  hdisk3         00c937e038ac7fe5                 Valhalla
  hdisk4         00c937e038ac8e46                 Valhalla
```

2.2.4 Create the GMVGs

Now that the local and remote “physical” volumes are available, we can configure mirroring by creating the GMVGs using both a local and remote “physical” volume. As we will be configuring Asynchronous mode GLVM, we will be following the async mode prerequisite which requires the GMVG be created:

- As a scalable VG using mirror pools.
- Defined with a “superstrict” policy.
- Not to be automatically activated.

Example 2-13 shows the command line option.

Example 2-13 Create the volume group

```
# mkvg -f -S -M s -n -y glvm_vg hdisk1 hdisk2
```

This can also be done using SMIT:

```
smitty _mksvg
```

Example 2-14 shows the SMIT screen.

Example 2-14 Create Volume Group with local disk and RPV Client

```
                               Add a  Volume Group

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                       [Entry Fields]
  VOLUME GROUP name                                  [glvm1]
  Physical partition SIZE in megabytes                                       +
* PHYSICAL VOLUME names                              [hdisk1 hdisk3]         +
  Force the creation of a volume group?               no                     +
  Activate volume group AUTOMATICALLY                 no                     +
    at system restart?
  Volume Group MAJOR NUMBER                          [80]                    +#
  Create VG Concurrent Capable?                       no                     +
  Max PPs per VG in units of 1024                     32                     +
  Max Logical Volumes                                 256                    +
  Enable Strict Mirror Pools                          Superstrict            +
  Infinite Retry Option                               no                     +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Bad block relocation needs to be turned off for each GMVG (see Example 2-15 for the command).

Example 2-15 Turn off bad block relocation

```
# chvg -b n glvm1
```

Once each physical volume has been assigned to a VG, they can be added to their respective mirror pools as shown in Example 2-16.

Example 2-16 Add disks to their respective mirror pool

```
# chpv -p Asgard hdisk1
# chpv -p Asgard hdisk2
# chpv -p Valhalla hdisk3
# chpv -p Valhalla hdisk4
```

Example 2-17 shows the mirror pool details for one disk.

Example 2-17 Display mirror pool details

```
# lspv hdisk1
PHYSICAL VOLUME:    hdisk1                   VOLUME GROUP:     glvm1
PV IDENTIFIER:      00c9388038acab89 VG IDENTIFIER
00c937e000004b0000000189426f0b74
PV STATE:           active
STALE PARTITIONS:   0                        ALLOCATABLE:      yes
PP SIZE:            16 megabyte(s)           LOGICAL VOLUMES:  0
TOTAL PPs:          1274 (20384 megabytes)   VG DESCRIPTORS:   2
FREE PPs:           1274 (20384 megabytes)   HOT SPARE:        no
```

```
USED PPs:           0 (0 megabytes)          MAX REQUEST:      512 kilobytes
FREE DISTRIBUTION:  255..255..254..255..255
USED DISTRIBUTION:  00..00..00..00..00
MIRROR POOL:        Asgard                   ENCRYPTION:       no
```

The mirror pool can also be set using SMIT:

smit chpv → Enter physical volume name.

This is shown in Example 2-18.

Example 2-18 Setting the mirror pool by way of SMIT

```
                   Change Characteristics of a Physical Volume
Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                [Entry Fields]
* Physical volume NAME                           hdisk1
  Allow physical partition ALLOCATION?           yes                    +
  Physical volume STATE                          active                 +
  Set hotspare characteristics                   n                      +
  Set Mirror Pool                               [Asgard]                 +
  Change Mirror Pool Name                       []
  Remove From Mirror Pool                                               +



F1=Help                 F2=Refresh              F3=Cancel               F4=List
F5=Reset                F6=Command              F7=Edit                 F8=Image
F9=Shell                F10=Exit                Enter=Do
```

The resulting configuration is shown in Example 2-19.

Example 2-19 Displaying the mirror pool configuration

```
lsmp -A glvm1
VOLUME GROUP:       glvm1               Mirror Pool Super Strict: yes
MIRROR POOL:        Asgard              Mirroring Mode:          SYNC
MIRROR POOL:        Asgard              Mirroring Mode:          SYNC
```

In this example we are replicating file systems so we will need to create both a jfs2log logical volume and the jfs2 logical volume for the file system, each with a copy in both mirror pools. Inline logs can also be used.

2.2.5 Create the LVs and the file systems

The logical volumes are created using 2 copies, one in each mirror pool using:

- Superstrict allocation policy
- Passive MWC
- Bad block relocation turned off

Example 2-20 on page 17 shows the command line option to create the logical volumes.

Example 2-20 Create logical volumes for synchronous replication

```
# mklv -c 2 -t jfs2log -y ulv11 -p copy1=Asgard -p copy2=Valhalla -b n -w p -s s glvm1 1
# mklv -c 2 -t jfs2 -y ulv21 -p copy1=Asgard -p copy2=Valhalla -b n -w p -s s glvm1 60
```

This can also be done using SMIT:

smitty mklv → Enter the volume group name

Example 2-21 shows using a jfs2log logical volume rather than inline logs.

Example 2-21 Create the logical volume using SMIT

```
                               Add a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                  [Entry Fields]
  Logical volume NAME                                 [ulv11]
* VOLUME GROUP name                                    glvm1
* Number of LOGICAL PARTITIONS                        [1]                   #
  PHYSICAL VOLUME names                               [hdisk1 hdisk32]      +
  Logical volume TYPE                                 [jfs2log]            +
  POSITION on physical volume                          middle              +
  RANGE of physical volumes                            minimum             +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                  []                    #
    to use for allocation
  Number of COPIES of each logical                     2                   +
    partition
  Mirror Write Consistency?                            passive             +
  Allocate each logical partition copy                 superstrict         +
    on a SEPARATE physical volume?
  RELOCATE the logical volume during                   yes                 +
    reorganization?
  Logical volume LABEL                                [ulv11]
  MAXIMUM NUMBER of LOGICAL PARTITIONS                [512]                 #
  Enable BAD BLOCK relocation?                         no                  +
  SCHEDULING POLICY for writing/reading                parallel            +
    logical partition copies
  Enable WRITE VERIFY?                                 no                  +
  File containing ALLOCATION MAP                      []
  Stripe Size?                                        [Not Striped]        +
  Serialize IO?                                        no                  +
 Mirror Pool for First Copy                           Asgard              +
 Mirror Pool for Second Copy                          Valhalla            +1
 Mirror Pool for Third Copy                                               +
  Infinite Retry Option                                no                  +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Similarly create a logical volume for the JFS2 file system data and format the jfslog logical volumes. Then create the file system using the jfslog and data logical volumes.

Repeat the above steps for the remaining two disks (hdisk2 and hdisk4) then create glvm2, its logical volumes and file system.

2.2.6 Create the cache logical volumes

If configuring asynchronous GLVM, create the cache logical volume (using type **aio_cache**), one for each site.

For asynchronous replication the updates to be mirrored to the Asgard site will be stored in the cache in the Valhalla site (Valhalla mirror pool) and when mirroring is being done from the Asgard site the Asgard mirror pool will be used. This can be defined using the command line as shown in Example 2-22.

Example 2-22 Create the aio_cache logical volume for each site

```
# mklv -c 1 -t aio_cache -y glvm1_val_ca -p copy1=Asgard -b n -w p glvm1 5
# mklv -c 1 -t aio_cache -y glvm1_asg_ca -p copy1=Valhalla -b n -w p glvm1 5
```

You can also create the cache for Asgard in Valhalla mirror pool using SMIT:

smitty mklv → Enter the volume group name.

See Example 2-23.

Example 2-23 Creating the aio_cache logical volume for Asgard in the Valhalla mirror pool

```
                               Add a Logical Volume

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

[TOP]                                                   [Entry Fields]
  Logical volume NAME                                [glvm1_asg_ca]
* VOLUME GROUP name                                   glvm1
* Number of LOGICAL PARTITIONS                       [5]                        #
  PHYSICAL VOLUME names                              [hdisk2]                  +
  Logical volume TYPE                                [aio_cache]               +
  POSITION on physical volume                         middle                   +
  RANGE of physical volumes                           minimum                  +
  MAXIMUM NUMBER of PHYSICAL VOLUMES                 []                         #
    to use for allocation
  Number of COPIES of each logical                    2                        +
    partition
  Mirror Write Consistency?                           passive                  +
  Allocate each logical partition copy                yes                      +
    on a SEPARATE physical volume?
  RELOCATE the logical volume during                  yes                      +
    reorganization?
  Logical volume LABEL                               [glvm1_asg_ca]
  MAXIMUM NUMBER of LOGICAL PARTITIONS               [512]                      #
  Enable BAD BLOCK relocation?                        no                       +
  SCHEDULING POLICY for writing/reading               parallel                 +
    logical partition copies
  Enable WRITE VERIFY?                                no                       +
  File containing ALLOCATION MAP                     []
  Stripe Size?                                       [Not Striped]             +
  Serialize IO?                                       no                       +
  Mirror Pool for First Copy                          Vallhala            +
  Mirror Pool for Second Copy                                             +
  Mirror Pool for Third Copy                                              +
  Infinite Retry Option                               no                  +

F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

If using SMIT, then create Valhalla’s aio_cache logical volume in mirror pool Asgard.

The resulting configuration can be shown with **lsvg** as in Example 2-24.

Example 2-24 GMVG logical volumes

```
# lsvg -l glvm1
glvm1:
LV NAME             TYPE       LPs     PPs     PVs  LV STATE      MOUNT POINT
ulv11               jfs2log    1       2       2    closed/syncd  N/A
ulv12               jfs2       60      120     2    open/syncd    /glvm1
glvm1_val_ca        aio_cache  5       5       1    closed/syncd  N/A
glvm1_asg_ca        aio_cache  5       5       1    closed/syncd  N/A
```

Before any change is made to the flow of replication, all activity on the GMVGs must be quiesced, that is all applications stopped, the file systems unmounted and the volume groups varied off.

Note: In the above scenario, there was no I/O activity in the geographically mirrored file systems, so there was no need to change the preferred read policy. This must be set before mounting any file systems and commencing any I/O in production. See 2.2.7, “Stop activity on first server in the primary data center” on page 19

2.2.7 Stop activity on first server in the primary data center

Since the first site shares the replicated LUNs between two servers, the RPV clients need to be defined on both servers. As the RPV Servers in Valhalla point to the Asgard alias address, no further changes need to be made there.

Before making any change, perform the following tasks:

- Stop any activity in the GMVG file systems.
- Unmount the GMVG file systems.

  ```
  umount /glvm1
  umount /glvm2
  ```

- Deactivate the GMVGs.

  ```
  varyoffvg glvm1
  varyoffvg glvm2
  ```

- Stop the RPV Clients.
  - Run **rmdev -l hdisk3** to put hdisk3 in a Defined state.
  - Run **rmdev -l hdisk4** to put hdisk4 in a Defined state.

2.2.8 Configure the RPV clients on the second server in the primary data center

The first step will be to move the primary data center alias to the second server and then, as with the first server, create the two RPV clients pointing to the RPV servers in the secondary data center as shown in Example 2-9 on page 13.

Once the hdisks are available, import the volume groups – do not forget to define the major numbers for consistency, for example:

```
importvg -y glvm1 -V 80 hdisk2
```

```
importvg -y glvm2 -V 90 hdisk1
```

2.2.9 Stop activity on the second server in the primary site

The system has now been configured for I/O replication from Asgard to Valhalla. The next step is to configure the RPV servers and clients for the replication from Valhalla to Asgard.

The same steps as for the first server should be followed:

- Stop any activity in the GMVG file systems.
- Unmount the GMVG file systems.
  ```
  umount /glvm1
  umount /glvm2
  ```
- Deactivate the GMVGs.

  ```
  varyoffvg glvm1
  varyoffvg glvm2
  ```
- Stop the RPV clients.
  - Run **rmdev -l hdisk3** to put hdisk3 in a Defined state.
  - Run **rmdev -l hdisk4** to put hdisk4 in a Defined state.

2.2.10 Stop the RPV servers at the secondary data center

Once activity has stopped in the primary data center, the RPV servers in the secondary data center can be stopped, See Example 2-25

Example 2-25 Stopping RPV servers

```
# rmdev -l rpvserver0
rpvserver0 Defined
# rmdev -l rpvserver1
rpvserver1 Defined
```

2.2.11 Create the RPV servers on the first server in the primary data center

As shown in the previous sections, the RPV Servers can be created from the command line (see Example 2-26) or using **SMIT** – similar to what is shown in Example 2-6 on page 12.

Example 2-26 Creating RPV Server

```
# mkdev -c rpvserver -s rpvserver -t rpvstype -a \
rpvs_pvid=00c9388038acc305 00c9388038acab89 -a client_addr='192.168.200.30' \
-a auto_online='n'
```

```
rpvserver0 Available
rpvserver1 Available
```

2.2.12 Create the RPV clients on the server in the secondary data center

Again this can be performed using the command line, see Example 2-9 on page 13 or using **SMIT**

Example 2-27 Creating RPV Client

```
# mkdev -c disk -s remote_disk -t rpvclient -a pvid=’00c9388038acc305000000
00c9388038acab890000000, \
-a server_addr='192.168.200.20' -a local_addr='192.168.200.30' \
-a io_timeout='180'
hdisk3 Available
hdisk4 Available
```

2.2.13 Import the GMVGs on the server in the secondary data center

We now have all the LUNs available on the server in the secondary data center for the GMVGs, so their definitions can be imported.

2.2.14 Stop activity on the server at the in the secondary data center

For availability, the second server in the primary data center may need to run the RPV servers, so deactivate the configuration in the secondary data center before creating the RPV server definitions on the second server by:

- Stopping any activity in the GMVG file systems.
- Unmount the GMVG file systems.
- Deactivate the GMVGs.

  varyoffvg glvm1
  varyoffvg glvm2
- Stop the RPV clients.
  - Run **rmdev -l hdisk3** to put hdisk3 in a Defined state.
  - Run **rmdev -l hdisk4** to put hdisk4 in a Defined state.

2.2.15 Create the RPV servers on the second server in the primary data center

Before creating the RPV servers on the second server using the shared LUNs, the RPV servers on the first server need to be stopped as shown in Example 2-28 and the GLVM alias moved to the second server.

Example 2-28 Stop the RPV servers

```
# rmdev -l rpvserver0
rpvserver0 Defined
# rmdev -l rpvserver1
rpvserver1 Defined
```

As above the RPV Servers can be created from the command line – see Example 2-29 – or using **SMIT**, similar to what is shown in Example 2-6 on page 12.

Example 2-29   Creating RPV server

```
# mkdev -c rpvserver -s rpvserver -t rpvstype -a \
rpvs_pvid=00c9388038acc305 00c9388038acab89 -a client_addr='192.168.200.30' \
-a auto_online='n'
rpvserver0 Available
rpvserver1 Available
```

2.2.16 Start the RPV clients in the secondary data center

Restart the RPV clients in the secondary data center as in Example 2-30 and activate the volume groups.

Example 2-30   Restart the RPV clients using the RPV servers from the 2nd Server

```
# mkdev -l hdisk3
hdisk3 Available
# mkdev -l hdisk4
hdisk4 Available
# varyonvg glvm1
# varyonvg glvm2
```

If the two GMVGs are activated successfully, the configuration of the RPV servers on the second server in the primary data center is correct.

2.2.17 Stop the RPV clients and then the RPV servers

If replication is to be resumed from the primary data center, the current replication will need to be gracefully stopped and the RPV server/client pairs started up in the opposite direction.

2.2.18 Setting preferred read

AIX LVM has a setting (-R) for controlling the preferred copy to read. Obviously for GLVM configurations, the setting of the preferred read is critical to performance as you do not want each read to go to the remote site. Each LV must have the preferred read set to the local mirror pool before the volume group is activated. To set the preferred read copy run **chlv -R <mirror copy> logical volume.**

For example if active at the Valhalla site, run **chlv -R 2 ulv11.**

The resulting configuration is shown in Example 2-31.

Example 2-31   Showing the preferred read setting when Valhalla is active

```
# lslv  ulv11
LOGICAL VOLUME:     ulv11                  VOLUME GROUP:   glvm1
..<snip>..
INFINITE RETRY:     no                     PREFERRED READ: 2
DEVICESUBTYPE:      DS_LVZ
COPY 1 MIRROR POOL: Asgard
COPY 2 MIRROR POOL: Valhalla
COPY 3 MIRROR POOL: None
```

```
ENCRYPTION:        no
```

When the preferred read is set, all reads will be made to the preferred pool if it is available. If the preferred pool is not available, reads will follow the scheduling policy (parallel, parallel write with sequential read, parallel write with round robin read, sequential). In the typical GLVM configuration, that will mean a read from the remote copy.

2.2.19 Verification of RPV client with respect to GLVM

After initial setup, ensure that the PV state is active and network connectivity is correct. To check the GLVM configuration use the commands shown in Example 2-32.

Example 2-32 Running verification of GLVM configuration

```
# lsglvm -c
Checking Volume Group glvm2
#   Site      Copy Physical Volumes
#Asgard         PV1 hdisk2
Valhalla       PV2 hdisk4
Checking Logical Volume ulv21
Checking Logical Volume ulv12
Checking Logical Volume glvm2_val_ca
Checking Logical Volume glvm2_asg_ca
Checking Volume Group glvm1
#   Site      Copy Physical Volumes
#Asgard         PV1 hdisk1
Valhalla       PV2 hdisk3
Checking Logical Volume ulv11
Checking Logical Volume ulv12
Checking Logical Volume glvm1_val_ca
Checking Logical Volume glvm1_asg_ca
```

This can also be done using SMIT:

```
smit glvm_utils
 → Geographically Mirrored Volume Groups.
 → Verify Mirror Copy Site Locations for a Volume Group.
 → Choose the Volume Group.
```

2.2.20 Changing GLVM mirroring modes

GLVM mirroring modes can be easily changed as long as the requirements for asynchronous configuration are met.

Changing mode from synchronous to asynchronous

Assuming that the aio_cache has been created for each mirror pool, all that is required is to change the property of the mirror pool to asynchronous.

> **Important:** Asynchronous mirror pools have one additional property, that is the High Water Mark. This variable sets the percentage of the cache that can be used before the cache is “full”. Remember that once the cache is “full”, new write requests will be synchronous until space is cleared in the cache.

To change the Mirror Pool:

```
chmp -A  -m Asgard -c glvm1_asg_ca -h 75 glvm1
chmp -A  -m Valhalla -c glvm1_val_ca -h 75 glvm1
```

This can also be done using SMIT:

```
smit glvm_utils
 → Geographically Mirrored Volume Groups.
 → Manage Geographically Mirrored Volume Groups with Mirror Pools.
 → Configure Mirroring Properties of a Mirror Pool.
 → Convert to Asynchronous Mirroring for a Mirror Pool.
```

Select:

- The mirror pool.
- The LV cache.

Set:

- The high water mark for the cache (%).
  TIP: SMIT requires the leading 0 (3 digits).

Repeat for the other Mirror Pool.

Listing the status of the glvm1 volume group now shows it as asynchronous as shown in Example 2-33.

Example 2-33 Display the asynchronous configuration for one GMVG

```
# lsmp -AL glvm1
VOLUME GROUP:       glvm_vg            Mirror Pool Super Strict: yes

MIRROR POOL:        Asgard             Mirroring Mode:          ASYNC
ASYNC MIRROR STATE: inactive           ASYNC CACHE LV:          glvm1_asg_ca
ASYNC CACHE VALID:  yes                ASYNC CACHE EMPTY:       yes
ASYNC CACHE HWM:    75                 ASYNC DATA DIVERGED:     no

MIRROR POOL:        Valhalla           Mirroring Mode:          ASYNC
ASYNC MIRROR STATE: active             ASYNC CACHE LV:          glvm1_val_ca
ASYNC CACHE VALID:  yes                ASYNC CACHE EMPTY:       no
ASYNC CACHE HWM:    75                 ASYNC DATA DIVERGED:     no
```

Repeat these steps for glvm2.

Changing asynchronous to synchronous

To change GLVM operation to synchronous mode:

```
chmp -S  -m Asgard glvm1
chmp -S  -m Valhalla glvm1
```

Repeat for glvm2.

2.3 Useful lsglvm options

The following examples show useful **lsglvm** options to display GMVG and mirror pool status.

lsglvm

Example 2-34 shows the output from **lsglvm** with no flags and displaying the remote PV details.

Example 2-34 lsglvm output

```
# lsglvm
#Volume Group   Logical Volume    RPV          PVID          Site
glvm1           glvm1_asg_ca    hdisk3       00c937e038ac7fe5   Valhalla
glvm1           ulv11           hdisk3       00c937e038ac7fe5   Valhalla
glvm1           ulv12           hdisk3       00c937e038ac7fe5   Valhalla
glvm2           glvm2_asg_ca    hdisk4       00c937e038ac8e46   Valhalla
glvm2           ulv21           hdisk4       00c937e038ac8e46   Valhalla
glvm2           ulv22           hdisk4       00c937e038ac8e46   Valhalla
```

lsglvm -p

Example 2-35 shows the **lsglvm** output with the mirror pool details (**-p** flag).

Example 2-35 lsglvm showing mirror pool details for remote PV (async and sync)

```
# lsglvm -p
glvm1: (Asynchronously mirrored)
# Logical Volume  RPV          PVID                Site          Mirror Pool
glvm1_asg_ca      hdisk3       00c937e038ac7fe5    Valhalla      Valhalla
ulv11             hdisk3       00c937e038ac7fe5    Valhalla      Valhalla
ulv12             hdisk3       00c937e038ac7fe5    Valhalla      Valhalla

glvm2: (Synchronously mirrored)
# Logical Volume  RPV          PVID                Site          Mirror Pool
glvm2_asg_ca      hdisk4       00c937e038ac8e46    Valhalla      Valhalla
ulv21             hdisk4       00c937e038ac8e46    Valhalla      Valhalla
ulv22             hdisk4       00c937e038ac8e46    Valhalla      Valhalla
```

lsglvm -m

Example 2-36 shows **lsglvm** with site and PV mapping for each LV.

Example 2-36 lsglvm with mapping for each LV

```
# lsglvm -m
# Table of All Physical Volumes in all Geographic Logical Volumes
#   Site      Copy Physical Volumes
glvm1
ulv11
Asgard         PV1 hdisk1
Valhalla       PV2 hdisk3
ulv12
Asgard         PV1 hdisk1
Valhalla       PV2 hdisk3
glvm1_val_ca
Asgard         PV1 hdisk1
glvm1_asg_ca
Valhalla       PV1 hdisk3
glvm2
ulv21
Asgard         PV1 hdisk2
```

```
Valhalla       PV2 hdisk4
ulv22
Asgard         PV1 hdisk2
Valhalla       PV2 hdisk4
glvm2_val_ca
Asgard         PV1 hdisk2
glvm2_asg_ca
Valhalla       PV1 hdisk4
```

3

Planning, sizing and tuning

This chapter provides guidance on planning, sizing and tuning your GLVM environment. The important components for planning and sizing the implementation are:

– Understanding the prerequisites for GLVM, independent of whether you are using it for migration from on-premises to cloud or for DR within the cloud.

– Having a good understanding of your application’s I/O profile and the underlying file system structure.

– Ensuring the network is stable and provides sufficient bandwidth to meet the application workload.

The following topics are covered in this chapter:

- “General Planning and Tuning Guidance”
- “GLVM and AIX requirements and limitations”
- “Additional limitations when using GLVM”
- “Enabling compression”
- “General recommendations”
- “Planning CPU, memory and network”
- “Further tuning tips”
- “GLVM tuning options”
- “Tuning summary”
- “Integration with Power HA SystemMirror Enterprise Edition”

3.1 General Planning and Tuning Guidance

The following is a summary of the recommendations taken from *Asynchronous Geographic Logical Volume Mirroring Best Practices for Cloud Deployment*, REDP-5665. For additional details you can refer to the Redpaper. It should also be noted that many of these recommendations are generic, each application and resulting setup will differ.

The monitoring recommendations will help the reader with both sizing the original configuration as well as assist in recognizing what needs to be changed over time as workloads and applications evolve. The most common change we have observed is that the application throughput increases leading to problems caused by insufficient network bandwidth.

When planning your implementation, it is strongly recommended that the current systems I/O be monitored at the very least for 1 week, but preferably for one application cycle (for example a month). Also record peak CPU, memory and network usage. Section 5.1, “Monitoring” on page 48 discusses some of the tools that can be used for both the initial planning as well as the ongoing monitoring.

If you are using GLVM for a production disaster recovery solution and not just doing a migration, it is highly recommended that you use PowerHA SystemMirror to help in the configuration and management of your environment. PowerHA integrates management of GLVM and simplifies the daily management tasks required to keep your environment running. For details on the use of PowerHA see section 3.10, “Integration with Power HA SystemMirror Enterprise Edition” on page 40,

3.2 GLVM and AIX requirements and limitations

Consider the following requirements and limitations when implementing GLVM in an AIX environment:

- Only two sites are supported. AIX LVM supports 3 copies in a logical volume, so a maximum of two copies can be at one site with a third copy at a second site.
- As with standard AIX LVM mirroring there is no requirement on the type and size of the LUNs that make up the GMVG, as long as the LUNs are supported by AIX and there is sufficient space in the remote LUNs for a copy of each Logical Volume and the cache.
- Volume Groups must be configured as scalable Volume Groups.
- The rootvg cannot be geographically mirrored.
- The inter-disk allocation policy must be set to superstrict.
- Mirror pools are required for asynchronous replication and they are also recommended for synchronous replication.
- Asynchronous mirrored volume groups cannot contain an active paging space, and it is recommended that synchronous mirrored VGs do not either.
- Synchronous mode VGs can be configured as enhanced concurrent mode, but this is not required.
- Split mirror function cannot be run on asynchronous mode VGs.
- GMVGs should not be configured to activate automatically.
- Bad block relocation must be turned off for asynchronous mode VGs and for each logical volume.
- GLVM site names must match PowerHA site names when integrated with PowerHA.
- IPSec can be configured to secure the RPV Client/Server traffic.
- 1MB space required in /usr to install.
- ICMP and port 6192 (both TCP and UDP) must be open between sites.

Note: Some of these requirements will necessitate a short outage if being implemented on an existing system, so must be planned for.

3.3 Additional limitations when using GLVM

GLVM imposes the following limitations:

- AIX Live Kernel update cannot be performed with asynchronous GMVGs. They must first be converted to synchronous mode before running the update.
- The volume group cannot be a snapshot volume group.
- An aio_cache logical volume cannot be removed or reduced if the GMVG is configured in asynchronous mode. It must first be converted to synchronous mode before any change and then converted back to asynchronous mode.
- Concurrent access is not supported.

3.4 Enabling compression

In Power7+ IBM introduced an acceleration unit for cryptography and Active Memory Expansion (AME). AME was introduced to allow a section of the LPAR’s memory to be compressed to allow the OS to access more memory than is physically available, and the NX842 acceleration unit took the load from the CPU. With GLVM, the use of the accelerator unit enables compression of network packets in both directions with little or no performance impact, but requires the installation of the AME license. To enable AME, enter the activation code on the hardware management console (HMC).

Note: AME is activated on all PowerVS servers, so this step is not required in the IBM Cloud.

To use the compression tunable parameter, ensure that the following prerequisites are met:

- Using Power7+ and above as mentioned above.
- The RPV client and the RPV server are running AIX version 7.2.5, or later with all the latest RPV device drivers.
- The RPV server and the RPV client are IBM Power Systems servers with NX842 acceleration units. If either of the client or server do not have the accelerator unit, there will be a performance impact.
- The AME activation code for the server has been entered.
- The compression tunable parameter is enabled on both the primary and DR servers, so that hardware compression will be used for compressing packets in both directions.

3.5 General recommendations

The following recommendations apply to both synchronous and asynchronous configurations

- GLVM is a clustered environment – ensure that configurations, settings and important system files are consistent across sites.
- GLVM requires mirror write consistency (MWC) to be set to passive.

  When MWC is set to passive, the volume group logs when the logical volume has been opened. After a crash when the volume group is varied on, an automatic force sync of the logical volume is started. This means a full sync of the mirrored volume group to the DR site will occur – this could add a significant amount of time to the recovery, depending on the size of the volume group and the network speed. For more information on MWC refer to this IBM documentation topic.

- Do not set file systems to mount automatically.
- Do not set the RPV servers or clients to start automatically.
- Use hardware compression of network packets.
- Update change control procedures to ensure that both sites are kept consistent.
- Set AIX mirror write consistency (MWC) to passive.
- Set the RPV I/O timeout sufficient to avoid any issues with network speed or dropouts. Do not set too low, as there must be sufficient time for the data in the cache to be synchronized.

  We recommend not setting the timeout to be less than twice the average round trip time for a packet. Tuning will depend on the network reliability and latency, with the time being long enough to allow for cache synchronization and for GLVM not to respond to a network interruption, while not being so long so that GLVM does not respond to a real failure.

  This value can only be changed when the disk is in a defined state. The default value is 180 seconds.
- Set the read preference to the local mirror pool for stand-alone GLVM, PowerHA will manage with site affinity setting. This must be set prior to activating the VG.
- Turn off quorum for each GMVG.
- For on premise workloads follow the AIX / Storage vendor’s recommendations for storage tuning variables (for example: the disk *queue_depth*, the adapter *num_cmd_elems*).
- If your on-premises server has more than 900 disks being mirrored by GLVM, increase the *lg_term_dma* for each fiber adapter to 0x8000000.

3.5.1 Asynchronous recommendations

These recommendations apply to asynchronous mode GMVGs.

- Plan the cache size carefully (see section 3.7.3, “Planning the cache” on page 37 for more details).
- A number of important improvements were introduced in AIX 7.2.5.

> **Note:** IBM strongly recommends using AIX at 7.2.5 or above to be able to take advantage of the new features at that level.

- It is recommended to configure the LVM asynchronous cache I/O physical buffer pool and the volume group physical buffer pool to improve performance and avoid I/O hangs. Each logical volume write can be divided into multiple remote physical I/Os. These I/Os are based on the application I/O size and the LVM LTG size because each remote physical write must perform the cache-logical volume write. Therefore, tune the aio_cache_pbuf_count by increasing it to be slightly greater than the expected maximum total parallel remote writes. Use the lvmo command as:

  ```
  lvmo -v <GMVG> -o aio_cache_pbuf_count = <new value>
  ```

  Also monitor with **`vmstat -v`** for increasing number of blocked I/Os (see below).
- Asynchronous GLVM has a restriction on the number of RPV Clients supported per LPAR. See Table 3-1 on page 31 for details.

Table 3-1 Max number of RPV Clients

Networks per RPV client	Max number of RPV clients
1	1020
2	510
3	340
4	255

Note: This table is only included for completeness as it is recommended to use network availability either through the VIO Server layer or by using Etherchannel rather than configuring multiple GLVM networks. Both those options have a shorter and more efficient code path for handling failover.

3.6 Planning CPU, memory and network

The following sections cover the sizing requirements for your solution with our recommendations for CPU, memory and network connectivity.

3.6.1 CPU

Typically we recommend configuring the CPU entitlement for the 95% of the analyzed usage. For GLVM it is recommended:

- if LPAR is less than 1 core, add 0.25 core.
- if LPAR is greater than 1 core, add 0.5 core.

However, these are rule of thumb recommendations and the CPU usage, particularly the percentage of entitled capacity that is consumed (*%entc)* should be monitored using the tools covered in this publication.

3.6.2 Memory

In our testing we did not observe any significant increase in memory usage due to GLVM, so tune for your application requirements, not GLVM. However GLVM does use pinned memory, so a minimum of 8GB is recommended.

3.6.3 Networks

The sizing of the network bandwidth is critical to the operation of GLVM and an understanding of the application's I/O profile is also required. Any peaks in I/O greater than the network bandwidth, will mean that more data is being written to the cache, than can be cleared. If these peaks are sustained for any significant period, the cache may fill and GLVM will revert to synchronous mode until space in the cache can be freed. Thus the network bandwidth must be sufficient to transfer all the data in the largest I/O peak without the cache filling.

Other network considerations:

- It is critical to the operation of GLVM that the network is reliable and stable. Lost packets and connectivity issues can lead to timeouts and the RPV server declared unavailable.

- Multiple network paths and Etherchannel / VIO Server redundancy is recommended.
- Firewall changes may impact the operation of GLVM.

Table 3-2 can be used as guidance for planning for network requirements across sites for Asynchronous GLVM.

> **Note:** These requirements are minimal and do not take into consideration I/O Peaks. Customers must review workload requirements and plan accordingly.

Table 3-2 Network sizing guidelines

Data change rate per day	Network speed and bandwidth requirements
Less than 1 TB	1 Gb/sec or higher
1 to 10 TB	5 Gb/sec or higher
10 TB or higher	10 Gb/sec or higher

3.6.4 PowerVS network connectivity

To mirror from your on-premises location to PowerVS or to mirror between PowerVS data centers require that you define networking connections in the IBM Cloud. IBM offers both Cloud Direct Link Dedicated and Cloud Direct Link Connect between your data center and IBM Power Cloud[1], using either dedicated fiber link, or connectivity through a service provider. Within the IBM Cloud, IBM uses Direct Link (2.0) Connect offering 50, 100, 200, 500 Mbps, and 1, 2, 5, and 10 Gbps. Also 25, 40, 50, and 100 Gbps are available using Partner interconnects or network to network interfaces[2].

For further information, IBM cloud documents have detailed information about direct links[3].

3.6.5 Network tuning

Should extra bandwidth be required, AIX and the IBM PowerVM® layer support Etherchannel to combine networks, If network reliability is a concern, it is recommended to have a minimum of 2 separate networks, using different providers, if possible, with different hardware and different physical paths.

It is also recommended to follow AIX network tuning good practice and the tuning recommendations appropriate for your network / adapter etc, in Table 3-3.

Table 3-3 Specific network tuning recommendations

Variable	Recommended value
rfc1323	1
tcp_sendspace	2MB
tcp_recvspace	2MB
udp_sendspace	1MB

[1] https://cloud.ibm.com/docs/dl?topic=dl-get-started-with-ibm-cloud-dl
[2] https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-ordering-direct-link-connect
[3] https://cloud.ibm.com/docs/dl?topic=dl-dl-about

Variable	Recommended value
udp_recvspace	1MB
sb_max	4MB
mtu	9000
mtu_bypass	on
jumbo_frame	enabled
flow_ctrl	on
chksum_offload	on
large_send	on
large_receive	on
tcp_nodelayack	on
sack	1

For more information there is a good update on the IBM Support pages by Darshan Patel. While it covers more generic tuning for WAN networks on AIX, it does give a good background for each of the important tunables[4].

It should be noted that some of these tunables can be set per interface, for example *tcp_recvspace*, however the *sb_max* should be set to roughly two times this value and has to be set system wide - discussed further below.

The **no** command is used to view change these settings, with the exception of the first 5 variables which should be set on the actual interface. Example 3-1 shows the use of the **no** command.

Example 3-1 The use of no command to display a setting

```
# no -o sack
sack = 0
```

Example 3-2 shows the use of the **no** command to get additional details about a network setting.

Example 3-2 The use of no command to show options for a setting

```
# no -L sack
----------------------------------------------------------------------------------------
NAME                      CUR    DEF    BOOT   LVUP   MIN    MAX    UNIT           TYPE
     DEPENDENCIES
----------------------------------------------------------------------------------------
sack                      0      0      0      0      0      1      boolean            C
----------------------------------------------------------------------------------------
```

Example 3-3 on page 34 shows changing a setting and making that value stay through the next reboot.

[4] https://www.ibm.com/support/pages/what-basic-tcp-tunings-are-recommended-improve-performance-wan-connections-between-aix-virtual-machines

Example 3-3 The use of no command to change a setting now and for next reboot

```
# no -p -o sack=1
Setting sack to 1
Setting sack to 1 in nextboot file
Change to tunable sack, will only be effective for future connections
```

We have seen many tuning guides recommending that the TCP and UDP send and receive spaces are configured through the **no** command. It is recommend to set *rfc1323* and these buffers at the interface level for the required adapters, using the **chdev** command and ensure that the **no** option *use_isno* is set (default is on). See Example 3-4 for an example.

Example 3-4 The use of chdev command to change an interface

```
# chdev -l  en0 -a tcp_sendspace=462144
en0 changed
Host:/# lsattr -El en0|grep -E "rfc|space"
rfc1323                       Enable/Disable TCP RFC 1323 Window Scaling    True
tcp_recvspace                 Set Socket Buffer Space for Receiving         True
tcp_sendspace 462144          Set Socket Buffer Space for Sending           True
```

Good practice recommends that one should only increase many of the above tunables gradually and test after each change, rather than just applying the maximum value.

Note: For our PowerVS test, a 5GB/sec transit gateway was used between the two data centers. Doing a load test, values for *tcp_sendspace* and *tcp_recvspace* up to 50MB were tried, However we found that for day-to-day operations, 2-5MB was sufficient.

Important: Remember to stop and start the *inetd* daemon after changing any of the buffer sizes for any of the adapters in order for the changes to take effect.

3.7 Further tuning tips

Regularly monitor using **netstat -v** and look for:

- Packets dropped increasing.
- No Resource errors.
- Hypervisor receive failures.

If any of these values are steadily increasing, use **netstat -v** to check the Tiny, Small, Medium, Large, Huge buffer usage. If the maximum allocated for any buffer is equal to the number of buffers, then we recommend doubling the number of buffers of that type and monitoring.

Some administrators recommend increasing the minimum to equal the maximum, as this will save time spent in the allocation of new buffers if required. However, we prefer to set the minimum 5 or so below the maximum, as it is then possible to tell if all the buffers of that type are being used and therefore the maximum needs to be further increased. Example 3-5 on page 35 shows the use of the **netstat** command.

Example 3-5 The output of the netstat command to display packet information and buffer usage

```
# netstat -v
..<snip> ..
ETHERNET STATISTICS (ent4) :
Device Type: Virtual I/O Ethernet Adapter (l-lan)
Hardware Address: fa:16:3e:e8:4d:83
Elapsed Time: 13 days 21 hours 55 minutes 10 seconds

Transmit Statistics:                          Receive Statistics:
--------------------                          -------------------
Packets: 5800                                 Packets: 5710
Bytes: 370440                                 Bytes: 495732
Interrupts: 0                                 Interrupts: 5710
Transmit Errors: 0                            Receive Errors: 0
Packets Dropped: 0                            Packets Dropped: 0
                                              Bad Packets: 0
Max Packets on S/W Transmit Queue: 0
S/W Transmit Queue Overflow: 0
Current S/W+H/W Transmit Queue Length: 0

Broadcast Packets: 107                        Broadcast Packets: 5336
Multicast Packets: 2                          Multicast Packets: 0
No Carrier Sense: 0                           CRC Errors: 0
DMA Underrun: 0                               DMA Overrun: 0
Lost CTS Errors: 0                            Alignment Errors: 0
Max Collision Errors: 0                       No Resource Errors: 0
Late Collision Errors: 0                      Receive Collision Errors: 0
Deferred: 0                                   Packet Too Short Errors: 0
SQE Test: 0                                   Packet Too Long Errors: 0
Timeout Errors: 0                             Packets Discarded by Adapter: 0
Single Collision Count: 0                     Receiver Start Count: 0
Multiple Collision Count: 0
Current HW Transmit Queue Length: 0

General Statistics:
-------------------
No mbuf Errors: 0

..<snip>..IBM PowerHA SystemMirror for AIX Cookbook, SG24-7739
IBM PowerHA SystemMirror for AIX Cookbook, SG24-7739
IBM PowerHA SystemMirror for AIX Cookbook, SG24-7739
IBM PowerHA SystemMirror for AIX Cookbook, SG24-7739


Hypervisor Send Failures: 0
  Receiver Failures: 0
  Send Errors: 0
Hypervisor Receive Failures: 0

..<snip>..

Receive Informatio Receive Buffers
    Buffer Type              Tiny    Small   Medium    Large     Huge
    Buffer Size               512     2048    16384    32768    65536
    Low threshold               0        0        0        0        0
    Pool low mark             512      512      128       32       32
    Min Buffers              2044     2044      252       62       62
    Max Buffers              2048     2048      256       64       64
    Allocated                2046     2044      252       62       62
    Registered               2046     2044      252       62       62
    Mapped                   2046     2044      252       62        62
    History
      Max Allocated          2046     2044      252       62        62
      Lowest Registered      2044     2044      252       61        62
      Low threshold drops       0        0        0        0         0

..<snip>..
```

3.7.1 Storage and file system planning

There are not many requirements for planning storage, other than ensuring that there is sufficient storage on the remote site to allow for a copy of all the local logical volumes.

However there is one important consideration, if using Power in the IBM Cloud. IBM throttles the I/O by limiting the number of IOPs per GB of the LUN, The amount depending on the tier of the storage:

- Tier 3 allows 3 IOPs/GB
- Tier 1 allows 10 IOPs/GB

Thus you may need to size the LUNs based on the expected IOPs, not just there size of the local LUNs.

While it is important to plan around GLVM and the network, it is also important to examine application I/O and the usage of the replicated file systems. Asynchronous GLVM, from the application perspective, is not granular – there is one cache per volume group. It is therefore important to plan the mix of file systems and/or processes/applications that are sharing the volume group as their I/O will be sharing the same cache.

If the application uses a JFS2 file system in a cached I/O mode, the file VMM cache can use up most of the memory (90% by default). Check that the system has enough memory to handle system wide operations other than the file VMM cache. It is often recommended to have 4 - 6 GB memory outside the file VMM cache. Other methods can be used to reduce the memory footprint used by the file VMM cache if required[5].

If the application is not accessing the same VMM cached data multiple times (data needed only once or re-accessed after long time), the JFS2 filesystem can be mounted with the release behind option enabled. This will releases pages after a large sequential read or write operation.

To set on an unmounted file system (/fs):

```
mount -o rbr,rbw /fs
```

To set while a file system is mounted (/fs):

```
mount -o remount,rbrw /fs
```

To have the setting persist across reboots and exportvg/importvg (/fs):

```
chfs -a “options=-o rbrw” /fs
```

Advanced AIX users will be able to further modify AIX’s use of the file cache with support from the IBM Support Centre.

[5] https://www.ibm.com/docs/en/aix/7.2?topic=performance-vmm-page-replacement-tuning

3.7.2 Tuning using vmstat

Regularly monitor the system using **vmstat -v** and check for any steady increase in I/Os blocked by a shortage of resources. If this is observed, it is recommended to double the size of the relevant buffer and monitor. For example if disk I/Os blocked with no *pbuf* steadily increases between observations, then increase the **pv_buf_count**. Repeat this process until the number of blocked I/Os is stable. Example 3-6 shows the use of the **vmstat** command.

Example 3-6 The use of the **vmstat** *command*

```
Host:/# vmstat -v
<snip>
                 32 pending disk I/Os blocked with no pbuf
                  0 paging space I/Os blocked with no psbuf
               1912 filesystem I/Os blocked with no fsbuf
                  0 client filesystem I/Os blocked with no fsbuf
                304 external pager filesystem I/Os blocked with no fsbuf
               45.0 percentage of memory used for computational pages
```

3.7.3 Planning the cache

As already noted, the size of the cache is critical to the operation of asynchronous GLVM:

- If the cache is too big, then there is the potential of losing a large amount of data in the event of a disaster.
- If the cache is too small, then there is the risk that the cache will fill, and GLVM will revert to synchronous operations until space is freed.

Note: Remember that there is one cache per volume group and it must be of sufficient size to hold all the I/O that could take place on all the logical volumes in the time it takes for an update to be made to the remote LUNs and the acknowledgment is returned.

Points to consider when planning the size of the aio_cache:

- Prioritize the importance of your data and file systems.
- If possible plan the layout of your Volume Groups so that each VG contains similar priority data - as they will share the same cache. One does not want temporary files or less critical I/O flooding a cache that is needed by more critical data.
- Plan operations and batch jobs to fit within the network bandwidth. This may include staggering batch jobs/write intensive jobs where possible.
- Analyze the peaks in your historical I/O profile. A useful exercise is to load the I/O writes in KB per time slice into a spreadsheet. Then estimate the amount of data that the network will be able to transfer in each time slice. For each time slice, where the amount of data is greater than what the network can transmit, add the excess amount of data to the next time slice and continue to the end of the analysis period. This will help estimate how long it will take I/O peaks to drain through the network at the given network bandwidth, and thus help size the cache. Figure 3-1 on page 38 shows the impact of I/O exceeding the network bandwidth.
- The percentage of the aio_cache used is set when the VG is changed to asynchronous mode and this value can be changed.
- The network bandwidth is not fixed, but will be the largest proportion of the cost of the DR solution. Too small will lead to performance problems, while too big is a waste.
- Regularly review I/O patterns to ensure that growth has not necessitated an increase in network bandwidth.

Note: 1K of updates for the remote site will use roughly 2K of the cache. So in a disaster with a 5GB cache, there is the potential of losing 2.5GB of updates.

In Figure 3-1 we demonstrate a case where the I/O rate exceeds the network bandwidth, which will lead to issues with the mirroring solution.

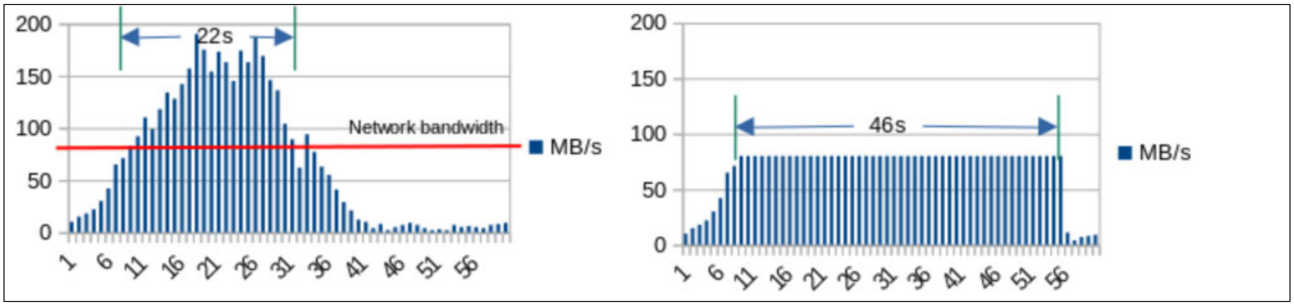


Figure 3-1 Demonstrating I/O exceeding the network bandwidth

3.8 GLVM tuning options

The **rpvutil** command was introduced in AIX 7.2.5 and is used to configure the RPV client mirror pool. The following options are available:

rpv_net_monitor=1|0

Setting *rpv_net_monitor* to 1 enables monitoring the RPV network so that the RPV client detects any network failures and attempts to resume operation after the network recovers.
The RPV client detects network interface states based in the network driver tracked states and attempts to resume the network after interface recovery. Similarly, when interfaces go down, the RPV client identifies and stops data mirroring. However, if the network interface state is up but remote servers are not reachable over the network, the RPV client *io_timeout* is used
The default is 0 (disabled).

compression=1|0

Compresses I/O data packets before they are sent from the client to the server. As discussed above this requires the use of the NX842 acceleration unit to avoid an impact on performance.
The default is 0 (disabled).

io_grp_latency=1-32768

Sets the maximum expected delay in milliseconds before receiving the I/O acknowledgment for an asynchronous mirror pool. A lower value can be set to improve I/O performance, but at the cost of higher CPU consumption.
Testing in the lab with the default *io_grp_latency* of 10 ms produced 45 thousand IOPS, but when *io_grp_latency* was reduced to 3 ms, the IOPS increased by 62% to 73 thousand IOPS.
The default delay value is 10 ms.

nw_sessions=1-99

The number of sessions in a new tunable (available in AIX 7.2.5.2) that controls the number of RPV sessions (sender and receiver threads) to be configured per network.
Depending on the network latency (and if compression is enabled), we found that a value of 40-80 dramatically improved the transfer rate. However it has been observed that increasing

this value beyond this value has lead to performance degradation.
The default value is 1.

cf_tmr_feature=1|0
: This setting enables or disables the cache full timer feature and was introduced in AIX 7.3.
If this feature is enabled, the rpvutil command starts a timer when the I/O buffer cache is full. Then all the subsequent I/O requests are buffered internally, and if the timer expires without space in the cache, all the internally buffered I/O requests are invalidated, the application threads released and the physical volumes are marked stale. This allows for a faster response if the cache fills, but it does mean that **syncvg** needs to be run each time it does and the whole logical partition will have to be synchronized, not just the updates.
The default is 0 (disabled).

cf_tmr_value=2-30
: This setting sets the timeout value in seconds for the cache full timer feature.
The default value is 10 seconds.

The following examples demonstrate the use of the **rpvutil** command. Example 3-7 demonstrates showing all of the settings where Example 3-8 shows only a single setting.

Example 3-7 The use of rpvutil command to show all settings

```
# rpvutil -a
rpv_net_monitor = 0
compression = 0
nw_sessions = 1
cf_tmr_feature = 0
cf_tmr_value = 10
```

Example 3-8 The use of rpvutil command to show one setting

```
# rpvutil -o nw_sessions
nw_sessions = 1
```

In Example 3-9 we use change a setting using **rpvutil**.

Example 3-9 The use of rpvutil command to change a setting

```
# rpvutil -o nw_sessions=3
Setting nw_sessions to 3
```

3.9 Tuning summary

Table 3-4 on page 40 below contains the recommendations from Ravi Shankar’s Best Practices guide, with notes from our testing. As noted above, some of the values have been set for an extreme load, so may be larger than required for general implementations. As above it should be noted that good practice dictates that if you are making changes, you increase the relevant tunable by smaller amounts and monitor.

Table 3-4 Tuning recommendations

Sub system	Tunable	Range	Default value	Value used	Comments
LVM	io_grp_latency	1 - 20ms	10ms	3 ms	
	LTG size			512KB with hardware compression enabled	Inherited from max_transfer of disk
	MWCC	Active / Passive / Disabled	Active	Passive	
	aio_cache_buf_count	512-16384		16384	
AIX disk subsystem	queue_depth	8-256	40	256	Based on storage vendor's recommendation
	max_transfer	< 16MB	0x80000 (512 KB)	512KB with hardware compression enabled	
AIX Networking	tcp_sendspace		128KB	50MB	While this is a recommendation, we found that 2-5MB was sufficient
	tcp_recvspace		64KB	50MB	While this is a recommendation, we found that 2-5MB was sufficient
	sack	0-1	0	1	
	tcp_nodelayack	0-1	0	1	
	rfc1323	0-1	1	1	

Note: Do not reduce the *max_transfer* size for a remote device while there is still data in the cache as it may cause remote I/O failures.

3.10 Integration with Power HA SystemMirror Enterprise Edition

This section looks at the advantages of having PowerHA manage the GLVM configuration and the steps required to bring an existing GLVM replicated environment under PowerHA control.

It is not the intention of this document to detail the PowerHA installation and configuration as these details are contained in the PowerHA document library and the cookbook discussed below. It is however recommended to use a recent version on PowerHA (generally most organizations prefer version “n-1”), at least at PowerHA version 7.2.

3.10.1 GLVM with PowerHA management

IBM strongly recommends using GLVM under the control of PowerHA as it simplifies the management, monitoring and troubleshooting of the configuration. In particular:

- Performs check of GLVM environment prior to starting RPV Servers and Clients.
- Ensures GLVM components start on the correct site, and only one site can access the data areas at one time.
- Sets preferred read to the local mirror pool copy.
- PowerHA / CAA monitors the health of the remote nodes.
- PowerHA / CAA monitors the health of the replication networks.
- Will suspend replication if the remote site is unreachable.
- Will resume replication when the remote site becomes available again, thus helping to reduce time to synchronize.
- Manages around planned and unplanned outages.
- Has a GUI for simplified installation and management.
- Helps manage and recover from data divergence.

In summary PowerHA with GLVM will help prevent data corruption.

> **Note:** Remember that CSPOC should not be used to manage or modify a GMVG.

The following steps outline what is required to integrate an operating GLVM environment under PowerHA control:

1. Configure PowerHA cluster with nodes and sites, ensuring that the GLVM site names and PowerHA site names match.
2. Configure replication network XD_DATA.
3. Add alias addresses to each replication network adapter.
4. Change RPV Clients / Servers to use replication addresses.
5. Configure RPV Clients and Servers for replication both directions.
6. Import each GMVG into all servers where it will be used to ensure that the ODM is consistent and up to date (use consistent major numbers).
7. Discover disks and networks.
8. Synchronize the cluster.
9. Add GMVGs to Resource Groups, set forced activation and choose data divergence recovery options.

For more details regarding PowerHA and GLVM, we recommend the latest *IBM PowerHA SystemMirror for AIX Cookbook*, SG24-7739.[6]

[6] https://www.redbooks.ibm.com/redpieces/pdfs/sg247739.pdf

4

Migration to the cloud

There are a number of ways to migrate from on-premises to the IBM Cloud, and he final choice may be a combination of the following options – depending on the application, the time available and allowed outage windows.

This chapter discusses different options to consider when migrating an application currently running on-premises to run in a cloud provider. The options discussed are done so in reference to the IBM Cloud, but many of them are applicable to other cloud vendors as well.

The following topics are covered in this chapter:

- “Replication options”
- “IBM Cloud Object Storage (ICOS)”
- “IBM Aspera”
- “Stand alone GLVM replication to PowerVS”

4.1 Replication options

There are a number of replication options that can be utilized for moving data from your on-premises location to a cloud. The following list provides a high level summary of some of those options:

- Hardware based replication:
  - Cloud Object Storage
- General:
  - **`rsync`** or other file replication
  - file replication using IBM Aspera®
- AIX:
  - GLVM
- IBM I:
  - Geographic Mirroring
- Database replication:
  - IBM Db2® has HADR
  - Oracle has Dataguard, etc.
  - Log shipping is an option for many databases.

We describe some of these options in more detail in the following sections.

4.2 IBM Cloud Object Storage (ICOS)

ICOS can be used to store data from on-premises and then be transferred to your PowerVS environment. Data can be files, **`mksysb`** or **`savevg`** images[1] and NIM can be used to build PowerVS instances. If PowerVC is used, OVA images can be created and transferred to ICOS and then directly used to create new LPARs.

4.3 IBM Aspera

IBM Aspera is a licensed product that can be used to improve the speed of transferring data from on-premises to Power in the Cloud[2]. IBM Aspera takes a different approach to tackling the challenges of moving large amounts of data over global wide area network connections (WANs). Rather than optimize or accelerate data transfer, Aspera eliminates underlying bottlenecks using a breakthrough transport technology, fully utilizing available network bandwidth to maximize speed and quickly scale up with no theoretical limit.

BM Aspera provides a fast alternative to FTP server software for reliable and secure file transfer and delivery. Aspera eliminates the bottlenecks and risks associated with FTP as a decades-old technology to move the largest files and data sets at maximum speed, even over long distances, by fully utilizing available bandwidth. In addition, Aspera's unique adaptive rate control delivers this high performance without impacting business-critical network traffic by sharing bandwidth to accommodate existing FTP, web browsing, and other TCP traffic.

The Redbook *Integration of IBM Aspera Sync with IBM Spectrum Scale: Protecting and Sharing Files Globally*, REDP-5527[3] provides a good overview of the use of Aspera even though its primary intent is describing how to integrate Aspera with Spectrum Scale.

[1] https://www.ibm.com/cloud/object-storage/faq
[2] https://www.ibm.com/products/aspera
[3] https://www.redbooks.ibm.com/abstracts/redp5527.html

4.4 Stand alone GLVM replication to PowerVS

GLVM provides an excellent option for migrating an existing AIX system to a cloud location. Since GLVM is built in to AIX and is based on standard AIX Logical Volume Manager functions, it can be simpler and safer to use compared to having to install additional products into your environment – the fact that there is no additional software licensing to consider is positive.

If you are migrating an existing LPAR from on-premises to PowerVS, the following steps will be required.

1. Configure the LPAR to meet your requirements for CPU, memory and storage. Ensure the storage meets your application I/O requirements.
2. Configure network with sufficient bandwidth between on-premises and PowerVS (See 3.6.3, “Networks” on page 31 for guidance).
3. Create a mksysb image of the existing LPAR and copy it to PowerVS to be used as the LPAR image.
4. Start the LPAR utilizing the mksysb image.
5. Verify that the volume group has been created as a scalable VG[4]. This is the only requirement that will require an outage if not met and you are on AIX 7.2 or below. You may also need to free up some physical partitions before the conversion.
6. Check the remaining asynchronous GLVM prerequisites – See 3.2, “GLVM and AIX requirements and limitations” on page 28.
7. Configure a cache for the PowerVS mirror pool. There is no need for a cache for the local mirror pool if there is no intention to replicate back from PowerVS to on-premises.
8. Configure and tune the replication network, ensure that the GLVM ports are open.
9. Assign GLVM site names and create a RPV Server for each replicated LUN in PowerVS.
10. Create an on-premises RPV Client for each PowerVS RPV Server.
11. Add each RPV Client (hdisk) to the appropriate local Volume Group.
12. Convert the local mirror pool to asynchronous and set the percentage of cache to use. It is recommended to set to 50-75% as this can be increased without requiring a re-mount of the file systems.
13. Start a background sync of each logical volume. Depending on I/O and network bandwidth, it may be a good idea to stagger these to reduce total network usage. Use syncvg command which has options by LV, PV or VG and the number of parallel threads can be set. In our testing, we found that 4-6 threads gave a good throughput. For example, to run for each LV:

    ```
    syncvg -P 6 -l lv_name
    ```

    In the example below, our network throughput was about 14GB/s, so one could expect that when using this network 500GB of data could take at least 10 hours. However we recommend that you do your own testing, tuning and use calculations based on the amount of data you have to sync.

14. The **syncvg** command can also be used to query the processes, pause, restart or terminate each sync process as required.
15. Once the synchronization of the stale partitions has completed, GLVM will continue to asynchronously replicate changes as they are made to the remote site.

[4] https://www.ibm.com/support/pages/aix-volume-group-limitations-and-types-small-big-scalable-and-how-convert-between-them#:~:text=This%20can%20accommodate%20up%20to,AIX%205.2%20or%20lower%20versions.

Note: Some customers have experienced issues during this initial synchronization of the stale partitions. This generally is a result of network sizing. Because this is the first time their network sizing is being tested and the network load includes the extra data transfer due to the initial synchronization on top of the application I/O.

We therefore recommend that the application, GLVM (caches in particular), network bandwidth and application performance be monitored carefully during this period.

When time to switch over, on-premises:

1. Stop the applications and wait for the outstanding I/Os to flush (rpvstat).
2. Unmount the replicated file systems.
3. Deactivate (varyoffvg) the GMVGs.
4. Put the RPV clients into defined state.

On the PowerVS instances:

1. Stop and remove the RPV Servers.
2. If the GMVGs have not been imported yet, import each VG (choosing a unique major number if planning to integrate with PowerHA), without activating.
3. Force activation (varyonvg) the GMVGs.
4. Remove the on-premises mirror pool copy from each LV in each GMVG.
5. Remove each aio_cache.
6. Remove each missing PV from each GMVG.
7. Mount each file system.
8. Start each application.

5

Monitoring, maintenance and Troubleshooting

This Section gives an overview of some of the monitoring requirements and some of the GLVM administration / maintenance tasks.

As with any complex environment IBM recommends having test / development / preproduction environments which replicate the production environment. This will enable thorough testing of any firmware / operating system / application change prior to those changes being implemented in production. Remember in Australia, we do not TIP[1].

The following topics are covered in this chapter:

- “Monitoring”
- “Maintenance”
- “Changing the size of the cache”
- “Troubleshooting”

[1] Test In Production

5.1 Monitoring

The key metrics to monitor for GLVM are I/O for the replicated volumes, AIX and memory buffers, GLVM cache usage and system errors. There are a number of useful commands to monitor the usage of these resources. These can be used in the planning phase and for ongoing monitoring once operational.

gmdsizing
This is the original tool that was supplied with HAGeo and is still installed in the samples directory with PowerHA. This is mentioned for completeness, but is not recommended as there are now better tools (gmvgstat, iostat, lvstat, nmon, etc)

lvmstat
reports I/O stats for logical partitions and reports on any buffer shortage

iostat
Useful I/O monitoring tool, particularly as it reports IOPS, which can be used to calculate cache size.

topas and nmon
Both useful display tools with data collection options

iPerf
iperf is an open source tool for testing network throughput.[2]

Grafana/InfluxDB
Both supported on AIX and a useful tool to collect and display performance data. A good example can be seen with nmonchart.[3]

If you are looking for a low impact way to just measure the network performance, we recommend using FTP as shown in Example 5-1. If FTP is disabled in your environment you may need to enable it in */etc/inetd.conf*, and in your firewalls (and do not forget to disable it after testing). Other tools such as secure copy (scp) or iPerf could also be used.

Example 5-1 Using ftp to measure network throughput (showing 14260 KB/s)

```
ftp> put "|dd if=/dev/zero bs=32k count=1000" /dev/null
200 PORT command successful.
150 Opening data connection for /dev/null.
1000+0 records in
1000+0 records out
226 Transfer complete.
32768000 bytes sent in 2.244 seconds (1.426e+04 Kbytes/s)
local: |dd if=/dev/zero bs=32k count=1000 remote: /dev/null
```

GLVM also provides a number of commands to report on GLVM activity. A number of our customers have collected this data into an influxDB database and then presented the results using Grafana which they found to be a useful way to show the network throughput, GLVM operations and the cache usage into one display.

The following examples show the output of some commands that can be used to display the GLVM stats, but for more options look at the relevant AIX man pages. In particular the output for **gmvgstat** and **rpvstat** are shown. Both of these commands can also be used to report results on regular intervals for ongoing monitoring.

Commands to display the configuration are covered in Chapter 2, “Configuring GLVM” on page 7.

[2] https://www.ibm.com/support/pages/ibm-aix-performance-analysis-using-iperf
[3] https://github.com/aguther/nmonchart

5.1.1 General GMVG statistics

This section shows the use of the **gmvgstat** command to display details on the status of your GMVG environment.

Display the size and sync details for the GMVG.

In Example 5-2 we show how to check the size of a GMVG and to see its synchronization status.

Example 5-2 Output of the gmvgstat command

```
# gmvgstat
GMVG Name         PVs  RPVs  Tot Vols   St Vols   Total PPs   Stale PPs  Sync
---------------  ----  ----  --------  --------  ----------  ----------  ----
glvm1               1     1         2         0        2548           0  100%
glvm2               1     1         2         0        2548           0  100%
```

Display the detailed status of the GMVG

In Example 5-3 we use different options to display detailed statistics about each GMVG. This shows the synchronization status of the GMVGs. If the GMVG is not fully synchronized, it will display the amount of data that is pending updates in the remote copy.

Example 5-3 GMVG detailed status

```
# gmvgstat -rt
Geographically Mirrored Volume Group Information         08:55:15 PM 18 Jul 2023
------------------------------------------------                          glvm1
                                                                          Asgard
GMVG Name         PVs  RPVs  Tot Vols   St Vols   Total PPs   Stale PPs  Sync
---------------  ----  ----  --------  --------  ----------  ----------  ----
glvm1               1     1         2         0        2548           0  100%

Remote Physical Volume Statistics:

                      Comp Reads  Comp Writes Comp KBRead  Comp KBWrite Errors
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite
------------------ -- ----------- ----------- ------------ ------------ ------
hdisk3              1          48           6          788          752      0
                                0           0            0            0

glvm2               1     1         2         0        2548           0  100%

Remote Physical Volume Statistics:

                      Comp Reads  Comp Writes Comp KBRead  Comp KBWrite Errors
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite
------------------ -- ----------- ----------- ------------ ------------ ------
hdisk4              1          48           0          788            0      0
                                0           0            0            0
```

5.1.2 Statistics for synchronous GMVGs

This section shows using different commands used to see the status of synchronous GMVGs.

Example 5-4 shows the general statistics for 2 synchronous GMVGs.

Example 5-4 rpvstat for synchronous GMVG

```
# rpvstat

Remote Physical Volume Statistics:

                      Comp Reads  Comp Writes Comp KBRead  Comp KBWrite Errors
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite
------------------ -- ----------- ----------- ------------ ------------ ------
hdisk4              1           0        1726            0       815752      0
                                0           0            0            0
hdisk3              1           0        2060            0       978676      0
                                0           0            0            0
```

Display the details of the remote physical volumes with a network summary. See Example 5-5.

Example 5-5 Using **`rpvstat`** *to display the remote physical volumes*

```
# rpvstat -m

Remote Physical Volume Statistics:

                         Maximum      Maximum     Maximum       Maximum  Total
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite Retries
------------------ -- ----------- ----------- ------------ ------------ -------
hdisk4              1           0          26            0         4096       0
hdisk3              1           0          26            0         4096       0

Network Summary:
    192.169.200.30              0          26            0         3584       0
```

Displaying the details for the local RPV clients can be seen in Example 5-6.

Example 5-6 Using **`rpvstat`** *to show RPV client and network statistics by RPV client*

```
# rpvstat -n

Remote Physical Volume Statistics:

                      Comp Reads  Comp Writes Comp KBRead  Comp KBWrite Errors
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite
------------------ -- ----------- ----------- ------------ ------------ ------
hdisk4              1           0        1725            0       815748      0
                                0           0            0            0
 192.169.200.30:1   Y           0        1725            0       815748      0
                                0           0            0            0
hdisk3              1           0        2054            0       978636      0
                                0           0            0            0
 192.169.200.30:1   Y           0        2054            0       978636      0
                                0           0            0            0
```

5.1.3 Statistics for asynchronous GMVGs

Display the asynchronous statistics for the GMVG using the command shown in Example 5-7.

Example 5-7 Using rpvstat to display the asynchronous mirror statistics

```
# rpvstat -A

Remote Physical Volume Statistics:

                Completd Completed   Cached   Cached      Pending  Pending
                Async    Async       Async    Async       Async    Async
RPV Client   ax Writes   KB Writes   Writes   KB Writes   Writes   KB Writes
------------ -- -------- ----------- -------- ----------- -------- -----------
hdisk3        A    28395      113580        1           4        0           0
hdisk4        A    33750     2684584        0           0        0           0
```

The global statistics for the system can be used to show the number of times the asynchronous cache was full. This is seen in Example 5-8.

Example 5-8 Using rpvstat to display the global statistics

```
Host:/# rpvstat -G

Remote Physical Volume Statistics:

GMVG name .................................... glvm1
AIO total commit time (ms) .................. 301902
Number of committed groups .................. 28378
Total committed AIO data (KB) ............... 127777
Average group commit time (ms) .............. 10
AIO data committed per sec (KB) ............. 0
AIO total complete time (ms) ................ 15509641
Number of completed groups .................. 28378
Total completed AIO data (KB) ............... 127777
Average group complete time (ms) ............ 546
AIO data completed per sec (KB) ............. 0
Number of groups read ....................... 27490
Total AIO data read (KB) .................... 247546
Total AIO cache read time (ms) .............. 14012
Average group read time (ms) ................ 0
AIO data read per sec (KB) .................. 8000
Number of groups formed ..................... 28378
Total group formation time (ms) ............. 291816
Average group formation time (ms) ........... 10
Number of cache fulls detected .............. 0
Total cache usage time (ms) ................. 1892557980
Total wait time for cache availability (ms) .. 0
Total AIO write data in transit (KB) ........ 0
GMVG name .................................... glvm2
AIO total commit time (ms) .................. 459807
Number of committed groups .................. 28715
Total committed AIO data (KB) ............... 2700872
Average group commit time (ms) .............. 16
AIO data committed per sec (KB) ............. 5000
AIO total complete time (ms) ................ 15829914
```

```
Number of completed groups .................. 28715
Total completed AIO data (KB) ............... 2700872
Average group complete time (ms) ............ 551
AIO data completed per sec (KB) ............. 0
Number of groups read ....................... 27375
Total AIO data read (KB) .................... 265181
Total AIO cache read time (ms) .............. 1998610
Average group read time (ms) ................ 73
AIO data read per sec (KB) .................. 0
Number of groups formed ..................... 28715
Total group formation time (ms) ............. 294429
Average group formation time (ms) ........... 10
Number of cache fulls detected .............. 99
Total cache usage time (ms) ................. 1892553320
Total wait time for cache availability (ms) .. 61700
Total AIO write data in transit (KB) ........ 0
```

Using the **rpvstat** command to show the number of times the cache full was detected and the group times for each GMVG. This is demonstrated in Example 5-9.

Example 5-9 Using **rpvstat** *to display the group times and cache full occurrences*

```
# rpvstat -g

Remote Physical Volume Statistics:

            Avg Group     Avg Group    Avg Group    Avg Group    No.of Cache
GMVG Name   form. time    Commit time  Compl time   read time    Fulls detected
----------  ----------   ------------  ----------   ---------    --------------
glvm1               10            10         546           0                  0
glvm2               10            16         551          73                 99
```

Showing further details of the asynchronous writes and space free in the cache. This can be seen in Example 5-10.

Example 5-10 Using **rpvstat** *to show details of asynchronous writes*

```
Host:/# rpvstat -C

Remote Physical Volume Statistics:

                                     Max    Pending    Total  Max
                    Total Async      Cache  Cache      Cache  Cache    Cache Free
GMVG Name        ax Writes           Util % Writes     Wait % Wait     Space KB
---------------- -- -------------- ------ ---------- ------ ------- ----------
glvm1             A          28398   0.12          0   0.00     333      60927
glvm2             A          33752 100.00          0   0.83       8      60927
```

The system error log will also report cache full events. An example is shown in Example 5-11.

Example 5-11 Example of a cache full event in the system error log

```
LABEL:          RPVC_CACHE_FULL
IDENTIFIER:     07C6CE33

Date/Time:       Wed Feb 22 18:19:43 CST 2023
```

```
Sequence Number: 387
Machine Id:      00C938904B00
Node Id:         glvm1
Class:           S
Type:            INFO
WPAR:            Global
Resource Name:   glvm1_val_ca

Description
RPV cache device is running low on available space.

Probable Causes
There is not enough free space on cache device to accomodate new data.
There is less than minimum percentage of available space in the cache device.

Failure Causes
The cache size is insufficient.
There was a problem with the data mirroring network.

        Recommended Actions
        Increase cache device size.

Detail Data
Reason
cache device is full
```

Displaying the details of the remote physical volumes with a network summary is shown in Example 5-12.

Example 5-12 Using **rpvstat** *to display the remote physical volumes*

```
# rpvstat -m

Remote Physical Volume Statistics:

                          Maximum      Maximum     Maximum       Maximum  Total
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite Retries
------------------ -- ----------- ----------- ------------ ------------ -------
hdisk3              1           5           2          512          512       0
hdisk4              1           6           2          537          512       0

Network Summary:
     192.169.200.30             6          50          537      4775804       0
```

Displaying the details for the local RPV clients is shown in Example 5-13.

Example 5-13 Using **rpvstat** *to show RPV client and network statistics by RPV client*

```
# rpvstat -n

Remote Physical Volume Statistics:

                       Comp Reads  Comp Writes Comp KBRead  Comp KBWrite Errors
RPV Client         cx Pend Reads  Pend Writes Pend KBRead  Pend KBWrite
------------------ -- ----------- ----------- ------------ ------------ ------
hdisk3              1       22808       28579      1427436       151411      2
```

```
                              0          0           0           0
  192.169.200.30:1  Y     22808      28579     1427436      151411      2
                              0     551615           0   854775804
hdisk4              1     22796   70933931     1426682     2722031      1
                              0          0           0           0
  192.169.200.30:1  Y     22796      33931     1426682     2722031      1
                              0  709551615           0   854775804
```

5.2 Maintenance

This section looks at some general maintenance tips and then a few specific examples of GLVM maintenance tasks.

5.2.1 Tips

We have the following tips and recommendations for setting up and maintaining your GLVM environment (additional details can be found in section 3.5, “General recommendations” on page 29):

- In a stand-alone GLVM environment, validate that all the backup disks in the secondary sites are in an active state before bringing the volume group online.

  During the online recovery of the volume group, if the RPV device driver detects that the RPV server is not online, it marks the cache as failed and all subsequent I/Os are treated as synchronous. In this state, each locally modified partition is marked as stale.

  To convert back to asynchronous mode after the problem is rectified, convert the mirror pool to synchronous mode and then back to asynchronous mode by using the **`chmp`** command. The volume group will then need to be resynchronized to update the stale partitions.

- Always remember to set the preferred read for each logical volume before activating a volume group on a new site if using standalone GLVM.
- If you decide to increase your local availability by having two copies of each logical volume at one site – to protect against local storage failure – remember that GLVM does not coalesce writes.

  This means that any update from the site with one LV copy will require two writes over the network to the site with two copies. This must be taken into account when planing network sizing and recovering from failures.

- When an asynchronous GMVG it brought online, it performs a cache recovery. If the node had halted abruptly previously (for example, because of a power outage) it is possible that the cache is not empty. In this case, cache recovery can take some time, depending upon amount of data in the cache and the network speed.

  No application writes are allowed to complete during the time cache recovery is in progress to ensure consistency at remote site. In this case, the application users will observe a significant pause. Therefore, plan for some downtime during the cache recovery operation to ensure the recovery synchronization of the residual data.

  Similarly, after a site failure, asynchronous mirror state on remote site is inactive. After integrating back with the primary site, the mirror pool must be converted to synchronous first and then, back to asynchronous to continue to mirror asynchronously.

- Some of the LVM metadata-related operations (see below) require synchronous I/O operations across sites to ensure that the LVM metadata is correct on both sites. You can perform these types of synchronous I/O operations only when previously buffered data in the *cachelv* logical volume is transferred completely to the recovery site. Therefore, these type of operations can take a long time while waiting for the buffered data to be transferred to the target site.

  If you need faster operations, plan to perform the synchronous I/O operations when the residual buffer data in the *cachelv* logical volume is minimal. You can use the **`rpvstat -C`** command to check the residual buffer data in the *cachelv* disks.

  The following operations might take time to complete because of the residual buffer data:

  – Reduction of logical volume size or reduction of file system size.
  – Removal of logical volume.
  – Closing the GLVM that supports asynchronous mirroring.

- GLVM requires mirror write consistency (MWC) to be set to passive. When MWC is set to passive, the volume group logs that the logical volume has been opened. After a crash when the volume group is varied on, an automatic force sync of the logical volume is started. This means a full sync of the mirrored volume group to the DR site will occur which could be a significant amount of time depending on the size of the volume group and the network speed.

  For more information on MWC refer to this IBM documentation topic.

5.2.2 Selected maintenance task descriptions

This section provides guidance on some specific maintenance tasks that you will encounter in managing your GLVM environment. In the following descriptions, it has been assumed that the file systems are not set to mount automatically as we recommended earlier.

> **Important:** Remember that if you are using PowerHA, CSPOC cannot be used to make changes to GMVGs.

Making any change to a GMVG

Any change to the GMVG (adding a file system, changing the size of a file system, adding a logical volume etc) will not be reflected in the ODM on any of the other nodes in the cluster that share the GMVG. Thus after any change, the following steps must be followed:

On local (active) site

1. Unmount all the file systems and deactivate the GMVG (**`varyoffvg <GMVG>`**).
2. Stop the RPV clients for the GMVG on this node (**`rmdev -l hdiskN)`**.
3. If the LUNs are shared between nodes at this site:
   a. Start the RPV clients on the other node at this site (**`mkdev -l hdiskN`**).
   b. run a learning import for the GMVG (**`importvg -L <GMVG>`**).
   c. Deactivate the GMVG (**`varyoffvg <GMVG>`**).
   d. Stop the RPV clients on that node (**`rmdev -l hdiskN)`**.

On the remote site

4. Stop the currently active RPV servers for the GMVG (**`rmdev -l rpvserverN`**).

On the local site

5. Start the RPV servers for the GMVG on this node (**`mkdev -l rpvserverN`**).

On the remote site

6. For each remote node in the cluster that shares this GMVG:
   a. Start the RPV clients on that node (**mkdev -l hdiskN**).
   b. run a learning import for the GMVG (**importvg -L <GMVG>**).
   c. Deactivate the GMVG (**varyoffvg <GMVG>**).
   d. Stop the RPV clients on that node (**rmdev -l hdiskN)**.

On the local site

7. Stop the RPV servers on the initial node (**rmdev -l rpvserverN**).

On the remote site

8. Start the RPV servers on the node you will replicate to (**mkdev -l rpvserverN**).

On the local site

9. Start the RPV clients locally (**mkdev -l hdiskN)**.
10. Activate the GMVG and mount the file systems (**varyonvg <GMVG>**).

Steps to recover from network or failure of the remote site

In order to be able to recover this failure, it is important to understand what will happen in this scenario. Initially, until GLVM reaches the RPV *io_timeout* setting (default 180 seconds) without an acknowledgment from the remote site, all I/O will be written to the cache.

Once the timeout is reached:

- All writes in the cache will be discarded.
- The logical partitions which had updates in the cache will be marked stale.
- All new writes will result in those local partitions also being marked stale.
- However; if the cache fills in this period, all writes will pause until the timeout is reached.

Once the remote site is available or the network has recovered, each stale partition will have to be replicated over the network, not just the bytes that were changed – so expect a much greater network utilization until production and DR are brought back into sync. This will put a greater load on the network and potentially limit the amount of updates the application can make before the systems are synchronized.

If **lsvg -p** shows the remote disks as missing and **lsvg -l** shows stale logical volumes:

1. Confirm that the RPV Servers on the remote host are available.
2. Confirm that the replication network is up.
3. Resume each RPV Client:

```
chdev -l hdiskN -a resume=yes
```

4. Reactivate each Volume Group:

```
varyonvg <GMVG>
```

5. Sync operations should now have resumed.

Convert GMVG to synchronous mode from asynchronous

To convert the GMVG to synchronous mode, we change the mirror pool itself to synchronous with the following commands:

1. **chmp -S -m <local_mirror_pool> <gmvg_name>**
2. **chmp -S -m <remote_mirror_pool> <gmvg_name>**

> **Note:** GMVGs can be synchronous in one direction and asynchronous in the reverse.

Convert GMVG to asynchronous mode from synchronous

To convert the GMVG to asynchronous mode, we change the mirror pool itself to asynchronous and set the high water mark of the cache with the following commands:

1.
   ```
   chmp -A -m <local_mirror_pool>  -c <local_cache> -h <high_water_mark_%>
   <gmvg_name>
   ```
2.
   ```
   chmp -A -m <remote_mirror_pool>  -c <remote_cache> -h <high_water_mark_%>
   <gmvg_name>
   ```

The *high_water_mark* is the percentage of the cache that will be used.

Adding new local / remote storage

The following steps should be followed to add a new Physical Volume to a GMVG:

1. Assign new LUNs to both the local and remote server.
2. Create the rpvservers on the local server as available.
3. Create the matching rpvclients on the remote server as available.
4. Add the remote rpvclients (hdisks) to the remote mirror pool.
5. Remove (rmdev -l <devname>) the rpvclients and then the rpvservers.
6. Create the remote rpvservers as available.
7. Active the local rpvclients as available.
8. Add the local rpvclients to the local mirror pool.
9. Add the remote physical volumes to the remote mirror pool
10. Add the local disks and rpvclients to the VG.
11. At a suitable time, not in the too distant future, follow the steps shown in "Making any change to a GMVG" on page 55.

Adding a new file system

The following steps should be followed to add a new file system (or raw logical volume if required)

1. On the node where the GMVG is active, create a new logical volume using the required settings, in the local mirror pool.
2. Add a copy to the logical volume in the remote mirror pool.
3. Create a file system using the new logical volume.
4. Mount the file system.
5. At a suitable time, not in the too distant future, follow the steps shown in "Making any change to a GMVG" on page 55

Expanding file system or Logical Volume

The procedure to expand a GMVG logical volume or file system/logical volume, is the same as that for a standard VG, with the only exception is that there will need to be an update of the ODM on the remote server.

This can be performed by following the steps shown in "Making any change to a GMVG" on page 55 when the DR site is next activated, or during the next outage.

Changing the size of the cache

The size of the cache can be modified by changing the size of the cache-logical volume (aio_cache), or the High Water Mark for the cache. If modifying the size of the aio_cache logical volume, the mirror pool must be converted to synchronous mode, so plan to perform this during a period of minimum activity.

To modify the size of the cache, complete the following steps:

1. Convert the mode from asynchronous to synchronous.

```
chmp -S -m <mirror_pool> <volume_group>
```

2. Change the size of the aio_cache_lv.
3. Convert back to asynchronous mode by using the modified aio_cache logical volume.

```
chmp -A -m <mirror_pool> -c <aio_cache_lv> -h <high water mark> <volume 
group>
```

4. Repeat these steps for the other mirror pool.

If modifying the usage of the aio_cache logical volume:

- Change the high water mark for each mirror pool

```
chmp -A -m <mirror_pool> -c <aio_cache_lv> -h <new high water mark> <volume 
group>
```

Managing data divergence

Data divergence occurs only with asynchronous GLVM when you are attempting to start GLVM on one site, while data exists in the cache on the remote site. If you are using PowerHA then it will assist in the management and recovery from data divergence, however for a stand alone implementation, it will have to be done manually.

The issue of data divergence is dealt with in the Redbook and the IBM GLVM publication in more detail, but in summary you will need to consider the following points before activating the GMVG:

- How much data is in the cache?
- Can the cache be recovered?
- Can the data be rebuilt, do manual records exist?
- Can you wait for the other site be brought on line, and if so, how long will it take?

A number of important changes were made to the **varyonvg** command to handle mirror pools and assist with recovering asynchronous mirror pools from disasters.

The following flags were added to **varyonvg**:

-k loc | rem

This option allows the user to specify which copy of the data to preserve in cases of data divergence:

- loc - keep the data in the local mirror pool
- rem - keep the data in the RPV mirror pool

-o

The **varyonvg** command with the -k flag will fail if it detects that the latest data is not in the chosen copy – it contains stale partitions. This flag will force the **varyonvg** with the choice of local or remote,

-d

If the remote cache is inaccessible and the system thinks that it contains data, then the **varyonvg** will fail with a warning. In this case, this flag is used to force the activation with potentially back level (stale) data.

Changing the active site

The details of this process were covered in Chapter 2, “Configuring GLVM” on page 7. However the steps at a high level are:

- On Site 1 (the active site):
  – Stop I/O activity on the active site.

  - Deactivate the GMVGs. This step may take a while if there is a large amount of data in the cache.
  - Stop the RPV clients.
- On Site 2:
  - Stop the RPV servers.
- On Site 1:
  - Start the RPV servers.
- On Site 2:
  - Start the RPV clients.
  - For each logical volume, set the preferred read copy
  - Activate the GMVGs.
  - Mount the file systems.

5.3 Troubleshooting

Much of the discussion around troubleshooting has been focused on network bandwidth and monitoring of the asynchronous cache, but there also other important considerations:

5.3.1 Firewalls

As noted in section 3.2, “GLVM and AIX requirements and limitations” on page 28, ICMP and ports 6192 TCP/UDP must be open between the two servers. PowerHA has a much larger list of ports that need to be open – see Appendix A, “PowerHA SystemMirror network ports” on page 61.

The IBM Redbook IBM Power Systems Virtual Server Guide for IBM AIX and Linux has example configurations and more detail around setting up networking between IBM Cloud data centers.

> **Note:** A customer experienced issues with GLVM when closing a GLVM firewall session, so it is strongly recommended not to make any firewall changes while GLVM is operational.

5.3.2 Cache failure

If there is any I/O failure for the aio_cache Logical Volume, the LVM will mark all physical partitions in the mirror pool as stale. This will require a new aio_cache to be created in the pool. Then the pool will need to be converted to synchronous before it can be converted back to asynchronous using the new aio_cache. Following this, the Volume Group will need to be resynchronized.

5.3.3 Changes in the VGDA on any node in the cluster

As only synchronous GMVGs support enhanced concurrent volume groups, any changes made to a GMVG on one node, will not be reflected on any other node, so the ODM on all the other nodes that share this volume group will have to be updated. This is done by a “Learning” activation of the volume group (**`varyonvg -L`**). The volume group cannot be active on the host where this command is executed, and all the disks must be available and unlocked. It is not safe to use **`varyonvg -bu`** to break any locks, so IBM recommends checking each node to ensure disks are active.

5.3.4 System performance

As with any AIX performance problem, follow your standard procedures and involve IBM support as required.

5.3.5 Using syslog

As with all AIX daemons / subsystems, syslog can be activated with different levels of logging (emerg/panic,alert,crit,err(or),warn(ing),notice,info,debug) – see /etc/syslog.conf.

5.3.6 PowerHA issues

PowerHA troubleshooting is already well covered in the IBM Documentation[4] and the *IBM PowerHA SystemMirror for AIX Cookbook*, SG24-7739

5.3.7 Data collection

Initially it is always good practice to collect a snap, and an snap -e if using PowerHA. The options chosen for snap are dictated by the nature of the problem, but for GLVM, it is recommended to collect at a minimum:

- -t (tcpip)
- -f (file system)
- -L (LVM)
- -g (general)

See Appendix B, “Sample data collection script” on page 63 for some examples of scripts that can be used to delve deeper into GLVM problems.

[4] https://www.ibm.com/docs/en/powerha-aix/7.2?topic=edition-planning

A

PowerHA SystemMirror network ports

Table A-1 shows all the ports used by PowerHA SystemMirror.

Table A-1 PowerHA ports

Port	Protocol	Description
22	TCP	Used for Secure Socket (SSH) configuration
657	TCP	Used for Resource Monitoring and Control (RMC) communication.
657	UDP	Used for Resource Monitoring and Control (RMC) communication.
2049	TCP	Used for Network File System (NFS) Tiebreaker (NFSv4) in PowerHA SystemMirror
4098	UDP	Used for cluster multi-cast communication. (CAA)
6174	TCP	The clinfo_client daemon uses this port number for the clstat utility and for other clinfo applications
6175	TCP	The clm_smux daemon uses this port number for the sysinfod SNMP multiplexing (SMUX) peer operations of the Simple Network Management Protocol (SNMP)

Port	Protocol	Description
6176	TCP	The clinfo_deadman daemon uses this port number for clinfo monitoring operations
6181	TCP	The clcomd daemon uses this port for cluster configuration. PowerHA SystemMirror verification checks for the caa_cfg entry in the /etc/services file. (CAA)
6191	TCP	The clcomd daemon uses this port number during the migration process from an older release of PowerHA SystemMirror
6192	TCP	This port is used for Remote physical volumes (RPV) client-server communication
6270	TCP	The clsmuxpd daemon uses this port number for SNMP operation
8080	TCP	Used for PowerHA SystemMirror GUI Server
8081	TCP	Used for PowerHA SystemMirror GUI Agent
12601	TCP	Reserved by RSCT for the future purpose
16191	TCP	Used for clcomd daemon communication. (CAA)
42112	TCP	Used for cluster unicast communication. (CAA)

B

Sample data collection script

These sample scripts are used to collect further data for delving into GLVM problems - collecting syslog, kernel traces etc.

> **Note:** These scripts are tailored for a specific problem and are only included as an example of what data can be collected for different types of troubleshooting.

Example B-1 shows steps to collect detailed data for GLVM, including syslog and AIX traces.

Example B-1 Description of the steps to collect detailed syslog and trace data for GLVM

```
Please collect the logs in to corresponding client/server directories.

- Enable glvm syslogs (On client and server nodes)
   Edit /etc/syslog and add
      kern.debug /tmp/syslog.out rotate size 1024k files 10
      kern.info /tmp/syslog.info.out rotate size 1024k files 10
      kern.crit /tmp/syslog.crit.out rotate size 1024k files 10

     Create the log file in case it does not exist:
      touch /tmp/syslog.out
      touch /tmp/syslog.info.out
      touch /tmp/syslog.crit.out

     Restart the syslog daemon.refresh -s syslogd

- Start traces (On client and server nodes)

   . start the kernel trace
      # trace -p -n -a -C all -r PURR -j
11F,200,4B0,106,101,104,107,10B,221,1037,4E3,4A6 -T 209715200 -L 419430400 -o
trace.raw

- Start the glvmsnap.sh script, by exporting the SNAPDIR (only on the client
nodes)
   - <Pls provide the script location>
   - export SNAPDIR=<location>
   - Run the script

- Run the test case. (Try to touch multiple files) (This should be run on client)
   - . recreate the problem by running touch command in a loop, then as soon as
they hit one taking 2-3 second (or more) to complete,

- Stop the traces (On client and server)
    - # nice --20 trcstop

- stop the rpvstat processes by killing them manually (Only on Client)
   - ps -ef | grep rpvstat
      -      root 42336692        1   0 09:20:16  pts/1  0:00 rpvstat -G -i 5
      -      root 47907102        1   0 09:20:16  pts/1  0:00 rpvstat -N -t -i 5
      -      root 49480084        1   0 09:20:16  pts/1  0:00 rpvstat -C -t -i 5
      -      root 49545498        1   0 09:20:16  pts/1  0:00 rpvstat -A -t -i 5
      -      root 50069880        1   0 09:20:16  pts/1  0:00 rpvstat -n -i 5

   - kill each process

- . collect the other files
     # /usr/bin/gennames > gennames.out 2>&1
     # LDR_CNTRL=MAXDATA=0x80000000 /usr/bin/gensyms > trace.syms
     # /bin/trcnm > trace.nm
     # /bin/cp /etc/trcfmt trace.fmt
     # snap -r
     # snap -gL

- Collect the below  syslogs, traces files. (On client and server)
   -  /tmp/syslog.out
   -  /tmp/syslog.info.out
   - /tmp/syslog.crit.out
   - trace.raw

- Collect the  glvm stats (Only On client node)
   - Files will be in $SNAPDIR/glvm/*

- Collect the info of the vg that is being used. (Collect on client)
    - lsmp -A <vgname> > "$SNAPDIR/glvm/lsmp-A.out"
    - lsvg -M <vgname> > "$SNAPDIR/glvm/lsvg-M.out"
```

The script in Example B-2 can be registered with the AIX **snap** tool to add GLVM data collection to **snap**.

Example B-2 glvmsnap.sh

```
#!/bin/ksh93
# IBM_PROLOG_BEGIN_TAG
# This is an automatically generated prolog.
#
# bos73D src/bos/usr/sbin/glvm/utils/glvmsnap.sh 1.1
#
# Licensed Materials - Property of IBM
#
# Restricted Materials of IBM
#
```

```
# COPYRIGHT International Business Machines Corp. 2022
# All Rights Reserved
#
# US Government Users Restricted Rights - Use, duplication or
# disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
#
# IBM_PROLOG_END_TAG
###########################################################################################
############
# This script should be registered to snap tool. It must be installed to
# /usr/lib/ras/snapscripts.
#
# Command to register the script is:
# snap -z ADD "product_name=glvm" "class=glvm_class" \
            "command_path=/usr/lib/ras/snapscripts/vmsnap"
#
# Command to run the script is:
# snap -z "product_name=glvm"
#
# Command to un-register the script is:
# snap -z DELETE "product_name=glvm"
#
# Logs related to rpv server & client are collected
# compressed together in a pax file, which can be extracted to any aix
# node for debugging.
#
# Command to view the contents of pax file: zcat filename.pax.Z | pax -v
# Command to extract the contents of pax file: zcat filename.pax.Z | pax -r
###########################################################################################


################################ MAIN Main main ###################################
[[ "$VERBOSE_LOGGING" == "high" ]] && set -x


lslpp -l |grep glvm > /tmp/glvm_inst.out

if (( $? == 0 ))
then
   echo "GLVM is installed on the node .. Gathering the relevant data \n"
else
   echo "GLVM is not installed on the node .. nothing to gather \n"
   return 1
fi

mkdir -p $SNAPDIR/glvm
#echo rpvstat -A -t -i 5
rpvstat -A -t -i 5  > "$SNAPDIR/glvm/rpvstat-A.out" &

#echo rpvstat -C -t -i 5
rpvstat -C -t -i 5 > "$SNAPDIR/glvm/rpvstat-C.out" &

#echo rpvstat -N -t -i 5
rpvstat -N -t -i 5 > "$SNAPDIR/glvm/rpvstat-N.out" &

#echo rpvstat -G -i 5
rpvstat -G -i 5 > "$SNAPDIR/glvm/rpvstat-g.out" &

#echo rpvstat -n -i 5
rpvstat -n -i 5 > "$SNAPDIR/glvm/rpvstat-n.out" &
```

```
lsrpvserver > "$SNAPDIR/glvm/rpvsrv.out"
lsrpvclient > "$SNAPDIR/glvm/rpvclnt.out"
gmvgstat -t > "$SNAPDIR/glvm/gmvgstat.out"
rpvutil -a > "$SNAPDIR/glvm/rpvutil.out"

netstat -in > "$SNAPDIR/glvm/netstat.out"
lspv > "$SNAPDIR/glvm/lspv.out"
lspv -P > "$SNAPDIR/glvm/lspv-P.out"

# Removing existing redundant files
#echo "Removing existing redundant files if present in $SNAPDIR." | tee -a $SCRIPTLOG >&3
#rm -rf $SNAPDIR/glvm.pax.Z
#rm -rf $SNAPDIR/script.log $SNAPDIR/other $SNAPDIR/testcase

# Storing and compressing all the collected logs in a pax file.
#pax -xpax -w  $SNAPDIR/glvm | compress  > /tmp/glvm.pax.Z
#mv /tmp/glvm.pax.Z $SNAPDIR/
```

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

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.

- *Asynchronous Geographic Logical Volume Mirroring Best Practices for Cloud Deployment, REDP-5665*
- *PowerHA SystemMirror for AIX Cookbook, SG24-7739*
- *IBM Power Virtual Server Guide for IBM AIX and Linux, SG24-8512*
- *IBM PowerHA SystemMirror V7.2.3 for IBM AIX and V7.22 for Linux, SG24-8434*
- *High Availability and Disaster Recovery Options for IBM Power Cloud and On-Premises, REDP-5656*

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Migration strategies to IBM Cloud:
  https://cloud.ibm.com/docs/power-iaas?topic=power-iaas-migration-strategies-power
- *IBM Documentation for Geographic Logical Volume Manager (GLVM):*
  https://www.ibm.com/docs/en/powerha-aix/7.2?topic=concepts-geographic-logical-volume-manager-glvm
- Storage recommendation for AIX: Performance improvements by tuning queue depth:
  https://techchannel.com/SMB/11/2018/storage-recommendations-aix-performance
- Basic recommended TCP tuning to improve performance of WAN connections between AIX virtual Machines:
  https://www.ibm.com/support/pages/what-basic-tcp-tunings-are-recommended-improve-performance-wan-connections-between-aix-virtual-machines
- Replicating data to the IBM Cloud – GLVM:
  https://belisama-services.has.coffee/glvm-overview
- Nigel Griffiths: Using Grafana and InfluxDB to capture and monitor nmon performance data:
  https://nmon.sourceforge.io/pmwiki.php?n=Site.Njmon
- IBM Support steps to install InfluxDB and Grafana:
  https://www.ibm.com/support/pages/aix-installing-influxdb-18-and-grafana-7

- IBM documentation has a full set of documentation for GLVM
  https://www.ibm.com/docs/en/powerha-aix/7.2?topic=edition-planning

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

REDP-5717-00

ISBN DocISBN

Printed in U.S.A.