

Implementing IBM Spectrum Scale

Dino Quintero
Puneet Chaudhary
Brian Herr
Steven Normann
Marcelo Ramos
Richard Rosenthal
Robert Simon
Marian Tomescu
Richard Wale



Cloud

Storage





International Technical Support Organization

Implementing IBM Spectrum Scale

November 2015

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

First Edition (November 2015)

This edition applies to IBM AIX 7.1 TL03 SP5, Red Hat Enterprise Linux 7.1, IBM Spectrum Scale 4.1.0.8, 4.1.1.0 and 4.1.1.1.

This document was created or updated on December 1, 2015.

© Copyright International Business Machines Corporation 2015. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	v
Trademarks	vi
IBM Redbooks promotions	vii
Preface	ix
Authors	ix
Now you can become a published author, too!	x
Comments welcome	xi
Stay connected to IBM Redbooks	xi
Chapter 1. Introduction to IBM Spectrum Scale 4.1.1	1
1.1 Overview of IBM Spectrum Scale	2
1.2 New features and enhancements	3
1.2.1 Installation toolkit	3
1.2.2 Multi protocol data access	3
1.2.3 Performance Monitoring Tool	4
1.2.4 Cygwin 64-bit version requirement for Windows cluster nodes	5
Chapter 2. IBM Spectrum Scale implementation	7
2.1 Lab environment setup	8
2.2 Installing IBM Spectrum Scale	8
2.3 Creating the IBM Spectrum Scale cluster	8
2.4 Allocate shared disks	10
2.5 Configuring quorum	11
2.6 Starting the IBM Spectrum Scale cluster	12
2.7 Creating an IBM Spectrum Scale file system	13
2.8 Mounting an IBM Spectrum Scale file system	13
2.9 Cluster startup and shutdown	14
2.10 Adding a network-based IBM Spectrum Scale client	16
2.10.1 Install additional prerequisite RPMs	16
2.10.2 Check name resolution	16
2.10.3 Install the IBM Spectrum Scale product filesets	16
2.10.4 Generate SSH keys and exchange with existing AIX cluster nodes	17
2.10.5 Extend the IBM Spectrum Scale cluster	17
2.10.6 Start IBM Spectrum Scale on a new node	18
2.11 Updating the existing IBM Spectrum Scale cluster from 4.1.0 to 4.1.1	19
2.11.1 Update the first AIX cluster node	19
2.11.2 Update the second AIX cluster node	21
2.11.3 Update the client cluster node	21
2.11.4 Final update steps	23
Chapter 3. Case scenario: Protocol node	25
3.1 Configuring protocol nodes	26
3.1.1 Changing node license	26
3.1.2 Installing prerequisites	26
3.1.3 Installing the IBM Spectrum Scale protocol rpms	27
3.1.4 Enabling Cluster Export Services and configuring CES IP	28

3.2 Creating the SMB export	29
3.2.1 Testing SMB access	30
Chapter 4. Case scenario: IBM Spectrum Scale AFM-based disaster recovery	33
4.1 Introduction to AFM-based disaster recovery	34
4.2 AFM-based disaster recovery: Implementation scenario	35
4.2.1 IBM Spectrum Scale installation planning.	37
4.2.2 Preparing the environment - installing the prerequisites.	38
4.2.3 Failover and failback.	58
4.2.4 Failback to new primary	63
4.2.5 Protocols disaster recovery (new on 4.1.1).	70
4.3 Hadoop configuration using shared storage	71
Appendix A. New commands	77
Commands	78
Installation toolkit	78
Appendix B. Performance monitoring.	81
Installing mmperfmon components	82
Firewall recommendations for the performance monitoring tool	84
Related publications	85
IBM Redbooks	85
Online resources	85
Help from IBM	85

Notices

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

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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


Trademarks

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

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

AIX®
AIX/ESA®
developerWorks®
GPFS™
IBM®
IBM Elastic Storage™

IBM Spectrum™
IBM Spectrum Scale™
Power Systems™
POWER7®
PowerHA®
PowerVM®

Redbooks®
Redpaper™
Redbooks (logo) ®
Storwize®
Symphony®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

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

Find and read thousands of IBM Redbooks publications

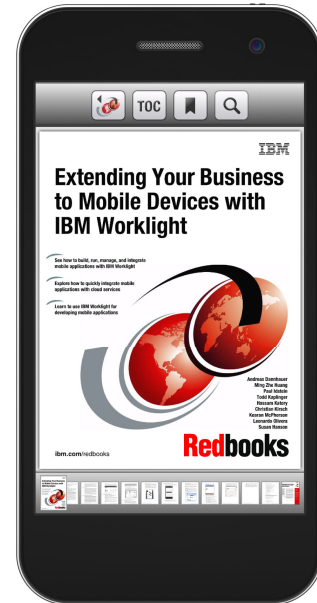
- ▶ Search, bookmark, save and organize favorites
- ▶ Get up-to-the-minute Redbooks news and announcements
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



Download
Now

iOS



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redpaper™ publication describes IBM Spectrum™ Scale, which is a scalable, high-performance data and file management solution, built on proven IBM General Parallel File System (GPFS™) technology. Providing reliability, performance and scalability, IBM Spectrum Scale™ can be implemented for a range of diverse requirements.

This publication can help you install, tailor, and configure the environment, which is created from a combination of physical and logical components: hardware, operating system, storage, network, and applications. Knowledge of these components is key for planning an environment. However, to appreciate potential benefit first requires a simpler understanding of what IBM Spectrum Scale actually provides.

This publication illustrates several example deployments and scenarios to demonstrate how IBM Spectrum Scale can be implemented.

This paper is for technical professionals (consultants, technical support staff, IT architects, and IT specialists). These professionals are responsible for delivering cost-effective cloud services and big data solutions, helping to uncover insights among client data and be able to take actions to optimize business results, product development, and scientific discoveries.

Authors

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

Dino Quintero is a Complex Solutions Project Leader and an IBM Level 3 Certified Senior IT Specialist with the ITSO in Poughkeepsie, New York. His areas of knowledge include enterprise continuous availability, enterprise systems management, system virtualization, technical computing, and clustering solutions. He is an Open Group Distinguished IT Specialist. Dino holds a Master of Computing Information Systems degree and a Bachelor of Science degree in Computer Science from Marist College.

Puneet Chaudhary is a Technical Solutions Architect working with the IBM Elastic Storage™ Server and IBM Spectrum Scale solutions. He has worked with IBM GPFS, now Spectrum Scale, for many years.

Brian Herr is a Software Engineer for IBM Spectrum Scale. He has been with IBM since 1983 working mostly in the area of High Performance Computing. He has been part of the GPFS Native RAID development team since 2008.

Steven Normann is a Senior Software Engineer in MTS working in Poughkeepsie, New York. He has worked with IBM since 1984. He currently is a Team Leader for IBM Spectrum Scale in the Software Technical Support Group, which supports the High Performance Clustering software.

Marcelo Ramos is a Senior Software Support Specialist at IBM Brazil. He has been working with AIX® and related products since 1998. His areas of expertise include implementation and advanced support for AIX, IBM PowerVM®, IBM PowerHA®, and GPFS. He also has SAN and Storage related skills.

Richard Rosenthal is a Senior Software Engineer for IBM Spectrum Scale. He has been with IBM since 1979, and in development since the mid-1990s, working mostly in the area of High Performance Computing. He has been part of the GPFS Native RAID development team since 2008.

Robert Simon is a Senior Software Engineer for IBM Spectrum Scale. He has been with IBM since 1987 working in Software Support (Level 2) for VM, IBM AIX/ESA®, PSSP (HPC), GPFS and GPFS Native RAID.

Marian Tomescu has 16 years of experience as an IT Specialist and currently works for IBM Global Technologies Services in Romania and has ten years of experience in IBM Power Systems™. Marian has a Master's degree in Electronics Images, Shapes and Artificial Intelligence from Polytechnic University- Bucharest, Electronics and Telecommunications, in Romania.

Richard Wale is a Senior IT Specialist working at the IBM Hursley Lab, UK. He holds a B.Sc. (Hons) degree in Computer Science from Portsmouth University, England. He has over 14 years of experience supporting AIX. His areas of expertise include IBM Power Systems, PowerVM, AIX, and IBM i.

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

Richard Conway, David Bennin
ITSO, Poughkeepsie Center

Wesley Jones
IBM Poughkeepsie, New York

Luis Bolinches
IBM Finland

Now you can become a published author, too!

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

Find out more about the residency program, browse the residency index, and apply online at:
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, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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



Introduction to IBM Spectrum Scale 4.1.1

IBM Spectrum Scale is a scalable, high-performance data and file management solution that is built on proven IBM General Parallel File System (GPFS) technology. Providing reliability, performance and scalability, IBM Spectrum Scale can be implemented for a range of diverse data and file management requirements.

This publication benefits those who are unfamiliar with IBM Spectrum Scale and equally those who are existing users.

This chapter summarizes IBM Spectrum Scale and introduces some of the features that were announced with IBM Spectrum Scale 4.1.1.

The following topics are discussed in this chapter:

- Overview of IBM Spectrum Scale
- New features and enhancements

1.1 Overview of IBM Spectrum Scale

Deployments of IBM Spectrum Scale are created from a combination of physical and logical components: hardware, operating system, storage, network and applications. Knowledge of these components is key for planning an environment. However, appreciating the potential benefit first requires a simpler understanding of what IBM Spectrum Scale actually provides.

Fundamentally IBM Spectrum Scale delivers a clustered file system, providing concurrent access from a number of servers. This model differs from a traditional client and server model, which has many clients and one server. With IBM Spectrum Scale, the relationship can be many (servers) to many (clients) and the clients can also be servers. The connectivity and relationship to the file system varies depending on the role of a given cluster node.

How you choose to implement the clustered file system will depend on your application and availability requirements. For more detail about the components and terminology of an IBM Spectrum Scale cluster, see *IBM Spectrum Scale (formerly GPFS), SG24-8254*.

Essentially a *cluster* is a configuration of server nodes, storage, and networking. The four typical cluster configurations are as follows:

- ▶ All cluster nodes are directly attached to a common (shared) set of storage.
- ▶ Some cluster nodes are directly attached to the storage; the remainder nodes are clients.
- ▶ A cluster is spread across one or more locations.
- ▶ Data is shared between clusters.

The first two configurations are dictated by what data is stored within the cluster and how that data is consumed. The final two configurations illustrate some of the resilience and business continuity possibilities. A deployment can equally be a combination of one or more of the four types.

Another type is a *shared nothing configuration*. This is an IBM Spectrum Scale feature starting with version 3.5 named GPFS-FPO. For details about the topologies, also see *IBM Spectrum Scale (formerly GPFS), SG24-8254*.

IBM Spectrum Scale supports cluster nodes on multiple hardware platforms and operating systems. For a current list of supported platforms and operating systems, see the IBM Spectrum Scale FAQ at the following website:

<http://ibm.co/10bb606>

Important aspects to understand and appreciate are the high-level concepts, before you try to interpret the supported configurations. Although a cluster is a logical combination of nodes and associated elements, what is appropriate for one deployment, might not be suitable for another. A deployment can be as simple or as complex as your requirements demand.

Appreciating the flexibility and scalability provided by IBM Spectrum Scale is another important consideration. A cluster could initially be deployed as a simple 2-node implementation. As the footprint of the hosted application grows, additional cluster nodes can be easily added to improve performance. A cluster can host a single clustered file system or there can be many. The configuration can be enhanced as a whole or given components can be integrated into a subset of the existing cluster.

The quantity and depth of documentation can appear overwhelming for a prospective new user of IBM Spectrum Scale. For such an audience, we recommend starting with a simple scenario to build familiarity of the fundamental components and their implementation. This context will provide the foundation to help you decide how you want to implement and leverage IBM Spectrum Scale and therefore which features you need to learn more about.

One such example is a 2-node Linux cluster, as detailed in the *IBM Spectrum Scale (formerly GPFS), SG24-8254*. In Chapter 2, “IBM Spectrum Scale implementation” on page 7 we detail a similar implementation of a deployment with two AIX nodes and a single Linux node.

1.2 New features and enhancements

IBM Spectrum Scale 4.1.1 was released in June 2015. This section highlights the new features and changes. The published list of changes are on the “Summary of changes” page in the IBM Knowledge Center for IBM Spectrum Scale:

<http://ibm.co/1VqYGFr>

Additional details of changes with IBM Spectrum Scale 4.1.1 and subsequent fix packs are in the announcement forum at the IBM developerWorks® website:

<http://ibm.co/1N62c3H>

1.2.1 Installation toolkit

A new command facilitates automating the installation and administration of a cluster. This new feature leverages *Chef* to distribute and orchestrate commands across the cluster. Cluster nodes can be added, configured, and upgraded from a central server. At the time this publication was written, this feature was available with IBM Spectrum Scale Standard Edition or later, running on Red Hat Enterprise Linux 7.1. For a broader overview of this new feature, see “Installation toolkit” on page 78.

For more information, see the **spectrumscale** command description at the “IBM Spectrum Scale Administration and Programming information” page in the IBM Knowledge Center:

<https://ibm.biz/BdHqqV>

1.2.2 Multi protocol data access

IBM Spectrum Scale 4.1.1 Standard and Advanced editions provide additional file and object access methods, enabling users to consolidate multiple sources of data efficiently in one global name space. This new functionality is also called Cluster Export Services (CES).

The additional protocol access methods that are integrated with IBM Spectrum Scale are file access, by using NFS, and object access by using OpenStack Swift. Although these server technologies are open source-based, their integration adds value to the user by providing the ability to scale and use IBM Spectrum Scale clustering technology for high availability. It also saves costs by allowing clients that do not have IBM Spectrum Scale to access data.

To enable the new protocols, some nodes (at least two are recommended) in the IBM Spectrum Scale cluster must be designated as protocol nodes (or CES nodes) from which non IBM Spectrum Scale clients can access data residing and managed by IBM Spectrum Scale. Protocol nodes need a server license. At the time this publication was written, protocol nodes also needed to all run on Linux on Power (in big endian mode) or Intel.

Different packaging for IBM Spectrum Scale, with or without protocols, are available if you do not want to accept additional license terms. Although some of the provided components are open source, the specific provided packages must be used.

Along with the **spectrumscale** command, some new commands are introduced to enable functionality of the protocols. The commands are **mmces**, **mmuserauth**, **mmnfs**, **mm smb**, **mmobj**, and **mmperfmon**, along with **mmdumpperfdata** and **mmprotocoltrace** for data collection and tracing. Some commands, like **mm1scluster**, **mmchnode**, and **mmchconfig**, are expanded to provide new functionalities; **gpfs.snap** now includes data-gathering about protocols.

For an example scenario of configuring protocol node, see 3.1, “Configuring protocol nodes” on page 26. An alternative implementation method, using the new **spectrumscale** command, is documented in the IBM Spectrum Scale Wiki:

<http://ibm.co/1J15Lvs>

Relevant information about protocols is at the following IBM Knowledge Center page:

<http://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html>

Consider the following information:

- ▶ The NFS functionality that is provided with Cluster Export Services (CES) cannot coexist with Clustered NFS function (CNFS). If you want to use SMB and Object function integrated with CES, you have to migrate from CNFS to CES NFS. As you plan for that migration please note that the CNFS failure group functionality is not currently available with CES NFS.
- ▶ Several IBM Spectrum Scale configuration aspects have not been explicitly tested with the protocol function:
 - Local Read Only Cache.
 - Active File Management.
 - NSD server functionality and storage attached to a protocol node. The suggestion is for protocol nodes not to take on these functions.
 - If you have a File Placement Optimizer (FPO) configuration, and if you want to use integrated protocol function, the protocol nodes should be nodes that are not FPO disk servers.
 - Protocol nodes also cannot export remote mounted file systems.
- ▶ The protocol software includes open source components of NFS server (Ganesha), SMB (Samba) and Openstack Swift (from IBM Cloud Manager). Use only the versions of these components provided in order to use the integration with CES (failover and monitoring) or it will not be supported.

1.2.3 Performance Monitoring Tool

Available with IBM Spectrum Scale Standard Edition or later, the **mmperfmon** command reports performance information about an existing cluster. The command collects and collates metrics from the cluster nodes and provides a range of predefined queries to integrate and display the data.

For more details about the **mmperfmon** command, see these locations:

- ▶ Appendix B, “Performance monitoring” on page 81
- ▶ The IBM Spectrum Scale Administration and Programming information page in the IBM Knowledge Center:

<http://ibm.co/1hf1EtZ>

1.2.4 Cygwin 64-bit version requirement for Windows cluster nodes

Starting with IBM Spectrum Scale 4.1.1, the 32-bit version of Cygwin is no longer supported on Windows nodes. Cygwin must be upgraded to the 64-bit version before an existing cluster node is upgraded from IBM Spectrum Scale 4.1.0 to 4.1.1. For more information, see the Installing Cygwin topic at the IBM Knowledge Center for IBM Spectrum Scale:

<http://ibm.co/1KUpiV4>



IBM Spectrum Scale implementation

This chapter describes a high-level sequence of steps to install and create a two-node IBM Spectrum Scale cluster on AIX:

1. Install IBM Spectrum Scale.
2. Create an IBM Spectrum Scale cluster.
3. Create Network Shared Disks (NSD).
4. Create and mount an IBM Spectrum Scale file system.

The example in this chapter illustrates the requirements of a simple cluster, and the minimal effort necessary to implement the fundamental environment.

The following topics are discussed in this chapter:

- ▶ Lab environment setup
- ▶ Installing IBM Spectrum Scale
- ▶ Creating the IBM Spectrum Scale cluster
- ▶ Allocate shared disks
- ▶ Configuring quorum
- ▶ Starting the IBM Spectrum Scale cluster
- ▶ Creating an IBM Spectrum Scale file system
- ▶ Mounting an IBM Spectrum Scale file system
- ▶ Cluster startup and shutdown
- ▶ Adding a network-based IBM Spectrum Scale client
- ▶ Updating the existing IBM Spectrum Scale cluster from 4.1.0 to 4.1.1

2.1 Lab environment setup

Our environment is made of the following infrastructure:

- ▶ Two LPARs (gpfs2740 and gpfs2750) hosted on separate IBM POWER7® machines, configured with N-Port Virtualization (NPIV) adapters. The network traffic is routed through a Virtual I/O Server (VIOS) LPAR on each machine. Both physical machines are hosted on the same subnet.
- ▶ Both POWER7 machines are connected to the same IBM Storwize 7000 storage subsystem.
- ▶ Both LPARs are installed with AIX 7.1 (7100-03-05) into a single 30 GB disk. Both OpenSSH and OpenSSL packages are installed.
- ▶ Both LPARs can resolve each other's host name and IP address.
- ▶ `/etc/ssh/sshd_config` is updated to **PermitRootLogin**. SSH keys are generated and exchanged between both LPARs to allow keyed authentication.
- ▶ Each of the two LPARs has access to the same set of ten 10 GB SAN LUNs.

2.2 Installing IBM Spectrum Scale

This step assumes that you have the required installable IBM Spectrum Scale filesets. The exact filesets you must install are dictated by which edition (standard or enterprise) you are installing. The filesets are standard Backup File Format (BFF) and are installed with **installp** or **smitty installp** command. They can also be installed from a NIM server if one exists within your environment.

Note: The license agreement must be accepted during the installation.

This part of the process installs only the filesets. It does not activate any configuration or start any IBM Spectrum Scale processes.

2.3 Creating the IBM Spectrum Scale cluster

For simplicity, we updated the path to include `/usr/lpp/mmfs/bin/` to save prefixing commands with the directory path.

Note: Based on our experience, we suggest applying the latest service level of IBM Spectrum Scale prior to creating the cluster. Updates can be downloaded from the IBM Fix Central web page:

<http://www.ibm.com/support/fixcentral/>

Create a text file listing the host names of both LPARs as shown in Example 2-1. In our case, we defined both as quorum nodes.

Example 2-1 Example nodes.list file

```
root@gpfs2740: /> cat /etc/gpfs/nodes.list
gpfs2740:quorum-manager
gpfs2750:quorum-manager
```

By using the `nodes.list` file in Example 2-1 on page 8, we create the cluster with the `mmcrcluster` command, as shown in Example 2-2. Although `--ccr` is the default that is used in the cluster creation, if not there, the command uses `ccr` anyway.

Example 2-2 Creating IBM Spectrum Scale cluster

```
root@gpfs2740: /> mmcrcluster -N /etc/gpfs/nodes.list --ccr-enable -r /usr/bin/ssh -R /usr/bin/scp -C GPFS.SPECTRUM -U SCALE
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license designations.
                Use the mmchlicense command to designate licenses as needed.
mmcrcluster: Propagating the cluster configuration data to all
                affected nodes. This is an asynchronous process.
```

As the output suggests, the configuration is being distributed and applied to all nodes. You issue the `mmcrcluster` command only on one of the nodes. As directed by the output in Example 2-2, the next step is issuing the `mmchlicense` command to suitably license the cluster nodes for the installed edition of IBM Spectrum Scale. This command is shown in Example 2-3.

Example 2-3 Licensing the installed edition

```
root@gpfs2740: /> mmchlicense server -N gpfs2740,gpfs2750

The following nodes will be designated as possessing GPFS server licenses:
    gpfs2740
    gpfs2750
Please confirm that you accept the terms of the GPFS server Licensing Agreement.
The full text can be found at www.ibm.com/software/sla
Enter "yes" or "no": yes
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
                affected nodes. This is an asynchronous process.
```

As Example 2-3 shows, the `mmchlicense` command must be issued only on one of the cluster nodes because the command is distributed and applied to all specified cluster nodes. To verify that the cluster is successfully created on both nodes, issue the `mmclscluster` command as shown in Example 2-4.

Example 2-4 Verifying successful cluster creation

```
root@gpfs2740: /> mmclscluster

GPFS cluster information
=====
GPFS cluster name:      GPFS.SPECTRUM
GPFS cluster id:       11300267533712031687
GPFS UID domain:       SCALE
Remote shell command:  /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:       CCR

Node  Daemon node name  IP address      Admin node name  Designation
-----
1     gpfs2740             172.16.20.125   gpfs2740         quorum-manager
2     gpfs2750             172.16.20.128   gpfs2750         quorum-manager
```

As this point in the installation sequence, the cluster is created and two nodes are added into the configuration. However the cluster is not yet active and no storage is provisioned.

2.4 Allocate shared disks

As summarized in 2.1, “Lab environment setup” on page 8, both LPARs are zoned to ten secondary disks. In this way, they are considered shared disks and in an IBM Spectrum Scale deployment, which signifies the LPARs as Network Shared Disk (NSD) servers.

Although ten secondary disks are available, we initially define four of them to the cluster. The **mmcrnsd** command is used to add the disks to the cluster configuration. This command requires an input file with the specific configuration details. Example 2-5 shows the syntax used for our example. For a description of the required stanza (Example 2-6 on page 11), see the **mmcrnsd** command page in the Administration and Programming Reference Guide or at the following page:

<https://ibm.biz/BdHqq5>

Example 2-5 The mmcrnsd example stanza file

```
root@gpfs2740:/> cat /etc/gpfs/disks.nsd
```

```
%nsd:
```

```
device=/dev/hdisk1
nsd=NSD001
servers=gpfs2740,gpfs2750
usage=dataAndMetadata
failureGroup=1
pool=system
```

```
%nsd:
```

```
device=/dev/hdisk2
nsd=NSD002
servers=gpfs2750,gpfs2740
usage=dataAndMetadata
failureGroup=1
pool=system
```

```
%nsd:
```

```
device=/dev/hdisk3
nsd=NSD003
servers=gpfs2740,gpfs2750
usage=dataAndMetadata
failureGroup=1
pool=system
```

```
%nsd:
```

```
device=/dev/hdisk4
nsd=NSD004
servers=gpfs2750,gpfs2740
usage=dataAndMetadata
failureGroup=1
pool=system
```

With the stanza file (Example 2-5 on page 10) the `mmcrnsd` command can be called to define the four shared disks to the cluster nodes. Example 2-6 lists the output from that command.

Example 2-6 *mmcrnsd output*

```
root@gpfs2740: /> mmcrnsd -F /etc/gpfs/disks.nsd
mmcrnsd: Processing disk hdisk1
mmcrnsd: Processing disk hdisk2
mmcrnsd: Processing disk hdisk3
mmcrnsd: Processing disk hdisk4
mmcrnsd: Propagating the cluster configuration data to all
        affected nodes. This is an asynchronous process.
```

Our example has two cluster nodes, and the same set of disks are visible to both. In larger deployments, an appropriate approach might be to have multiple groups of shared disks, visible to certain groups of nodes.

The creation of NSDs and propagation across the cluster can be verified by using the `mm1snsd` command as shown in Example 2-7. The output confirms that four NSDs were created and are served by both LPARs.

Example 2-7 *NSD creation verification*

```
root@gpfs2740: /> mm1snsd -L -a
```

File system	Disk name	NSD volume ID	NSD servers

(free disk)	NSD001	AC10147D55C901F0	gpfs2740,gpfs2750
(free disk)	NSD002	AC10147D55C901F3	gpfs2740,gpfs2750
(free disk)	NSD003	AC10147D55C901F7	gpfs2740,gpfs2750
(free disk)	NSD004	AC10147D55C901FB	gpfs2740,gpfs2750

2.5 Configuring quorum

Our created cluster has only two cluster nodes. If either node fails (or is accidentally shut down) quorum will not be met and the cluster will go offline. To mitigate against this, we configure some of our NSDs to also be tiebreaker disks. One, two, or three disks can be configured as tiebreakers. If your disks are provided from different sides of your storage infrastructure, it would be a benefit when deciding which disks to use as tiebreakers. Configuring a disk as a tiebreaker does not reduce the available disk capacity (Example 2-8).

Example 2-8 *Configuring tiebreaker disks*

```
root@gpfs2740: /> mmchconfig tiebreakerDisks="NSD001;NSD002;NSD003"
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
        affected nodes. This is an asynchronous process.
```

```
root@gpfs2740: /> mmgetstate -L -a -s
```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks

1	gpfs2740	1	2	2	active	quorum node
2	gpfs2750	1	2	2	active	quorum node

Summary information

```
-----
Number of nodes defined in the cluster:      2
Number of local nodes active in the cluster: 2
Number of remote nodes joined in this cluster: 0
Number of quorum nodes defined in the cluster: 2
Number of quorum nodes active in the cluster: 2
Quorum = 1*, Quorum achieved
```

```
root@gpfs2740:~/> mmlsconfig
Configuration data for cluster GPFS.SPECTRUM:
```

```
-----
clusterName GPFS.SPECTRUM
clusterId 11300267533712031687
autoload no
uidDomain SCALE
dmapiFileHandleSize 32
minReleaseLevel 4.1.0.4
ccrEnabled yes
tiebreakerDisks NSD001;NSD002;NSD003
adminMode central
```

```
File systems in cluster GPFS.SPECTRUM:
```

```
-----
(none)
```

Configuring the tiebreaker disks in our cluster allows us to shut down one of the nodes while the cluster remains active and available.

2.6 Starting the IBM Spectrum Scale cluster

At this point in the sequence, the created cluster still remains offline. Before we can progress with the final steps of the configuration, the cluster must be started. The **mmstartup** command starts the required daemons and subsystems on each defined cluster nodes and bring the cluster entity up to an online operational state.

Note: The default behavior of IBM Spectrum Scale cluster is not to automatically restart the daemons and subsystems during the boot sequence of a cluster node. Such behavior is expected in deployments where hosted applications must be started first. However, an environment variable is provided to facilitate restart during the boot sequence if preferred. For more information, see Example 2-14 on page 15.

Example 2-9 shows our created cluster starting and its current status verified.

Example 2-9 Cluster startup and verification

```
root@gpfs2740:~/> mmstartup -a
Wed Aug 19 17:16:00 EDT 2015: mmstartup: Starting GPFS ...
root@gpfs2740:~/>
```

2.7 Creating an IBM Spectrum Scale file system

The **mmcrfs** command is used to create an IBM Spectrum Scale file system. Similar to the steps illustrated in 2.4, “Allocate shared disks” on page 10, the **mmcrfs** command takes an input file containing the disks on which to create the file system. In our case, all four NSDs are used, and, by default, the command uses all available space and creates a 40 GB file system. As shown in Example 2-10, we create a file system named `/shared`.

Example 2-10 Creating an IBM Spectrum Scale file system

```
root@gpfs2740:/> mmcrfs msFS -F /etc/gpfs/disks.nsd -T /shared
```

The following disks of `msFS` will be formatted on node `gpfs2740`:

```
NSD001: size 10240 MB
NSD002: size 10240 MB
NSD003: size 10240 MB
NSD004: size 10240 MB
```

Formatting file system ...

Disks up to size 105 GB can be added to storage pool system.

Creating Inode File

Creating Allocation Maps

Creating Log Files

Clearing Inode Allocation Map

Clearing Block Allocation Map

Formatting Allocation Map for storage pool system

Completed creation of file system `/dev/msFS`.

`mmcrfs`: Propagating the cluster configuration data to all affected nodes. This is an asynchronous process.

As with other commands in this example, **mmcrfs** needs to be run only on a single node. The configuration changes are automatically propagated out to other nodes. The file system is successfully created, but not mounted.

2.8 Mounting an IBM Spectrum Scale file system

Mount the previously created file system by using the **mmmount** command. The **-a** parameter, shown in Example 2-11, requests that the file system be mounted on all nodes.

Note: Remember to use the appropriate IBM Spectrum Scale commands and not resort to similarly named standard operating systems commands. The IBM Spectrum Scale commands operate at the cluster level, as opposed to only the single server.

Example 2-11 Mounting an IBM Spectrum Scale file system

```
root@gpfs2740:/> mmmount /shared -a
```

```
Tue Aug 11 13:48:47 EDT 2015: mmmount: Mounting file systems ...
```

```
root@gpfs2740:/> df -g
```

Filesystem	GB	blocks	Free	%Used	Iused	%Iused	Mounted on
/dev/hd4		0.25	0.06	75%	10169	39%	/
/dev/hd2		2.19	0.10	96%	42341	61%	/usr
/dev/hd9var		0.41	0.13	69%	6472	17%	/var
/dev/hd3		0.12	0.12	4%	58	1%	/tmp
/dev/hd1		0.03	0.03	2%	5	1%	/home
/dev/hd11admin		0.12	0.12	1%	5	1%	/admin
/proc		-	-	-	-	-	/proc

```

/dev/hd10opt      0.31      0.18  44%    6970    15% /opt
/dev/livedump     0.25      0.25   1%       4     1% /var/adm/ras/livedump
/dev/msFS         40.00     39.37   2%    4038     7% /shared
root@gpfs2750: /> df -g
Filesystem      GB blocks   Free %Used    Iused %Iused Mounted on
/dev/hd4         0.25      0.06  75%    10154   39% /
/dev/hd2         2.19      0.10  96%   42344   61% /usr
/dev/hd9var      0.41      0.13  69%    6474   17% /var
/dev/hd3         0.12      0.12   4%       55     1% /tmp
/dev/hd1         0.03      0.03   2%       5      1% /home
/dev/hd11admin   0.12      0.12   1%       5      1% /admin
/proc            -          -    -         -     - /proc
/dev/hd10opt     0.31      0.18  44%    6971   15% /opt
/dev/livedump    0.25      0.25   1%       4     1% /var/adm/ras/livedump
/dev/msFS        40.00     1.49  97%    4038     7% /shared

```

In Example 2-11 on page 13, consider how the /shared file system appears on both nodes, like any other normal file system does. Note that the created file system is also referenced in /etc/filesystems, as shown in Example 2-12.

Example 2-12 Example /etc/filesystems entry

```

/shared:
    dev          = /dev/msFS
    vfs          = mmfs
    nodename     = -
    mount        = mmfs
    type         = mmfs
    account      = false
    options      = rw,mtime,atime,dev=msFS

```

The mounted /shared file system can now be accessed simultaneously from either NSD node. Files, directories and data can be read, written and updated. If appropriate you could install any required applications on the two nodes and start to interact with the file system.

Note: For more detailed documentation about the recommended cluster installation and configuration sequence for IBM Spectrum Scale 4.1.1, see the IBM Knowledge Center:
<http://ibm.co/1KQJ1ar>

2.9 Cluster startup and shutdown

We previously noted (in 2.6, “Starting the IBM Spectrum Scale cluster” on page 12) that, by default, an IBM Spectrum Scale cluster does not automatically start during system boot of the cluster nodes. If having automatic startup is more preferable in your environment, this behavior can be set by updating a cluster attribute as shown in Example 2-13. This is a cluster-level attribute and therefore does not need to be set on each cluster node.

Example 2-13 Updating the autoloading attribute

```

root@gpfs2740: /> mmchconfig autoloading=yes
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
              affected nodes. This is an asynchronous process.

```

On some occasions, the IBM Spectrum Scale processes on a cluster node must be shut down; for example, operating system or application maintenance needs to initiate a system reboot. In such situations, the cluster daemons can be cleanly shut down (on a node) by the use of the **mmshutdown** command. The command can either be issued on the specific cluster node or can remotely initiate the request from another cluster node. Example 2-14 shows one of our pair of nodes having the IBM Spectrum Scale daemons stopped and restarted, and how this is reflected in the cluster status.

Example 2-14 Shutdown and startup sequence

```

root@gpfs2750: /> mmshutdown
Tue Aug 11 13:59:16 EDT 2015: mmshutdown: Starting force unmount of GPFS file
systems
forced unmount of /shared
Tue Aug 11 13:59:21 EDT 2015: mmshutdown: Shutting down GPFS daemons
Shutting down!
'shutdown' command about to kill process 2359316
Tue Aug 11 13:59:28 EDT 2015: mmshutdown: Finished

root@gpfs2740: /> mmgetstate -L -a -s

Node number  Node name      Quorum  Nodes up  Total nodes  GPFS state  Remarks
-----
1      gpfs2740      1        1        2        active    quorum node
2      gpfs2750      0        0        2        down      quorum node

Summary information
-----
Number of nodes defined in the cluster:      2
Number of local nodes active in the cluster:  1
Number of remote nodes joined in this cluster: 0
Number of quorum nodes defined in the cluster: 2
Number of quorum nodes active in the cluster: 1
Quorum = 1*, Quorum achieved

root@gpfs2740: /> mmstartup -a
Tue Aug 11 14:12:32 EDT 2015: mmstartup: Starting GPFS ...
gpfs2740: The GPFS subsystem is already active.
root@gpfs2740: /> mmgetstate -L -a -s

Node number  Node name      Quorum  Nodes up  Total nodes  GPFS state  Remarks
-----
1      gpfs2740      1        1        2        active    quorum node
2      gpfs2750      1        0        2        arbitrating quorum node

Summary information
-----
Number of nodes defined in the cluster:      2
Number of local nodes active in the cluster:  2
Number of remote nodes joined in this cluster: 0
Number of quorum nodes defined in the cluster: 2
Number of quorum nodes active in the cluster: 2
Quorum = 1*, Quorum achieved

root@gpfs2740: /> mmgetstate -L -a -s

```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs2740	1	2	2	active	quorum node
2	gpfs2750	1	2	2	active	quorum node

Summary information

```

-----
Number of nodes defined in the cluster:      2
Number of local nodes active in the cluster:  2
Number of remote nodes joined in this cluster: 0
Number of quorum nodes defined in the cluster: 2
Number of quorum nodes active in the cluster: 2
Quorum = 1*, Quorum achieved

```

On AIX cluster nodes, one potential addition is to update `/etc/rc.shutdown` to include a call to `mmshutdown` to facilitate a clean exit from the cluster during a system reboot or shutdown.

2.10 Adding a network-based IBM Spectrum Scale client

In this section, we extend our two-node cluster with a third node. This Linux based node will be configured as a network-based IBM Spectrum Scale client. That means the node will be part of the cluster, but will not have direct connection to the shared storage and therefore will not be a NSD server.

Note: A network-based IBM Spectrum Scale client node is also known as an *application node*.

In our scenario, our Linux node (named *rhel2750*) is another LPAR that is hosted on one of our existing POWER7 systems. The LPAR is installed with Red Hat Enterprise Linux 7.1. The sequence of steps to install and configure IBM Spectrum Scale is similar to those documented previously in this chapter. However, some steps (like the NSD configuration) are not required here because the LPAR does not have direct access to the shared disks. Also, an additional step is required on Linux (compared to AIX) nodes. In the following sequence, we summarize the first few fundamental steps and then detail the latter steps.

2.10.1 Install additional prerequisite RPMs

Linux had already been installed on the LPAR. For the most up-to-date list of prerequisite software, see the IBM Spectrum Scale FAQ:

<http://ibm.co/10bb606>

2.10.2 Check name resolution

Confirm that all three nodes can resolve each other by host name and IP address. Amend `/etc/hosts` where required or update the DNS.

2.10.3 Install the IBM Spectrum Scale product filesets

Install the collection of RPM filesets required for your edition of IBM Spectrum Scale.

2.10.4 Generate SSH keys and exchange with existing AIX cluster nodes

For details about Secure Shell (SSH) keys, including how to generate and exchange them within your cluster nodes, see the following web page:

<http://www.ibm.com/developerworks/aix/library/au-sshsecurity/>

Also ensure that unprompted access is available between all three nodes.

2.10.5 Extend the IBM Spectrum Scale cluster

This two-step process adds the new node to the existing cluster and licenses the installed software, as shown in Example 2-15.

Example 2-15 Extending an existing cluster

```
root@gpfs2740:/etc/gpfs> mmlscluster

GPFS cluster information
=====
GPFS cluster name:      GPFS.SPECTRUM
GPFS cluster id:        11300267533712031687
GPFS UID domain:        SCALE
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR

Node  Daemon node name  IP address      Admin node name  Designation
-----
1     gpfs2740              172.16.20.125   gpfs2740         quorum-manager
2     gpfs2750              172.16.20.128   gpfs2750         quorum-manager

root@gpfs2740:/etc/gpfs> mmaddnode -N rhel2750
Thu Aug 13 16:43:15 EDT 2015: mmaddnode: Processing node rhel2750
mmaddnode: Command successfully completed
mmaddnode: Warning: Not all nodes have proper GPFS license designations.
      Use the mmchlicense command to designate licenses as needed.
mmaddnode: Propagating the cluster configuration data to all
      affected nodes. This is an asynchronous process.
root@gpfs2740:/etc/gpfs> mmlscluster

=====
Warning:
This cluster contains nodes that do not have a proper GPFS license
designation. This violates the terms of the GPFS licensing agreement.
Use the mmchlicense command and assign the appropriate GPFS licenses
to each of the nodes in the cluster. For more information about GPFS
license designation, see the Concepts, Planning, and Installation Guide.
=====

GPFS cluster information
=====
GPFS cluster name:      GPFS.SPECTRUM
GPFS cluster id:        11300267533712031687
GPFS UID domain:        SCALE
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR

Node  Daemon node name  IP address      Admin node name  Designation
-----
```

```

1 gpfs2740      172.16.20.125 gpfs2740      quorum-manager
2 gpfs2750      172.16.20.128 gpfs2750      quorum-manager
3 rhel2750      172.16.20.131 rhel2750

```

```

root@gpfs2740:/etc/gpfs> mmchlicense server -N rhel2750

The following nodes will be designated as possessing GPFS server licenses:
    rhel2750
Please confirm that you accept the terms of the GPFS server Licensing Agreement.
The full text can be found at www.ibm.com/software/sla
Enter "yes" or "no": yes
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
    affected nodes. This is an asynchronous process.
root@gpfs2740:/etc/gpfs> mmlscluster

GPFS cluster information
=====
GPFS cluster name:      GPFS.SPECTRUM
GPFS cluster id:        11300267533712031687
GPFS UID domain:        SCALE
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR

Node  Daemon node name  IP address      Admin node name  Designation
-----
1    gpfs2740             172.16.20.125   gpfs2740         quorum-manager
2    gpfs2750             172.16.20.128   gpfs2750         quorum-manager
3    rhel2750             172.16.20.131   rhel2750

```

2.10.6 Start IBM Spectrum Scale on a new node

Example 2-16 shows how to start IBM Spectrum Scale on a new node.

Example 2-16 Starting IBM Spectrum Scale on a new node

```

root@gpfs2740:/> mmstartup -N nodename
Thu Aug 13 16:47:00 EDT 2015: mmstartup: Starting GPFS ...
gpfs2750: The GPFS subsystem is already active.
gpfs2740: The GPFS subsystem is already active.
root@gpfs2740:/> mmgetstate -L -a -s

```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs2740	1	2	3	active	quorum node
2	gpfs2750	1	2	3	active	quorum node
3	rhel2750	1	2	3	active	

```

Summary information
-----
Number of nodes defined in the cluster:      3
Number of local nodes active in the cluster:  3
Number of remote nodes joined in this cluster: 0
Number of quorum nodes defined in the cluster: 2
Number of quorum nodes active in the cluster: 2
Quorum = 1*, Quorum achieved

```

2.11 Updating the existing IBM Spectrum Scale cluster from 4.1.0 to 4.1.1

In this section, we step through upgrading our configured 3-node cluster from IBM Spectrum Scale 4.1.0. to 4.1.1.

For sample environments that consist only of a cluster of IBM Spectrum Scale nodes, the following high-level steps are required:

1. Download required filesets from IBM Fix Central.
2. On a cluster node, cleanly shut down any applications that depend on IBM Spectrum Scale.
3. Cleanly shut down IBM Spectrum Scale on the cluster node.
4. Update the installed IBM Spectrum Scale product filesets.
5. Restart IBM Spectrum Scale on the cluster node.
6. Repeat these steps for each node.
7. Update the cluster configuration data.

Upgrading one node at a time allows the cluster to remain operational and available during the upgrade process. The sequence in which to upgrade the cluster nodes in a particular environment might be significant. If an environment additionally contains IBM Spectrum Scale client nodes or external clients (like NFS clients), then additional steps and planning might be required.

In our sample environment, the update is implemented in the following sequence:

1. Update the first cluster node.
2. Update the second cluster node.
3. Update the cluster client node.
4. Update the cluster configuration data.

Updating the cluster nodes in a staggered fashion preserves the client node access to the configured IBM Spectrum Scale file system. Naturally the update to the client node will impact its own access to the file system.

2.11.1 Update the first AIX cluster node

As shown in Example 2-17, our IBM Spectrum Scale cluster is online and available.

Example 2-17 Cluster status prior to the update

```
root@gpfs2740: /> mmgetstate -L -a
```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs2740	1	2	3	active	quorum node
2	gpfs2750	1	2	3	active	quorum node
3	rhe12750	1	2	3	active	

We downloaded the required filesets from IBM Fix Central and made them available to the cluster node. At this point, we must cleanly shut down any hosted applications. Next, we cleanly shut down IBM Spectrum Scale processes on the particular node, as shown in Example 2-18 on page 20.

Example 2-18 Shut down IBM Spectrum Scale

```
root@gpfs2740: /> mmshutdown -N gpfs2740
Mon Sep  7 15:47:23 EDT 2015: mmshutdown: Starting force unmount of GPFS file
systems
gpfs2740: forced unmount of /shared
Mon Sep  7 15:47:28 EDT 2015: mmshutdown: Shutting down GPFS daemons
gpfs2740: Shutting down!
gpfs2740: 'shutdown' command about to kill process 4522098
gpfs2740: Master did not clean up; attempting cleanup now
gpfs2740: Mon Sep  7 15:48:30.323 2015: [N] mmfsd is shutting down.
gpfs2740: Mon Sep  7 15:48:30.324 2015: [N] Reason for shutdown: mmfsadm shutdown
command timed out
gpfs2740: Mon Sep  7 15:48:30 EDT 2015: mmcommon mmfsdown invoked. Subsystem:
mmfs Status: down
gpfs2740: Mon Sep  7 15:48:30 EDT 2015: mmcommon: Unmounting file systems ...
Mon Sep  7 15:48:34 EDT 2015: mmshutdown: Finished

root@gpfs2750: /> mmgetstate -L -a
```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs2740	0	0	3	down	quorum node
2	gpfs2750	1	1	3	active	quorum node
3	rhel2750	1	1	3	active	

We unload the kernel extensions by calling the **mmfsenv** command, shown in Example 2-19.

Example 2-19 Unload kernel extensions

```
root@gpfs2740: /> mmfsenv -ummfsenv -u
/usr/lpp/mmfs/bin/mmfskxunload: module /usr/lpp/mmfs/bin/mmfs unloaded.
```

We update the installed IBM Spectrum Scale filesets. This can be performed by using any appropriate method for example **installp** or using NIM. Example 2-20 lists which filesets were updated in our case.

Example 2-20 Updated filesets

Installation Summary

Name	Level	Part	Event	Result
gpfs.docs.data	4.1.1.0	SHARE	APPLY	SUCCESS
gpfs.docs.data	4.1.1.1	SHARE	APPLY	SUCCESS
gpfs.base	4.1.1.0	USR	APPLY	SUCCESS
gpfs.base	4.1.1.0	ROOT	APPLY	SUCCESS
gpfs.base	4.1.1.1	USR	APPLY	SUCCESS
gpfs.base	4.1.1.1	ROOT	APPLY	SUCCESS
gpfs.gskit	8.0.50.40	USR	APPLY	SUCCESS
gpfs.msg.en_US	4.1.1.0	USR	APPLY	SUCCESS
gpfs.msg.en_US	4.1.1.1	USR	APPLY	SUCCESS
gpfs.ext	4.1.1.0	USR	APPLY	SUCCESS
gpfs.ext	4.1.1.1	USR	APPLY	SUCCESS
gpfs.crypto	4.1.1.0	USR	APPLY	SUCCESS
gpfs.crypto	4.1.1.1	USR	APPLY	SUCCESS

gpfs.docs.data	4.1.1.1	SHARE	COMMIT	SUCCESS
gpfs.base	4.1.1.1	USR	COMMIT	SUCCESS
gpfs.base	4.1.1.1	ROOT	COMMIT	SUCCESS
gpfs.gskit	8.0.50.40	USR	COMMIT	SUCCESS
gpfs.ext	4.1.1.1	USR	COMMIT	SUCCESS
gpfs.msg.en_US	4.1.1.1	USR	COMMIT	SUCCESS
gpfs.crypto	4.1.1.1	USR	COMMIT	SUCCESS

We restart the IBM Spectrum Scale processes using the **mmstartup** command. Example 2-21 illustrates the startup and verification that the node has successfully rejoined the cluster.

Example 2-21 Cluster node startup and verify

```
root@gpfs2740: /> mmstartup -N gpfs2740
Mon Sep 7 16:06:03 EDT 2015: mmstartup: Starting GPFS ...
root@gpfs2740: />
```

```
root@gpfs2750: /> mmgetstate -L -a
```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs2740	1	2	3	active	quorum node
2	gpfs2750	1	2	3	active	quorum node
3	rhel2750	1	2	3	active	

2.11.2 Update the second AIX cluster node

The same sequence of steps is followed to update the second cluster node. There is no impact to the client node's connection to the IBM Spectrum Scale file system because the nodes were updated in sequence.

2.11.3 Update the client cluster node

The third node in our example cluster is the Linux client node. The same high-level steps are required as undertaken for the two AIX cluster nodes, with several additions:

- ▶ Prior to issuing **mmshutdown** against the Linux client node, we use **mmumount** to cleanly unmount the IBM Spectrum Scale file system.
- ▶ We gain an additional step (to reboot Linux prior to restarting IBM Spectrum Scale) after reviewing the output from **mmfsenv**.
- ▶ After updating the Linux filesets, we suggest rebuilding the IBM Spectrum Scale portability layer.

Example 2-22 shows the output from the **mmunmount**, **mmshutdown**, and **mmfsenv** commands.

Example 2-22 Clean shutdown of Linux client

```
[root@rhel2750 ~]# mmumount /shared
Mon Sep 7 16:40:57 EDT 2015: mmumount: Unmounting file systems ...
```

```
root@gpfs2740: /> mmshutdown -N rhel2750
Mon Sep 7 16:42:12 EDT 2015: mmshutdown: Starting force unmount of GPFS file systems
Mon Sep 7 16:42:17 EDT 2015: mmshutdown: Shutting down GPFS daemons
rhel2750: Shutting down!
rhel2750: 'shutdown' command about to kill process 22910
```

```

rhel2750: Unloading modules from /lib/modules/3.10.0-229.el7.ppc64/extra
rhel2750: Unloading module mmfs26
rhel2750: Unloading module mmfslinux
Mon Sep 7 16:42:25 EDT 2015: mmshutdown: Finished

```

```

[root@rhel2750 ~]# mmfsenv -u
Unloading modules from /lib/modules/3.10.0-229.el7.ppc64/extra
Unloading module tracedev
rmmmod: ERROR: Module tracedev is in use
mmfsenv: Error unloading module tracedev.

```

The documentation states that if **mmfsenv** reports that it cannot unload the kernel extensions, a reboot will be required after the installation. In certain scenarios, identifying and stopping the cause is possible if a reboot is not necessary.

The required RPM filesets were installed. The next step, shown in Example 2-23, was to rebuild the portability layer.

Example 2-23 mmbuildgpl output

```

[root@rhel2750 ptf]# mmbuildgpl
-----
mmbuildgpl: Building GPL module begins at Mon Sep 7 16:54:14 EDT 2015.
-----
Verifying Kernel Header...
  kernel version = 31000229 (3.10.0-229.el7.ppc64, 3.10.0-229)
  module include dir = /lib/modules/3.10.0-229.el7.ppc64/build/include
  module build dir   = /lib/modules/3.10.0-229.el7.ppc64/build
  kernel source dir  = /usr/src/linux-3.10.0-229.el7.ppc64/include
  Found valid kernel header file under /usr/src/kernels/3.10.0-229.el7.ppc64/include
Verifying Compiler...
  make is present at /bin/make
  cpp is present at /bin/cpp
  gcc is present at /bin/gcc
  g++ is present at /bin/g++
  ld is present at /bin/ld
Verifying Additional System Headers...
  Verifying kernel-headers is installed ...
    Command: /bin/rpm -q kernel-headers
    The required package kernel-headers is installed
make World ...
make InstallImages ...
-----
mmbuildgpl: Building GPL module completed successfully at Mon Sep 7 16:54:31 EDT 2015.
-----

```

We rebooted Linux, then IBM Spectrum Scale restarted automatically (based on our configuration) on reboot, and the client rejoined the cluster as shown in Example 2-24.

Example 2-24 Post-reboot cluster status

```

root@gpfs2740: /> mmgetstate -L -a

```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs2740	1	2	3	active	quorum node
2	gpfs2750	1	2	3	active	quorum node
3	rhel2750	1	2	3	active	

2.11.4 Final update steps

After updating all cluster nodes and clients to the same level, the following step must be completed to migrate the cluster configuration data and enable the use of any added functionality.

Example 2-25 details the output from `mmlsconfig` before and after the issuing the `mmchconfig` command.

Example 2-25 Output from mmlsconfig and mmchconfig

```
root@gpfs2740: /> mmlsconfig
Configuration data for cluster GPFS.SPECTRUM:
-----
clusterName GPFS.SPECTRUM
clusterId 11300267533712031687
uidDomain SCALE
dmapifileHandleSize 32
minReleaseLevel 4.1.0.4
ccrEnabled yes
tiebreakerDisks NSD001;NSD002;NSD003
autoload yes
adminMode central

File systems in cluster GPFS.SPECTRUM:
-----
/dev/msFS

root@gpfs2740: /> mmchconfig release=LATEST
Verifying that all nodes in the cluster are up-to-date ...
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
              affected nodes. This is an asynchronous process.
root@gpfs2740: /> mmlsconfig
Configuration data for cluster GPFS.SPECTRUM:
-----
clusterName GPFS.SPECTRUM
clusterId 11300267533712031687
uidDomain SCALE
dmapifileHandleSize 32
ccrEnabled yes
tiebreakerDisks NSD001;NSD002;NSD003
autoload yes
minReleaseLevel 4.1.1.0
adminMode central

File systems in cluster GPFS.SPECTRUM:
-----
/dev/msFS
```

Our sample cluster has now been updated to version 4.1.1 and we are in a position to take advantage of new functionality and features.



Case scenario: Protocol node

This chapter contains the step-by-step deployment of two protocol nodes in a mixed AIX and Linux IBM Spectrum Scale cluster.

This chapter also details the configuration steps to share a directory on the IBM Spectrum Scale structure using SMB protocol.

The following topic is discussed in this chapter:

- ▶ Configuring protocol nodes
- ▶ Creating the SMB export

3.1 Configuring protocol nodes

Using Chapter 2, “IBM Spectrum Scale implementation” on page 7, configure two additional Linux nodes on the cluster, and proceed to configure the nodes `rhel3750` and `rhel3740` as cluster export nodes for the SMB protocol, as shown in Example 3-1.

Example 3-1 Nodes configured in the Spectrum Scale (GPFS) cluster

```
# mmlsnode -a
GPFS nodeset      Node list
-----
GPFS              gpfs3740 gpfs3750 rhel3750 rhel3740
```

3.1.1 Changing node license

You must change the license in the nodes because the Cluster Export Services (CES) nodes must be designated as *servers* on the IBM Spectrum Scale cluster. Example 3-2 shows the command to change the licensing configuration.

Example 3-2 Changing node licenses

```
# mmchlicense server -N rhel3740,rhel3750
The following nodes will be designated as possessing server licenses:
    rhel3740
    rhel3750
Please confirm that you accept the terms of the IBM Spectrum Scale server
Licensing Agreement.
The full text can be found at www.ibm.com/software/sla
Enter "yes" or "no": yes
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

3.1.2 Installing prerequisites

Some RPM packages must be installed as prerequisites for the NFS-Ganesha, SMB, and Swift RPMs. See the Red Hat documentation for how to create an RPM repository from the DVD media or installation ISO file.

Example 3-3 shows the `yum` command to install the prerequisites.

Example 3-3 Installing protocols prerequisites

```
# yum install nfs-utils boost-regex PyQt4
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use
subscription-manager to register.
Package boost-regex-1.53.0-23.el7.ppc64 already installed and latest version
Resolving Dependencies
--> Running transaction check
---> Package PyQt4.ppc64 0:4.10.1-13.el7 will be installed
--> Processing Dependency: sip-api(9) >= 9.2 for package:
PyQt4-4.10.1-13.el7.ppc64
```

```
--> Processing Dependency: libphonon.so.4()(64bit) for package:
PyQt4-4.10.1-13.e17.ppc64
(...)
Installed:
  PyQt4.ppc64 0:4.10.1-13.e17 nfs-utils.ppc64 1:1.3.0-0.8.e17
Dependency Installed:
(...)
Complete!
```

3.1.3 Installing the IBM Spectrum Scale protocol rpms

Chapter 1, “Introduction to IBM Spectrum Scale 4.1.1” on page 1 indicates that IBM Spectrum Scale protocols use open source packages. If these packages are already installed in the nodes with different versions, these packages must be removed and then installed from the IBM Spectrum Scale installation directory.

After downloading the IBM Spectrum Scale software, and executing the extracted file for accepting the license agreement in the `/usr/lpp/mmfs/<Version>/` directory, you now have the following directories:

- ▶ `ganesha_rpms`
- ▶ `gpfs_rpms`
- ▶ `installer`
- ▶ `object_rpms`
- ▶ `smb_rpms`
- ▶ `zimon_rpms`

Because our cluster is already installed, go to the `smb_rpms`, `zimon_rpms`, and `ganesha_rpms` directories and run the `rpm` command to install all GPFS SMB protocol filesets, performance sensor, and collectors used by `mmperfmon` plus `nfs-ganesha` RPMs as shown in Example 3-4.

Example 3-4 Installing protocol RPMs

```
# rpm -ivh gpfs.smb*.ppc64.rpm
Preparing... ##### [100%]
Updating / installing...
 1:gpfs.smb-0:4.2.1_gpfs_27-1.e17 ##### [100%]

# rpm -ivh gpfs.gss.pmsensors-4.1.0-8.e17.ppc64.rpm \
gpfs.gss.pmcollector-4.1.0-8.e17.ppc64.rpm
Preparing... ##### [100%]
Updating / installing...
 1:gpfs.gss.pmcollector-4.1.0-8.e17 ##### [ 50%]
 2:gpfs.gss.pmsensors-4.1.0-8.e17 ##### [100%]

# rpm -ivh nfs-ganesha*.ppc64.rpm nfs-ganesha-gpfs*.ppc64.rpm \
nfs-ganesha-utils*.ppc64.rpm
Preparing... ##### [100%]
Updating / installing...
 1:nfs-ganesha-2.2.0-0.2ibm1.e17 ##### [ 33%]
 2:nfs-ganesha-gpfs-2.2.0-0.2ibm1.e17 ##### [ 67%]
 3:nfs-ganesha-utils-2.2.0-0.2ibm1.e17 ##### [100%]
```

3.1.4 Enabling Cluster Export Services and configuring CES IP

Before enabling the Cluster Export Services (CES), define the CES shared root (cesSharedRoot) which is required for storing CES shared configuration data, protocol recovery, and for some other protocol specific purpose. The shared root is part of the cluster export configuration and is shared between the protocols. Every CES node requires access to the path configured as shared root.

The cesSharedRoot is monitored by **mmcesmonitor**. If the shared root is not available, the CES node list (**mmces node list**) will show "no-shared-root" and a failover is triggered.

Example 3-5 shows the command to define the cesSharedRoot and after the output of the **mmclsconfig** command.

Example 3-5 Defining the cesSharedRoot

```
# mmchconfig cesSharedRoot=/gpfs2
mmchconfig: Command successfully completed
mmchconfig: 6027-1371 Propagating the cluster configuration data to all
    affected nodes. This is an asynchronous process.
# mmclsconfig
Configuration data for cluster GPFS.SPECTRUM:
-----
clusterName GPFS.SPECTRUM
clusterId 2261951228035752105
autoload no
uidDomain SCALE
dmapifileHandleSize 32
ccrEnabled yes
tiebreakerDisks NSD001;NSD002;NSD003
minReleaseLevel 4.1.1.0
cesSharedRoot /gpfs2
adminMode central
(...)
```

After installing the filesets, as shown in Example 3-6, you can see the commands to enable CES on the nodes.

Example 3-6 Enabling CES

```
# mmchnode --ces-enable -N rhel3740,rhel3750
Mon Aug 17 14:44:34 EDT 2015: mmchnode: Processing node rhel3740
Mon Aug 17 14:44:35 EDT 2015: mmchnode: Processing node rhel3750
mmchnode: Propagating the cluster configuration data to all
    affected nodes. This is an asynchronous process.
```

Then, add an IP address that will be used for the clients to access the new protocol. The IP address must be a different address than the IBM Spectrum Scale node address already in use. The address should be resolvable to a name, so after adding the IP address that is associated with a name on all nodes `/etc/hosts`, run the commands (Example 3-7) to add the CES address and enable the SMB service.

Example 3-7 Adding and checking the CES address

```
# mmces address add --ces-ip 172.16.20.136
# mmces address list
```

Node	Daemon	node name	IP address	CES IP address list

```

3   rhel3750                               172.16.20.132    172.16.20.136
4   rhel3740                               172.16.20.135

```

```

# mmces service enable SMB
rhel3740: Redirecting to /bin/systemctl start  gpfs-ctdb.service
rhel3750: Redirecting to /bin/systemctl start  gpfs-ctdb.service
rhel3740: Wait for ctdb to become ready. State=STARTUP
rhel3750: Wait for ctdb to become ready. State=FIRST_RECOVERY
rhel3740: Wait for ctdb to become ready. State=STARTUP
rhel3750: Wait for ctdb to become ready. State=STARTUP
rhel3750: mmchconfig: Command successfully completed
rhel3750: mmchconfig: Propagating the cluster configuration data to all
rhel3750:   affected nodes. This is an asynchronous process.
rhel3750: Redirecting to /bin/systemctl start  gpfs-smb.service
rhel3740: Redirecting to /bin/systemctl start  gpfs-smb.service
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
            affected nodes. This is an asynchronous process.

```

```

# mmlscluster --ces

```

```

GPFS cluster information
=====

```

```

    GPFS cluster name:      GPFS.SPECTRUM
    GPFS cluster id:       14565899132146859664

```

```

Cluster Export Services global parameters
-----

```

```

    Shared root directory:      /gpfs02
    Enabled Services:           SMB
    Log level:                   0
    Address distribution policy: even-coverage

```

Node	Daemon node name	IP address	CES IP address list
3	rhel3750	172.16.20.132	172.16.20.136
4	rhel3740	172.16.20.135	None

The address is associated with one of the nodes as seen in the output of the `mmces address` command (Example 3-7 on page 28). This is the highly available address that is managed by the IBM Spectrum Scale cluster services. You can also see the output of the `mmlscluster --ces` command with the SMB enabled.

3.2 Creating the SMB export

The product manuals have a recommendation for creating SMB exports on independent filesets, so Example 3-8 shows the steps to create a fileset in an IBM Spectrum Scale export.

Example 3-8 Creating the fileset

```

# mmcrfileset FS02 fileset --inode-space=new
Fileset fileset created with id 1 root inode 131075.
# mmlinkfileset FS02 fileset -J /gpfs02/fileset
Fileset fileset linked at /gpfs02/fileset

```

```
# mkdir /gpfs02/fileset/smb
```

SMB exports must have ACL configured to support NFSv4. An authentication service must be configured. Example 3-9 shows these three steps. The permissions of the directory used as the SMB export are also adjusted.

Example 3-9 Changing ACL, enabling authentication, and creating the export

```
# mmchfs FS02 -k nfs4
```

```
# mmuserauth service create --type userdefined --data-access-method file
File authentication configuration completed successfully.
```

```
# mmsmb export add smbexport /gpfs02/fileset/smb
```

```
mmsmb export add: The SMB export was created successfully.
```

```
# mmsmb export list
```

export	path	guest ok	smb encrypt
smbexport	/gpfs02/fileset/smb	no	auto

```
# chmod 777 /gpfs02/fileset/smb
```

Note: Production environments usually have an Active Directory, LDAP, or NIS+ service available to authenticate users. Our test environment used a simple file-based authentication.

SMB does not use the operating system authentication. When using authentication that is file-based, different users and passwords must be declared. The `gpfs.smb` fileset gives all necessary tools to manage local authentication. Example 3-10 on page 30 shows local users (`smbusr1` and `smbusr2`) on the `/etc/passwd` file and the command to *add* and declare password to those users.

Example 3-10 SMB users

```
# tail -n5 /etc/passwd
```

```
rpc:x:32:32:Rpcbind Daemon:/var/lib/rpcbind:/sbin/nologin
```

```
rpcuser:x:29:29:RPC Service User:/var/lib/nfs:/sbin/nologin
```

```
nfsnobody:x:65534:65534:Anonymous NFS User:/var/lib/nfs:/sbin/nologin
```

```
smbusr1:x:500:1001::/home/smbusr1:/bin/bash
```

```
smbusr2:x:501:1002::/home/smbusr2:/bin/bash
```

```
# /usr/lpp/mmfs/bin/smbpasswd -a smbusr1
```

```
New SMB password:
```

```
Retype new SMB password:
```

```
Added user smbusr1.
```

```
# /usr/lpp/mmfs/bin/smbpasswd -a smbusr2
```

```
New SMB password:
```

```
Retype new SMB password:
```

```
Added user smbusr2.
```

3.2.1 Testing SMB access

Using a Windows workstation, on the address bar of the browser point to the address that is associated with the CES cluster to see the `smbexport` that was just created (Figure 3-1).

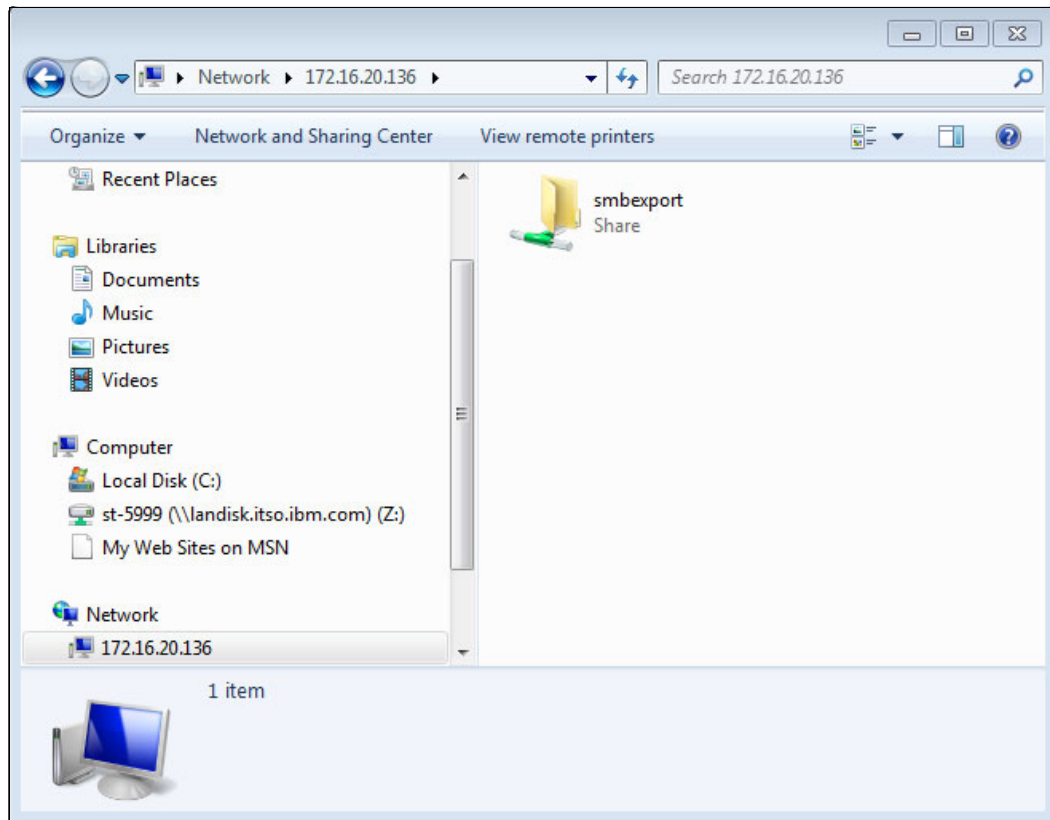


Figure 3-1 smbexport share on Windows Explorer

After clicking on the **smbexport**, the typical Windows authentication panel opens. Using the user and password that was created with the **smbpasswd** command, copy a file to the share as shown in Figure 3-2.

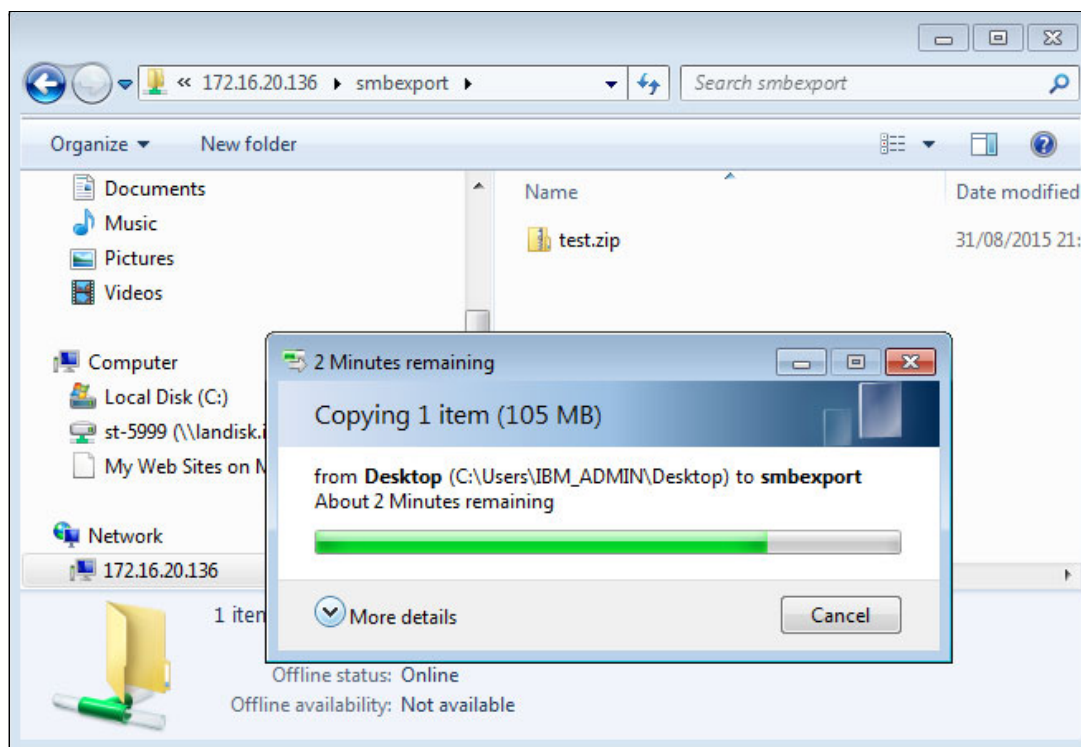


Figure 3-2 Copying the file

In the protocol nodes, you can see the files copied from the Windows workstation, as shown in Example 3-11.

Example 3-11 Files in Spectrum Scale

```
# ls -l /gpfs02/fileset/smb
total 117272
-rwxr--r--. 1 smbusr2 smbusr2 9321173 Jul 30 13:21 test1.zip
-rwxr--r--. 1 smbusr1 smbusr1 110761076 Oct 26 2013 test.zip
```

The authenticated user on the workstation will be the owner of the files in the node. Workstations and servers can now access the data by using the native CIFS protocol, and have all the availability and scalability advantages of the typical Spectrum Scale clusters.



Case scenario: IBM Spectrum Scale AFM-based disaster recovery

This chapter provides information about configuration scenarios using the IBM Spectrum Scale active file management (AFM)-based disaster recovery feature and IBM Spectrum Scale new feature integration and interoperability methods.

The following topics are discussed in this chapter:

- ▶ Introduction to AFM-based disaster recovery
- ▶ AFM-based disaster recovery: Implementation scenario
- ▶ Hadoop configuration using shared storage

This chapter also presents step-by-step configurations, active file management async disaster recovery implementation considerations using NFS and Network Shared Disk protocols.

4.1 Introduction to AFM-based disaster recovery

The active file management feature of IBM Spectrum Scale continues to expand its data protection configuration options for disaster recovery (DR) scenarios by introducing the IBM Spectrum Scale feature AFM-based disaster recovery (AFM async DR).

AFM async DR uses asynchronous data replication between two sites. The replication mechanism is based on AFM fileset level replication DR capability with a strict one-to-one active-passive model. In this model, the sites are represented as being primary and secondary, and the replicated filesets are mapped one-to-one.

As long as the active file management masks wide area network latencies and outages by using IBM Spectrum Scale to cache massive data sets and providing asynchronous data movement between Cache and Home, the AFM async DR feature is specialized on the accomplishment of business recovery objectives, targeting disaster recovery scenarios by replicating all data asynchronously from a primary site to a secondary site.

A disaster recovery solution takes into account two important business-critical parameters:

- ▶ **Recovery time objective (RTO):** Represents the amount of time required for an application to fail over and be operational when a disaster occurs.
- ▶ **Recovery point objective (RPO):** Represents the point in time relative to the failure for which data for an application might be lost.

The RPO and RTO parameters provide the required input for the necessary disaster recovery planning and solution sizing. In accordance with the RTO and RPO values, the solutions parameters can be defined, and the required system resources, the volume of data generated for a given time and the network bandwidth, can practically provide all the solution requirements to fulfill the business criteria.

In AFM async DR, the applications are currently running on the primary site using a read-write fileset. The secondary site (the DR site) should be read-only and not be used for direct writes under normal conditions.

The primary and secondary filesets that are configured for async DR can be created so they are independent of each other in terms of storage and network configurations. The AFM async DR allows use of an existent or a new fileset having the configured mode as primary (the replication data source). The target fileset located on the secondary site should be an empty fileset. After the filesets are identified or created, an AFM async DR relationship can be established.

How it works

The way it works is that a consistent point-in-time view of the data in the primary fileset is propagated inline to the secondary fileset with the use of fileset-based snapshots (psnaps).

AFM async DR uses the RPO snapshots to maintain consistency between primary and secondary, specifying the frequency of taking these snapshots and sending alerts when the set RPO is not achieved. These automatic RPO snapshots are based on the afmRPO setting.

At any point in time, two RPO snapshots apart from psnap0 are retained on both sides. Deletions from the primary fileset are done and queued when new RPO snapshots arrive.

RPO snapshots can be created also by the user taking into account that these snapshots are not deleted as part of a subsequent RPO snapshot delete process. The command used for RPO snapshot creation is `mmsnap create` with the `--rpo` option.

The RPO snapshot pushes all of the snapshot data in primary to the secondary fileset so that the fileset data is consistent across the sites, then takes a snapshot of the corresponding secondary fileset data. This results in a pair of peer snapshots, one each at the primary and secondary filesets, that refer to the same consistent copy.

What is transferred

All file user data, metadata (including user-extended attributes except inode number and atime), hard links, renames, and clones from the primary are replicated to the secondary. All file system and fileset related attributes such as user, group and fileset quotas, replication factors, dependent filesets from the primary are not replicated to the secondary.

Data replication between primary and secondary sites is performed asynchronously, and only the modified blocks for data and metadata are transferred.

Minimum RPO time is set to 15 minutes and is set as the fileset attribute `afmRPO=15` on the primary site, and indicates the amount of data loss that can be tolerated in the event of failure. Each RPO interval triggers a fileset or cache level snapshot, which results in the file systems being quiesced.

AFM async DR provides capabilities for failback and failover between sites for the configured filesets. When a failover occurs, the secondary site's filesets data can be restored to the state of last consistent RPO snapshot, which is the default, or, by using the `--norestore` option, the data can be available on the secondary site as is.

Note: Current considerations for using AFM and AFM async DR functions can be found at the following web page:

<http://ibm.co/1Ky1WnC>

4.2 AFM-based disaster recovery: Implementation scenario

Disaster recovery scenarios can be complex and directly dependent on the number of systems and applications that are required to interact when a failover occurs to the disaster recovery site. The data availability and data consistency on the secondary site is mandatory.

Starting with IBM Spectrum Scale 4.1.1, AFM async DR offers a specialized mechanism to provide a full disaster recovery solution for the IBM Spectrum Scale file systems.

In this scenario, an AFM async DR configuration uses IBM Spectrum Scale protocol. On each site, an IBM Spectrum Scale cluster is configured with a file system and an associated fileset, as shown in Figure 4-1 on page 36. The primary site holds data that must be replicated and available on the secondary site based on RPO and RTO values.

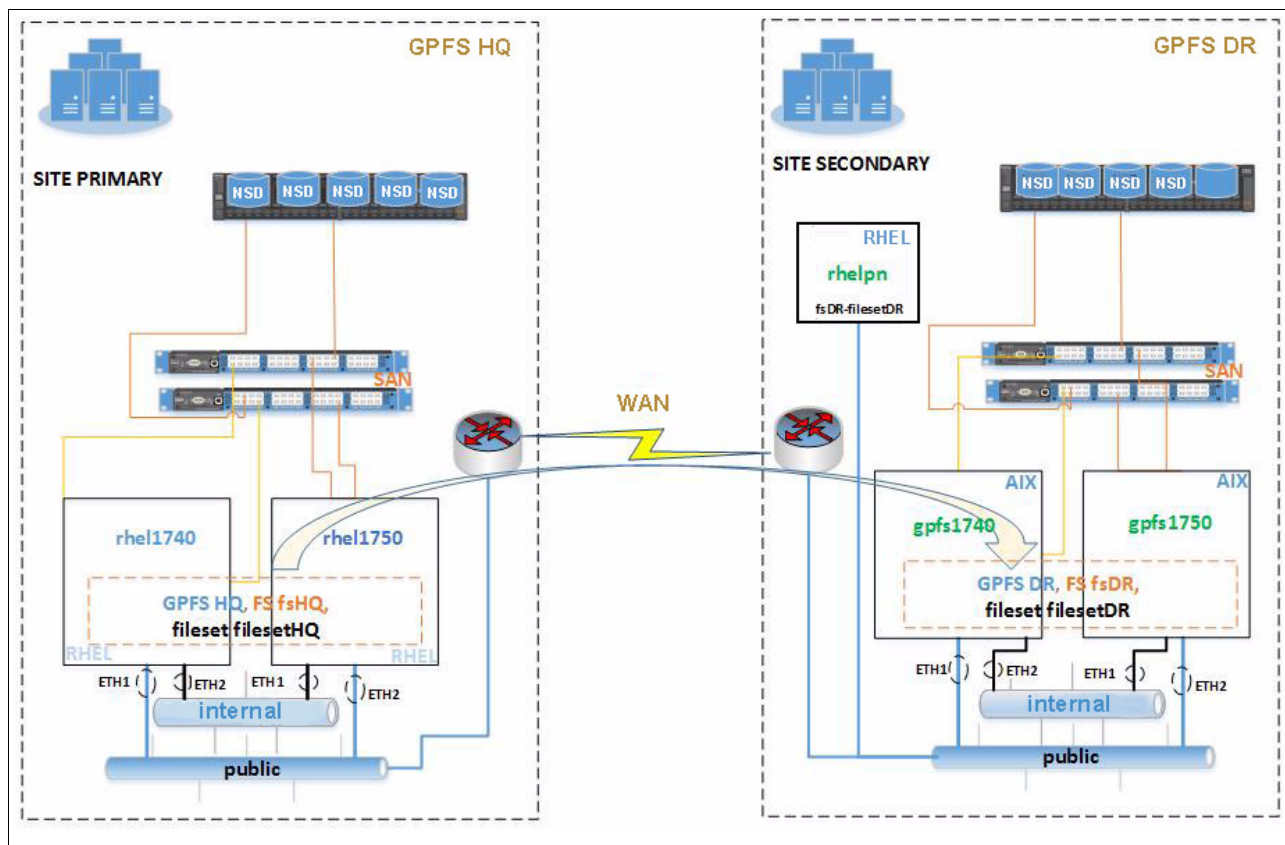


Figure 4-1 AFM async DR: Two sites, one fileset replication

The scenario aims to accomplish a complete fileset failover from the primary site to the secondary site. Two IBM Spectrum Scale clusters are configured, based on Red Hat Enterprise Linux PPC 64 on the primary site and AIX on the secondary site. On the secondary site, only one RHEL machine is configured as gateway. For redundancy, the number of gateway nodes per site should be at least two.

The LPARs are configured on two IBM Power 770 servers with each LPAR allocated two CPUs and 16 GB RAM. The LPARs are connected to SAN by NPIV through the Virtual I/O Server (VIOS). The storage used in our configuration is IBM Storwize® v7000.

Table 4-1 indicates details of the LPARs.

Table 4-1 LPARs details

LPAR	RHEL1740	RHEL1750	GPFS1740	GPFS1750	RHELPN
OS	RHEL 7.1 PPC64	RHEL 7.1 PPC64	AIX 7.1	AIX 7.1	RHEL 7.1 PPC64
IP	172.16.20.133	172.16.20.130	172.16.20.124	172.16.20.127	172.20.16.20.138
Shared disks (NSDs)	0035, 0033, 0036, 003a	0035, 0033, 0036, 003a	000C, 000D, 000E, 000F	000C, 000D, 000E, 000F	
GPFS file system or fileset	HQ/filesetHQ	HQ/filesetHQ	DR/filesetDR	DR/filesetDR	DR/filesetDR
Role	quorum-manager	quorum-manager	quorum-manager	quorum-manager	client, gateway, quorum Protocol-Node

Considering the operating systems installed, the next configuration steps are as follows:

1. Installation planning
2. Preparing the environment
3. IBM Spectrum Scale installation and configuration on the RHEL and AIX nodes
4. Configuring User Equivalence on each cluster pair
5. Configuring the Spectrum Scale clusters
6. Planning cluster file systems configuration
7. Identifying disks and configuring NSD disks for each LPAR in the cluster pair
8. Configuring the file systems
9. Configuring remote cluster authentication or configuring NFS
10. Configuring AFM Async DR
 - Creating and configuring filesets for AFM Async DR
 - Convert filesets for AFM Async DR
11. Failover to secondary site
12. Failback to primary site
13. Monitoring
14. Protocols disaster recovery

4.2.1 IBM Spectrum Scale installation planning

Through planning is required for any solution deployment. IBM Spectrum Scale does not fall in a different category. The application requirements dictates in the most part how the backend components which relies upon are configured. IBM Spectrum Scale offers flexibility, scalability and various types of configurations to fulfil the applications requirements.

IBM Spectrum Scale requires planning by identifying these items:

- ▶ Application specifics such as storage access pattern, I/O sizes, number of I/Os, volume of data required, and growing factors.
- ▶ The appropriate configuration for the Spectrum Scale cluster configuration mode.
- ▶ Network configuration: Administrative network, daemon communication, protocols network.
- ▶ Storage configuration: Disk configuration, size of the disks, NSD configurations.
- ▶ File system configuration
- ▶ Advanced Spectrum Scale features such as encryption
- ▶ Optimization for the Spectrum Scale features enablement
- ▶ Tuning for performance
- ▶ Security

Regarding those items, you can find more information in the following resources:

- ▶ IBM Knowledge Center for IBM Spectrum Scale
<http://ibm.co/1QbrHhC>
- ▶ IBM Spectrum Scale FAQ
<http://ibm.co/1ysyp8w>
- ▶ IBM Spectrum Scale Wiki
<http://ibm.co/1aKwtP5>
- ▶ *IBM Spectrum Scale (formerly GPFS), SG24-8254*
<http://www.redbooks.ibm.com/abstracts/sg248254.html>

4.2.2 Preparing the environment - installing the prerequisites

The first step for the IBM Spectrum Scale installation is to validate and verify the IBM Spectrum Scale installation prerequisites on the target operating systems. The information regarding the compatibility details and prerequisite requirements for various operating systems are located at the following web page:

<http://ibm.co/1MHeAqs>

In our environment, the prerequisites for Red Hat Enterprise Linux (RHEL) are shown in Example 4-1. The RHEL installation has been performed as *Server with the GUI*.

Example 4-1 For this setup install packages for RHEL PPC 64

```
yum install gcc-c++.ppc64 gcc.ppc64 cpp.ppc64 kernel-devel m4 ksh
Dependencies Resolved
```

Dependencies Resolved

Package	Arch	Version	Repository	Size
Installing:				
cpp	ppc64	4.8.3-9.el7	dvd	6.9 M
gcc	ppc64	4.8.3-9.el7	dvd	15 M
gcc-c++	ppc64	4.8.3-9.el7	dvd	8.2 M
kernel-devel	ppc64	3.10.0-229.el7	dvd	9.9 M
ksh	ppc64	20120801-22.el7	dvd	845 k
m4	ppc64	1.4.16-9.el7	dvd	256 k
Installing for dependencies:				
glibc-devel	ppc64	2.17-78.el7	dvd	1.1 M
glibc-headers	ppc64	2.17-78.el7	dvd	645 k
kernel-headers	ppc64	3.10.0-229.el7	dvd	2.3 M
libmpc	ppc64	1.0.1-3.el7	dvd	52 k
libstdc++-devel	ppc64	4.8.3-9.el7	dvd	1.5 M
mpfr	ppc64	3.1.1-4.el7	dvd	210 k

Transaction Summary

Install 6 Packages (+6 Dependent packages)

```
[root@rhel1750 4.1.1]# rpm -ivh gpfs.msg.en_US-4.1.1-0.noarch.rpm gpfs.ext-4.1.1-0.ppc64.rpm
gpfs.gskit-8.0.50-40.ppc64.rpm gpfs.base-4.1.1-0.ppc64.rpm gpfs.gpl-4.1.1-0.noarch.rpm
gpfs.docs-4.1.1-0.noarch.rpm
[root@rhel1740 4.1.1]# rpm -ivh gpfs.msg.en_US-4.1.1-0.noarch.rpm gpfs.ext-4.1.1-0.ppc64.rpm
gpfs.gskit-8.0.50-40.ppc64.rpm gpfs.base-4.1.1-0.ppc64.rpm gpfs.gpl-4.1.1-0.noarch.rpm
gpfs.docs-4.1.1-0.noarch.rpm
Preparing... ##### [100%]
Updating / installing...
 1:gpfs.base-4.1.1-0 ##### [ 17%]
 2:gpfs.ext-4.1.1-0 ##### [ 33%]
 3:gpfs.gpl-4.1.1-0 ##### [ 50%]
 4:gpfs.docs-4.1.1-0 ##### [ 67%]
 5:gpfs.gskit-8.0.50-40 ##### [ 83%]
 6:gpfs.msg.en_US-4.1.1-0 ##### [100%]
```

Note: Installing the IBM Spectrum Scale packages without having all the prerequisite packages in place will give you a message listing the missing packages. Then, you must install the missing packages.

Installing IBM Spectrum Scale packages

Installing IBM Spectrum packages is a straightforward process. Depending on your configuration, whether or not the X server is installed, you can choose how the packages are decompressed. The IBM Spectrum Scale packages include an installer, which requires that the licence be accepted. Therefore, if the X server is running, the installer will open the licence acceptance window in the X window; otherwise the `--text-only` (double-dash) switch must be added when you decompress the package. In Example 4-2, the IBM Spectrum Scale package is decompressed with the `--text-only` switch.

Example 4-2 IBM Spectrum Scale license acceptance

```
[root@rhel1750 kits]# ./Spectrum_Scale_install-4.1.1.0_ppc64_advanced --text-only
```

```
Extracting License Acceptance Process Tool to /usr/lpp/mmfs/4.1.1 ...
```

```
tail -n +479 ./Spectrum_Scale_install-4.1.1.0_ppc64_advanced | /bin/tar -C /usr/lpp/mmfs/4.1.1 -xvz  
--exclude=*rpm --exclude=*tgz --exclude=*deb 2> /dev/null 1> /dev/null
```

```
Installing JRE ...
```

```
tail -n +479 ./Spectrum_Scale_install-4.1.1.0_ppc64_advanced | /bin/tar -C /usr/lpp/mmfs/4.1.1 --wildcards -xvz  
ibm-java*tgz 2> /dev/null 1> /dev/null
```

```
Invoking License Acceptance Process Tool ...
```

```
/usr/lpp/mmfs/4.1.1/ibm-java-ppc64-71/jre/bin/java -cp /usr/lpp/mmfs/4.1.1/LAP_HOME/LAPApp.jar  
com.ibm.lex.lapapp.LAP -l /usr/lpp/mmfs/4.1.1/LA_HOME -m /usr/lpp/mmfs/4.1.1 -s /usr/lpp/mmfs/4.1.1 -text_only  
International Program License Agreement
```

Part 1 - General Terms

BY DOWNLOADING, INSTALLING, COPYING, ACCESSING, CLICKING ON AN "ACCEPT" BUTTON, OR OTHERWISE USING THE PROGRAM, LICENSEE AGREES TO THE TERMS OF THIS AGREEMENT. IF YOU ARE ACCEPTING THESE TERMS ON BEHALF OF LICENSEE, YOU REPRESENT AND WARRANT THAT YOU HAVE FULL AUTHORITY TO BIND LICENSEE TO THESE TERMS. IF YOU DO NOT AGREE TO THESE TERMS,

* DO NOT DOWNLOAD, INSTALL, COPY, ACCESS, CLICK ON AN "ACCEPT" BUTTON, OR USE THE PROGRAM; AND

* PROMPTLY RETURN THE UNUSED MEDIA, DOCUMENTATION, AND

Press Enter to continue viewing the license agreement, or enter "1" to accept the agreement, "2" to decline it, "3" to print it, "4" to read non-IBM terms, or "99" to go back to the previous screen.

1

License Agreement Terms accepted.

```
Extracting Product RPMs to /usr/lpp/mmfs/4.1.1 ...
```

```
tail -n +479 ./Spectrum_Scale_install-4.1.1.0_ppc64_advanced | /bin/tar -C /usr/lpp/mmfs/4.1.1 --wildcards -xvz  
gpfs.base-4.1.1-0.ppc64.rpm gpfs.crypto-4.1.1-0.ppc64.rpm gpfs.docs-4.1.1-0.noarch.rpm gpfs.ext-4.1.1-0.ppc64.rpm  
gpfs.gpl-4.1.1-0.noarch.rpm gpfs.gskit-8.0.50-40.ppc64.rpm gpfs.hadoop-2-connector-4.1.1-0.ppc64.rpm  
gpfs.msg.en_US-4.1.1-0.noarch.rpm manifest 2> /dev/null 1> /dev/null
```

- gpfs.base-4.1.1-0.ppc64.rpm
- gpfs.crypto-4.1.1-0.ppc64.rpm
- gpfs.docs-4.1.1-0.noarch.rpm
- gpfs.ext-4.1.1-0.ppc64.rpm
- gpfs.gpl-4.1.1-0.noarch.rpm
- gpfs.gskit-8.0.50-40.ppc64.rpm
- gpfs.hadoop-2-connector-4.1.1-0.ppc64.rpm
- gpfs.msg.en_US-4.1.1-0.noarch.rpm

- manifest

Removing License Acceptance Process Tool from /usr/lpp/mmfs/4.1.1 ...

rm -rf /usr/lpp/mmfs/4.1.1/LAP_HOME /usr/lpp/mmfs/4.1.1/LA_HOME

Removing JRE from /usr/lpp/mmfs/4.1.1 ...

rm -rf /usr/lpp/mmfs/4.1.1/ibm-java*tgz

=====

The packages are extracted in /usr/lpp/mmfs/4.1.1.

Licenses and extracted packages

The extracted packages in our case represent all the packages that are required for the highest available IBM Spectrum Scale license.

The licenses available in IBM Spectrum Scale are as follows:

- ▶ **GPFS Express Edition**
Available on AIX, Linux, and Windows. Provides the base GPFS functions.
- ▶ **GPFS Standard Edition**
Available on AIX, Linux, and Windows. Provides extended features in addition to the base GPFS functions that are provided in the GPFS Express Edition.
 - On AIX and Linux, the extended features include Information Lifecycle Management (ILM), active file management (AFM), and Clustered NFS (CNFS).
 - On Windows, the extended features include limited ILM.
- ▶ **GPFS Advanced Edition**
Available on AIX and Linux. Provides high-level data protection using the GPFS cryptographic subsystem. For additional information, see the “Encryption” topic in the *GPFS: Advanced Administration Guide*.

The extracted packages are shown in Table 4-2

Table 4-2 Extracted packages and description

Package	Description
gpfs.base-4.1.1-0.ppc64.rpm	General Parallel File System File Manager
gpfs.crypto-4.1.1-0.ppc64.rpm	General Parallel File System Cryptographic subsystem
gpfs.docs-4.1.1-0.noarch.rpm	General Parallel File System Server Manpages and Documentation
gpfs.ext-4.1.1-0.ppc64.rpm	General Parallel File System Extended Features
gpfs.gpl-4.1.1-0.noarch.rpm	General Parallel File System Open Source Modules The GPFS portability layer is a loadable kernel module that allows the GPFS daemon to interact with the operating system.
gpfs.gskit-8.0.50-40.ppc64.rpm	General Parallel File System GSKit Cryptography Runtime

Package	Description
gpfs.hadoop-2-connector-4.1.1-0.ppc64.rpm	General Parallel File System Hadoop connector 1.x
gpfs.msg.en_US-4.1.1-0.noarch.rpm	General Parallel File System server message catalog; US English

Install the IBM Spectrum Scale packages according to the owned license

The next step is the IBM Spectrum Scale package installations. Depending on the owned license type, the packages are installed with **rpm** commands as shown in Example 4-3. In our case, the AFM async DR feature is installed, and all the IBM Spectrum Scale packages are installed.

Note: For AFM async DR, the IBM Spectrum Scale advanced edition license is required.

AFM primary/secondary filesets cannot be created in the currently installed version.
mmcrfileset: Command failed. Examine previous error messages to determine cause.

The gpfs.crypto-4.1.1-0.ppc64.rpm package is installed:

```
[root@rhel1740 4.1.1]# rpm -ivh gpfs.crypto-4.1.1-0.ppc64.rpm
Preparing...                               ##### [100%]
Updating / installing...
 1:gpfs.crypto-4.1.1-0                      ##### [100%]
*****
Please restart GPFS daemon to use GPFS encryption functionality
*****
```

Example 4-3 Installing GPFS packages

```
[root@rhel1740 4.1.1]# rpm -ivh gpfs.msg.en_US-4.1.1-0.noarch.rpm
gpfs.ext-4.1.1-0.ppc64.rpm gpfs.gskit-8.0.50-40.ppc64.rpm
gpfs.base-4.1.1-0.ppc64.rpm gpfs.gpl-4.1.1-0.noarch.rpm
gpfs.docs-4.1.1-0.noarch.rpm
Preparing...                               ##### [100%]
Updating / installing...
 1:gpfs.base-4.1.1-0                        ##### [ 17%]
 2:gpfs.ext-4.1.1-0                        ##### [ 33%]
 3:gpfs.gpl-4.1.1-0                        ##### [ 50%]
 4:gpfs.docs-4.1.1-0                      ##### [ 67%]
 5:gpfs.gskit-8.0.50-40                   ##### [ 83%]
 6:gpfs.msg.en_US-4.1.1-0                 ##### [100%]
```

On the Linux operating system, IBM Spectrum Scale requires a loadable kernel module that allows the IBM Spectrum Scale daemon to interact with the operating system. The IBM Spectrum Scale portable layer is built by the **mmbuildgp1** command as shown in “Build the IBM Spectrum Scale portability layer” on page 48.

Note: IBM Spectrum Scale Portable Layer is updated whenever Red Hat Linux Kernel is updated or a new IBM Spectrum Scale fix is applied.

IBM Spectrum Scale requires name resolution for all the nodes in the cluster and also the network time for all cluster nodes to be synchronized with the network time server.

Our cluster hosts file and NTP server is shown in Example 4-4.

Example 4-4 hosts file and NTP configuration

```
[root@rhel1pn 4.1.1]# cat /etc/hosts
172.16.20.133    rhel1740.local rhel1740
172.16.20.130    rhel1750.local rhel1750
172.16.20.124    gpfs1740.local gpfs1740
172.16.20.127    gpfs1750.local gpfs1750
172.16.20.138    rhel1pn.local rhel1pn
```

```
NTP Client /etc/ntp.conf
[root@rhel1750 ~]# cat /etc/ntp.conf
server 172.16.20.41 prefer
```

SSH user equivalence is a required step for the IBM Spectrum Scale configuration. Administrative commands and configuration synchronization across the cluster nodes is performed by using a remote shell. Use SSH instead of the RSH protocol.

SSH and network setup are general IBM Spectrum Scale prerequisites, especially when initiating the installation of the SMB, NFS, or Object protocols through the installation toolkit, which must be able to communicate by way of an internal or external network with all protocol nodes to be installed.

All nodes require SSH keys to be set up so that installation toolkit can run remote commands without password authentication.

The SSH configuration procedure is detailed in 2.10.4, “Generate SSH keys and exchange with existing AIX cluster nodes” on page 17.

With the network, user equivalence, and NTP server configured, proceed by identifying the disks that will be part of the file system as they were mapped from storage.

In our configuration, the LPARs disks are provided by IBM Storwize V7000.

The information regarding the disk systems that were tested with IBM Spectrum Scale is at the following web page:

<http://ibm.co/1JrNxNb>

Note: At the time this publication was written, placing IBM Spectrum Scale metadata on thinly provisioned or compressed volumes is not supported when using IBM Storwize V7000, V3500, V3700, and SAN Volume Controller for both AIX and Linux operating systems.

There are no special requirements for Fibre Channel switches, only those that are required by the specific operating system host attachment.

Information regarding IBM Spectrum Scale supported multipath drivers on the supported operating systems are found at the following web page:

<http://ibm.co/1JbW6Ix>

In this section, two IBM Spectrum Scale clusters are configured. One AIX and another one on RHEL PPC64. The AIX LPARs use Subsystem Device Driver Path Control Module (SDDPCM) as multipath driver as long Red Hat Enterprise Linux 7.1 uses DM-MP, its native multipath driver.

IBM Storwize v7000 disk allocations

In our environment, an IBM Storwize v7000 storage system is used. The disks are allocated to the hosts after preliminary steps are performed as follows:

- LPARs are created and use NPIV configuration. For each LPAR, the NPIV HBA *worldwide port name (WWPN) ports* are recorded for the further zoning configuration. The easiest way is to identify the corresponding WWPNs in the HMC by using the web interface or command-line interface (CLI) as shown in Example 4-5; or, IOINFO support can be used. For more information, see the following web page:

<http://ibm.co/1NMi6wB>

Example 4-5 Identifying WWPNs from HMC by CLI

```
hscroot@hmc8:~> lssyscfg -r prof -m 740-2 -F virtual_fc_adapters --filter
"lpar_names=GPFS1_740"
""16/client/2/p740_2_vio2/16/c050760582a20228,c050760582a20229/0"",""15/client
/1/p740_2_vio1/15/c050760582a20222,c050760582a20223/0""
```

- Identifying the IBM Storwize V7000 FC ports on SAN

The V7000 WWPN ports can be found by using either the CLI (Example 4-6) or the GUI (Figure 4-2 on page 44).

Example 4-6 identifying v7000 WWPNs using cli

```
IBM_Storwize:V7000-SYS-1:admin>svcinfo lspportfc
id fc_io_port_id port_id type port_speed node_id node_name WWPN nportid status attachment
cluster_use
0 1 1 fc 8Gb 1 node1 500507680215F77F 011200 active switch
local_partner
1 2 2 fc 8Gb 1 node1 500507680225F77F 011200 active switch
local_partner
2 3 3 fc N/A 1 node1 500507680235F77F 000000 inactive_unconfigured switch
local_partner
3 4 4 fc N/A 1 node1 500507680245F77F 000000 inactive_unconfigured none
local_partner
12 1 1 fc 8Gb 2 node2 500507680215F780 011300 active switch
local_partner
13 2 2 fc 8Gb 2 node2 500507680225F780 011300 active switch
local_partner
14 3 3 fc N/A 2 node2 500507680235F780 000000 inactive_unconfigured switch
local_partner
15 4 4 fc N/A 2 node2 500507680245F780 000000 inactive_unconfigured none
local_partner
```

Network

Management IP
Addresses

Service IP Addresses

Ethernet Ports

iSCSI

Fibre Channel
Connectivity

Fibre Channel Ports

Fibre Channel Ports

Each port is configured identically across all nodes in the system. The connection determines with which systems.

Actions		Filter		
ID	System Connection	Node	WWPN	
1				
1	Any	1(Upper)	500507680215F77F	
1	Any	2(Lower)	500507680215F780	
2				
2	Any	1(Upper)	500507680225F77F	
2	Any	2(Lower)	500507680225F780	
3				
3	Any	1(Upper)	500507680235F77F	
3	Any	2(Lower)	500507680235F780	
4				
4	Any	1(Upper)	500507680245F77F	
4	Any	2(Lower)	500507680245F780	

Figure 4-2 Identifying v7000 wwpn ports

- ▶ For two fabrics SAN (fabric1 and fabric 2), the zoning configuration is performed by considering the following rules:
 - On fabric 1: One zone consisting of one LPAR WWPN port (initiator) with one storage WWPN (target) from node 1 and a secondary zone with the storage WWPN (target) from node 2.
 - On fabric 2: One zone consisting of one LPAR WWPN port (initiator) with one storage WWPN (target) from node 1 and a secondary zone with the storage WWPN (target) from node 2 link.
 - Depending on requirements, the suggested number of paths per disk is four and maximum paths of eight.
- ▶ The LPARs are configured as hosts on the Storwize V7000.
- ▶ The disks are presented to the hosts, as shown in Figure 4-3 on page 45.

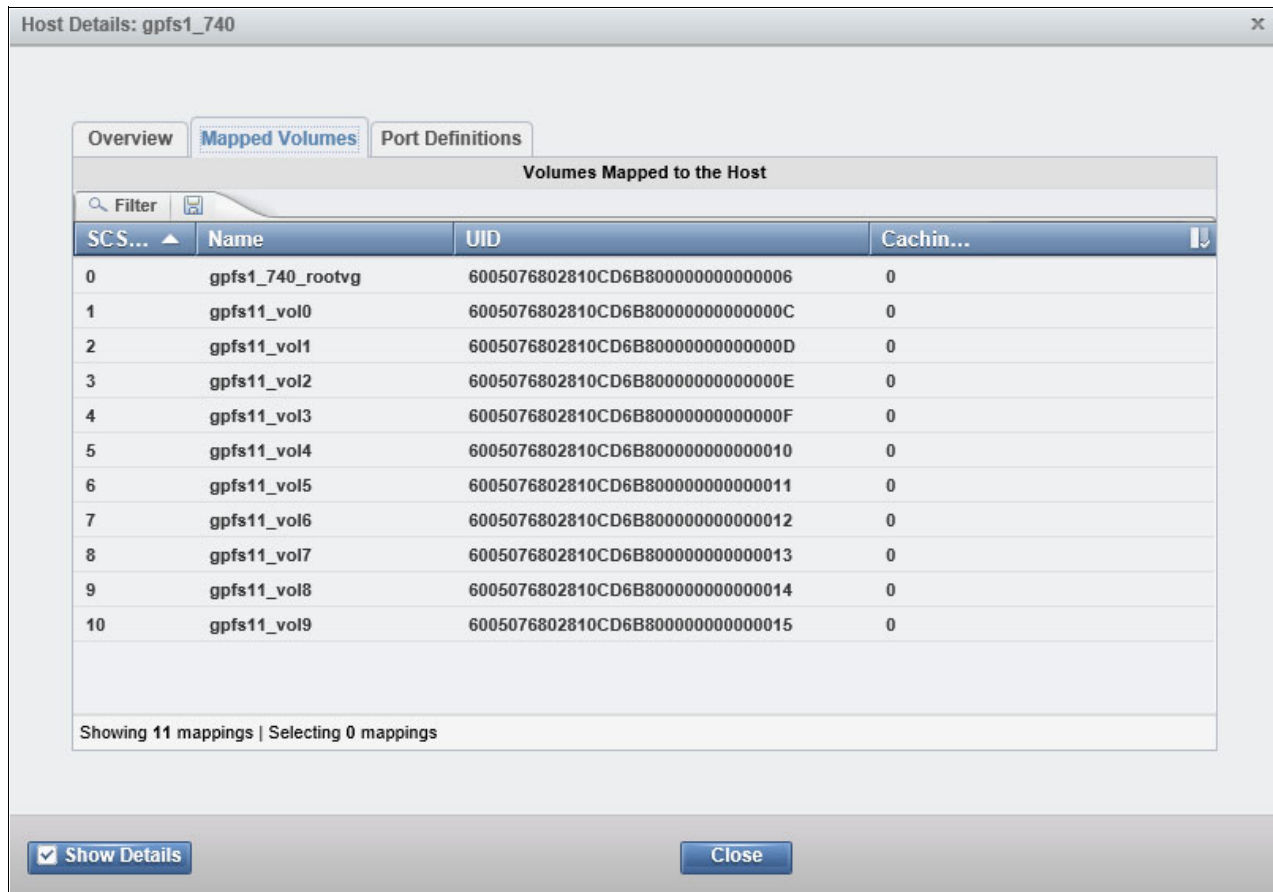


Figure 4-3 Disks assigned to AIX host

The LPAR configuration for using NPIV is in *IBM PowerVM Best Practices*, SG24-8062:
<http://www.redbooks.ibm.com/abstracts/sg248062.html>

IBM Spectrum Scale AIX disks configuration

The multipath driver used with the AIX operating system is SDDPCM driver. The disk parameters and paths are queried by using the **pcmpath** command, as shown in Example 4-7. The serial ID shown in the example is on the IBM Storage v7000. In this way you can identify the allocated LUN in storage and the disk on the AIX.

Example 4-7 Disks displayed on AIX

```
root@gpfs1750: /> pcmpath query device
```

```
Total Dual Active and Active/Asymmetric Devices : 10
```

```
DEV#: 1 DEVICE NAME: hdisk1 TYPE: 2145 ALGORITHM: Load Balance
SERIAL: 6005076802810CD6B80000000000000C
```

```
=====
Path#    Adapter/Path Name      State   Mode    Select   Errors
0        fscsi0/path0            OPEN    NORMAL   8408     0
1*       fscsi0/path1            OPEN    NORMAL   84       0
```

```
DEV#: 2 DEVICE NAME: hdisk2 TYPE: 2145 ALGORITHM: Load Balance
SERIAL: 6005076802810CD6B80000000000000D
```

```
=====
Path#      Adapter/Path Name      State   Mode    Select   Errors
  0*        fscsi0/path0             OPEN   NORMAL    84        0
  1          fscsi0/path1             OPEN   NORMAL   8906        0

DEV#:   3  DEVICE NAME: hdisk3  TYPE: 2145  ALGORITHM: Load Balance
SERIAL: 6005076802810CD6B80000000000000E
=====
Path#      Adapter/Path Name      State   Mode    Select   Errors
  0          fscsi0/path0             OPEN   NORMAL   8736        0
  1*        fscsi0/path1             OPEN   NORMAL    84        0

DEV#:   4  DEVICE NAME: hdisk4  TYPE: 2145  ALGORITHM: Load Balance
SERIAL: 6005076802810CD6B80000000000000F
=====
Path#      Adapter/Path Name      State   Mode    Select   Errors
  0*        fscsi0/path0             OPEN   NORMAL    84        0
  1          fscsi0/path1             OPEN   NORMAL   212        0

.....<< snippet>>.....
```

More details about SDDPCM driver installation and configuration, and the AIX host attachment using IBM Storage Systems are at the following web page:

<http://www.ibm.com/support/docview.wss?uid=ssg1S4001363>

Disk configurations for RHEL PPC64

In our scenario, the second cluster is configured on two Linux LPARs, running RHEL for IBM Power Systems.

The hosts running on Linux on Power using Fibre Channel Communications are defined on IBM Storwize v7000 storage as Generic hosts. The disks are allocated at hosts in the same way as for AIX systems. More details about host configuration and disk allocation using IBM Storwize v7000 Storage System are in *Implementing the IBM Storwize V7000 V7.4*, SG24-7938:

<http://www.redbooks.ibm.com/abstracts/sg247938.html>

The multipath software used for IBM Storwize v7000 disks is the native Linux Device Mapper Multipath (DMMP) driver. Details regarding multipath configuration at the Linux operating system level are at the following web page:

<http://ibm.co/1Q0K4FW>

The disks are configured to have the same alias across the nodes. The multipath configuration for Red Hat Linux version 7.1 for Power PC64 is shown in Example 4-8.

Example 4-8 Multipath configuration on each RHEL cluster nodes

```
multipathd> show config
defaults {

    find_multipaths yes
    user_friendly_names yes
    polling_interval 30
    .....<<snippet; only above settings are located on
multipath.conf>>.....
}
```


Build the IBM Spectrum Scale portability layer

This step is specific to Linux cluster nodes. The portability layer is a kernel module that allows IBM Spectrum Scale daemons to interact with the Linux operating system. These are the three ways to build the module:

- ▶ Use the **autoconfig** tool.
- ▶ Use the **spectrumscale** installation toolkit.
- ▶ Use the **mmbuildgpl** command.

The **mmbuildgpl** command was introduced in IBM Spectrum Scale 4.1.0.4. The command simplifies the build process, compared to the autoconfig method. The output from the command is shown in Example 4-9.

Example 4-9 mmbuildgpl output

```
[root@rhel2750]# /usr/lpp/mmfs/bin/mmbuildgpl
-----
mmbuildgpl: Building GPL module begins at Thu Aug 13 16:38:38 EDT 2015.
-----
Verifying Kernel Header...
  kernel version = 3100099 (3.10.0-229.el7.ppc64, 3.10.0-229)
  module include dir = /lib/modules/3.10.0-229.el7.ppc64/build/include
  module build dir   = /lib/modules/3.10.0-229.el7.ppc64/build
  kernel source dir  = /usr/src/linux-3.10.0-229.el7.ppc64/include
  Found valid kernel header file under
/usr/src/kernels/3.10.0-229.el7.ppc64/include
Verifying Compiler...
  make is present at /bin/make
  cpp is present at /bin/cpp
  gcc is present at /bin/gcc
  g++ is present at /bin/g++
  ld is present at /bin/ld
make World ...
make InstallImages ...
-----
mmbuildgpl: Building GPL module completed successfully at Thu Aug 13 16:38:54 EDT
2015.
-----
```

Creating and configuring AIX and Linux IBM Spectrum Scale clusters

Because all the nodes meet all the prerequisites, we proceed further by creating and configuring the IBM Spectrum Scale cluster on AIX and also on RHEL for IBM Power Systems. The cluster configuration steps are described in Chapter 2, “IBM Spectrum Scale implementation” on page 7 and are similar for both types of operating systems. Example 4-10 lists both cluster configurations.

Example 4-10 Cluster configurations

RHEL IBM Spectrum Scale configuration - primary site

```
[root@rhel1750 ~]# mmlscluster
```

GPFS cluster information

=====

```
GPFS cluster name:      RHEL_CLUSTER.rhel1750
GPFS cluster id:        7179332175107162005
```

```
GPFS UID domain:      RHEL_CLUSTER.rhel1750
Remote shell command: /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:      CCR
```

Node	Daemon node name	IP address	Admin node name	Designation
1	rhel1750	172.16.20.130	rhel1750	quorum-manager
2	rhel1740	172.16.20.133	rhel1740	quorum-manager

```
[root@rhel1750 ~]# mmlsconfig
Configuration data for cluster RHEL_CLUSTER.rhel1750:
```

```
-----
clusterName RHEL_CLUSTER.rhel1750
clusterId 7179332175107162005
dmapiFileHandleSize 32
minReleaseLevel 4.1.1.0
ccrEnabled yes
autoload yes
adminMode central
```

```
File systems in cluster RHEL_CLUSTER.rhel1750:
```

```
-----
A
IX IBM Spectrum Scale configuration - secondary site
```

```
root@gpfs1750:/work>mmcrcluster -N gpfsDR.txt --ccr-enable -p gpfs1750 -s gpfs1740 -r
/usr/bin/ssh -R /usr/bin/scp -C gpfs.DR.site -A <
mmcrcluster: Performing preliminary node verification ...
mmcrcluster: Processing quorum and other critical nodes ...
mmcrcluster: Finalizing the cluster data structures ...
mmcrcluster: Command successfully completed
mmcrcluster: Warning: Not all nodes have proper GPFS license designations.
Use the mmchlicense command to designate licenses as needed.
mmcrcluster: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

```
root@gpfs1750:/work> mmchlicense server --accept -N gpfs1750,gpfs1740
```

```
The following nodes will be designated as possessing GPFS server licenses:
```

```
gpfs1740
gpfs1750
mmchlicense: Command successfully completed
mmchlicense: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
```

```
root@gpfs1750:/> mmlsconfig
Configuration data for cluster gpfs.DR.site:
```

```
-----
clusterName gpfs.DR.site
clusterId 9334245562478638495
autoload yes
dmapiFileHandleSize 32
ccrEnabled yes
minReleaseLevel 4.1.1.0
```

```
adminMode central
```

```
File systems in cluster gpfs.DR.site:
```

```
-----
```

```
root@gpfs1750:/> mmlscluster
```

```
GPFS cluster information
```

```
=====
```

```
GPFS cluster name:      gpfs.DR.site
GPFS cluster id:        9334245562478638495
GPFS UID domain:        gpfs.DR.site
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR
```

Node	Daemon node name	IP address	Admin node name	Designation
1	gpfs1740	172.16.20.124	gpfs1740	quorum-manager
2	gpfs1750	172.16.20.127	gpfs1750	quorum-manager

Network Shared Disks configuration

For each IBM Spectrum Scale cluster, a stanza file was populated with the corresponding disk names and attributes in order to create NSD disks.

On AIX, four disks are planned to be used in the IBM SPECTRUM Scale file system configuration. The stanza file used as input for `mmcrnsd` command for NSDs creation is shown in Example 4-11.

Example 4-11 NSD Stanza file and NSD creation

```
root@gpfs1750:/work> cat disks.gpfs
```

```
%nsd:
```

```
device=/dev/hdisk1
nsd=DRNSD001
servers=gpfs1740,gpfs1750
usage=dataAndMetadata
```

```
%nsd:
```

```
device=/dev/hdisk2
nsd=DRNSD002
servers=gpfs1750,gpfs1740
usage=dataAndMetadata
```

```
%nsd:
```

```
device=/dev/hdisk3
nsd=DRNSD003
servers=gpfs1740,gpfs1750
usage=dataAndMetadata
```

```
%nsd:
```

```
device=/dev/hdisk4
nsd=DRNSD004
servers=gpfs1750,gpfs1740
usage=dataAndMetadata
```

```
root@gpfs1750:/work> mmcrnsd -F "disks.gpfs"
```

```
mmcrnsd: Processing disk hdisk1
```

```
mmcrnsd: Processing disk hdisk2
mmcrnsd: Processing disk hdisk3
mmcrnsd: Processing disk hdisk4
mmcrnsd: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
root@gpfs1750:/work> mmlsnsd -X
```

Disk name	NSD volume ID	Device	Devtype	Node name	Remarks
DRNSD001	AC10147F55CE36DA	/dev/hdisk1	hdisk	gpfs1740	server node
DRNSD001	AC10147F55CE36DA	/dev/hdisk1	hdisk	gpfs1750	server node
DRNSD002	AC10147F55CE36DE	/dev/hdisk2	hdisk	gpfs1740	server node
DRNSD002	AC10147F55CE36DE	/dev/hdisk2	hdisk	gpfs1750	server node
DRNSD003	AC10147F55CE36E3	/dev/hdisk3	hdisk	gpfs1740	server node
DRNSD003	AC10147F55CE36E3	/dev/hdisk3	hdisk	gpfs1750	server node
DRNSD004	AC10147F55CE36E7	/dev/hdisk4	hdisk	gpfs1740	server node
DRNSD004	AC10147F55CE36E7	/dev/hdisk4	hdisk	gpfs1750	server node

On the Linux system, the NSD creation procedure needs verification steps because IBM Spectrum Scale version 4.1 introduced NSDv2 format for the IBM Spectrum Scale disks. The stanza file used for NSD creation is shown in Example 4-13 on page 52. The NSD's device points to /dev/dm-xx device. The disks can be identified by using the `scsi_id` command as shown in Example 4-12 and it is mentioned in the *IBM Spectrum Scale (formerly GPFS), SG24-8254*. The `scsi_id` command retrieves the device identification number (page 0x83), which matched the Storwize v7000 volume's UID.

Example 4-12 Identifying disk serial

```
for MPATHDISK in `ls -l /dev/dm-?`; do echo "$MPATHDISK SERIAL:";
/usr/lib/udev/scsi_id --page=0x83 --whitelisted --device=$MPATHDISK; done
/dev/dm-0 SERIAL:
36005076802810cd6b800000000000036
/dev/dm-1 SERIAL:
36005076802810cd6b800000000000038
/dev/dm-2 SERIAL:
36005076802810cd6b80000000000003c
/dev/dm-3 SERIAL:
36005076802810cd6b800000000000034
/dev/dm-4 SERIAL:
36005076802810cd6b800000000000035
/dev/dm-5 SERIAL:
36005076802810cd6b800000000000035
/dev/dm-6 SERIAL:
36005076802810cd6b800000000000039
/dev/dm-7 SERIAL:
36005076802810cd6b80000000000002a
/dev/dm-8 SERIAL:
36005076802810cd6b80000000000003b
/dev/dm-9 SERIAL:
36005076802810cd6b80000000000003a
```

Identifying the LUN's serial number allows us to complete the correct device for the NSD corresponding disk in the stanza file.

Example 4-13 NSD Stanza file for Linux

```
%nsd:
    device=/dev/dm-1
    nsd=NSD001
    servers=rhel1750,rhel1740
    usage=dataAndMetadata
%nsd:
    device=/dev/dm-2
    nsd=NSD002
    servers=rhel1750,rhel1740
    usage=dataAndMetadata
%nsd:
    device=/dev/dm-3
    nsd=NSD003
    servers=rhel1750,rhel1740
    usage=dataAndMetadata
%nsd:
    device=/dev/dm-5
    nsd=NSD004
    servers=rhel1750,rhel1740
    usage=dataAndMetadata
%nsd:
    device=/dev/dm-6
    nsd=NSD005
    servers=rhel1750,rhel1740
    usage=dataAndMetadata
```

Note: The /dev/mapper/devices does not work on NSD definitions.

File system configuration on RHEL and AIX Spectrum Scale clusters

A new file system is created on the IBM Spectrum Scale primary site. Example 4-14 shows the file system creation, relying upon three NSDs with the mount point in /HQ.

Example 4-14 Creating fsHQ on primary site - Spectrum Scale RHEL cluster

```
[root@rhel1750 work]# mmcrfs fsHQ "NSD001;NSD002;NSD003" -A yes --filesetdf
-T /HQ -B 1M
```

The following disks of fsHQ will be formatted on node rhel1740:

```
NSD001: size 10240 MB
NSD002: size 10240 MB
NSD003: size 10240 MB
```

Formatting file system ...

Disks up to size 411 GB can be added to storage pool system.

Creating Inode File

Creating Allocation Maps

Creating Log Files

Clearing Inode Allocation Map

Clearing Block Allocation Map

Formatting Allocation Map for storage pool system

Completed creation of file system /dev/fsHQ.

mmcrfs: Propagating the cluster configuration data to all affected nodes.

This is an asynchronous process.

In this scenario, using AFM-based disaster recovery feature, our goal is to asynchronously replicate data held in one fileset (primary) to the DR site. The target (secondary) on the DR site is the location where the data is replicated and is also a fileset in the IBM Spectrum Scale configuration. The primary can be connected to the secondary by using NFSv3 or IBM Spectrum Scale.

The AFM-based disaster configuration can be performed in the following situations:

- ▶ New primary and secondary filesets are created using `mmcrfileset`, independent of each other.
- ▶ An existing independent IBM Spectrum Scale (GPFS) fileset can be converted to a primary or secondary.
- ▶ A working AFM single writer (SW) or independent writer (IW) relationship can be converted to a primary or secondary relationship.

Each scenario has configuration steps that are detailed in the following web page:

<http://ibm.co/1VNIzP5>

In this section, we focus on fileset replication using AFM-based disaster recovery with new primary and secondary independent filesets.

The steps for establishing the async DR replication relationship are as follows:

1. Creating the filesets
2. Linking primary using `mmlinkfileset` command
3. Updating secondary
4. Linking secondary and updating NFS export (using GPFS protocol, the secondary file system should be mounted on the primary site)

On an AFM cluster that uses the native IBM Spectrum Scale protocol for defining an AFM target, gateway nodes can be mapped to any other node in the same cache cluster. In the absence of a mapping definition, all gateway nodes will be used for the I/O. The requirement is to have at least one node with the gateway role. In our case, node `rhel1740` is promoted as gateway, as shown in Example 4-15.

Example 4-15 Adding the gateway node role

```
[root@rhel1750 ~]# mmchnode -N rhel1740 --gateway
Fri Aug 14 15:24:19 EDT 2015: mmchnode: Processing node rhel1740
mmchnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@rhel1750 work]# mmlscluster
```

GPFS cluster information

```
=====
GPFS cluster name:      RHEL_CLUSTER.rhel1750
GPFS cluster id:       7179332175107162005
GPFS UID domain:       RHEL_CLUSTER.rhel1750
Remote shell command:  /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:       CCR
```

Node	Daemon node name	IP address	Admin node name	Designation
1	rhel1750	172.16.20.130	rhel1750	quorum-manager

2 rhel1740 172.16.20.133 rhel1740 quorum-manager-gateway

The second node can also be configured as a gateway:

```
[root@rhel1750 ~]# mmchnode -N rhel1750 --gateway
Fri Aug 14 15:25:19 EDT 2015: mmchnode: Processing node rhel1750
mmchnode: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@rhel1750 work]# mmlscluster
```

GPFS cluster information

=====

```
GPFS cluster name:      RHEL_CLUSTER.rhel1750
GPFS cluster id:        7179332175107162005
GPFS UID domain:        RHEL_CLUSTER.rhel1750
Remote shell command:   /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:        CCR
```

Node	Daemon node name	IP address	Admin node name	Designation
1	rhel1750	172.16.20.130	rhel1740	quorum-manager
2	rhel1740	172.16.20.133	rhel1750	quorum-manager-gateway

Following the steps (in Example 4-15 on page 53), we proceed with the independent fileset creation on the file system fsHQ. The fsHQ file system is mounted and a new fileset filesetHQ is created, as shown in Example 4-16. The filesetHQ fileset has as an AFM target the filesetDR fileset on DR, uses the IBM Spectrum Scale protocol for afmtarget and also a unique Primary ID is generated.

Example 4-16 Creating fileset filesetHQ

```
[root@rhel1750 work]# mmmount fsHQ -a
Fri Aug 14 14:10:39 EDT 2015: mmmount: Mounting file systems ...
[root@rhel1750 4.1.1]# mmcrfileset fsHQ filesetHQ --inode-space=new -p
afmtarget=gpfs:///gpfs/fsDR/filesetDR -p afmnode=primary --inode-limit=1024000 -p
afmAsyncDelay=15 -p afmRPO=5
Fileset filesetHQ created with id 1 root inode 131075.
Primary Id (afmPrimaryId) 7179332175107162005-1485AC1055CE259B-1
```

Note: The AFM-based DR feature is available only with the advanced IBM Spectrum Scale license. If the crypto package is not installed, the following message is displayed:

AFM primary/secondary filesets cannot be created in the currently installed version.
mmcrfileset: Command failed. Examine previous error messages to determine cause.

Because the AFM-based DR is available starting with IBM Spectrum Scale 4.1.1, you must have the file system at minimum version 14.20. Therefore, after an IBM Spectrum Scale upgrade, the file system must be also upgraded by using the `mmchfs <fs name> -V full` command.

The fileset attributes are displayed in Example 4-17.

Example 4-17 Fileset filesetHQ attributes

```
[root@rhell1750 4.1.1]# mmlsfileset fsHQ filesetHQ -L --afm
Filesets in file system 'fsHQ':

Attributes for fileset filesetHQ:
=====
Status                               Unlinked
Path                                 --
Id                                   1
Root inode                           131075
Parent Id                             --
Created                               Fri Aug 14 13:44:00 2015
Comment
Inode space                           1
Maximum number of inodes              807936
Allocated inodes                      66816
Permission change flag                chmodAndSetacl
afm-associated                         Yes
Target                                gpfs:///gpfs/fsDR/filesetDR
Mode                                   primary
Async Delay                           15
Recovery Point Objective               5 minutes
Last pSnapId                          0
Number of Gateway Flush Threads       4
Primary Id                            7179332175107162005-1485AC1055CE259B-1
```

Note: The fileset attributes can be changed with the `mmchfileset` command as for example the `afmAsyncDelay` or `afmRPO`:

```
mmchfileset fsHQ filesetHQ -p afmAsyncDelay=15 -p afmRPO=15
```

The next step in our configuration is linking the `filesetHQ` fileset into the location we want, as shown in Example 4-18.

Example 4-18 Linking fileset to /HQ/filesetHQ

```
[root@rhell1740 /]# mmlinkfileset fsHQ filesetHQ -J /HQ/filesetHQ
Fileset filesetHQ linked at /HQ/filesetHQ
First snapshot name is psnap0-rpo-1485AC1055CE259B-1
Flushing dirty data for snapshot filesetHQ::psnap0-rpo-1485AC1055CE259B-1...
Quiescing all file system operations.
Snapshot filesetHQ::psnap0-rpo-1485AC1055CE259B-1 created with id 1.
```

The secondary fileset is also created and linked on the DR site cluster, as shown in Example 4-19.

Example 4-19 Create secondary fileset

```
root@gpfs1750:/work> /usr/lpp/mmfs/bin/mmcrfileset fsDR filesetDR --inode-space=new
--inode-limit=1024000 -p afmMode=secondary -p afmMode=secondary -p
afmPrimaryId=7179332175107162005-1485AC1055CE259B-1
Fileset filesetDR created with id 1 root inode 131075.
root@gpfs1750:/work> mmlsfileset fsDR filesetDR -L --afm
Filesets in file system 'fsDR':
```

Attributes for fileset filesetDR:

=====

Status	Unlinked
Path	--
Id	1
Root inode	131075
Parent Id	--
Created	Fri Aug 14 15:08:38 2015
Comment	
Inode space	1
Maximum number of inodes	807936
Allocated inodes	66816
Permission change flag	chmodAndSetacl
afm-associated	Yes
Associated Primary ID	7179332175107162005-1485AC1055CE259B-1
Mode	secondary
Last pSnapId	0

```
root@gpfs1750:/> mmlinkfileset fsDR filesetDR -J /DR/filesetDR
Fileset filesetDR linked at /DR/filesetDR
```

```
root@gpfs1750:/> mmlsfileset fsDR filesetDR -L --afm
Filesets in file system 'fsDR':
```

Attributes for fileset filesetDR:

=====

Status	Linked
Path	/DR/filesetDR
Id	1
Root inode	131075
Parent Id	0
Created	Fri Aug 14 16:22:03 2015
Comment	
Inode space	1
Maximum number of inodes	807936
Allocated inodes	66816
Permission change flag	chmodAndSetacl
afm-associated	Yes
Associated Primary ID	7179332175107162005-1485AC1055CE259B-1
Mode	secondary
Last pSnapId	0

The next step is to configure remote cluster authentication and, on the primary site, remotely mounting the file system (secondary) from the DR site as described on “Configuring remote access cluster authentication” on page 61.

Note: Protocol support is not available in a multi-cluster configuration.

To validate the configuration, the **gpfsperf** tool is compiled on the RHEL system, as shown in Example 4-20.

Example 4-20 Compiling gpfsperf

```
[root@rhel1740 ~]# cd /usr/lpp/mmfs/samples/perf/
[root@rhel1740 perf]# make
cc -c -O -DGPFS_LINUX -DGPFS_ARCH_PPC64 -m64 -I/usr/lpp/mmfs/include gpfsperf.c -o gpfsperf.o
cc -O -DGPFS_LINUX -DGPFS_ARCH_PPC64 -m64 gpfsperf.o irreg.o -lpthread -lrt -lgpfs -o gpfsperf
```

We start creating files on the /fsHQ/filesetHQ file system by using **gpfsperf** and **dd** tools, as shown in Example 4-21.

Example 4-21 Creating files on primary file system

```
[root@rhel1740 ~]# /usr/lpp/mmfs/samples/perf/gpfsperf -r 10K -n 4096M
create seq /HQ/filesetHQ/file4
/usr/lpp/mmfs/samples/perf/gpfsperf create seq /HQ/filesetHQ/file4
  recSize 10K nBytes 4194300K fileSize 100M
  nProcesses 1 nThreadsPerProcess 1
  file cache flushed before test
  not using direct I/O
  offsets accessed will cycle through the same file segment
  not using shared memory buffer
  not releasing byte-range token after open
  no fsync at end of test
  Data rate was 216795.41 Kbytes/sec, thread utilization 1.000
[root@rhel1740 ~]# dd if=/dev/zero of=/HQ/filesetHQ/lg bs=1M count=1024
1024+0 records in
1024+0 records out
1073741824 bytes (1.1 GB) copied, 0.396967 s, 2.7 GB/s
```

The replication status is validated as shown in Example 4-22.

Example 4-22 AFM-based DR replication status

[root@rhel1740 filesetHQ]# mmadmctl fsHQ getstate -j filesetHQ				
Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length
Queue numExec				
-----	-----	-----	-----	-----
filesetHQ	gpfs:///gpfs/fsDR/filesetDR	Dirty	rhel1740	2
40971				
[root@rhel1740 filesetHQ]# mmadmctl fsHQ getstate -j filesetHQ				
Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length
Queue numExec				
-----	-----	-----	-----	-----
filesetHQ	gpfs:///gpfs/fsDR/filesetDR	Active	rhel1740	0
40973				

Note: The gateway node must have mounted the remote file system. Otherwise the unmounted Cache Mode status is displayed.

The RPO validation is verified by using the command shown in Example 4-23.

Example 4-23 RPO verification

```
[root@rhel1750 filesetHQ]# mmfssnapshot fsHQ -j filesetHQ
Snapshots in file system fsHQ:
Directory          SnapId      Status  Created                Fileset
psnap0-rpo-1485AC1055CE259B-1 1          Valid   Fri Aug 14 16:07:58 2015 filesetHQ
psnap-rpo-1485AC1055CE259B-1-15-09-13-11-19-18 24        Valid   Sun Sep 13 11:19:18 2015 filesetHQ
psnap-rpo-1485AC1055CE259B-1-15-09-13-11-29-19 25        Valid   Sun Sep 13 11:29:19 2015 filesetHQ
```

4.2.3 Failover and fallback

Disaster recovery procedures imply uncontrolled or controlled applications fail over to the disaster site and also fail back when the primary site is recovered.

In this section, we simulate a primary site failure, promoting the secondary IBM Spectrum Scale fileset as being primary, practically reversing the replication direction. The failover is not performed automatically but can be automated by scripting with the entire stack of the verifying/starting application failover procedure of the production environment in the DR site.

The failover is performed by converting the secondary, acting as primary, with choosing whether the latest snapshot on the secondary is restored or not. While acting as primary, the RPO snapshots are temporarily disabled, and the data is not replicated to another site until a new primary is set up or the old primary is reinstalled and this acting primary is converted back to a secondary.

In the controlled failover, the primary site is simulated as going down by unlinking the fileset, as shown in Example 4-24.

Example 4-24 Unlink primary

```
[root@rhel1750 filesetHQ]# mmunlinkfileset fsHQ filesetHQ -f
Fileset filesetHQ unlinked.
[root@rhel1750 filesetHQ]#
```

On the secondary site, convert the secondary to acting as primary (Example 4-25).

Example 4-25 Failover - Convert secondary to acting as primary

```
root@gpfs1750:/> mmfctl fsDR failoverToSecondary -j filesetDR
mmfctl: failoverToSecondary restoring from psnap
psnap-rpo-1485AC1055CE259B-1-15-09-13-11-49-19
[2015-09-13 11:52:10] Restoring fileset "filesetDR" from snapshot
"psnap-rpo-1485AC1055CE259B-1-15-09-13-11-49-19" of filesystem "/dev/fsDR"

[2015-09-13 11:52:12] Scanning inodes, phase 1 ...
[2015-09-13 11:52:13] 197888 inodes have been scanned, 100% of total.
[2015-09-13 11:52:13] There's no data changes since the restoring snapshot,
skipping restore.
[2015-09-13 11:52:13] Restore completed successfully.
[2015-09-13 11:52:13] Clean up.
```

```
Primary Id (afmPrimaryId) 9334245562478638495-AC10147C55CE3871-1
Fileset filesetDR changed.
Promoted fileset filesetDR to Primary
```

The status of the fileset after promoting as the primary is shown in Example 4-26.

Example 4-26 Fileset status after failover

```
root@gpfs1750: /> mmlsfileset fsDR filesetDR -L --afm
Filesets in file system 'fsDR':
```

Attributes for fileset filesetDR:

=====

Status	Linked
Path	/DR/filesetDR
Id	1
Root inode	131075
Parent Id	0
Created	Fri Aug 14 16:22:03 2015
Comment	
Inode space	1
Maximum number of inodes	807936
Allocated inodes	66816
Permission change flag	chmodAndSetacl
afm-associated	Yes
Target	--
Mode	primary
Async Delay	15
Recovery Point Objective	disable
Last pSnapId	0
Number of Gateway Flush Threads	4
Primary Id	9334245562478638495-AC10147C55CE3871-1

We generate sample data from the acting primary by writing two files (Example 4-27).

Example 4-27 Generating files in primary (secondary promoted as primary)

```
root@gpfs1740: /DR/filesetDR> for i in 1 2 3 4 5 6; do dd if=/dev/zero
of=/DR/filesetDR/file_acting_primary_$i bs=1M count=100;done
100+0 records in
100+0 records out
.....<<snippet>>>>>>>>
100+0 records in
100+0 records out
root@gpfs1750: /> ls -l /DR/filesetDR|grep acti
-rw-r--r-- 1 root system 104857600 Sep 13 11:54 file_acting_primary_1
-rw-r--r-- 1 root system 104857600 Sep 13 11:54 file_acting_primary_2
```

The **AsyncDelay** parameter is set to the default value of 15 minutes and RPO disabled. Considering that the old primary site is back, we link the fileset on the old primary (Example 4-28).

Example 4-28 Link old primary at primary site

```
[root@rhel1750 HQ]# mmlinkfileset fsHQ filesetHQ -J /HQ/filesetHQ
Fileset filesetHQ linked at /HQ/filesetHQ
```

Failback to the old primary steps are as follows:

1. Start the failback process.
2. Apply the differential from the acting primary.
3. Complete the failback process.
4. Change the secondary.

Starting the failback process, we restore the primary to the contents from the last RPO on the primary before the disaster. The assumption is that because the old primary is back up, all RPOs prior to the disaster are available. Example 4-29 shows the start of the activation process for the filesethQ fileset.

Example 4-29 Failback to primary

```
[root@rhel1750 /]# mmadmctl fshq failbackToPrimary -j filesethQ --start
Fileset filesethQ changed.
mmadmctl: failbackToPrimary restoring from psnap
failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49
[2015-09-13 14:06:43] Restoring fileset "filesethQ" from snapshot
"failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49" of filesystem "/dev/fshq"

[2015-09-13 14:06:45] Scanning inodes, phase 1 ...
[2015-09-13 14:06:46] 197888 inodes have been scanned, 100% of total.
[2015-09-13 14:06:46] There's no data changes since the restoring snapshot, skipping restore.
[2015-09-13 14:06:46] Restore completed successfully.
[2015-09-13 14:06:46] Clean up.
```

Now the old primary is in read-only mode. We can start syncing all differences from the secondary (now primary) to the old primary. The command is shown in Example 4-30.

Note: While applying the updates, for the final synchronization of the primary site (old primary), the applications are stopped at the DR site.

Example 4-30 Applying updates

```
[root@rhel1750 filesethQ]# mmadmctl fshq applyUpdates -j filesethQ
[2015-09-13 14:23:12] Getting the list of updates from the acting Primary...
[2015-09-13 14:23:22] Applying the 6 updates...
[2015-09-13 14:23:24] 6 updates have been applied, 100% of total.
mmadmctl: Creating the failback psnap locally. failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-14-23-09
Flushing dirty data for snapshot filesethQ::failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-14-23-09...
Quiescing all file system operations.
Snapshot filesethQ::failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-14-23-09 created with id 29.
mmadmctl: Deleting the old failback psnap. failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49
Invalidating snapshot files in filesethQ::failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49...
Deleting files in snapshot filesethQ::failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49...
100.00 % complete on Sun Sep 13 14:23:26 2015 ( 66816 inodes with total 2 MB data processed)
Invalidating snapshot files in filesethQ::failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49/F/...
Delete snapshot filesethQ::failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-13-08-49 complete, err = 0
```

Note: If a replication is scheduled to be performed from the DR site to the primary site, at least one gateway node must exist on the DR site. Momentary Gateway nodes cannot be configured on AIX. More information about gateway nodes is at the AFM architecture page:

<http://ibm.co/1QyD5nV>

The data is synchronized across the sites and we convert the acting primary back to the secondary also reestablishing the relationship for the fileset, as shown in Example 4-31.

Example 4-31 Fileset on primary is read-write

```
[root@rhel1750 filesetHQ]# mmafmctl fsHQ failbackToPrimary -j filesetHQ --stop
Fileset filesetHQ changed.
```

At this moment, the fileset data has been replicated to the old primary, the fileset is in read-write mode, and the primary site is ready for starting applications.

We proceed by changing the secondary with its previous role, reestablishing the relationship as shown in Example 4-32.

Example 4-32 Convert the acting primary back to secondary

```
root@gpfs1750:/DR/filesetDR> mmchfileset fsDR filesetDR -p
afmmode=secondary,afmPrimaryID=7179332175107162005-1485AC1055CE259B-1
Fileset filesetDR changed.
```

We also verify the replication status by creating a file on filesetHQ as shown in Example 4-33.

Example 4-33 Verifying replication HQ-->DR

```
On Primary cluster
[root@rhel1750 filesetHQ]# dd if=/dev/zero of=/HQ/filesetHQ/final_to_orig bs=1M
count=4096
4096+0 records in
4096+0 records out
4294967296 bytes (4.3 GB) copied, 12.3815 s, 347 MB/s
[root@rhel1750 filesetHQ]# ls -l |grep final
-rw-r--r--. 1 root root 4294967296 Sep 13 14:44 final_to_orig
[root@rhel1750 filesetHQ]# mmafmctl fsHQ getstate -j filesetHQ
```

Fileset Name	Fileset Target	Cache State
Gateway Node	Queue Length	Queue numExec
filesetHQ	gpfs:///gpfs/fsDR/filesetDR	Active
rhel1740	0	4098

On the cluster DR we observe that the file has been transferred:

```
root@gpfs1750:/DR/filesetDR> ls -l |grep final
-rw-r--r--. 1 root system 4294967296 Sep 13 14:44 final_to_orig
```

Configuring remote access cluster authentication

To access a file system from another IBM Spectrum Scale cluster, proper authorization and grants are required for the corresponding resource; in our case, this is one or many file systems. The authentication procedure is described in the following web page:

<http://ibm.co/1KEKAv6>

In our setup, the authorization is provided for accessing the fsDR file system by RHEL_CLUSTER.rhel1750 as primary site, as shown in Example 4-34.

Example 4-34 Remote cluster file system access

```
root@gpfs1750:/work> mmauth show
Cluster name:      RHEL_CLUSTER.rhel1750
Cipher list:      AUTHONLY
SHA digest:       c014744b5f8c41aff0b608bb705ae08bb5a9b5059982515d5f32c26101dd0408
File system access: fsDR      (rw, root allowed)

Cluster name:      gpfs.DR.site (this cluster)
Cipher list:      AUTHONLY
SHA digest:       7f57db5999617b7b5ff22a6a0dbad4d68934e1864bc8ce822998d71aab7eb443
SHA digest (new):  fd1f60db9aea8a384638440419974bc332ae5c1c29a71eb2c4d6486431357c4d
File system access: (all rw)
```

On the primary site, the file system we mount the fsDR file system (Example 4-35).

Example 4-35 Mounting remote cluster file system fsDR on primary site

```
[root@rhel1740 work]# mmremotefs add /dev/fsDR -f /dev/fsDR -C gpfs.DR.site
-T /gpfs/fsDR
mmremotefs: Propagating the cluster configuration data to all
affected nodes. This is an asynchronous process.
[root@rhel1750 work]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/rhel-root	27G	6.4G	21G	24%	/
devtmpfs	5.7G	0	5.7G	0%	/dev
tmpfs	5.8G	0	5.8G	0%	/dev/shm
tmpfs	5.8G	19M	5.8G	1%	/run
tmpfs	5.8G	0	5.8G	0%	/sys/fs/cgroup
/dev/mapper/mpatha2	497M	160M	338M	33%	/boot
/dev/fsHQ	30G	6.6G	24G	22%	/HQ
/dev/fsDR	30G	6.6G	24G	22%	/gpfs/fsDR

Configuring AFM using NFS

The Network File System (NFS) protocol can be used instead of the IBM Spectrum Scale (GPFS) protocol when AFM is used. Because the AFM-based disaster recovery is a specialized AFM feature it can rely also on NFS exports. For more information about tuning parameters, see the “Tuning active file management home communications” web page:

<http://ibm.co/li9mlbC>

Refer to the following information resources if you plan to use NFS for deploying AFM-based disaster recovery:

- IBM Spectrum Scale Frequently Asked Questions and Answers:

<http://ibm.co/1JH0huE>

- IBM Spectrum Scale Wiki:

<http://ibm.co/1aKwTP5>

- *IBM Spectrum Scale (formerly GPFS), SG24-8254:*

<http://www.redbooks.ibm.com/abstracts/sg248254.html>

4.2.4 Failback to new primary

In this scenario, we consider that the primary site is lost; practically we use a new fileset for replication from the DR to HQ. The IBM Spectrum Scale primary cluster name remains the same, only the new fileset is named FSQHNEW on the same file system HQ. We are positioned at the step before Example 4-28 on page 59, which was to bring up the primary site with the original fileset in place. Now, instead having the original fileset, we create another one as shown in Example 4-36.

The steps are as follows:

1. Create the Spectrum Scale fileset on primary (old primary - HQ site)
2. Convert the primary fileset (now on DR)

Example 4-36 Create a new fileset on primary site and delete previous one

```
[root@rhel1750 /]# mmlsfileset fsHQ filesetHQ -d
Collecting fileset usage information ...
Filesets in file system 'fsHQ':
Name                Status    Path                                Data (in KB)
filesetHQ           Linked   /HQ/filesetHQ                     16195840
[root@rhel1750 /]# mmlssnapshot fsHQ
Snapshots in file system fsHQ:
Directory           SnapId    Status  Created                Fileset
psnap-rpo-1485AC1055CE259B-1-15-09-13-11-49-19 27        Valid   Sun Sep 13 11:49:26 2015 filesetHQ
failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-14-23-09 29        Valid   Sun Sep 13 14:23:25 2015
filesetHQ
psnap-rpo-1485AC1055CE259B-1-15-09-13-14-48-56 30        Valid   Sun Sep 13 14:48:56 2015 filesetHQ
[root@rhel1750 /]# mmppsnap fsHQ delete -s psnap-rpo-1485AC1055CE259B-1-15-09-13-11-49-19 -j filesetHQ
mmppsnap: The peer snapshot psnap-rpo-1485AC1055CE259B-1-15-09-13-11-49-19 is deleted successfully.
[root@rhel1750 /]# mmppsnap fsHQ delete -s failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-14-23-09 -j
filesetHQ
mmppsnap: The peer snapshot failback-psnap-rpo-1485AC1055CE259B-1-15-09-13-14-23-09 is deleted successfully.
[root@rhel1750 /]# mmppsnap fsHQ delete -s psnap-rpo-1485AC1055CE259B-1-15-09-13-14-48-56 -j filesetHQ
mmppsnap: The peer snapshot psnap-rpo-1485AC1055CE259B-1-15-09-13-14-48-56 is deleted successfully.
[root@rhel1750 /]# mmlssnapshot fsHQ
No snapshots in file system fsHQ
[root@rhel1750 /]# mmunlinkfileset fsHQ filesetHQ -f
Fileset filesetHQ unlinked.
[root@rhel1750 /]# mmdelfileset fsHQ filesetHQ -f
Checking fileset ...
Checking fileset complete.
Deleting user files ...
 100.00 % complete on Sun Sep 13 15:36:37 2015 (    66816 inodes with total    261 MB data processed)
Deleting fileset ...
Fileset filesetHQ deleted.

[root@rhel1750 /]# mmcrfileset fsHQ filesetHQNEW --inode-space=new
Fileset filesetHQNEW created with id 1 root inode 131075.
```

The initial attributes of the filesetDR fileset are shown in Example 4-37.

Example 4-37 filesetDR attributes

```
root@gpfs1750:/DR/filesetDR> mmlsfileset fsDR filesetDR -L --afm
Filesets in file system 'fsDR':
```

Attributes for fileset filesetDR:

=====

Status	Linked
Path	/DR/filesetDR
Id	1
Root inode	131075
Parent Id	0
Created	Fri Aug 14 16:22:03 2015
Comment	
Inode space	1
Maximum number of inodes	807936
Allocated inodes	66816
Permission change flag	chmodAndSetacl
afm-associated	Yes
Target	--
Mode	primary
Async Delay	15
Recovery Point Objective	disable
Last pSnapId	0
Number of Gateway Flush Threads	4
Primary Id	9334245562478638495-AC10147C55CE3871-1

We modify the DR fileset to primary, as shown in Example 4-38.

Example 4-38 Converting filesetDR to primary

```
root@gpfs1750:/> mmafmctl fsDR convertToPrimary -j filesetDR
--afmtarget=gpfs:///gpfs/fsHQ/filesetHQNEW --inband
Checking for any special files. This may take a while...
Converting GPFS fileset to AFM primary fileset...
Primary Id (afmPrimaryId) 9334245562478638495-AC10147C55CE3871-1
Fileset filesetDR changed.
Setting up a Primary and Secondary relation...
Data will be moved to secondary via AFM
psnap will be taken at secondary after data movement completes
```

The new attributes of filesystem are shown below:

```
root@gpfs1750:/> mmlsfileset fsDR filesetDR -L --afm
Filesets in file system 'fsDR':
```

Attributes for fileset filesetDR:

=====

Status	Linked
Path	/DR/filesetDR
Id	1
Root inode	131075
Parent Id	0
Created	Fri Aug 14 16:22:03 2015
Comment	

Inode space	1
Maximum number of inodes	807936
Allocated inodes	66816
Permission change flag	chmodAndSetacl
afm-associated	Yes
Target	gpfs:///gpfs/fsHQ/filesetHQNEW
Mode	primary
Async Delay	15
Recovery Point Objective	15 minutes
Last pSnapId	0
Number of Gateway Flush Threads	4
Primary Id	9334245562478638495-AC10147C55CE3871-1

Note: Cluster authentication is already configured.

We proceed with fileset modification on the primary site for establishing a relationship, indicating the AFM Primary ID, as shown in Example 4-39, and also linking the file system.

Example 4-39 Fileset modification

```
[root@rhel1750 /]# mmchfileset fsHQ filesetHQNEW -p afmmode=secondary -p
afmPrimaryID=9334245562478638495-AC10147C55CE3871-1
Fileset filesetHQNEW changed.
[root@rhel1750 /]# mmlinkfileset fsHQ filesetHQNEW -J /HQ/filesetHQNEW
Fileset filesetHQNEW linked at /HQ/filesetHQNEW
```

We see that the status of the replication is on PrimInitInProg, as shown in Example 4-40.

Example 4-40 Replication status

```
root@gpfs1750:/DR/filesetDR> mmafmctl fsDR getstate -j filesetDR
```

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
filesetDR	gpfs:///gpfs/fsHQ/filesetHQNEW	PrimInitInProg	rhelpn.local	23	211

After taking psnap0, the primary is ready for use, as shown in Example 4-41.

Example 4-41 Fileset ready for use

```
root@gpfs1750:/DR/filesetDR> mmafmctl fsDR getstate -j filesetDR
```

Fileset Name	Fileset Target	Cache State	Gateway Node	Queue Length	Queue numExec
filesetDR	gpfs:///gpfs/fsHQ/filesetHQNEW	Active	rhelpn.local	0	234

The files are compared in both sites as shown in Example 4-42.

Example 4-42 The files on the systems

```
At destination
[root@rhel1750 filesetHQNEW]# ll
total 16396288
-rw-r--r--. 1 root root 1073741824 Sep 13 16:02 1g
-rw-r--r--. 1 root root 4294967296 Sep 13 16:03 HQ1_4G
-rw-r--r--. 1 root root 4294967296 Sep 13 16:03 HQ_4G
-rw-r--r--. 1 root root 104857600 Sep 13 16:02 file4
-rw-r--r--. 1 root root 104857600 Sep 13 16:01 file_acting_primary_1
-rw-r--r--. 1 root root 104857600 Sep 13 16:01 file_acting_primary_2
-rw-r--r--. 1 root root 1048576000 Sep 13 16:02 file_acting_primary_3
```

```
-rw-r--r--. 1 root root 1048576000 Sep 13 16:02 file_acting_primary_4
-rw-r--r--. 1 root root 104857600 Sep 13 16:01 file_acting_primary_5
-rw-r--r--. 1 root root 104857600 Sep 13 16:01 file_acting_primary_6
-rw-r--r--. 1 root root 104857600 Sep 13 16:03 file_acting_primary_7
-rw-r--r--. 1 root root 104857600 Sep 13 16:03 file_acting_primary_8
-rw-r--r--. 1 root root 4294967296 Sep 13 16:03 final_to_orig
```

At source

```
root@gpfs1750:/DR/filessetDR> ls -l
```

```
total 32792577
```

```
drwxr-xr-x   3 root    42949671      4096 Sep 13 14:42 .afm
drwx-----  2 root    system        4096 Sep 13 10:39 .ptrash
dr-xr-xr-x   2 root    system       32768 Dec 31 1969 .snapshots
-rw-r--r--   1 root    system    1073741824 Sep 13 11:49 lg
-rw-r--r--   1 root    system    4294967296 Sep 13 11:42 HQ1_4G
-rw-r--r--   1 root    system    4294967296 Sep 13 11:42 HQ_4G
-rw-r--r--   1 root    system    104857600 Sep 13 11:46 file4
-rw-r--r--   1 root    system    104857600 Sep 13 11:54 file_acting_primary_1
-rw-r--r--   1 root    system    104857600 Sep 13 11:54 file_acting_primary_2
-rw-r--r--   1 root    system    1048576000 Sep 13 11:56 file_acting_primary_3
-rw-r--r--   1 root    system    1048576000 Sep 13 11:56 file_acting_primary_4
-rw-r--r--   1 root    system    104857600 Sep 13 13:10 file_acting_primary_5
-rw-r--r--   1 root    system    104857600 Sep 13 13:10 file_acting_primary_6
-rw-r--r--   1 root    system    104857600 Sep 13 16:02 file_acting_primary_7
-rw-r--r--   1 root    system    104857600 Sep 13 16:03 file_acting_primary_8
-rw-r--r--   1 root    system    4294967296 Sep 13 14:44 final_to_orig
```

Note: More descriptions of cases that use AFM-based DR are at the following web page:

<http://ibm.co/1VWiJsc>

Persistent Reservation

Although the Persistent Reservation feature is not new in IBM Spectrum Scale 4.1.1, 4.1.0, or 3.5, this feature is an important one to consider. See the “SCSI-3 Persistent Reservation” topic in *IBM Spectrum Scale (formerly GPFS), SG24-8254*:

<http://www.redbooks.ibm.com/abstracts/sg248254.html>

Depending on the storage capabilities used in implementation, IBM Spectrum Scale can reduce cluster recovery time and perform a faster failover when the cluster is instructed to use Persistent Reservation (PR) for the disks that support this feature. PR allows the stripe group manager to “fence” disks during node failover by removing the reservation keys for that node. This configuration has a specific environment and requires the file systems to be created on disks capable of NSD SCSI-3. In contrast, non-PR disk failovers cause the system to wait until the disk lease expires. IBM Spectrum Scale allows file systems to have a mix of PR and non-PR disks.

Note: The SCSI-3 Persistent Reserve IBM Spectrum Scale configuration can be performed only with the IBM Spectrum Scale cluster shut down on all its nodes. You can switch to IBM Spectrum Scale configuration to use Persistent Reservation following the configuration procedure as is described next.

In our environment, we take advantage of SCSI-3 Persistent Reservation configuration because the IBM storage Storwize V7000 is used. The list of available storage types that support SCSI-3 Persistent Reservation including their supported minimum firmware level are listed at the following web page:

<http://ibm.co/1IFVklw>

IBM Spectrum Scale configuration procedures for using SCSI-3 Persistent Reservation for fast failover consists in the following steps:

1. Validating storage compatibility and the minimum required firmware level for IBM Spectrum Scale supported configuration.
2. IBM Spectrum Scale daemon is shut down on every cluster node.
3. Depending on the IBM Spectrum Scale version used, tiebreaker disk Persistent Reservation configuration validation is required. IBM Spectrum Scale versions 3.5.0.11, 4.1.0.4, and later, support tiebreaker disk configuration with Persistent Reservation.
4. Configure the disks at AIX operating system level to use Persistent Reservation. Disk parameters that must be modified are **reserve_policy** and **PR_key_value**.
5. Modifying IBM Spectrum Scale configuration parameters for fast failure detection and using Persistent Reservation.
6. Validate configuration.
7. Starting up the IBM Spectrum Scale cluster.

Considerations for using Persistent Reservation support in IBM Spectrum Scale are at the following web page:

<http://ibm.co/1VuIksa>

Next, we configure IBM Spectrum Scale to use Persistent Reservation. In our example, we have two nodes sharing four disks in IBM Spectrum Scale configuration, and configured as NSDs. According to the previous configuration procedure, we stop IBM Spectrum Scale daemons by using the **mmshutdown** command on all nodes, as shown in the Example 4-43. The existing IBM Spectrum Scale configuration does not use Persistent Reservation but can be easily migrated to SCSI-3 Persistent Reservation.

Example 4-43 Shutting down IBM Spectrum Scale daemons on all nodes: IBM Spectrum Scale 2-node cluster

```
root@gpfs1750:/> mmshutdown -a
Tue Sep 1 17:39:07 EDT 2015: mmshutdown: Starting force unmount of GPFS file systems
gpfs1740: forced unmount of /DR
gpfs1750: forced unmount of /DR
Tue Sep 1 17:39:12 EDT 2015: mmshutdown: Shutting down GPFS daemons
gpfs1750: Shutting down!
gpfs1740: Shutting down!
gpfs1740: 'shutdown' command about to kill process 12845112
gpfs1750: 'shutdown' command about to kill process 12189928
Tue Sep 1 17:39:19 EDT 2015: mmshutdown: Finished
```

Because IBM Spectrum Scale daemons are stopped, the required disk parameters to activate them to use Persistent Reservation of PR shared type can now be changed. This reservation type means that a LUN can be accessed by many hosts based on host-owned Persistent Reservation key. SCSI-3 reservation helps in avoiding “split-brain” scenarios and also on identifying and evicting a dead path in case of failure for a certain LUN.

The multipath disk driver used in our scenario is SDDPCM but also native AIX MPIO PCM driver is supported with the minimum required version of AIX 6.1 TL7 with SP4.

IBM Spectrum Scale automatically modifies all required disk parameters just by issuing `mmchconfig usePersistentReserve=yes` command. There is not required to modify manually any disk parameters since IBM Spectrum Scale takes care of all required disks configuration. By activating this configuration, the existing IBM Spectrum Scale disks in the actual configuration are changed and also the added new disks will inherit this property automatically if it is supported.

However, the IBM Spectrum Scale disk parameters can also be changed manually by configuring the disk `reserve_policy` attribute to `PR_shared` and also assigning a value for `PR_Key_value`. The `PR_Key_value` is chosen to be the node number in our case. The value can be chosen as a hex number and it must be different for every cluster node.

Note: When the `PR_Key_value` is not specified when the disk attributes are changed for `reserve_policy=PR_shared`, you are not able to change the `reserve_policy` for a disk.

On the Storwize v7000 storage side, only the minimum firmware configuration level is required to support SCSI-3 reservation.

On every cluster's node, the `hdisks` parameter is changed as shown in Example 4-44.

Example 4-44 Changing PR_Shared and PR_key attribute values

```
root@gpfs1750: /> for i in 1 2 3 4; do chdev -l hdisk$i -a reserve_policy=PR_shared -a
PR_key_value=0001;done
hdisk1 changed
hdisk2 changed
hdisk3 changed
hdisk4 changed
```

```
root@gpfs1750: /> for i in 1 2 3 4; do echo hdisk$i;lsattr -El hdisk$i |egrep
'reserve_policy|PR_key_value';done
hdisk1
PR_key_value      0001                Reserve Key                True
reserve_policy    PR_shared              Reserve Policy              True
hdisk2
PR_key_value      0001                Reserve Key                True
reserve_policy    PR_shared              Reserve Policy              True
hdisk3
PR_key_value      0001                Reserve Key                True
reserve_policy    PR_shared              Reserve Policy              True
hdisk4
PR_key_value      0001                Reserve Key                True
reserve_policy    PR_shared              Reserve Policy              True
```

In our environment, the `hdisk` numbers are the same but they might differ in other systems. Modifying disk attributes must be consistent across a cluster's nodes on the IBM Spectrum Scale disks.

Note: Persistent Reserve is supported on IBM Spectrum Scale tiebreaker disks with GPFS V3.5.0.21, or later, and IBM Spectrum Scale V4.1.0.4, or later. For earlier levels, if Persistent Reserve is used over tiebreaker disks, contact IBM Spectrum Scale service for a fix for your level of code.

The IBM Spectrum Scale cluster parameters are modified for supporting Persistent Reservation and minimizing the amount of time that IBM Spectrum Scale will take for a node failure detection as shown in Example 4-45. Changing parameter attributes requires IBM Spectrum Scale to be down on all nodes. In contrast with non-PR environment, where the value for failureDetectionTime is a default of 35 seconds (representing the time that the IBM Spectrum Scale cluster manager will take to detect that a node did not renew its disk lease in the PR configuration), the parameter is configured at its minimum value of 10 seconds.

Example 4-45 Modifying IBM SPectrum Scale cluster parameters for using PR

```
root@gpfs1750: /> mmchconfig usePersistentReserve=yes
Verifying GPFS is stopped on all nodes ...
mmchconfig: Processing disk DRNSD001
mmchconfig: Processing disk DRNSD002
mmchconfig: Processing disk DRNSD003
mmchconfig: Processing disk DRNSD004
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
              affected nodes. This is an asynchronous process.

root@gpfs1750: /> mmchconfig failureDetectionTime=10
Verifying GPFS is stopped on all nodes ...
mmchconfig: Command successfully completed
mmchconfig: Propagating the cluster configuration data to all
              affected nodes. This is an asynchronous process.
```

The configured cluster parameters can be verified by using the **mmfsconfig** command as shown in Example 4-46.

Example 4-46 Validate IBM SS configuration

```
root@gpfs1750: /> mmfsconfig
Configuration data for cluster gpfs.DR.site:
-----
clusterName gpfs.DR.site
clusterId 9334245562478638495
autoload yes
dmapiFileHandleSize 32
ccrEnabled yes
minReleaseLevel 4.1.1.0
cipherList AUTHONLY
usePersistentReserve yes
failureDetectionTime 10
tiebreakerDisks DRNSD001
adminMode central
File systems in cluster gpfs.DR.site:
-----
/dev/fsDR

root@gpfs1750: /work> mmgetstate -aLs
```

Node number	Node name	Quorum	Nodes up	Total nodes	GPFS state	Remarks
1	gpfs1740	1	2	2	active	quorum node
2	gpfs1750	1	2	2	active	quorum node

```

Summary information
-----
```

```

Number of nodes defined in the cluster:      2
Number of local nodes active in the cluster: 2
Number of remote nodes joined in this cluster: 2
Number of quorum nodes defined in the cluster: 2
Number of quorum nodes active in the cluster: 2
Quorum = 1*, Quorum achieved

```

The cluster is now started and the disks are displayed with the attributes pr=yes as shown in Example 4-47.

Example 4-47 Displaying disks attributes

```
root@gpfs1750:/> mmlsnsd -X
```

Disk name	NSD volume ID	Device	Devtype	Node name	Remarks
DRNSD001	AC10147F55CE36DA	/dev/hdisk1	hdisk	gpfs1740	server node,pr=yes
DRNSD001	AC10147F55CE36DA	/dev/hdisk1	hdisk	gpfs1750	server node,pr=yes
DRNSD002	AC10147F55CE36DE	/dev/hdisk2	hdisk	gpfs1740	server node,pr=yes
DRNSD002	AC10147F55CE36DE	/dev/hdisk2	hdisk	gpfs1750	server node,pr=yes
DRNSD003	AC10147F55CE36E3	/dev/hdisk3	hdisk	gpfs1740	server node,pr=yes
DRNSD003	AC10147F55CE36E3	/dev/hdisk3	hdisk	gpfs1750	server node,pr=yes
DRNSD004	AC10147F55CE36E7	/dev/hdisk4	hdisk	gpfs1740	server node,pr=yes
DRNSD004	AC10147F55CE36E7	/dev/hdisk4	hdisk	gpfs1750	server node,pr=yes

The reservation keys for a disk are shown in Example 4-48. For each disk path we have the same registration value (two paths per node); also, two nodes are identified accessing the disk, with the keys managed by IBM Spectrum Scale.

Example 4-48 Displaying disk reservation keys and reservation disk type

```

root@gpfs1740:/> /usr/lpp/mmfs/bin/tsprrreadkeys hdisk1
Registration keys for hdisk1
1. 00006d00000000001
2. 00006d00000000002
3. 00006d00000000002
4. 00006d00000000001

root@gpfs1740:/> /usr/lpp/mmfs/bin/tsprrreadres hdisk1
reservation_info for hdisk1
00000000000000000
reservation_type = Write_Exclusive_All_Registrants

```

For information about troubleshooting Persistent Reserve errors, see the following website:

<http://ibm.co/1IXAHkQ>

4.2.5 Protocols disaster recovery (new on 4.1.1)

Implementing disaster recovery protocols also relies on the AFR-based disaster recovery feature. Protocols disaster recovery provides a solution to allow an IBM Spectrum Scale cluster to fail over to another cluster and fail back. The solution provides sample scripts that automate the setup and functionality for cluster disaster recovery.

For the procedure of setting up the DR cluster for disaster recovery using the sample script, see the following web page:

<http://ibm.co/1VaZJE0>

The following prerequisites are required for the DR cluster for disaster recovery in an IBM Spectrum Scale with protocols:

- ▶ IBM Spectrum Scale is installed and configured.
- ▶ Cluster Export Services are installed and configured, and the shared root file system is defined.
- ▶ All protocols are configured either on primary site or secondary site.
- ▶ All exports needed to be protected using AFM DR must have the same device and fileset name, and the same fileset link point on the DR cluster as defined on the primary cluster.
- ▶ IBM NFS4 stack must be configured on both clusters for the AFM DR transport of data.
- ▶ No data must be written to exports on DR cluster while cluster is acting only as a DR cluster, before a failover.

The following limitations apply to disaster recovery in an IBM Spectrum Scale cluster with protocols:

- ▶ All protocols exports protected for DR must be created from independent filesets as AFM based Disaster Recovery requirement.
- ▶ Nested independent filesets are not supported.
- ▶ Backup and restore of authentication configuration is not supported.
- ▶ On failover and failback or restore all clients need to disconnect and then reconnect.

Cluster Configuration Repository restoration depends on a function that can restore it in the following cases:

- ▶ A single broken quorum node.
- ▶ Loss of quorum in some supported disaster recovery environments, as they are documented in the *IBM Spectrum Scale Advanced Administration Guide*.
- ▶ Depending on enabled protocols on failover, failback or restore authentication might be removed and then reconfigured.

Configuration steps for each protocol can be done manually also. Setting up protocols disaster recovery in the IBM Spectrum Scale cluster are documented at this web page:

<http://ibm.co/1jbEmqA>

4.3 Hadoop configuration using shared storage

IBM Spectrum Scale version 4.1.1 extends Apache Hadoop support from File Place Optimizer (FPO) storage to shared storage.

IBM Spectrum Scale Hadoop connector is enhanced for supporting transparently both FPO based storage pool to leverage data locality and shared storage where locality information is not applicable.

IBM Spectrum Scale version 4.1.1 Hadoop Connector provides this support:

- ▶ Full support for Hadoop version 2.5.
- ▶ Support for Hadoop version 2.6 in compatibility mode (Hadoop file system APIs in 2.6 are not yet implemented).

The following resources provide information for planning and implementing Hadoop on a file system with FPO-enabled storage pools:

- ▶ *Deploying a big data solution using IBM Spectrum Scale*
<http://ibm.co/1FBmW0v>
- ▶ *Best Practices IBM Platform Symphony® and GPFS FPO Configuration and Tuning Guide*
<http://ibm.co/1iSWZP5>
- ▶ Hadoop support for IBM Spectrum Scale
<http://ibm.co/1FB03xf>

Using IBM Spectrum Scale shared storage eliminates the requirements of FPO license and takes advantage of existing cluster file systems when locality information is not required.

The following requirements are for implementing IBM Spectrum Scale Hadoop connector in an Apache Hadoop environment:

- ▶ JRE 1.7 or later versions. Use of OpenJDK and IBM JRE is suggested.
- ▶ IBM Spectrum Scale 4.1.1 or later versions.
- ▶ Processor x86_64 or ppc64.
- ▶ Linux.
- ▶ We suggest validating and rebuilding Hadoop Native Library as necessary when Hadoop distribution is downloaded directly from the Apache website on a 64-bit machine because the distribution is built on a 32-bit machine.

This section provides an example of installation and configuration of Hadoop single node using an IBM Spectrum Scale cluster file system in a shared storage configuration, and using RHEL version 6.6.

Considering the installed Hadoop version 2.6 and having its native libraries recompiled, the following steps are required in order to install and configure the Hadoop connector:

1. Install IBM Spectrum Scale packages on all Hadoop nodes.
2. Install Hadoop Connector using the `mmclsh` command or manually on each node.
3. Configure the Hadoop.

In our environment, the IBM Spectrum Scale cluster uses two nodes with designated quorum and manager roles, and the Hadoop node is configured as client. The cluster configuration is shown in Example 4-49.

Example 4-49 IBM Spectrum Scale cluster information

```
[root@rhel1 work]# mmclshcluster
```

```
GPFS cluster information
=====
```

```
GPFS cluster name:      gpfs.FPO
GPFS cluster id:        7508694778746743426
GPFS UID domain:        gpfs.FPO
Remote shell command:    /usr/bin/ssh
Remote file copy command: /usr/bin/scp
Repository type:         CCR
```

Node	Daemon node name	IP address	Admin node name	Designation
1	rhel1	192.168.229.200	rhel1	quorum-manager
2	rhel2	192.168.229.201	rhel2	quorum-manager
3	rhel8	192.168.229.207	rhel8	

The node rhel8 has assigned to it the client license having installed the GPFS packages by following the standard procedure. The installed GPFS packages are shown in Example 4-50.

Example 4-50 GPFS packages installed on Hadoop node

```
[root@rhel8 bigfs]# rpm -qa |grep gpfs
gpfs.msg.en_US-4.1.1-0.noarch
gpfs.base-4.1.1-0.x86_64
gpfs.hadoop-2-connector-4.1.1-0.x86_64
gpfs.gpl-4.1.1-0.noarch
gpfs.docs-4.1.1-0.noarch
gpfs.gskit-8.0.50-40.x86_64
```

The Hadoop connector package is installed. The next step is to install the connector, as shown in Example 4-51.

Example 4-51 Installing the Hadoop connector

```
[root@rhel8 ~]# /usr/lpp/mmfs/fpo/hadoop-2.5/install_script/deploy_connector.sh
-d Apache -v 2.6 install
DISTRIBUTION=apache
VERSION=2.6
ARCH=Linux-amd64-64
CONNECTOR_VERSION=2.5
CONNECTOR_DIR=/usr/lpp/mmfs/fpo/hadoop-2.5
JAR_FILE=hadoop-gpfs-2.5.jar
JAR_DIR=/home/hadoop/hadoop/share/hadoop/common
NATIVE_LIB_DIR=/usr/lib64
OOZIE_SERVER_DIR=
OOZIE_JAR_DIR=
SLIDER_JAR_DIR=
HADOOP_HOME=/home/hadoop/hadoop

Deploy connector
ln -f -s /usr/lpp/mmfs/fpo/hadoop-2.5/hadoop-gpfs-2.5.jar
/home/hadoop/hadoop/share/hadoop/common/hadoop-gpfs-2.5.jar succeeded.
ln -f -s /usr/lpp/mmfs/fpo/hadoop-2.5/libgpfs_hadoop.so /usr/lib64/libgpfs_hadoop.so
succeeded.
Deploy callbacks for connector
install -Dm755 /usr/lpp/mmfs/fpo/hadoop-2.5/gpfs-connector-daemon /var/mmfs/etc//
succeeded.
install -Dm755
/usr/lpp/mmfs/fpo/hadoop-2.5/install_script/gpfs-callback_start_connector_daemon.s
h /var/mmfs/etc// succeeded.
install -Dm755
/usr/lpp/mmfs/fpo/hadoop-2.5/install_script/gpfs-callback_stop_connector_daemon.sh
/var/mmfs/etc// succeeded.
mmfscalcallback: No callback identifiers were found.
mmfscalcallback: No callback identifiers were found.
Register callbacks.
/usr/lpp/mmfs/fpo/hadoop-2.5/install_script/gpfs-callbacks.sh --add succeeded.
/usr/lpp/mmfs/bin/mmhadoopctl connector start succeeded
```

Note: If Hadoop native libraries are not compiled for 64bit machine, libgpfs_hadoop.so is not linked correctly.

The procedure for installing Hadoop connector on all nodes (Example 4-52) can be performed by using the `mmdsh` command; this information is provided by the README file as part of the `gpfs.hadoop-2-connector` package.

Example 4-52 Install Connector on all nodes

```
mmdsh -N all "HADOOP_HOME=${HADOOP_HOME}
/usr/lpp/mmfs/fpo/hadoop-2.5/install_script/deploy_connector.sh -d apache -v 2.6 install"
```

The connector status is started and monitored by using the `mmhadoopctl` command as shown in Example 4-53.

Example 4-53 Starting Hadoop connector and monitor

```
[root@rhel8 bigfs]# /usr/lpp/mmfs/bin/mmhadoopctl connector start
Hadoop connector 'gpfs-connector-daemon' started.
[root@rhel8 bigfs]# /usr/lpp/mmfs/bin/mmhadoopctl connector status
Hadoop connector 'gpfs-connector-daemon' pid 23840 is already running
```

For all nodes `mmdsh` command is used:

```
mmdsh -N all /usr/lpp/mmfs/bin/mmhadoopctl connector status
```

Depending on the version of Hadoop that is used, the `core-site.xml.template` file, provided with the connector package, is reviewed and applied to Hadoop `core-site.xml` configuration.

The `core-site.xml` configuration as provided by the Hadoop connector documentation is shown in Example 4-54.

Example 4-54 core-site.xml configuration

```
- core-site.xml

<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>gpfs://</value>
  </property>
  <property>
    <name>fs.gpfs.impl</name>
    <value>org.apache.hadoop.fs.gpfs.GeneralParallelFileSystem</value>
  </property>
  <property>
    <name>fs.AbstractFileSystem.gpfs.impl</name>
    <value>org.apache.hadoop.fs.gpfs.GeneralParallelFs</value>
  </property>
  <property>
    <name>gpfs.mount.dir</name>
    <value>/gpfs/bigfs</value>
  </property>
  <property>
    <!-- Required when not using root user to run the job, please configure the groups
that will
        be privileged on filesystem. Multiple groups are seperated by comma -->
    <name>gpfs.supergroup</name>
    <value></value>
  </property>
  <property>
    <!--Optional. The default dfs.blocksize in hadoop is 128MB. It can be
override if any other blocksize is adopted.-->
```

The procedure is described in the README file that is provided with the IBM Spectrum Scale Hadoop connector.

In our case, the IBM Spectrum Scale file system attributes and its mount point are shown in Example 4-55.

Example 4-55 Spectrum Scale file system characteristics

```
[root@rhel8 work]# mmlsfs all
```

```
File system attributes for /dev/nonfp01:
```

```
=====
```

flag	value	description

-f	8192	Minimum fragment size in bytes
-i	4096	Inode size in bytes
-I	16384	Indirect block size in bytes
-m	1	Default number of metadata replicas
-M	2	Maximum number of metadata replicas
-r	1	Default number of data replicas
-R	2	Maximum number of data replicas
-j	cluster	Block allocation type
-D	nfs4	File locking semantics in effect
-k	all	ACL semantics in effect
-n	32	Estimated number of nodes that will mount file system
-B	262144	Block size
-Q	none	Quotas accounting enabled
	none	Quotas enforced
	none	Default quotas enabled
--perfilesset-quota	No	Per-fileset quota enforcement
--filesetdf	No	Fileset df enabled?
-V	14.23 (4.1.1.0)	File system version
--create-time	Sun Sep 27 14:29:05 2015	File system creation time
-z	No	Is DMAPI enabled?
-L	4194304	Logfile size
-E	Yes	Exact mtime mount option
-S	No	Suppress atime mount option
-K	whenpossible	Strict replica allocation option
--fastea	Yes	Fast external attributes enabled?
--encryption	No	Encryption enabled?
--inode-limit	65792	Maximum number of inodes
--log-replicas	0	Number of log replicas
--is4KAligned	Yes	is4KAligned?
--rapid-repair	Yes	rapidRepair enabled?
--write-cache-threshold	0	HAWC Threshold (max 65536)
-P	system	Disk storage pools in file system
-d	gennsd000;gennsd01;gennsd02;gennsd03	Disks in file system
-A	yes	Automatic mount option
-o	none	Additional mount options
-T	/gpfs/bigfs	Default mount point
--mount-priority	0	Mount priority

Next step is to start **yarn** as shown in Example 4-56.

Example 4-56 Starting yarn

```
[hadoop@rhel8 kit]$ start-yarn.sh
starting yarn daemons
starting resourcemanager, logging to /home/hadoop/hadoop/logs/yarn-hadoop-resourcemanager-rhel8.out
localhost: starting nodemanager, logging to /home/hadoop/hadoop/logs/yarn-hadoop-nodemanager-rhel8.out
```

Considering that initial permissions were granted for the hadoop user on /gpfs/bigfs file system, the IBM Spectrum Scale file system is available for Hadoop. We create and list the files on IBM Spectrum Scale file system by using the **hadoop fs** command (Example 4-57).

Example 4-57 Create and list files on Spectrum Scale using hadoop fs

```
[hadoop@rhel8 ~]$ hadoop fs -ls /
15/09/27 20:35:32 INFO GeneralParallelFileSystem.audit: allowed=true ugi=hadoop ip=null
cmd=getFileStatus src=gpfs:/ dst=null perm=null
15/09/27 20:35:32 INFO GeneralParallelFileSystem.audit: allowed=true ugi=hadoop ip=null
cmd=listStatus src=gpfs:/ dst=null perm=null
Found 2 items
drwxr-xr-x - hadoop hadoop 4096 2015-09-27 16:42 /user
drwxr-xr-x - hadoop hadoop 4096 2015-09-27 14:57 /work

[hadoop@rhel8 kit]$ hadoop fs -copyFromLocal /home/hadoop/kit/hadoop-2.6.0-src.tar.gz /
15/09/27 20:38:58 INFO GeneralParallelFileSystem.audit: allowed=true ugi=hadoop ip=null
cmd=getFileStatus src=gpfs:/ dst=null perm=null
15/09/27 20:38:58 INFO GeneralParallelFileSystem.audit: allowed=false ugi=hadoop ip=null
cmd=getFileStatus src=/hadoop-2.6.0-src.tar.gz dst=null perm=null
15/09/27 20:38:58 INFO GeneralParallelFileSystem.audit: allowed=false ugi=hadoop ip=null
cmd=getFileStatus src=/hadoop-2.6.0-src.tar.gz._COPYING_ dst=null perm=null
15/09/27 20:38:58 INFO GeneralParallelFileSystem.audit: allowed=true ugi=hadoop ip=null cmd=create
src=gpfs:/hadoop-2.6.0-src.tar.gz._COPYING_ dst=null perm=rw-r--r--
15/09/27 20:38:58 INFO GeneralParallelFileSystem.audit: allowed=true ugi=hadoop ip=null
cmd=getFileStatus src=gpfs:/hadoop-2.6.0-src.tar.gz._COPYING_ dst=null perm=null
15/09/27 20:38:58 INFO GeneralParallelFileSystem.audit: allowed=true ugi=hadoop ip=null cmd=rename
src=gpfs:/hadoop-2.6.0-src.tar.gz._COPYING_ dst=gpfs:/hadoop-2.6.0-src.tar.gz perm=null

Listing directory content:
[hadoop@rhel8 kit]$ hadoop fs -ls /
Found 4 items
-rw-r--r-- 1 hadoop hadoop 17523255 2015-09-27 20:38 /hadoop-2.6.0-src.tar.gz
drwxr-xr-x - hadoop hadoop 4096 2015-09-27 20:38 /kit
drwxr-xr-x - hadoop hadoop 4096 2015-09-27 16:42 /user
drwxr-xr-x - hadoop hadoop 4096 2015-09-27 14:57 /work
```

As an optional step, the connector log configuration is modified according to the README file provided in the IBM Spectrum Scale Hadoop connector package.



A

New commands

This appendix provides information about the new commands that are introduced in IBM Spectrum Scale 4.1.1 to enable the protocol functionality.

The commands are `mmces`, `mmuserauth`, `mmnfs`, `mmsmb`, `mmobj` and `mmperfmon`, along with `mmdumpperfdata` and `mmprotocoltrace` for data collection and tracing. Some commands such as `mm1scluster`, `mmchnode`, and `mmchconfig` are expanded to provide new functionalities including `gpfs.snap`, which now provides data-gathering about protocols.

This appendix also describes some of the functionalities of the **spectrumscale** toolkit.

The following topics are discussed in this appendix:

- ▶ Commands
- ▶ Installation toolkit

Commands

Table A-1 describes the new commands that are available with IBM Spectrum Scale 4.1.1

Table A-1 Commands

Command	Description
mmces	Refers to protocols used with Cluster Export Services (CES) and is used to manage, configure and show status of protocol addresses, services, node states, logging level, and load balancing.
mmuserauth	Creates, lists, and verifies authentication configuration for the CES. The command configures what protocols (Active Directory, LDAP, Kerberos, files, and others) users will use to access data stored on the GPFS file system using external protocols.
mmnfs	Creates, lists, removes, and configures NFS exports.
mmsmb	Creates, lists, removes, and configures SMB exports.
mmobj	Creates, lists, removes, and configures Object exports related to Swift services.
mmperfmon	Queries metrics from the performance metrics collector. The command can show Spectrum Scale related metrics, compare nodes and show metrics related to protocols and CES.
mmdumpperfdata	Runs all queries and computed metrics used on mmperfmon query for each cluster node, writes the output on CSV files, and archives all information in a .tgz file.
mmprotocoltrace	Starts, stops, and monitors tracing for CES protocols. The command is targeted to use with the SMB protocol and it states how it uses functions existing in other commands. This command might be deprecated in future releases.

Installation toolkit

After you download or extract the IBM Spectrum Scale software, and execute the extracted file for accepting the license agreement, you will find the **spectrumscale** command in the `/usr/lpp/mmfs/<Version>/installer` directory. This command *utility* can help you configure, install, and deploy an IBM Spectrum Scale cluster.

To work, the **spectrumscale** command depends on SSH keys. The command creates a definition file and works in two steps:

1. You add nodes and other configurations.
2. You use **spectrumscale install** to install the nodes and apply the defined configurations.

The steps work the same way to deploy protocols. First you define specific properties and what protocols to enable and then use **spectrumscale deploy** to deploy the protocols.

Table A-2 shows the options or sub-commands available for **spectrumscale**.

Table A-2 The spectrumscale command options

Options	Purpose
node add	Adds node specifications to the cluster definition file.
nsd add	Adds NSD specifications to the cluster definition file.
nsd clear	Clears all NSDs.
nsd delete	Removes a single NSD.
nsd list	Lists all NSDs currently in the configuration.
nsd modify	Modifies an NSD.
filesystem list	Lists all file systems that currently have NSDs assigned to them.
filesystem modify	Changes the block size and mount point of a file system.
auth file	Configures file authentication on protocols in the cluster definition file.
auth object	Specifies Object authentication on protocols in the cluster definition file.
config gpfs	Adds specific IBM Spectrum Scale properties to the cluster definition file.
install	Installs IBM Spectrum Scale on the configured nodes, creates a cluster, and creates NSDs.
config protocols	Provides details about the IBM Spectrum Scale environment to be used during protocol deployment.
config enable	Identifies which protocols are to be enabled during deployment.
config object	Defines object-specific properties to be applied during deployment.
deploy	Creates file systems and deploys protocols on your configured nodes.
upgrade	Upgrades the various components of the installation.

There are some limitations on the use of the toolkit on already installed clusters. For more information, see the “Understanding the **spectrumscale** options” topic at this web page:

<http://ibm.co/1MDhdk>

Also see the “Using the **spectrumscale** options to perform tasks: Explanations and examples” topic:

<http://ibm.co/1Nr0Sbg>



B

Performance monitoring

This appendix contains information and considerations regarding the IBM Spectrum Scale Performance Monitoring tool.

The following topics are discussed in this appendix:

- ▶ Installing mmperfmon components
- ▶ Firewall recommendations for the performance monitoring tool

Installing mmperfmon components

This appendix describes installation and configuration of the required packages for using the **mmperfmon** command to gather performance metrics from IBM Spectrum Scale and its protocols. This Performance Monitoring tool that is included with IBM Spectrum Scale can provide performance information after collecting the metrics from IBM Spectrum Scale and protocols.

The system is started by default and consists of three parts: collector, sensors, and proxies.

The **mmperfmon** command is included in the **gpfs.base** package. Installing and configuring all the components can be done either by using the **spectrumscale** command or by selectively installing only the required packages.

Besides the **mmperfmon** utility, the RPMs for collector and sensors must be installed. The **boost-regex.ppc64** RPM is a required prerequisite package. The prerequisite is installed as shown in Example B-1.

Example B-1 Installing boost-regex.ppc64 prerequisite

```
[root@rhel1750 cdrom]# yum install boost-regex.ppc64
Loaded plugins: langpacks, product-id, subscription-manager
This system is not registered to Red Hat Subscription Management. You can use subscription-manager to
register.
dvd | 4.1 kB 00:00:00
Resolving Dependencies
--> Running transaction check
---> Package boost-regex.ppc64 0:1.53.0-23.el7 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
boost-regex ppc64 1.53.0-23.el7 dvd 291 k

Transaction Summary
=====
Install 1 Package

Total download size: 291 k
Installed size: 2.6 M
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
Warning: RPMDB altered outside of yum.
Installing : boost-regex-1.53.0-23.el7.ppc64 1/1
Verifying : boost-regex-1.53.0-23.el7.ppc64 1/1

Installed:
boost-regex.ppc64 0:1.53.0-23.el7

Complete!
```

The required packages for sensors and collectors are in the `zimon_rpms` directory obtained by extracting the IBM Spectrum Scale Protocol package as shown in Example B-2.

Example B-2 Installing the sensors and collectors

```
[root@rhel1740 kits]#
./Spectrum_Scale_Protocol_Advanced-4.1.1.1-power-Linux.update
tail -n +480 ./Spectrum_Scale_Protocol_Advanced-4.1.1.1-power-Linux.update |
/bin/tar -C /usr/lpp/mmfs/4.1.1.1 --wildcards -xvz installer object_rpms
ganesha_rpms gpfs_rpms smb_rpms zimon_rpms manifest 2> /dev/null 1> /dev/null

- installer
- object_rpms
- ganesha_rpms
- gpfs_rpms
- smb_rpms
- zimon_rpms
- manifest
```

We install, from the `/usr/lpp/mmfs/4.1.1.1/zimon_rpms` folder, the following RPMs as shown in the Example B-3

- ▶ `gpfs.gss.pmsensors-4.1.0-8.el7`
- ▶ `gpfs.gss.pmcollector-4.1.0-8.el7`

Example B-3 installing pmsensors and pmcollector RPMs

```
[root@rhel1740 zimon_rpms]# rpm -ivh *gss*
Preparing... ##### [100%]
Updating / installing...
 1:gpfs.gss.pmsensors-4.1.0-8.el7 ##### [ 50%]
 2:gpfs.gss.pmcollector-4.1.0-8.el7 ##### [100%]
```

Configuring the sensors and collectors can be done automatically with a default configuration. The installer starts and stops the corresponding services on every cluster node as shown in Example B-4. On each node, certain sensors and collectors can be activated or deactivated in reporting to the indicated node.

Example B-4 Configuring sensors automatically

```
[root@rhel1750 zimon]# /opt/IBM/zimon/bin/installSensors.sh
Installing performance monitoring sensors on nodes rhel1740
rhel1750
Sensors will be reporting to rhel1750
rhel1740 Stopping pmsensors (via systemctl): [ OK ]
rhel1750 Stopping pmsensors (via systemctl): [ OK ]
RPM gpfs.gss.pmsensors-4.1.0-8.el7.ppc64.rpm is already installed.
RPM gpfs.gss.pmsensors-4.1.0-8.el7.ppc64.rpm is already installed.
Installation of ZIMonSensors.cfg on host rhel1740 successful!
Installation of ZIMonSensors.cfg on host rhel1750 successful!
Installation of ZIMonSensors.cfg.tmp as ZIMonSensors.cfg on rhel1750 successful
rhel1740 Starting pmsensors (via systemctl): [ OK ]
rhel1750 Starting pmsensors (via systemctl): [ OK ]
rhel1750 Starting pmcollector (via systemctl): [ OK ]
```

Note: The NTP service is must be configured correctly on all the IBM Spectrum Scale cluster nodes.

Having the packages installed, a manual configuration can be performed also on each node, modifying the following configuration files:

- For collectors: /opt/IBM/zimon/ZIMonCollector.cfg
- For sensors: /opt/IBM/zimon/ZIMonSensors.cfg

Also the associated **pmsensors** and **pmcollector** services must be restarted manually after reconfiguration. These operations to configure these files can be performed in one cluster node and distributed among the cluster nodes through SSH.

For an overview of the Performance Monitoring tool and, in particular, sensor activation and specific attributes, see the following page in the IBM Knowledge Center:

<http://ibm.co/1P87zyF>

Manually modifying the configuration files requires an indication of the collectors as shown in Example B-5.

Example B-5 Indicate the collectors on sensor configuration file

```
tail -n 5 /opt/IBM/zimon/ZIMonSensors.cfg
```

```
collectors =  
{  
    host="rhe11750"  
    port="4739"  
}
```

Firewall recommendations for the performance monitoring tool

Table B-1 shows the ports that are required to be opened on every cluster node.

Table B-1 .Ports to be opened on each cluster node

Port number	Protocol	Service name	Components involved in communication
4739	TCP and UDP	Performance monitoring tool	Intra-cluster
8123	TCP	Object metric collection	Intra-cluster
8124	TCP	Object metric collection	Intra-cluster
8125	UDP	Object metric collection	Intra-cluster
8126	TCP	Object metric collection	Intra-cluster
8127	TCP	Object metric collection	Intra-cluster

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.

- ▶ *IBM Spectrum Scale (formerly GPFS)*, SG24-8254

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:

- ▶ IBM Spectrum Scale 4.1.1 FAQ
[Http://ibm.co/1JH0huE](http://ibm.co/1JH0huE)
- ▶ IBM Spectrum Scale Wiki
<http://ibm.co/1aKWtP5>
- ▶ IBM Spectrum Scale Administration and Programming information
<http://ibm.co/1hf1EtZ>

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services



REDP-5254-00

ISBN 0738454656

Printed in U.S.A.

Get connected

