Redbooks
ibm.com/redbooks

IBM Elastic Storage Server Implementation Guide for Version 5.3

Common scenarios and use cases

Luis Bolinches

Puneet Chaudhary

Kiran Ghag

Poornima Gupte

Vasfi Gucer

Nikhil Khandelwal

Ravindra Sure

Redpaper

International Technical Support Organization

IBM Elastic Storage Server Implementation Guide

January 2019

REDP-5487-00

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

First Edition (January 2019)

This edition applies to IBM Elastic Storage Server (ESS) Version 5.3.

This document was created or updated on January 28, 2019.

Contents

Notices

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

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the 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 provide in any way it believes appropriate without incurring any obligation to you.

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®
GPFS™
IBM®
IBM Cloud™
IBM Elastic Storage™
IBM Spectrum™
IBM Spectrum Archive™
IBM Spectrum Protect™
IBM Spectrum Scale™
Linear Tape File System™
Power Systems™
POWER8®
Redbooks®
Redpaper™
Redbooks (logo) ®
WebSphere®

The following terms are trademarks of other companies:

ITIL is a Registered Trade Mark of AXELOS Limited.

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

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

Preface

This IBM® Redpaper™ publication introduces and describes the IBM Elastic Storage™ Server as a scalable, high-performance data and file management solution. The solution is built on proven IBM Spectrum™ Scale technology, formerly IBM General Parallel File System (GPFS™).

IBM Elastic Storage Servers can be implemented for a range of diverse requirements, providing reliability, performance, and scalability. This publication helps you to understand the solution and its architecture and helps you to plan the installation and integration of the environment. The following combination of physical and logical components are required:

- Hardware
- Operating system
- Storage
- Network
- Applications

This paper provides guidelines for several usage and integration scenarios. Typical scenarios include Cluster Export Services (CES) integration, disaster recovery, and multicluster integration.

This paper addresses the needs of technical professionals (consultants, technical support staff, IT Architects, and IT Specialists) who must deliver cost-effective cloud services and big data solutions.

Authors

A team of specialists from around the world produced this paper while they worked at the International Technical Support Organization, Poughkeepsie Center.

Luis Bolinches has been working with IBM Power Systems™ servers for over 15 years and has been with IBM Spectrum Scale™ (formerly known as IBM General Parallel File System (IBM GPFS) for over 8 years. He works 50% for IBM Lab Services in Nordic where he is the subject matter expert (SME) for HANA on IBM Power Systems, and the other 50% is on the IBM Spectrum Scale development team.

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.

Kiran Ghag is an IBM Storage Solution Architect, working with various clients at IBM India. He received his bachelors degree in Computer Engineering from Mumbai University. His current interests include software defined storage using IBM Spectrum family. Kiran joined IBM in 2013 as consultant with 10 years of experience in storage systems, currently helping IBM customers with storage solutions and storage infrastructure optimization.

Poornima Gupte works as a Senior Software Engineer at IBM India. She has a BE, Computer Science degree from Pune Institute of Computer Technology and MS, Computer Science degree from Binghamton University. She joined IBM in 2012 and has been working on IBM Spectrum Scale and IBM ESS as a developer.

Vasfi Gucer is an IBM Technical Content Services Project Leader with the Digital Services Group. He has more than 20 years of experience in the areas of systems management, networking hardware, and software. He writes extensively and teaches IBM classes worldwide about IBM products. His focus has been primarily on cloud computing, including cloud storage technologies for the last 6 years. Vasfi is also an IBM Certified Senior IT Specialist, Project Management Professional (PMP), IT Infrastructure Library (ITIL) V2 Manager, and ITIL V3 Expert.

Nikhil Khandelwal is a senior engineer with the IBM Spectrum Scale development team. He has over 15 years of storage experience on NAS, disk, and tape storage systems. He has led development and worked in various architecture roles. Nikhil currently is a part of the IBM Spectrum Scale client adoption and cloud teams.

Ravindra Sure works for IBM India as a Senior System Software Engineer. He has worked on developing workload schedulers for High Performance Computers, Parallel File Systems, Computing Cluster Network Management and Parallel Programming. He has strong engineering professional skills in distributed systems, parallel computing, C, C++, Python,shell scripting, MPI, and Linux.

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

A**nn Lund, Matt Lesher**
International Technical Support Organization, Poughkeepsie Center and Austin, TX

Steve Duersch, Brian Herr, Nariman Nasef, Aaron S Palazzolo, Richard Rosenthal, Mary Jane Zajac
IBM USA

Tomer Perry
IBM Israel

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 by using leading-edge technologies. Your efforts 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 on line through the following web page:

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 that is 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`

1

Introduction to the IBM Elastic Storage Server

This chapter introduces the IBM Elastic Storage Server solution, its characteristics, and where it fits in the business environments.

This chapter also describes some of the software and hardware characteristics of the Elastic Storage Server, the software Redundant Array of Independent Disks (RAID), and the building block concepts of the solution.

The following sections are presented in this chapter:

1.1 Elastic Storage Server

The IBM Elastic Storage Server is software-defined storage that combines IBM Spectrum Scale (formerly GPFS), IBM software RAID, and performance of IBM POWER8® architecture.

The building block-based solution of the IBM Elastic Storage Server delivers high performance, high availability, and scalable functionalities on clustered file systems for a wide variety of demanding applications.

IBM Spectrum Scale is required to access data on the Elastic Storage Server common repository infrastructure of shared file system through native clients. IBM Spectrum Scale's new capabilities include protocol nodes to allow object access data. They use OpenStack Swift or file access through Server Message Block (SMB) or Network File System (NFS).

1.2 Software RAID

The IBM Spectrum Scale RAID software that is used in the Elastic Storage Server solution runs on standard serial-attached SCSI (SAS) disks in just a bunch of disks (JBOD) arrays. The solution does not require or use any kind of external RAID controller or acceleration. The RAID functions are handled by the software. Use of SAS drives allows greater cost reduction. The solid-state drives (SSDs) option is also available when more performance is needed.

IBM Spectrum Scale RAID supports multiple RAID codes and distributes client data, redundancy information, and spare space across the disks. This approach ensures that if there is a physical disk loss, or even a group of physical disk losses, data availability is unchanged.

Spectrum Scale RAID implements an end-to-end checksum to detect and report faults, read or write errors, and other integrity problems of traditional RAID.

1.2.1 RAID codes

Spectrum Scale RAID in the Elastic Storage Server supports different data protection algorithms that can detect and correct up to three concurrent faults.

The options for RAID configuration are eight stripes of data plus 2 or 3 parity stripes. You can use Reed-Solomon codes or one stripe of data plus 2 or 3 replica stripes. The data plus parity or replica stripes, which are called *tracks*, are illustrated in Figure 1-1 on page 3.

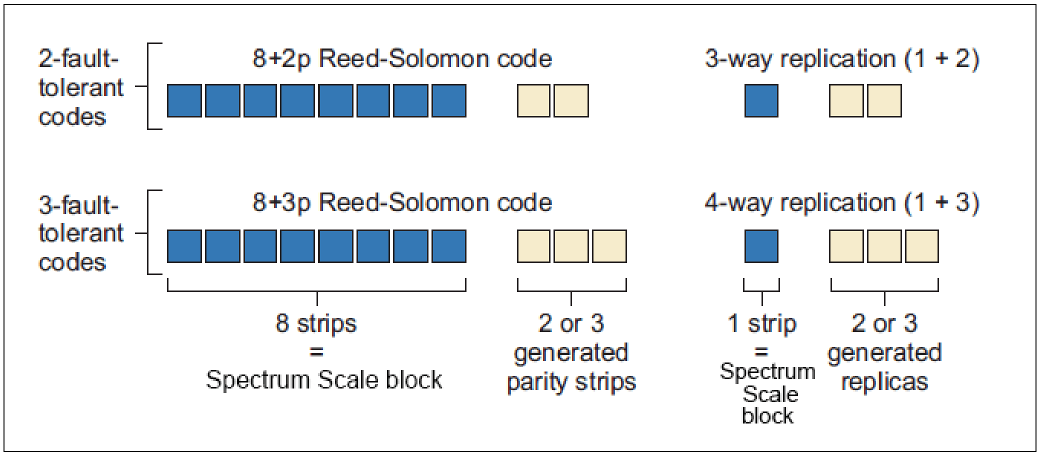

2-fault-tolerant codes
8+2p Reed-Solomon code
3-way replication (1 + 2)
3-fault-tolerant codes
8+3p Reed-Solomon code
4-way replication (1 + 3)
8 strips = Spectrum Scale block
2 or 3 generated parity strips
1 strip = Spectrum Scale block
2 or 3 generated replicas

Figure 1-1 RAID tracks

1.2.2 End-to-end checksum

IBM Spectrum Scale software on the client is used to access data on the Elastic Storage Server. This software is aware that the Spectrum Scale file system is based on Spectrum Scale RAID Network Shared Disks. During a write operation an 8-bytes checksum is calculated, appended to the data, and sent over the network to the Spectrum Scale RAID server. The checksum is verified. Then, Spectrum Scale RAID writes the data along with its checksum on the disks and logs the version number in its metadata.

When a read operation is requested, Spectrum Scale RAID verifies checksum and version on its metadata. If it is OK, it sends the data to the client. If it is not OK, the data is rebuilt based on parity or replication. Then, the data is sent to the client along with newly generated checksum.

The end-to-end checksum feature can prevent and correct silent disk errors or missing disk writes.

1.2.3 Declustered RAID

Spectrum Scale RAID implements its own data and spare disk layout scheme, which reduces the overhead to clients during a recovery from disk failure. To accomplish this, it does not leave all spare space in a single disk. Instead, it spreads or *declusters* user data, redundancy information, and spare space across all the disks of the array. Figure 1-2 on page 4 compares conventional 1+1 RAID layout with a declustered array.

For example, consider seven stripes of data on each disk. Figure 1-2 on page 4 shows the left three arrays of two disks in a replicated 1+1 configuration and a spare. On the left, you can see the data stripes that spread over all seven disks of the declustered array.

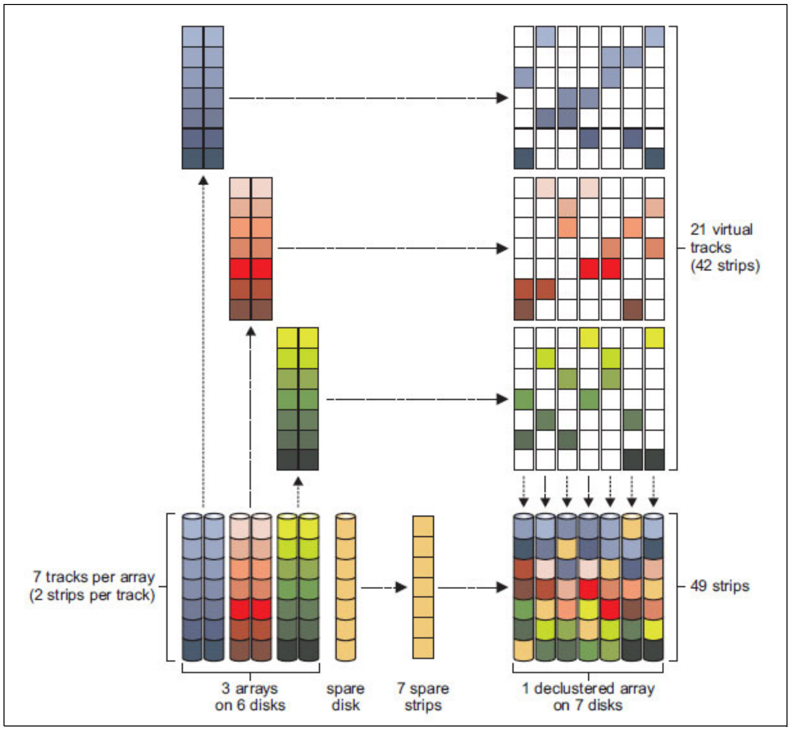

21 virtual tracks (42 strips)
7 tracks per array (2 strips per track)
49 strips
3 arrays on 6 disks
spare disk
7 spare strips
1 declustered array on 7 disks

Figure 1-2 Declustered array versus 1+1 array

In Figure 1-2, notice that on the Elastic Storage Server from v3.5 and even on GL6 models, only one declustered array is used. Figure 1-2 shows 348 HDD, which is a simplified model.

If one disk fails on the traditional 1+1, all data from the remaining disks of the array must be replicated to the spare disk. On the declustered array, the replication occurs on spare space of all the remaining disks. This approach can decrease the rebuild impact by as much as four times. Figure 1-3 shows a simplified model.

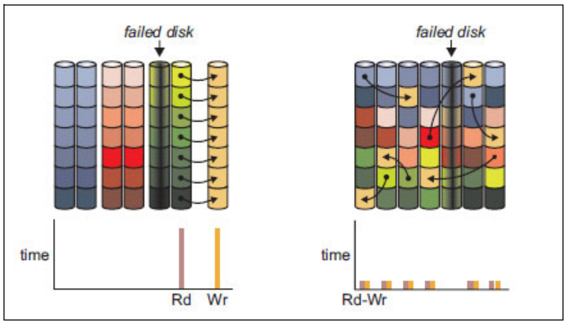


Figure 1-3 Array rebuild operation

Consider the case of an Elastic Storage Server that uses RAID 8+2 or two-way replication (1+2). If one disk is lost, the rebuild operation starts with low priority with even lower impact for the clients. With this array configuration, two concurrent disk losses must occur before the system treats the rebuild as critical to be run on high priority. Using 8+3 RAIDs or three-way replication (1+3), the rebuild operation becomes critical only when three concurrent disk losses occur in the same declustered array.

1.3 Building blocks

The *building block* is the minimum configuration of an Elastic Storage Server and is also the unit of expansion of the solution. If more space is needed, either a new building block is added to the solution, or it is expanded to a higher model.

Each building block consists of two IBM POWER8 servers and a specific number of storage enclosures (1, 2, 4, or 6), depending on the model.

The POWER8 servers are model S822L and the storage enclosures can be one of the following types:

- DCS3700 (4U 60 drives) units for models GL2, GL4, and GL6
- EXP24S SFF Gen2-bay drawer (2U 24 drives) for models GS1, GS2, GS4, or GS6.

The GS models can use 2.5" 10 K rpm HDD or 2.5" SSD. GL models can use 3.5" NL-SAS HDDs. Models GL and GS can be mixed to achieve specific needs for client applications that use the storage. For example, SSDs for metadata and NL-SAS for data storage.

Elastic Storage Server V5.3 introduced two new *hybrid* models to the ESS family: *GH models*. ESS Hybrid models place spinning disks and flash storage in a single ESS node, giving better storage density, and a smaller footprint at a lower cost. GH14 has one 2U drawer with 24 Solid State Drives (SSD) combined with four 5U drawers with a 7200 rpm spinning disk. The GH2R has two 2U drawers with four 5U drawers.

Like the GS models, the SSDs are either 3.84 TB or 15.3 TB capacities. The 5U drawers are similar to those in the GL models, either 4 TB, 8 TB, or 10 TB capacities.

A new Enterprise Slim Rack (S42) is now available to hold these. The S42 is available for all ESS orders, including the GS, GL, and new GH models. You can refer to the following announcement letter for more details: https://ibm.biz/BdYkqC.

Model upgrades: For clients with existing Power Systems servers, nondisruptive upgrades are introduced with Elastic Storage Server V5.3. With nondisruptive upgrades, the original data is preserved and accessible in place. Meanwhile, the added capacity of the new storage drawers is integrated into the original capacity without interruption. The existing data is rebalanced across both the new and old storage drawers, again without interruption to the Elastic Storage Server cluster. The new storage capacity is immediately available for use.

1.4 Value added

IBM Spectrum Scale RAID and support from IBM for the solution, contribute to the added value, including a tuned solution from the IBM services team.

IBM engineering and testing teams created a solution that provides greater performance and data availability. They included these design factors in the solution:

- adapter placement on the servers
- number of adapters
- number of disks on the drawers
- cabling
- number of drawers to the software versions
- the way data is placed on disks

Elastic Storage Server offers scalability from 40 TB to hundreds of petabytes. It supports 10-Gigabit Ethernet (10 GbE), 40-Gigabit Ethernet (40 GbE), 100-Gigabit Ethernet (100 GbE), and 100 Gb/s EDR InfiniBand.

Big data requires easy storage growth. The IBM Elastic Storage Server building block approach meets that requirement. Adding more storage servers adds to the overall capacity, bandwidth, and performance with a single name space.

By leveraging JBODs and IBM Power Systems servers, the Elastic Storage Server provides price/performance storage for analytics, genomics, video streaming, machine learning, metadata services, artificial intelligence, technical computing, and cloud computing environments.

The Elastic Storage Server's modern declustered RAID technology can recover from multiple disk failures in minutes, versus hours and days in older technology. As a result, the solution delivers predictable performance and data protection. 8+2 and 8+3 RAID protection and platter-to-client data protection are included with the Elastic Storage Server.

Various tools and scripts are used by the services team during the delivery. These items ensure that every piece of the solution integrates together and customer requirements are met.

For more information about the Elastic Storage Server, refer to the following websites:

http://www.ibm.com/systems/storage/spectrum/ess
https://ibm.biz/BdYkqW

2

Planning and integration

This chapter contains guidelines and considerations for proper planning, installation, and configuration of the IBM Elastic Storage Server.

This chapter presents configurations and integration considerations for a smooth Elastic Storage Server deployment into an existing or a new IT environment. The following topics are covered:

- 2.1, “Elastic Storage Server overview” on page 8
- 2.2, “Elastic Storage Server installation and upgrading” on page 8
- 2.3, “Elastic Storage Server networks” on page 8
- 2.4, “Elastic Storage Server: Storage parameters” on page 11
- 2.5, “Recommended guidelines for Spectrum Scale RAID” on page 21
- 2.6, “Elastic Storage Server file system configurations” on page 24
- 2.7, “Elastic Storage Server ordering” on page 28
- 2.8, “Elastic Storage Server client integration” on page 30
- 2.9, “Monitoring” on page 30

2.1 Elastic Storage Server overview

As shown in Chapter 1, “Introduction to the IBM Elastic Storage Server” on page 1, the Elastic Storage Server is a high-capacity and high-performance storage system. It combines IBM Power Systems servers, storage enclosures and drives, software (including IBM Spectrum Scale Redundant Array of Independent Disks (RAID)), and networking components.

The Elastic Storage Server V5.3 is an IBM file-based storage. Its core technology is the IBM Spectrum Scale V5.0.1.2, which includes IBM Spectrum Scale RAID combined with specific IBM Power Systems servers.

2.2 Elastic Storage Server installation and upgrading

The IBM Lab Services team can install an Elastic Storage Server building block as an included service part of acquisition. Alternatively, the customer’s IT team can do the installation.

The Elastic Storage Server documentation is at the following web page:

https://ibm.biz/BdYQuZ

The following documents provide information that you need for proper deployment, installation, and upgrade procedures for an IBM Elastic Storage System:

- IBM Elastic Storage Server: Planning for the system, service maintenance packages, and service procedures

  http://ibm.co/1LYNkBa

- IBM Elastic Storage Server: FAQs

  https://ibm.biz/BdYQuf

2.3 Elastic Storage Server networks

Network planning and configuration are important steps for a rapid and successful Elastic Storage Server deployment. This section provides recommendations and details regarding the Elastic Storage Server network communications setup and node-naming conventions as shown in Table 2-1 on page 9.

> **Note:** Network planning is an important part of the preparation for deploying an Elastic Storage Server.

Table 2-1 Network descriptions

Number	Description	Remarks
1a.	**Support network (for PPC64BE system):** ► This private network connects the HMC with the flexible service processor (FSP) of the EMS and the I/O server nodes. ► HMC uses this network to discover the EMS and the I/O server nodes. It also does hardware management activities, such as create and manage logical partitions, allocate resources, power control, and reboot. ► This is a private network between the HMC and FSPs. This network must not be seen by the operating system that runs in the node that is being managed (for example, EMS and I/O server node). ► Requires planning if the HMC is not part of the original order and supplied by the customer.	EMS (xCAT) and I/O server (FSP) are connected to the HMC through the support network. The HMC must be set to be the DHCP server for the support network.
1b.	**Support network (for PPC64LE system):** ► PPC64LE systems come without an HMC and EMS server itself connects to flexible server processor (FSP) of I/O server nodes. ► EMS server uses IPMI protocol to discover and communicate with IO Nodes. ► Each node should be assigned unique IP for the FSP interface from FSP subnet. ► This should be private network between the EMS and FSP interface or all nodes. ► It should not be seen by the operating system that runs in the IO nodes or client nodes.	Control Panel Function 30 can be used to display current FSP IP address on each node individually. Example: IP: 10.0.0.12
2.	**Management or provisioning network:** ► This network connects the EMS with the HMC and other I/O servers. ► This network is visible by the operating system that runs on the nodes. ► The EMS uses this network to communicate with the HMC and to discover I/O server nodes. ► This network is also used for provisioning the node and therefore deploys and installs the operating system in the I/O server nodes. No other DHCP servers can exist on this network. ► Requires additional planning if the EMS is not part of the order and building blocks are managed by an existing EMS.	The HMC, and I/O server (OS) nodes are connected to the EMS node through this network. The EMS will be the DHCP server in this network (although it does serve any IP address).
3.	**Public or clustering network** This network is for high-performance data access. This network in most cases can also be part of the clustering network.	This network is typically composed of 10 Gb, 40 Gb Ethernet, or InfiniBand.
4.	**Domain of the management network** This is used by the EMS for proper resolution of short host names and must be in lowercase.	Example: gpfs.net
5.	**HMC node IP address on the management network, short and long host name: (for PPC64LE system)** ► This IP address must be configured and the attached link to the network interface must be up. ► The EMS must be able to reach the HMC that uses this address. The long name and short name must be in lowercase. ► The host name must never end in "-enx" for any x.	Example: IP: 192.168.45.9 Shortname: hmc1 FQDN: hmc1.gpfs.net

Number	Description	Remarks
6.	**EMS node IP address**, short name and long host name (FQDN): ► This IP address must be configured and the link to the interface must be up. ► The management network must be reachable from this IP address. ► There is a valid gateway for this IP address. ► The long and short host name must be in lowercase. ► The host name must never end in "-enx" for any x.	Example: IP: 192.168.45.10 Shortname: ems1 FQDN: ems1.gpfs.net
7.	**IP address of the I/O server nodes**, short and long host name: ► This address is assigned to the I/O server nodes during node deployment. ► The I/O server nodes must be able to reach the management network through this address. ► The names that are defined here must match the name of the partition that is created for this node through the HMC. ► The long and short host name must be in lowercase. ► The host name must never end in "-enx" for any x.	Example: IO Server1: IP: 192.168.45.11 Shortname: gssio1 FQDN: gssio1.gpfs.net IO Server2: IP: 192.168.45.12 Shortname: gssio2 FQDN: gssio2.gpfs.net
8.	**EMS node management network interface:** This interface must have the IP address of item number 6 of this table. Only one IP address can be assigned for this interface. It can be obtained with the `ip addr` command.	Example: enP7p128s0f0
9.	**Prefix or Suffix for public/clustering network:** Management network typically runs over 1 Gb. Public and clustering runs on a high-speed network that is implemented on 10 Gb Ethernet, 40 Gb Ethernet, or InfiniBand network. It is customary to use host names for the high-speed network by using prefix and suffix of the actual host name. Do not use "-enx" for any x as a suffix.	Suffix example: -ib, -10G, -40G
10.	**High-speed cluster network IP address of the I/O server and the EMS nodes:** In the example, 172.10.0.11 is the IP address that the GPFS daemon uses for clustering. Corresponding short and long names are gssio1-ib and gssio1-ib.data.net, respectively. **Note:** Do not make changes in the `/etc/host` file for the high-speed network until the deployment is complete. Do not create or enable the high-speed network interface until the deployment is complete.	Example: 172.10.0.10 ems1-ib.gpfs.net ems1-ib 172.10.0.11 gssio1-ib.data.net gssio1-ib 172.10.0.12 gssio2-ib.data.net gssio2-ib
11.	**IP address and access credentials are required:** (uid/pw) of the switch that implements the xCAT management network. The switch must allow BOOTP to go through. Some switches generate excessive STP messages, which interferes with network boot. STP must be disabled to mitigate this.	None

It is highly recommended to use a redundant network communication both for private and public networks. Bonding interface can be created for the Ethernet or the InfiniBand network interfaces. Examples of bonding network configurations are provided in the Deploying the Elastic Storage Server documentation found at the following website:

https://ibm.co/2DWFs6V

2.4 Elastic Storage Server: Storage parameters

The Elastic Storage Server uses declustered arrays and implements as controller the IBM Spectrum Scale RAID. This controller decreases the rebuilding impact and client overhead of a conventional RAID because it stripes client data across all the storage nodes of a cluster. As a result, file system performance becomes less dependent on the speed of any single rebuilding.

This section presents the elements and associated storage parameters that you must account for to properly plan and size a solution.

2.4.1 Recovery group server parameters

To enable a Spectrum Scale cluster node as a recovery group server, the **nsdRAIDTracks** parameter for **mmchconfig** must be set to a nonzero value. And the GPFS daemon must be restarted on the node.

The **nsdRAIDTracks** parameter defines the maximum number of VDisk ("VDisks" on page 19) track descriptors that the server can have in memory at a given time. The volume of actual VDisk data that the server can cache in memory is governed by these factors:

- the size of the Spectrum Scale page pool on the server
- the value of the **nsdRAIDBufferPoolSizePct** configuration parameter

The **nsdRAIDBufferPoolSizePct** parameter defaults to 50% of the page pool on the server. A recovery group server should be configured with a substantial amount of page pool, on the order of tens of gigabytes. A recovery group server becomes a Network Shared Disk (NSD) server after NSDs are defined on the VDisks in the recovery group. So, the **nsdBufSpace** parameter also applies. (These servers are called *VDisk NSDs*, in this document.

The default for **nsdBufSpace** is 30% of the page pool. This value can be decreased to its minimum value of 10%, because the VDisk data buffer pool is used directly to serve the VDisk NSDs.

The VDisk track descriptors, as governed by **nsdRAIDTracks**, include such information as the RAID code, track number, and status. The descriptors also contain pointers to VDisk data buffers in the Spectrum Scale page pool, as governed by **nsdRAIDBufferPoolSizePct**. It is these buffers that hold the actual VDisk data and redundancy information.

Recovery group creation

Recovery groups are created by using the **mmcrrecoverygroup** command, which takes the following arguments:

- The name of the recovery group to create.
- The name of a stanza file that describes the declustered arrays and pdisks within the recovery group.
- The names of the Spectrum Scale cluster nodes that are the primary and, if specified, backup servers for the recovery group.

When a recovery group is created, the GPFS daemon must be running with the **nsdRAIDTracks** configuration parameter in effect on the specified servers.

Recovery group server failover

It is recommended that you assign two servers to a recovery group. When you do this, one server is the preferred and primary server for the recovery group and the other server is the backup server.

Only one server can serve the recovery group at any given time; this server is known as the *active recovery group server*. The server that is not currently serving the recovery group is the *standby server*. If the active recovery group server is unable to serve a recovery group, it relinquishes control of the recovery group and passes it to any available standby server.

The failover from the active to the standby server should be transparent to any Spectrum Scale file system that is using the VDisk NSDs in the recovery group. There is a pause in access to the file system data in the VDisk NSDs of the recovery group. During this pause, the recovery operation takes place on the new server. This server failover recovery operation involves the new server opening the component disks of the recovery group and playing back any logged RAID transactions.

The active server for a recovery group can be changed by the IBM Spectrum Scale RAID administrator by using the **mmchrecoverygroup** command. This command can also be used to change the primary and backup servers for a recovery group.

2.4.2 pdisks

The IBM Spectrum Scale RAID pdisk is an abstraction of a physical disk. A pdisk corresponds to exactly one physical disk, and belongs to exactly one declustered array within exactly one recovery group. Before we discuss how declustered arrays collect pdisks into groups, it is useful to describe the characteristics of pdisks.

A recovery group can contain a maximum of 512 pdisks. A declustered array within a recovery group can contain a maximum of 256 pdisks. The name of a pdisk must be unique within a recovery group. That is, two recovery groups can each contain a pdisk that is named disk10. But a recovery group cannot contain two pdisks that are named disk10, even if they are in different declustered arrays.

Typically, a pdisk is created with the **mmcrrecoverygroup** command. This command assigns the pdisk to a declustered array within a newly created recovery group. In unusual situations, pdisks can also be created and assigned to a declustered array of an existing recovery group with the **mmaddpdisk** command.

To create a pdisk, a stanza must be supplied to the **mmcrrecoverygroup** or **mmaddpdisk** commands that specify these values:

- the pdisk name
- the declustered array name to which it is assigned
- a block device special file name for the entire physical disk as it is configured by the operating system on the active recovery group server

A sample pdisk creation stanza follows:

```
%pdisk: pdiskName=c073d1
device=/dev/hdisk192
da=DA1
nPathActive=2
nPathTotal=4
```

Other stanza parameters might be present.

The device name for a pdisk must refer to the entire single physical disk. pdisks should not be created by using virtualized or software-based disks (for example, logical volumes, disk partitions, logical units from other RAID controllers, or network-attached disks). The exception to this rule is non-volatile RAM (NVRAM) volumes that are used for the log tip VDisk, which is described in “Log VDisks” on page 20. For a pdisk to be created successfully, the physical disk must be present and functional at the specified device name on the active server. The physical disk must also be present on the standby recovery group server, if one is configured.

The physical disk block device special name on the standby server is almost certainly different, and will be discovered automatically by IBM Spectrum Scale.

The attributes of a pdisk include the physical disk’s unique worldwide name (WWN), its field replaceable unit (FRU) code, and its physical location code. pdisk attributes can be displayed by using the **mmlspdisk** command. The pdisk device paths and the pdisk states are of particular interest.

You replace pdisks that fail and are marked for replacement by the disk hospital by using the **mmchcarrier** command. In unusual situations, pdisks can be added or deleted by using the **mmaddpdisk** or **mmdelpdisk** commands. Consider the case where a pdisk is deleted through replacement or the **mmdelpdisk** command. In this case, the pdisk abstraction ceases to exist only after all of the data that it contained is rebuilt onto spare space. (This is a requirement, even though the physical disk might have been removed from the system).

pdisks are normally under the control of IBM Spectrum Scale RAID and the disk hospital. However, in some situations the **mmchpdisk** command can be used to manipulate pdisks directly. For example, consider the scenario where a pdisk is removed temporarily to allow for hardware maintenance on other parts of the system. You can use the **mmchpdisk --begin-service-drain** command to drain the data before you remove the pdisk.

After you bring the pdisk back online, you can use the **mmchpdisk --end-service-drain** command to return the drained data to the pdisk.

> **Note:** This process requires that you have sufficient spare space in the declustered array for the data that is to be drained. If the available spare space is insufficient, it can be increased with the **mmchrecoverygroup** command.

pdisk paths

To the operating system, physical disks are made visible as block devices with device special file names, such as `/dev/sdbc` (on Linux) or `/dev/hdisk32` (on IBM AIX®). Most pdisks that IBM Spectrum Scale RAID uses are in JBOD arrays, except for the NVRAM pdisk that is used for the log tip VDisk. To achieve high availability and throughput, the physical disks of a JBOD array are connected to each server by multiple (usually two) interfaces. You set them up in a configuration that is known as multipath (or dualpath). When two operating system block devices are visible for each physical disk, IBM Spectrum Scale RAID refers to them as the paths to the pdisk.

In normal operation, the paths to individual pdisks are discovered by IBM Spectrum Scale RAID automatically. There are only two instances when a pdisk must be referred to by its explicit block device path name:

- During recovery group creation that uses the **mmcrrecoverygroup** command.
- When you add new pdisks to an existing recovery group with the **mmaddpdisk** command.

In both of these cases, only one of the block device path names as seen on the active server needs to be specified. Any other paths on the active and standby servers are discovered automatically.

For each pdisk, the **nPathActive** and **nPathTotal** stanza parameters can be used to specify the expected number of paths to that pdisk, from the active server and from all servers. This option allows the disk hospital to verify that all expected paths are present and functioning.

The operating system might be capable of internally merging multiple paths to a physical disk into a single block device. When IBM Spectrum Scale RAID is in use, the operating system multipath merge function must be disabled because IBM Spectrum Scale RAID itself manages the individual paths to the disk.

pdisk stanza format

pdisk stanzas have three mandatory parameters and five optional parameters, and look as follows:

```
%pdisk: pdiskName=PdiskName
device=BlockDeviceName
da=DeclusteredArrayName
[nPathActive=ExpectedNumberActivePaths]
[nPathTotal=ExpectedNumberTotalPaths]
[rotationRate=HardwareRotationRate]
[fruNumber=FieldReplaceableUnitNumber]
[location=PdiskLocation]
```

where:

pdiskName=PdiskName

Specifies the name of a pdisk.

device=BlockDeviceName

Specifies the name of a block device. The value that is provided for BlockDeviceName must refer to the block device as configured by the operating system on the primary recovery group server. Alternatively, the node name can be prefixed to the device block name.

Sample values for BlockDeviceName are shown here:

- For Linux: /dev/sdbc and //nodename/dev/sdbc
- For IBM AIX: hdisk32, /dev/hdisk32, and //nodename/dev/hdisk32

Only one BlockDeviceName needs to be used, even if the device uses multipath and has multiple device names.

da=DeclusteredArrayName

Specifies the DeclusteredArrayName in the pdisk stanza, which implicitly creates the declustered array with default parameters.

nPathActive=ExpectedNumberActivePaths

Specifies the expected number of paths for the connection from the active server to this pdisk. If this parameter is specified, the **mmlsrecoverygroup** and **mmlspdisk** commands display warnings in this situation: The number of paths does not match the expected number for a pdisk that should be functioning normally. If this parameter is not specified, the default is 0, which means “do not issue such warnings”.

Sample values are as follows:

- 2 for all pdisks that are in an Elastic Storage Server disk enclosure (or the IBM Power 775 Disk Enclosure)
- 1 for the NVRAM pdisk that is used for the log tip VDisk.

nPathTotal=ExpectedNumberTotalPaths

Specifies the expected number of paths for the connection from all active and backup servers to this pdisk. If this parameter is specified, the **mmlsrecoverygroup** and **mmlspdisk** commands display warnings in this situation: The number of paths does not match the expected number for a pdisk that should be functioning normally. If this parameter is not specified, the default is 0, which means “do not issue such warnings”.

Sample values are as follows:

- 4 for all pdisks in an Elastic Storage Server disk enclosure (or the IBM Power 775 Disk Enclosure)
- 1 for the NVRAM pdisk that is used for the log tip VDisk.

rotationRate=HardwareRotationRate

Specifies the hardware type of the pdisk: NVRAM, SSD, or a rotating HDD. The only valid values are as follows:

- the string NVRAM
- the string SSD
- a number 1025 - 65535 (inclusive) indicating the rotation rate in revolutions per minute for HDDs

For all pdisks that are used in an Elastic Storage Server disk enclosure (or the IBM Power 775 Disk Enclosure), there is no need to specify this parameter. Instead, the hardware type and rotation rate are determined from the hardware automatically. This parameter should be specified for the NVRAM pdisk on the Elastic Storage Server only. The default is to rely on the hardware to identify itself. Alternatively, you can leave the hardware type and rotation rate unknown if the hardware cannot identify itself.

A sample value is the string NVRAM for the NVRAM pdisk that is used for the log tip VDisk.

fruNumber=FieldReplaceableUnitNumber

Specifies the unit number for the field-replaceable unit (FRU) that is needed to repair this pdisk if it fails. For all pdisks that are used in an Elastic Storage Server disk enclosure (or the IBM Power 775 Disk Enclosure), there is no need to specify this parameter. This parameter is automatically determined from the hardware. For the NVRAM pdisk used in the log tip VDisk, the user can enter a string here. This string is displayed to service personnel when replacement of that pdisk is performed. However, setting this value for the NVRAM pdisk is not required. The service replacement procedure for that pdisk is specific to that particular type of hardware. The default is to rely on the hardware to identify itself. If the hardware is not able to identify itself, leave the FRU number unknown.

location=PdiskLocation

Specifies the physical location of this pdisk. For all pdisks used in an Elastic Storage Server disk enclosure (or the IBM Power 775 Disk Enclosure), there is no need to specify this parameter. The parameter is automatically determined from the hardware. For the NVRAM pdisk used in the log tip VDisk, the user can enter a string here, which is displayed in the output of **mmlspdisk**. The default is to rely on the location reported by the hardware, or leave the location unknown.

A sample value is SV21314035-5-1, which describes a pdisk in enclosure serial number SV21314035, drawer 5, slot 1.

pdisk states

IBM Spectrum Scale RAID maintains its view of a pdisk and its corresponding physical disk by using a pdisk state. The pdisk state consists of multiple keyword flags, which can be displayed by using the **mmlsrecoverygroup** or **mmlspdisk** commands. You can also use the Elastic Storage Server GUI to display pdisk states. The state of pdisks is displayed in these views: **Arrays** → **Physical**, **Monitoring** → **System**, and **Monitoring** → **System Details**.

In addition, information about pdisks with a negative state (disks that should be replaced, for example) is displayed in the **Monitoring** → **Events** view.

The pdisk state flags indicate in detail how IBM Spectrum Scale RAID is currently using or managing a disk. The state of a pdisk is also summarized in its user condition, as described at the end of this section.

In normal circumstances, the state of the vast majority of pdisks is represented by the sole keyword `ok`. This means that IBM Spectrum Scale RAID considers the pdisk to be healthy in this sense:

- The recovery group server is able to communicate with the disk.
- The disk is functioning normally.
- The disk can be used to store data.

The diagnosing flag is present in the pdisk state when the IBM Spectrum Scale RAID disk hospital suspects, or attempts to correct, a problem.

If IBM Spectrum Scale RAID is unable to communicate with a disk, the pdisk state includes the keyword `missing`. If a missing disk becomes reconnected and functions properly, its state changes back to ok. The read-only flag means that a disk has indicated that it can no longer safely write data. A disk can also be marked by the disk hospital as failing, perhaps due to an excessive number of media or checksum errors. When the disk hospital concludes that a disk is no longer operating effectively, it declares the disk to be dead.

The number of non-functioning (dead, missing, failing, or slow) pdisks might reach or exceed the replacement threshold of their declustered array. In this case, the disk hospital adds the flag replace to the pdisk state. This state indicates that physical disk replacement should be performed as soon as possible.

The state of a pdisk might indicate that it can no longer behave reliably. In this case, IBM Spectrum Scale RAID rebuilds the pdisk's data onto spare space on the other pdisks in the same declustered array. This is called *draining the pdisk*. A keyword in the pdisk state flags indicates that a pdisk is draining or has been drained. The flag systemDrain means that IBM Spectrum Scale RAID has decided to rebuild the data from the pdisk. The flag adminDrain means that the IBM Spectrum Scale RAID administrator issued the **`mmdelpdisk`** command to delete the pdisk.

IBM Spectrum Scale RAID uses pdisks to store user (Spectrum Scale file system) data, its own internal recovery group data, and VDisk configuration data. More pdisk state flags indicate when these data elements are not present on a pdisk. When a pdisk starts draining, IBM Spectrum Scale RAID first replicates the recovery group data and VDisk configuration data onto other pdisks. When this process is complete, the flags noRGD (no recovery group data) and noVCD (no VDisk configuration data) are added to the pdisk state flags. When the slower process of removing all user data is complete, the noData flag is added to the pdisk state.

To summarize, the vast majority of pdisks are in the ok state during normal operation. The ok state indicates the following state:

- The disk is reachable, functioning, and not draining.
- The disk contains user data and IBM Spectrum Scale RAID recovery group and VDisk configuration information.

A more complex example of a pdisk state is dead/systemDrain/noRGD/noVCD/noData for a single pdisk that has failed. This set of pdisk state flags indicates the following state for the pdisk:

- It was declared dead by the system.
- It was marked to be drained.
- All of its data (recovery group, VDisk configuration, and user) has been successfully rebuilt onto the spare space on other pdisks.

2.4.3 Declustered arrays

Declustered arrays are disjoint subsets of the pdisks in a recovery group. VDisks are created within declustered arrays, and VDisk tracks are declustered across all of an array's pdisks. A recovery group can contain up to 16 declustered arrays. A declustered array can contain up to 256 pdisks (but the total number of pdisks in all declustered arrays within a recovery group cannot exceed 512).

A pdisk can belong to only one declustered array. The name of a declustered array must be unique within a recovery group. That is, two recovery groups can each contain a declustered array that is named DA3, but a recovery group cannot contain two declustered arrays named DA3. The pdisks within a declustered array must all be of the same size and should all have similar performance characteristics.

Typically, you use the **`mmchrecoverygroup`** command to create a declustered array with its member pdisks and its containing recovery group. A declustered array can also be created by using the **`mmaddpdisk`** command. The command adds pdisks to a declustered array that does not yet exist in a recovery group.

You can delete a declustered array by deleting its last member pdisk, or by deleting the recovery group in which it resides. Any VDisk NSDs and VDisks within the declustered array must already have been deleted. There are no explicit commands to create or delete declustered arrays.

The main purpose of a declustered array is to segregate pdisks of similar performance characteristics and similar use. Because VDisks are contained within a single declustered array, mixing pdisks of varying performance within a declustered array would not use the disks optimally. In a typical IBM Spectrum Scale RAID system, the first declustered array contains SSD pdisks that are used for the log VDisk, or the log backup VDisk if configured. If the system is configured to use a log tip VDisk, another declustered array contains NVRAM pdisks for that VDisk. VDisks that are Spectrum Scale NSDs are then contained in one or more declustered arrays that use high-capacity HDDs or SSDs.

A secondary purpose of declustered arrays is to partition disks that share a common point of failure or unavailability, such as removable carriers that hold multiple disks. This issue comes into play when one considers that removing a multi-disk carrier to perform disk replacement also temporarily removes some good disks. The number of good disks that are removed might exceed the fault tolerance of the VDisk NSDs. This condition would cause temporary suspension of file system activity until the disks are restored. To avoid this condition, each disk position in a removable carrier should be used to define a separate declustered array. Disk position one defines DA1; disk position two defines DA2; and so on. Then, when a disk carrier is removed, each declustered array suffers the loss of just one disk. This loss is within the fault tolerance of any IBM Spectrum Scale RAID VDisk NSD.

Data spare space and VCD spares

When it runs with a failed pdisk in a declustered array, IBM Spectrum Scale RAID continues to serve file system I/O requests through two strategies:

- Using redundancy information about other pdisks to reconstruct data that cannot be read.
- Marking data that cannot be written to the failed pdisk as stale.

Meanwhile, to restore full redundancy and fault tolerance, the data on the failed pdisk is rebuilt onto data spare space. (Reserved unused portions of the declustered array that are declustered over all of the member pdisks). The failed disk is thereby drained of its data by copying it to the data spare space.

The amount of data spare space in a declustered array is set at creation time and can be changed later.

The data spare space is expressed in whole units equivalent to the capacity of a member pdisk of the declustered array. But the space is spread among all of the member pdisks. There are no dedicated spare pdisks.

This space setup has this implication:

1. A number of pdisks equal to the specified data spare space might fail.
2. The full redundancy of all of the data in the declustered array can be restored through a rebuild operation.

Users are not required to fill the space in the declustered array with VDisks. Instead, they can use the deallocated space as extra data spare space. This goal is accomplished by increasing the setting of the **dataSpares** parameter to the wanted level of resilience against pdisk failures.

At a minimum, each declustered array typically requires data spare space that is equivalent to the size of one member pdisk. The exceptions, which have zero data spares and zero VCD spares, are declustered arrays of these types:

- Non-volatile RAM disks that are used for a log tip VDisk.
- SSDs that are used for a log tip backup VDisk.

Because large declustered arrays have a greater probability of disk failure, the default amount of data spare space depends on the size of the declustered array. A declustered array with nine or fewer pdisks defaults to having one disk of equivalent data spare space. A declustered array with 10 or more disks defaults to having two disks of equivalent data spare space. These defaults can be overridden, especially when a declustered array is created. However, at a later point too much of the declustered array might already be allocated for use by VDisks. In this case, it might not be possible to increase the amount of data spare space.

In IBM Spectrum Scale RAID, VDisk configuration data (VCD) is stored more redundantly than VDisk content. Typically VCD is five-way replicated. When a pdisk fails, this configuration data is rebuilt at the highest priority, onto functioning pdisks. The redundancy of configuration data always must be maintained. IBM Spectrum Scale RAID does not serve a declustered array that lacks sufficient pdisks to store all configuration data at full redundancy. The declustered array parameter **vcdSpares** determines how many pdisks can fail and still have restoration of full VCD redundancy. The parameter reserves room on each pdisk for VDisk configuration data. When pdisk-group fault tolerance is used, the value of vcdSpares is set higher than the value of the **dataSpares** parameter. This configuration accounts for the expected failure of hardware failure domains.

Increasing VCD spares

When new recovery groups are created, the **mkrginput** script sets recommended values for VCD spares.

To increase the VCD spares for existing recovery groups, use the **mmchrecoverygroup** command.

Declustered array free space

The declustered array free space reported by the **mmlsrecoverygroup** command reflects the space available for creating VDisks. Spare space is not included in this value because it is not available for creating new VDisks.

pdisk free space

The pdisk free space reported by the **mmlsrecoverygroup** command reflects the actual number of unused data partitions on the disk. This includes spare space, so if a pdisk fails, these values decrease as data is moved to the spare space.

VDisks

VDisks are created across the pdisks within a declustered array. Each recovery group requires a special log home VDisk to function (along with other log-type VDisks, as appropriate for specific environments). See "Log VDisks" on page 20. All other VDisks are created for use as Spectrum Scale file system NSDs.

A recovery group can contain at most 64 VDisks. VDisks can be allocated arbitrarily among declustered arrays. VDisks are created with the **mmcrvdisk** command. The **mmdelvdisk** command destroys VDisks and all their contained data.

When you create a VDisk, you must specify the RAID code, block size, and VDisk size. You must also specify a name that is unique within the recovery group and the Spectrum Scale cluster. There are no adjustable parameters available for VDisks.

RAID code

Consider the type, performance, and space efficiency of the RAID codes that are used for VDisks. This information affects your choice of RAID code for a particular set of user data. See 1.2.1, "RAID codes" on page 2.

Spectrum Scale storage pools and policy-based data placement can be used to ensure that data is stored with appropriate RAID codes.

The VDisk block size must equal the Spectrum Scale file system block size of the storage pool where the VDisk is assigned. For replication codes, the supported VDisk block sizes are 256 KiB, 512 KiB, 1 MiB, and 2 MiB.

For Reed-Solomon codes, the supported VDisk block sizes are 512 KiB, 1 MiB, 2 MiB, 4 MiB, 8 MiB, and 16 MiB.

The *maxblocksize* configuration attribute of the IBM Spectrum Scale **mmchconfig** command must be set appropriately for all nodes. The value of *maxblocksize* must be greater than or equal to the maximum block size of the VDisks. For more information about this attribute, see the **mmchconfig** command description in *IBM Spectrum Scale: Administration and Programming Reference* at the following website:

http://ibm.co/20XE4oD

VDisk size

The maximum VDisk size is the total space available on the pdisks in the declustered array. This value takes into account the overhead of the RAID code, minus the following factors:

- spare space
- VDisk configuration data
- a small amount of space that is reserved as a buffer for write operations

IBM Spectrum Scale RAID rounds up the requested VDisk size as required. When you create a VDisk, you can specify to use all remaining space in the declustered array for that VDisk.

Log VDisks

IBM Spectrum Scale RAID uses log VDisks to quickly store internal information, such as event log entries, updates to VDisk configuration data, and certain data write operations. There are four types of log VDisks, as listed here. Among them, they can be created and destroyed in any order:

- Log home VDisk

  Every recovery group requires one log home VDisk to function. The log home VDisk must be created before any other non-log VDisks in the recovery group. It can be deleted only after all other non-log VDisks in the recovery group have been deleted. The log home VDisk is divided into four sublogs: Long-term event log, short-term event log, metadata log, and fast-write log, which logs small write operations.

- Log tip VDisk

  The log tip VDisk is appropriate for certain environments, but not required for all. It is a VDisk to which log records are initially written, then it is migrated to the log home VDisk. The intent is as follows:

  – Use a small, high-performance NVRAM device for the log tip.
  – Use a larger VDisk on conventional spinning disks for the log home VDisk.

  Fast writes to the log tip hide the latency of the spinning disks that are used for the main body of the log.

- Log tip backup VDisk

  The log tip backup VDisk is appropriate for certain environments, but not required for all. this log is used as an additional replica of the log tip VDisk when the log tip VDisk is two-way replicated on nonvolatile RAM disks. Ideally, the log tip backup VDisk provides a level of performance between that of NVRAM disks and that of spinning disks.

- Log reserved VDisk

  Log reserved VDisks are optional VDisks that are used when the log home disk is not allocated in its own declustered array. Log reserved VDisks have the same size as the log home VDisk and are used to equalize the space consumption on the data declustered arrays. Otherwise, they are unused.

Declustered array parameters

Declustered arrays have four parameters that can be set by using stanza parameters when you create a declustered array. The parameters can be changed with the **--declustered-array** option of the **mmchrecoverygroup** command. The following four parameters are available:

- dataSpares

  The number of disks worth of equivalent spare space that are used for rebuilding VDisk data if pdisks fail. This value defaults to one for arrays with nine or fewer pdisks, and two for arrays with 10 or more pdisks.

- vcdSpares

  The number of disks that can be unavailable while the IBM Spectrum Scale RAID server continues to function with full replication of VDisk configuration data (VCD). This value defaults to the number of data spares. To enable pdisk-group fault tolerance, this parameter is typically set to a larger value during initial system configuration. (For example, you give the parameter this value: (half of the number of pdisks in the declustered array + 1).

- replaceThreshold

  The number of disks that must fail before the declustered array is marked as having disks that must be replaced. The default is the number of data spares.

- scrubDuration

  The number of days over which all the VDisks in the declustered array is scrubbed for errors. The default is 14 days.

2.4.4 Typical configurations

The following list describes typical VDisk configurations in various recovery group environments:

- Elastic Storage Server without NVRAM disks

  In this configuration, a three-way replicated log tip VDisk is allocated on a declustered array that is made up of three SSDs. A four-way replicated log home VDisk is allocated in the first declustered array of HDDs.

- Elastic Storage Server with NVRAM disks

  In this configuration, a two-way replicated log tip VDisk is allocated on NVRAM disks, one from each of the servers.

  A log tip backup VDisk is allocated on a declustered array of one or more SSDs. This provides an additional copy of the log tip data, when one of the NVRAM disks is unavailable. If only one SSD is used, the log tip backup uses a RAID code of Unreplicated.

  A four-way replicated log home VDisk is allocated in the first declustered array of HDDs. A four-way replicated log reserved VDisk is allocated for each of the data declustered arrays that do not contain the log home VDisk.

- Elastic Storage Server with NVRAM disks, that use SSDs for data

  In this configuration, a two-way replicated log tip VDisk is allocated on NVRAM disks, one from each of the servers.

  All SSDs for a recovery group form a single declustered array, containing the log home VDisk and user data VDisks. No log tip backup disk is used.

- Power 775 configuration

  In this configuration, the log home VDisk is allocated on a declustered array that is made up of four SSDs. Only three-way and four-way replication codes are supported for the log home VDisk. Consider the typical system with four SSDs and with spare space equal to the size of one disk. In such a system, the three-way replication code would be used for the log home VDisk.

2.5 Recommended guidelines for Spectrum Scale RAID

In the Elastic Storage Server, the recommended guidelines for IBM Spectrum Scale RAID implementation are enforced as *de facto* standards. These standards allow modifications to configuration parameters for achieving the best storage performance. The Elastic Storage Server implementation relies on these factors:

- JBOD arrays
- the required redundancy protection and usable disk capacity
- the required spare capacity and maintenance strategy
- the ultimate Spectrum Scale file system configuration

The following IBM Spectrum Scale recommendations are fulfilled or suitable for the Elastic Storage Server as well:

- **A primary and backup server is assigned to each recovery group.**

  Each JBOD array is connected to two servers to prevent server failure. Each server has two independent paths to each physical disk to prevent path failure and provide higher throughput to the individual disks.

  Recovery group server nodes are designated Spectrum Scale manager nodes. They are dedicated to IBM Spectrum Scale RAID and not the run application workload.

- **Configure recovery group servers with a large VDisk track cache and a large page pool.**

  The **`nsdRAIDTracks`** configuration parameter tells IBM Spectrum Scale RAID how many VDisk track descriptors to cache in memory, not including the actual track data.

  In general, many VDisk track descriptors should be cached. The **`nsdRAIDTracks`** value for the recovery group servers should be 10000 - 60000. If the expected VDisk NSD access pattern is random across all defined VDisks and within individual VDisks, a larger value for **`nsdRAIDTracks`** might be warranted. If the expected access pattern is sequential, a smaller value might be sufficient.

  The amount of actual VDisk data (including user data, parity, and checksums) that can be cached depends on these factors:

  – the size of the Spectrum Scale page pool on the recovery group servers
  – the percentage of page pool that is reserved for IBM Spectrum Scale RAID

  The **`nsdRAIDBufferPoolSizePct`** parameter specifies what percentage of the page pool should be used for VDisk data. The default is 50%, but you can set it as high as 90% and as low as 10%. A recovery group server is also an NSD server and the VDisk buffer pool also acts as the NSD buffer pool. For this reason, the configuration parameter **`nsdBufSpace`** should be reduced to its minimum value of 10%.

  Consider this example:

  – You want a recovery group server that can cache 20000 VDisk track descriptors (**`nsdRAIDTracks`**).
  – The data size of each track is 4 MiB, using 80% (**`nsdRAIDBufferPoolSizePct`**) of the page pool,

  Thus, an approximate page pool size of 20000 * 4 MiB * (100/80) ~ 100000 MiB ~ 98 GiB would be required. It is not necessary to configure the page pool to cache all the data for every cached VDisk track descriptor. But this example calculation can guide you in determining appropriate values for **`nsdRAIDTracks`** and **`nsdRAIDBufferPoolSizePct`**.

- **Define each recovery group with at least one large declustered array.**

  A large declustered array contains enough pdisks to store the required redundancy of IBM Spectrum Scale RAID VDisk configuration data. This is defined as at least nine pdisks plus the effective spare capacity. A minimum spare capacity equivalent to two pdisks is strongly recommended in each large declustered array. You must also consider the code width of the VDisks. Here are some guidelines:

  – The effective number of non-spare pdisks must be at least as great as the largest VDisk code width.
  – A declustered array with two effective spares where 11 is the largest code width (8 + 3p Reed-Solomon VDisks) must contain at least 13 pdisks.
  – A declustered array with two effective spares in which 10 is the largest code width (8 + 2p Reed-Solomon VDisks) must contain at least 12 pdisks.

- **Define the log VDisks based on the type of configuration.**

  See 2.4.4, “Typical configurations” on page 21 and “Log VDisks” on page 20.

- **Determine the declustered array maintenance strategy.**

  Disks fail and need replacement, so a general strategy of deferred maintenance can be used. For example, failed pdisks in a declustered array are only replaced when the spare capacity of the declustered array is exhausted. To implement this, set the replacement threshold for the declustered array equal to the effective spare capacity. This strategy is useful in installations that have many recovery groups, where disk replacement might be scheduled weekly. Smaller installations might choose to configure IBM Spectrum Scale RAID to require disk replacement as disks fail, which means that the declustered array replacement threshold is set to 1.

- **Choose the VDisk RAID codes based on the Spectrum Scale file system usage.**

  The choice of VDisk RAID codes depends on these factors:

  - The level of redundancy protection that is required versus the amount of actual space that is required for user data
  - The ultimate intended use of the VDisk NSDs in a Spectrum Scale file system.

  Reed-Solomon VDisks are more space efficient. An 8 + 3p VDisk uses approximately 27% of actual disk space for redundancy protection and 73% for user data. An 8 + 2p VDisk uses 20% for redundancy and 80% for user data. Reed-Solomon VDisks perform best when the system is writing whole tracks (the Spectrum Scale block size) at one time. When partial tracks of a Reed-Solomon VDisk are written, parity recalculation must occur.

  Replicated VDisks are less space efficient. A VDisk with 3-way replication uses approximately 67% of actual disk space for redundancy protection and 33% for user data. A VDisk with 4-way replication uses 75% of actual disk space for redundancy and 25% for user data. The advantage of VDisks with N-way replication is that small or partial write operations are completed faster.

  For file system applications where write performance must be optimized, the preceding considerations affect your choice of file system:

  - Replicated VDisks are most suitable for use as Spectrum Scale file system metadataOnly NSDs.
  - Reed-Solomon VDisks are most suitable for use as Spectrum Scale file system dataOnly NSDs.

  The volume of the Spectrum Scale file system metadata is usually small (1 - 3%) relative to file system data. So the impact of the space inefficiency of a replicated RAID code is minimized. The file system metadata is typically written in small chunks, which takes advantage of the faster small and partial write operations of the replicated RAID code. Applications are often tuned to write file system user data in whole multiples of the file system block size. This approach works to the strengths of the Reed-Solomon RAID codes in space efficiency and speed.

  When segregating VDisk NSDs for file system metadataOnly and dataOnly disk usage, consider these guidelines:

  - You can create metadataOnly replicated VDisks with a smaller block size and assign them to the Spectrum Scale file system storage pool.
  - You can create the dataOnly Reed-Solomon VDisks with a larger block size and assign them to the Spectrum Scale file system data storage pools.
  - When you use multiple storage pools, you must install a Spectrum Scale placement policy to direct file system data to non-system storage pools.

  When write performance optimization is not important, it is acceptable to use Reed-Solomon VDisks as dataAndMetadata NSDs for better space efficiency.

2.6 Elastic Storage Server file system configurations

This section describes the required steps for IBM Spectrum Scale file system creation through use of the Elastic Storage Server command-line interface (CLI) commands. The descriptions take into account the Elastic Storage Server that is installed. All verification steps have been performed according to the Elastic Storage Server documentation.

Hardware verification steps assume that the software is installed in the I/O server nodes.

The system verification steps consist of information validation reported with the following commands:

1. **gssstoragequickcheck** checks the server, adapter, and storage configuration quickly.
2. **gssfindmissingdisks** checks the disk paths and connectivity.
3. **gsscheckdisks** checks for disk errors under various I/O operations.

The file system configuration is performed from the management server node as follows:

- **Run the gssgencluster command from the management server node.**

  Run the **gssgencluster** command on the management server to create the cluster. If you specify the **-G** option, this command creates a Spectrum Scale cluster and uses all of the nodes in the node group. You can also provide a list of names by using the **-N** option. The command assigns server licenses to each I/O server node, so it prompts for license acceptance (or use the **-accept-license** option). It applies the best-practice Spectrum Scale configuration parameters for a Spectrum Scale RAID-based NSD server. At the end of cluster creation, the SAS adapter firmware, storage enclosure firmware, and drive firmware are upgraded, if needed. To bypass the firmware update, specify the **--no-fw-update** option.

  > **Note:** This command might take some time to run.

- **Verify whether the cluster is active.**

  Log in to one of the I/O server nodes and verify that the cluster is created correctly. Run the **mmlscluster** command.

- **Create recovery groups.**

  The **gssgenclusterrgs** command creates the recovery groups (RGs) and the associated log tip VDisk, log backup VDisk, and log home VDisk. This command can create NSDs and file systems for simple configurations that require one file system. For more flexibility follow these guidelines:

  – Use the **gssgencluster** command to create only the recovery groups.
  – Use the **gssgenvdisks** command (the preferred method) to create data VDisks, metadata VDisks, NSDs, and file systems.

  For compatibility with an earlier version, the **gssgenclusterrgs** command continues to support VDisk, NSD, and file system creation.

  gssgenclusterrgs creates and saves the stanza files for the data and metadata VDisks and NSD. The stanza files are in the `/tmp` directory of the first node of the first building block with the following names:

  – node1_node2_vdisk.cfg.save
  – node1_node2_nsd.cfg.save

  These files can be edited for further customization.

If a customized recovery stanza file is available, it can be used to create the recovery group. The files must be on the first node (in the node list) of each building block in `/tmp`. Their names must be in the format xxxxL.stanza and yyyyR.stanza, where L is for the left recovery group and R is for the right recovery group. The name of the recovery group is derived from the I/O server node's short name (with prefix and suffix) by adding a prefix of rg_. When the **`--create-nsds`** option is specified, 1% of the space is left as reserved, by default. The remaining space is used to create the NSDs. You can select the amount of reserved space. The default is 1% of the total raw space. The percentage of reserved space is based on the total raw space (not on the available space) before any redundancy overhead is applied.

If the system already contains recovery groups and log VDisks (created in the previous steps), you use the appropriate options to skip their creation. This can be useful when NSDs are re-created (for a change in the number of NSDs or block size, for example).

> **Notes:**
>
> - This command might take some time to complete.
> - NSDs in a building block are assigned to the same failure group by default. If you have multiple building blocks, the NSDs defined in each building block have a different failure group for each building block. Carefully consider this information and change the failure group assignment when you are configuring the system for metadata and data replication.

- **Verify the recovery group configuration.**

  To view the details for one of the recovery groups, log on to one of the I/O server nodes and run the **`mmlsrecoverygroup`** command.

  Running **`mmlsrecoverygroup`** with no parameters lists all of the recovery groups in your Spectrum Scale cluster.

  For each recovery group:

  - NVR contains the NVRAM devices to use for the log tip VDisk.
  - SSD contains the SSD devices to use for the log backup VDisk.
  - DA1 contains the SSD or HDD devices to use for the log home VDisk and file system data.
  - DAn, where n > 1, depending on the Elastic Storage Server model and with v3.5 by default allowing only one DA. This parameter contains the SSD or HDD devices to use for file system data.

  > **Note:** In an SSD-only configuration, there is no SSD declustered array (DA) for log backup because DA1 to DAn are composed of SSD devices.

- **Create VDisk stanza.**

  Use the **`gssgenvdisks`** command to create the VDisk stanza file. By default, the VDisk stanza is stored in `/tmp/vdisk1.cfg`. Optionally, you can use the **`gssgenvdisks`** command to create VDisks, NSDs, and the file system on existing recovery groups. If no recovery groups are specified, all available recovery groups are used. You might run the command on the management server node (or any other node) that is not part of the cluster. In this case, a contact node that is part of the cluster must be specified. The contact node must be reachable from the node (the management server node, for example) where the command is run.

You can use this command to add a suffix to VDisk names, which can be useful when you create multiple file systems. A unique suffix can be used with a VDisk name to associate it with a different file system (examples to follow). The default reserve capacity is set to 1%. If the VDisk data block size is less than 8 M, the reserved space should be increased by decreasing the data VDisk block size.

This command can be used to create a shared-root file system for IBM Spectrum Scale protocol nodes.

> **Note:** NSDs that are in the same building block are given the same failure group by default. If file system replication is set to 2 (m=2 or r=2), there should be more than one building block. Otherwise, the failure group of the NSDs must be adjusted accordingly.

- **Reserved space configurations.**

  When all available space is allocated, you must increase the reserved space by decreasing data VDisk block size. A default reserved space of 1% works well for a block size of up to 4 MB. For a 2 MB block size, 2% should be reserved. For a 1 MB block size, reserved space should be increased to 3%.

- **Creating file systems by using the `gssgenvdisks` command.**

  Create two file systems, one with 20 TB (two VDisks, 10 TB each), and the other with 40 TB (two VDisks, 20 TB each) with a RAID code of 8+3p as shown in Example 2-1.

Example 2-1 Creating two file systems RAID code of 8+3p

```
[root@ems1 ~]# gssgenvdisks --contact-node gssio1 --create-vdisk --create-nsds
--create-filesystem
--vdisk-suffix=_fs1 --filesystem-name fs1 --data-vdisk-size 10240
2015-06-16T00:50:37.254906 Start creating vdisk stanza
vdisk stanza saved in gssio1:/tmp/vdisk1.cfg
2015-06-16T00:50:51.809024 Generating vdisks for nsd creation
2015-06-16T00:51:27.409034 Creating nsds
2015-06-16T00:51:35.266776 Creating filesystem
Filesystem successfully created. Verify failure group of nsds and change as
needed.
2015-06-16T00:51:46.688937 Applying data placement policy
2015-06-16T00:51:51.637243 Task complete.
Filesystem Size Used Avail Use% Mounted on
/dev/sda3 246G 2.9G 244G 2% /
devtmpfs 60G 0 60G 0% /dev
tmpfs 60G 0 60G 0% /dev/shm
tmpfs 60G 43M 60G 1% /run
tmpfs 60G 0 60G 0% /sys/fs/cgroup
/dev/sda2 497M 161M 336M 33% /boot
/dev/fs1 21T 160M 21T 1% /gpfs/fs1

[root@ems1 ~]# gssgenvdisks --contact-node gssio1 --create-vdisk --create-nsds
--create-filesystem --vdisk-suffix=_fs2 --filesystem-name fs2 --data-vdisk-size
20480 --raid-code 8+3p
2015-06-16T01:06:59.929580 Start creating vdisk stanza
vdisk stanza saved in gssio1:/tmp/vdisk1.cfg
2015-06-16T01:07:13.019100 Generating vdisks for nsd creation
2015-06-16T01:07:56.688530 Creating nsds
2015-06-16T01:08:04.516814 Creating filesystem
Filesystem successfully created. Verify failure group of nsds and change as
needed.
```

```
2015-06-16T01:08:16.613198 Applying data placement policy
2015-06-16T01:08:21.637298 Task complete.
Filesystem Size Used Avail Use% Mounted on
/dev/sda3 246G 2.9G 244G 2% /
devtmpfs 60G 0 60G 0% /dev
tmpfs 60G 0 60G 0% /dev/shm
tmpfs 60G 43M 60G 1% /run
tmpfs 60G 0 60G 0% /sys/fs/cgroup
/dev/sda2 497M 161M 336M 33% /boot
/dev/fs1 21T 160M 21T 1% /gpfs/fs1
/dev/fs2 41T 160M 41T 1% /gpfs/fs2
```

To display the VDisk information, run the **mmlsvdisk** command as shown in Example 2-2.

Example 2-2 VDisks configuration

```
[root@gssio1 ~]# mmlsvdisk
declustered block size
vdisk name RAID code recovery group array in KiB remarks
------------------ --------------- ------------------ ----------- ----------
rg_gssio1_hs_Data_8M_2p_1_fs1 8+2p rg_gssio1-hs DA1 8192
rg_gssio1_hs_Data_8M_3p_1_fs2 8+3p rg_gssio1-hs DA1 8192
rg_gssio1_hs_MetaData_8M_2p_1_fs1 3WayReplication rg_gssio1-hs DA1 1024
rg_gssio1_hs_MetaData_8M_3p_1_fs2 4WayReplication rg_gssio1-hs DA1 1024
rg_gssio1_hs_loghome 4WayReplication rg_gssio1-hs DA1 2048 log
rg_gssio1_hs_logtip 2WayReplication rg_gssio1-hs NVR 2048 logTip
rg_gssio1_hs_logtipbackup Unreplicated rg_gssio1-hs SSD 2048 logTipBackup
rg_gssio2_hs_Data_8M_2p_1_fs1 8+2p rg_gssio2-hs DA1 8192
rg_gssio2_hs_Data_8M_3p_1_fs2 8+3p rg_gssio2-hs DA1 8192
rg_gssio2_hs_MetaData_8M_2p_1_fs1 3WayReplication rg_gssio2-hs DA1 1024
rg_gssio2_hs_MetaData_8M_3p_1_fs2 4WayReplication rg_gssio2-hs DA1 1024
rg_gssio2_hs_loghome 4WayReplication rg_gssio2-hs DA1 2048 log
rg_gssio2_hs_logtip 2WayReplication rg_gssio2-hs NVR 2048 logTip
rg_gssio2_hs_logtipbackup Unreplicated rg_gssio2-hs SSD 2048 logTipBackup
```

- **Check the file system configuration.**

  Use the **mmlsfs** command to check the file system configuration. This command is run on one of the cluster nodes. If the management server node is not part of the cluster, use Secure Shell (SSH) to connect to one of the cluster nodes. Run the **mmlsfs all** command.

- **Mounting the file system.**

  Mount the file system by using the **mmmount** command (where `device` is the name of the file system): **mmmount device -a**.

- **Testing the file system.**

  Use the **gpfsperf** script to run some basic I/O tests on the file system to measure the performance of the file system by using a variety of I/O patterns. The **gpfsperf** script is included with Spectrum Scale. To run a basic I/O test by first sequentially creating a file, run this command:

  ```
  /usr/lpp/mmfs/samples/perf/gpfsperf create seq /gpfs/gpfs0/testfile1 -n 200G -r
  16M -th 4
  ```

- **Adding nodes to the cluster.**

  Use the **gssaddnode** command to add the management server node and additional I/O server nodes to the Elastic Storage Server cluster. The management server node is updated with the required RPMs during deployment and is prepared to join the cluster if needed. You can update the management server node by running the commands as shown in Example 2-3.

Example 2-3 Adding nodes to the cluster commands

```
updatenode ManagementServerNodeName -P gss_updatenode
reboot
updatenode ManagementServerNodeName -P gss_ofed
reboot
```

You must deploy the I/O server nodes properly and configure the high-speed network. You must complete that activity before you use the **gssaddnode** command to add these nodes to the Elastic Storage Server cluster. The **gssaddnode** command performs these operations:

– adds the nodes to the cluster,
– runs the product license acceptance tool,
– configures the nodes (by using gssServerConfig.sh or gssClientConfig.sh), and
– updates the host adapter, enclosure, and drive firmware.

Do not use **gssaddnode** to add non-Elastic Storage Server (I/O server or management server) nodes to the cluster. Use **mmaddnode** instead.

On the **gssaddnode** command, the **-N ADD-NODE-LIST** option specifies the list of nodes to add. For the management server node, it is that node’s host name. The **--nodetype** option specifies the type of node that is being added. For the management server node, the value is ems. This command must run on the management server node when that node is being added. This command can be also used to add I/O server nodes to an existing cluster.

2.7 Elastic Storage Server ordering

Whether you are evaluating or ordering the Elastic Storage Server solution, your action prompts IBM to do a technical delivery assessment (TDA). The assessment ensures that the proposed solution meets the customer requirements based on evaluation of the customer’s current IT environment. The assessment also ensures that the solution meets IBM specifications and results in the best possible implementation.

Depending on the model that you require, you can order the Elastic Storage Server in a minimum configuration. A minimum configuration can include elastic storage nodes, EMS, drawers, and disks on the condition that the HMC and network devices exist in the current IT environment. This scenario is recommended when another Elastic Storage Server exists in the current IT environment.

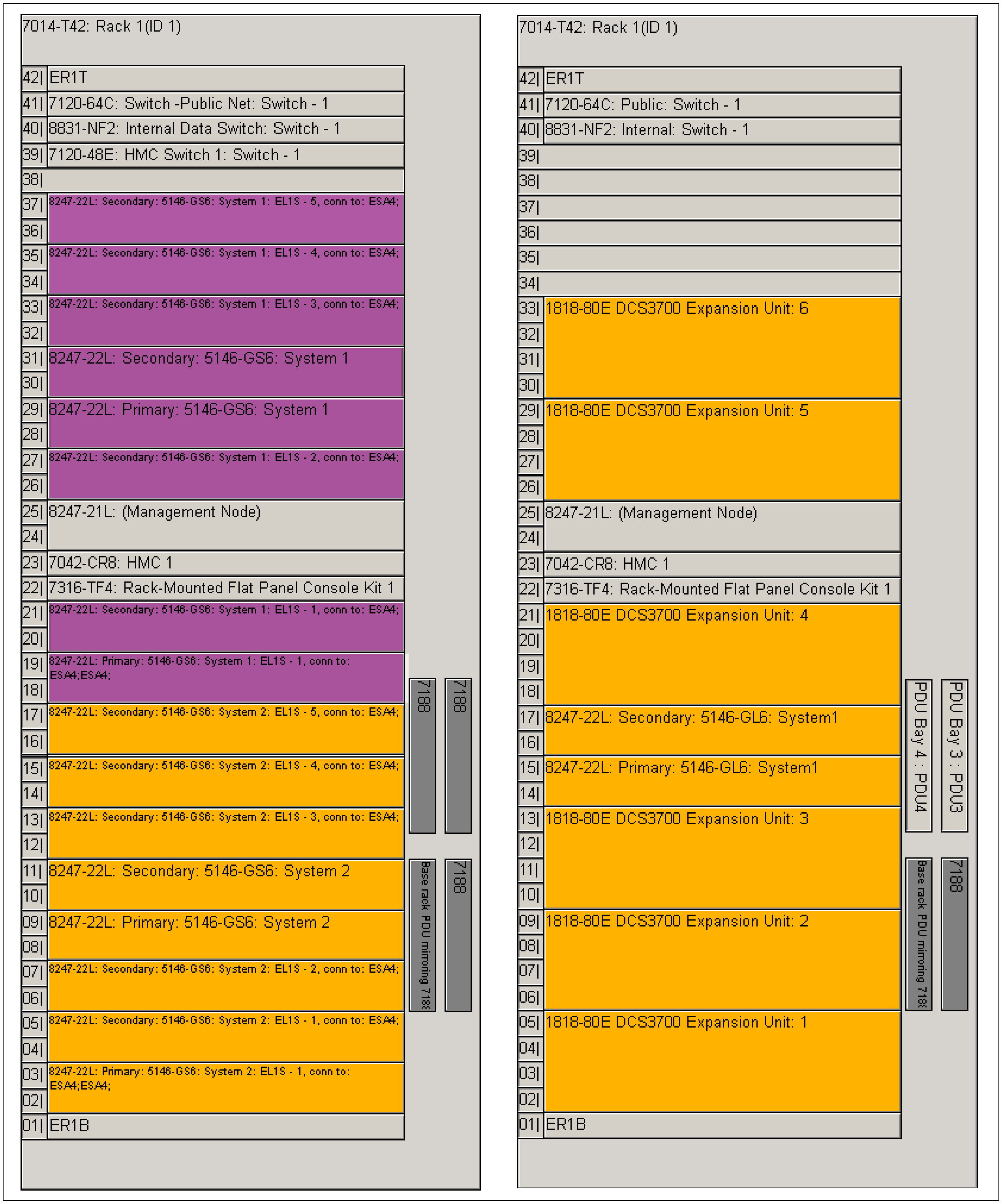
The recommended setup is as follows:

- A dedicated HMC both for the Elastic Storage Server storage nodes and platform console.
- A dedicated switch for private HMC network communication.

Usually, the Elastic Storage Server systems are delivered in a T42 rack, with these components:

- an HMC for the first Elastic Storage Server system
- a rack-mounted Flat Panel console kit
- network switches for public and internal communication, either for HMC network and internal Elastic Storage Server system network communication. The networks might also be provided by the customer as dedicated switches or configured VLANs.

Figure 2-1 on page 29 shows two Elastic Storage Servers GS6 occupying 32U into one rack IBM 7014-T42 with network components and also an Elastic Storage Server GL6.



7014-T42: Rack 1(ID 1)

42|	ER1T
41|	7120-64C: Switch -Public Net: Switch - 1
40|	8831-NF2: Internal Data Switch: Switch - 1
39|	7120-48E: HMC Switch 1: Switch - 1
38|	
37|	8247-22L: Secondary: 5146-GS6: System 1: EL1S - 5, conn to: ESA4;
36|	
35|	8247-22L: Secondary: 5146-GS6: System 1: EL1S - 4, conn to: ESA4;
34|	
33|	8247-22L: Secondary: 5146-GS6: System 1: EL1S - 3, conn to: ESA4;
32|	
31|	8247-22L: Secondary: 5146-GS6: System 1
30|	
29|	8247-22L: Primary: 5146-GS6: System 1
28|	
27|	8247-22L: Secondary: 5146-GS6: System 1: EL1S - 2, conn to: ESA4;
26|	
25|	8247-21L: (Management Node)
24|	
23|	7042-CR8: HMC 1
22|	7316-TF4: Rack-Mounted Flat Panel Console Kit 1
21|	8247-22L: Secondary: 5146-GS6: System 1: EL1S - 1, conn to: ESA4;
20|	
19|	8247-22L: Primary: 5146-GS6: System 1: EL1S - 1, conn to: ESA4;ESA4;
18|	
17|	8247-22L: Secondary: 5146-GS6: System 2: EL1S - 5, conn to: ESA4;
16|	
15|	8247-22L: Secondary: 5146-GS6: System 2: EL1S - 4, conn to: ESA4;
14|	
13|	8247-22L: Secondary: 5146-GS6: System 2: EL1S - 3, conn to: ESA4;
12|	
11|	8247-22L: Secondary: 5146-GS6: System 2
10|	
09|	8247-22L: Primary: 5146-GS6: System 2
08|	
07|	8247-22L: Secondary: 5146-GS6: System 2: EL1S - 2, conn to: ESA4;
06|	
05|	8247-22L: Secondary: 5146-GS6: System 2: EL1S - 1, conn to: ESA4;
04|	
03|	8247-22L: Primary: 5146-GS6: System 2: EL1S - 1, conn to: ESA4;ESA4;
02|	
01|	ER1B

7014-T42: Rack 1(ID 1)

42|	ER1T
41|	7120-64C: Public: Switch - 1
40|	8831-NF2: Internal: Switch - 1
39|	
38|	
37|	
36|	
35|	
34|	
33|	1818-80E DCS3700 Expansion Unit: 6
32|	
31|	
30|	
29|	1818-80E DCS3700 Expansion Unit: 5
28|	
27|	
26|	
25|	8247-21L: (Management Node)
24|	
23|	7042-CR8: HMC 1
22|	7316-TF4: Rack-Mounted Flat Panel Console Kit 1
21|	1818-80E DCS3700 Expansion Unit: 4
20|	
19|	
18|	
17|	8247-22L: Secondary: 5146-GL6: System1
16|	
15|	8247-22L: Primary: 5146-GL6: System1
14|	
13|	1818-80E DCS3700 Expansion Unit: 3
12|	
11|	
10|	
09|	1818-80E DCS3700 Expansion Unit: 2
08|	
07|	
06|	
05|	1818-80E DCS3700 Expansion Unit: 1
04|	
03|	
02|	
01|	ER1B

Figure 2-1 2 x Elastic Storage Servers GS6 and an Elastic Storage Server GL6 rack

2.8 Elastic Storage Server client integration

The Elastic Storage Server server access methods are similar to the ones for accessing an IBM Spectrum Scale cluster. Depending on the required configuration, the Elastic Storage Server can be accessed as an IBM Spectrum Scale Client, with a specific connector, or through dedicated protocol node servers.

When cNFS, Protocol Node services, the AFM gateway nodes, or the ISKLM key server nodes are required these conditions apply:

- One or more other servers should be configured for providing these services.
- These servers must run IBM Spectrum Scale at the necessary level for support.
- These servers must not run on the Elastic Storage Server server nodes.

2.9 Monitoring

This section presents some of the monitoring features based on the Elastic Storage Server GUI and command-line commands.

The Elastic Storage Server GUI is installed automatically during the Elastic Storage Server deployment process. The Elastic Storage Server GUI provides several monitoring functions.

The monitoring → System view

In this view, you see the status of all servers and enclosures and the drives in an enclosure. The servers and enclosures are displayed by using images that represent the way they are physically placed within racks. Click a server or enclosure to view the details of that particular hardware component.

From this view, you run a procedure to replace one or more failed drives within an enclosure.

The monitoring → System details view

This view displays hardware-related information for all servers and enclosures and the drives in an enclosure. In addition, this view displays many more hardware-related details, such as the status of subhardware components, for example, processors and fans. This view also provides a text search and filtering for unhealthy hardware.

The monitoring → Events view

This view displays events. From this view, you can monitor and troubleshoot errors in your system. The status icons can help you to quickly determine whether the event is informational, a warning, or an error.

To see detailed information about the event, click an event and select the Properties action. The event table displays the most recent events first. The status icons become gray if an error or warning has been fixed or if an informational message has been marked as read. The *Mark as read* action can be used on informational messages only.

The monitoring → Performance view

The Performance view shows the performance of file systems and block storage. The charting area allows these operations:

- panning (click and drag the chart or the timeline control at the bottom)
- zooming (by using the mouse wheel or resizing the timeline control).

Two charts can be displayed side by side to compare different objects and metrics or different time periods of the same chart. The timelines of the two charts can be linked together by using the **Link** button on the right side of the window. When the charts display two compatible metrics, you also can synchronize the Y-axis by using the **Settings** fly out at the bottom of the toolbar.

The monitoring → Capacity view

You can monitor the capacity of the file system, filesets, users, and user groups.

For each file system, the capacity is divided into the total disk capacity available, used disk capacity, and free disk capacity. You can view the capacity trend for a selected file system over a 30-day or 12-month time period. You can also view all file systems at the same time. Select one or more of the listed file systems to view them in a graph.

You can also display the data by percentage. That way, you see the capacity of filesets in a file system based on the nine largest filesets in the file system. This view shows the fileset capacity as reported by the Spectrum Scale quota capability.

You can also use the **`mmlsrecoverygroup`**, **`mmlspdisk`**, and **`mmpmon`** commands for monitoring. The IBM Spectrum Scale RAID event log is visible by using the **`mmlsrecoverygroupevents`** command as shown in Example 2-4.

Example 2-4 Monitoring commands

```
[root@c55f02n01 ~]# mmlsrecoverygroup

                     declustered
                     arrays with
 recovery group        vdisks     vdisks  servers
 ------------------  -----------  ------  -------
 rgL                           3       5  c55f02n01.gpfs.net,c55f02n02.gpfs.net
 rgR                           3       5  c55f02n02.gpfs.net,c55f02n01.gpfs.net

[root@c55f02n01 ~]# mmlsrecoverygroupevents
mmlsrecoverygroupevents: Missing arguments.
Usage:
   mmlsrecoverygroupevents RecoveryGroupName [-T] [--days days]
             [--long-term CEWID] [--short-term CEWID]
[root@c55f02n01 ~]# mmlsrecoverygroupevents rgL --days 1
Sat Oct  7 17:35:47.250 2015 c55f02n01 ST [I] Start scrubbing tracks of da_DA1_rgL.
Sat Oct  7 17:34:39.344 2015 c55f02n01 ST [I] End readmitting 1/3-degraded tracks of
da_DA1_rgL.
Sat Oct  7 17:34:39.309 2015 c55f02n01 ST [I] Start readmitting 1/3-degraded tracks of
da_DA1_rgL.
Sat Oct  7 17:34:39.308 2015 c55f02n01 ST [I] End readmitting 1/3-degraded tracks of
loghome_rgL.
Sat Oct  7 17:34:39.233 2015 c55f02n01 ST [I] Start readmitting 1/3-degraded tracks of
loghome_rgL.
Sat Oct  7 17:34:39.039 2015 c55f02n01 ST [I] Finished repairing RGD/VCD in RG rgL.
Sat Oct  7 17:34:38.067 2015 c55f02n01 ST [I] Start repairing RGD/VCD in RG rgL.
Sat Oct  7 17:34:38.067 2015 c55f02n01 ST [I] Finished repairing RGD/VCD in RG rgL.
.......................<<snippet>>.....................


[root@c55f02n01 ~]# mmlspdisk
mmlspdisk: Missing arguments.
Usage:
   mmlspdisk {all | RecoveryGroupName
```

```
      [--declustered-array DeclusteredArrayName | --pdisk PdiskName]}
      [--not-in-use | --not-ok | --replace]
[root@c55f02n01 ~]# mmlspdisk rgL --pdisk "n2s01"
pdisk:
   replacementPriority = 1000
   name = "n2s01"
   device = "//c55f02n01/dev/sda10"
   recoveryGroup = "rgL"
   declusteredArray = "NVR"
   state = "ok"
   capacity  = 2088763392
   freeSpace = 1904214016
   fru = ""
   location = ""
   WWN = ""
   server = "c55f02n01.gpfs.net"
   reads = 764
   writes = 114482633
   bytesReadInGiB = 1.381
   bytesWrittenInGiB = 678.420
   IOErrors = 18
   IOTimeouts = 1
   mediaErrors = 0
   checksumErrors = 0
   pathErrors = 0
   relativePerformance = 1.000
   dataBadness = 0.000
   rgIndex = 2
   userLocation = ""
   userCondition = "normal"
   VDiskhardware = "IBM IPR-10  68C08900  "
   hardwareType = NVRAM
```

3

Recommended practices for implementing Elastic Storage Server and IBM Spectrum Scale use cases

This chapter provides general guidelines for using Elastic Storage Server (ESS) and IBM Spectrum Scale.

This chapter includes the following sections:

- 3.1, “Introduction” on page 34
- 3.2, “Planning” on page 34
- 3.5, “Initial configuration” on page 37

3.1 Introduction

A good understanding of the use cases and planning ahead helps avoid problems along the way when you use any system. This concept applies to the use of ESS and Spectrum Scale. IBM provides a large amount of relevant documentation on the web. It is a good practice to spend enough time with that documentation to understand what the system is going to be used for.

There are many attributes of IBM Spectrum Scale file systems to consider when you develop a solution. Consider the issues for a typical genomics system:

- **Number of file systems:** Multiple file systems are typically required, which increases administrative overhead. But you need to make sure that you actually need more than one IBM Spectrum Scale file system.
- **Striping:** IBM Spectrum Scale stripes data across all available resources. When you have multiple file systems, you might want to isolate resources. However, for most workloads it is preferable for IBM Spectrum Scale to stripe across all available resources.

To balance these considerations, while still having a manageable number of file systems, Spectrum Scale provides the concept of *fileset* within a file system. This feature is explained in this chapter.

3.2 Planning

For guidelines on basic data center planning, see Planning for the system at IBM Knowledge Center.

When you install an ESS, the first place to look into is the Elastic Storage Server (ESS) product documentation at IBM Knowledge Center. The link not only provides up-to-date documentation, but also bulletins and alerts that are valuable when you are running ESS in your environment.

Another excellent resource is the IBM Spectrum Scale Frequently Asked Questions and Answers (FAQ). This FAQ is generic for Spectrum Scale, not specific to ESS. However, it is a valuable resource for the common questions about the product.

Spectrum Scale uses a file system object that is called a fileset. A fileset is a subtree of a file system namespace that in many respects behaves like an independent file system. Filesets provide a means of partitioning the file system to allow administrative operations at a finer granularity than the entire file system:

- Filesets can be used to define quotas on both data blocks and inodes.
- The owning fileset is an attribute of each file. This attribute can be specified in a policy to control initial data placement, migration, and replication of the file's data.
- Fileset snapshots can be created instead of creating a snapshot of an entire file system.

Spectrum Scale provides a powerful policy engine that can be used to process the entire file system in an efficient way. It can use any of the following attribute sets:

  – POSIX attributes (UID, GID, filename, atime, and so on)
  – Spectrum Scale attributes (fileset name, fileheat, and so on)
  – Any attribute that the file system admin wants to use with the use of extended attributes.

All these attributes give the storage administrator an extremely flexible policy engine that is built into the file system. Keep this capability in mind when you plan for file systems. For more information, see Information lifecycle management for IBM Spectrum Scale at IBM Knowledge Center.

3.3 Guidelines

ESS as a solution based on IBM Spectrum Scale has multiple ways to access and interact with the data. The same best practices for Spectrum Scale apply to ESS:

- Pre-planning. Involve IBM technical presales in the discussion.
- Do not mix different storages in the same storage pool.
- Understand the workload, including application logical block, patterns, and so on.
- Plan. When you purchase your first ESS, you receive up to 100 free hours of an onsite IBM Lab Services consultant for planning and installation.
- Unless you have hard evidence not to, use the defaults settings.
- Network is key. For more information, see 3.4, “Networking” on page 35.

3.4 Networking

Networking is one of the most important factors when you use Spectrum Scale in general. ESS as part of the Spectrum Scale family is no exception to this rule. You can never do enough planning in networking. Spectrum Scale runs over the network. So, the performance you can get from the network directly relates to the performance that you see from the file system.

On ESS, two network technologies are available to connect with the native Spectrum Scale protocol, called Network Shared Disk (NSD):

- Ethernet
- Remote Direct Memory Access (RDMA)

As a rule, regardless of the technology that is used, *do not share the daemon (data) network for other purposes*. This precaution is particularly important in Ethernet networks because the design of RDMA networks typically is flatter than Ethernet networks. RDMA networks have a reduced number of routers and switches because they typically connect devices to a single switch instead of separate switches.

IBM Spectrum Scale -- with ESS as part of the same family -- uses two networks: admin and daemon. They can be defined using the same network. However, in some cases it might be beneficial to split them, for example, to isolate the daemon network from other workloads.

3.4.1 Admin network

The Spectrum Scale admin network has these characteristics:

- Used for the execution of administrative commands.
- Requires TCP/IP.
- Can be the same network as the Spectrum Scale daemon network, or a different one.
- Establishes the reliability of Spectrum Scale.

3.4.2 Daemon network

The Spectrum Scale daemon network has these characteristics:

- Is used for communication between the mmfsd daemon of all nodes.
- Requires TCP/IP.
- In addition to TCP/IP, Spectrum Scale can be optionally configured to use RDMA for daemon communication. TCP/IP is still required if RDMA is enabled for daemon communication.
- Establishes the performance of Spectrum Scale, as determined by its bandwidth, latency, and reliability of the Spectrum Scale daemon network.

Both networks must be reliable to be able to have a reliable cluster. If TCP/IP packets are lost, there are significant variable latencies between the systems or other undesirable anomalies in the health of the network. The Spectrum Scale cluster and file systems suffer the consequences of these problems.

When you design the network, use a network that is as fast as the daemon network. If you are using RDMA, you can use IP over InfiniBand (IPoIB) for the IP part of the daemon network.

The admin network can use the same fast and reliable network as the daemon network. However, on ESS-only clusters, the system is delivered with at least one high-speed network and one management network as shown in the network diagram in Figure 3-1.

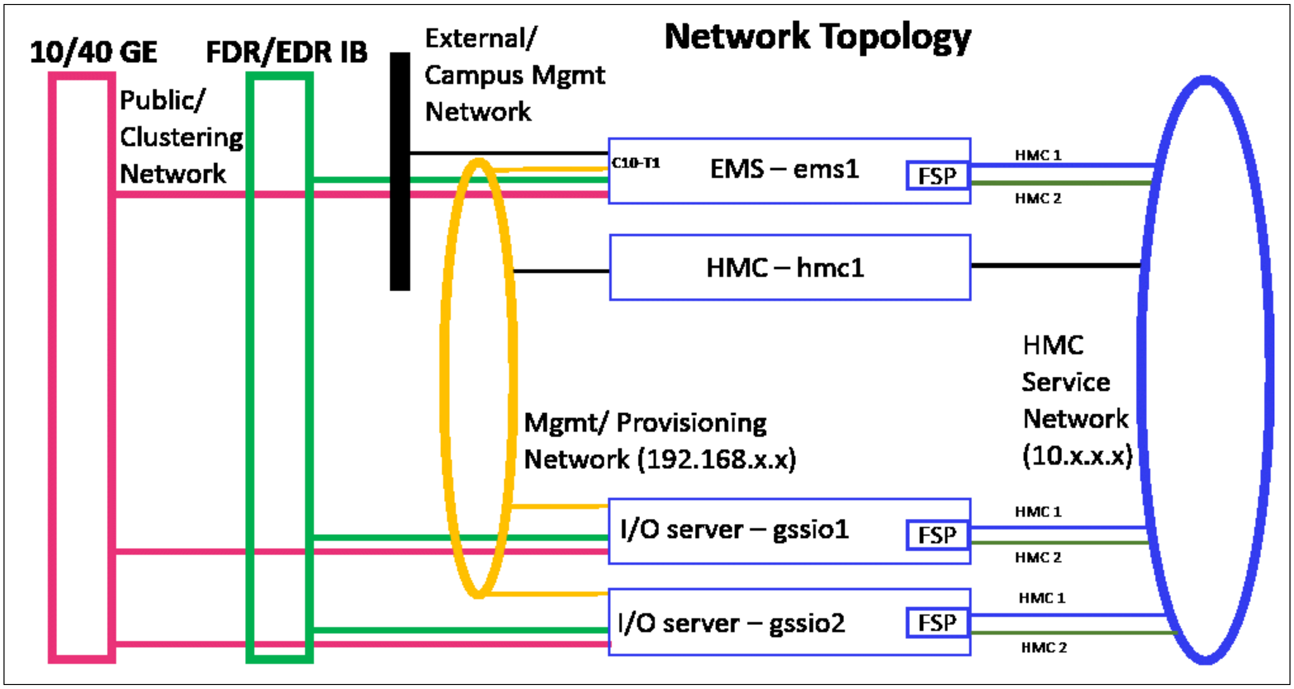


Figure 3-1 Second-generation ESS network diagram

The management network should be used in an ESS cluster as the admin network. The high-speed network, which is shown as 10/40 GE and FDE/EDR IB, should be used as the daemon network. For details about the ESS network requirements, see ESS networking considerations at IBM Knowledge Center.

In an existing cluster, you use the **mmlscluster** command to see what networks are used for admin and which for daemon. Example 3-1 shows the output of the **mmlslcuster** command when both networks are using different interfaces.

Example 3-1 mmlscluster with admin and daemon networks being the same

```
# mmlscluster

GPFS cluster information
========================
  GPFS cluster name: ITSO.stg.forum.fi.ibm.com
  GPFS cluster id:           8075176174583977671
  GPFS UID domain: ITSO.stg.forum.fi.ibm.com
  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   specscale01-hs     10.10.16.15  specscale01     quorum-manager-perfmon
   2   specscale02-hs     10.10.16.16  specscale02     quorum-manager-perfmon
   3   specscale03-hs     10.10.16.17  specscale03     quorum-manager-perfmon
```

Note: This example shows only the network output of **mmlscluster** and it is valid for all IBM Spectrum Scale products, including the ESS.

You might plan to extend the cluster with non-ESS parts. In that case, you must plan how the admin and daemon networks are going to interconnect in a reliable manner with enough performance to the ESS cluster nodes. That plan must include end-to-end design of those networks, switches, uplinks, networks, hops, routing if used, firewall configurations if used. Many other planning topics are beyond the scope of a storage product.

To identify possible issues on the network, create a baseline for the network after all planning and deployment is complete. IBM Spectrum Scale includes tools to create this baseline. In particular, use the **nsdperf** command. For more information about how to run the **nsdperf** command to create a baseline, see Testing network performance with nsdperf.

Every time that you change something on the network, check the results against your baseline. Tuning is an ongoing process that does not end on the day of delivery. Systems evolve, as do workloads. Therefore, it is important to regularly monitor the network for changes in performance and resilience. These factors have a direct and critical impact on any networked service, including IBM Spectrum Scale and ESS as part of the Spectrum Scale family.

3.5 Initial configuration

When you set up ESS for the first time in your environment, review the sample ESS client configuration script for your non-ESS nodes (including IO nodes and EMS).

3.5.1 Generic client script part of ESS

The script contains default Spectrum Scale configuration parameters for generic workloads. Review and adapt those parameters to fit your specific needs. For more information about Spectrum Scale parameters, see mmchconfig command at IBM Knowledge Center.

ESS includes the `gssClientConfig.sh` script, which is in the `/usr/lpp/mmfs/samples/gss/` directory. It can be used to add client nodes to the ESS cluster.

Note: When you use the `gssClientConfig.sh` script, do a dry run by using the `-D` parameter to review the settings without committing them.

For more information about the `gssClientConfig.sh` script, see Adding IBM Spectrum Scale nodes to an ESS cluster at IBM Knowledge Center.

3.5.2 Ongoing maintenance (upgrades)

A proper planning phase is critical for a successful upgrade. An ESS update/upgrade changes multiple layers in the system, including but not limited to the following items:

- Operating system (OS)
  - Kernel
  - Libraries
- Device Driver and firmware levels
  - Network stack
  - Storage subsystem
  - Node
  - Enclosure
  - Host bus adapter (HBA)
  - Drives (SSD, HDD, and so on)
- IBM Spectrum Scale and IBM Spectrum Scale RAID

Note: Before any system upgrade, back up all important data before you do the upgrade to avoid the possible loss of critical data.

You might make protocol nodes a part of the cluster. If so, plan for the IP addresses failovers that will occur and the consequences that these might have in the clients.

In addition to the components that the ESS upgrade involves, planning is needed for the components that ESS (and IBM Spectrum Scale) interacts with directly or indirectly. This list includes, but is not limited to, the following items:

- RDMA fabrics
- Applications - Middleware
- Users
- Other

This process might cause brief NFS, SMB, and Performance Monitoring outages.

ESS upgrade guidelines

Familiarize yourself with OS, device drivers and firmware levels, IBM Spectrum Scale, and IBM Spectrum Scale RAID levels. You find the supported levels in the release notes for IBM Elastic Storage Server (ESS) 5.3.1 under the “Package Information” section.

> **Note:** Be aware that the links and information referred to here are related to ESS version 5.3.1. Always check the latest information in IBM Knowledge Center, which is the official documentation.

When you use RDMA, check the MOFED levels that ESS supplies with the fabric levels and firmware that are currently in use. Make plans at the fabric level before you apply the ESS update/upgrade. For information about the supported levels of Mellanox, see the Mellanox support matrix.

For application levels, contact the software vendor for recommendations if applicable.

IBM Spectrum Scale matrix support

Confirm the IBM Spectrum Scale version that is included in each ESS version. The ESS versions are listed in Planning for the system. You should check that the versions on non-ESS nodes are compatible with Spectrum Scale version that is delivered with the ESS. The best place to look for the compatibility across versions is the IBM Spectrum Scale Frequently Asked Questions and Answers (FAQ).

For ESS-supported update/upgrade paths, see Supported upgrade paths at IBM Knowledge Center.

Preferred upgrade flow

For the ESS building blocks, see Upgrading Elastic Storage Server at IBM Knowledge Center.

The link gives you a starting plan for the upgrade process. It covers the ESS parts only. So, the plan is not complete until you account for the rest of the parts that interact with ESS, and add these to an overall plan.

Your cluster might have other nodes that are not part of the ESS, such as the following node types:

- Protocol Nodes
- Active File Management (AFM) gateways
- Transparent Cloud Tier (TCT)
- IBM Spectrum Protect™ nodes
- IBM Spectrum Archive™

See Upgrading a cluster containing ESS and protocol nodes at IBM Knowledge Center.

Final comments

This chapter is not intended to scare you about update or upgrades. Instead, the goal is to make you aware that proper planning, as with anything in technology, is important to reach success. This requirement is even more important as interactions between interconnected systems become more common and complex. Remember that when you do an ESS update or upgrade, you can always involve IBM to help you with it.

4

Multicluster integration with ESS

This chapter describes the implementation and recommended practices for multicluster integration with ESS and contains the following sections:

4.1 Introduction

Multicluster ESS setups allow connections to different clusters. You have these options:

- Allow clusters to access one or more file systems belonging to a different cluster.
- Mount file systems that belong to other clusters on condition that the correct authorization is set up.

A multicluster setup is useful for sharing data across clusters.

The use cases for sharing data across multiple ESS clusters include these scenarios:

- Separating the data ingest cluster from data analytics cluster.
- Collaboration of data across different departments.
- Isolating the storage cluster from the application clients.

It is possible to share data across multiple clusters within a physical location or across locations. Clusters are most often attached by using a LAN, but they might also include a SAN.

4.2 Planning

When you plan for a multicluster setup, it is important to consider the use case for setting up multiple clusters. In other words, consider the reason for sharing of data. This consideration helps you to determine the type of access that you need to set up across clusters for data sharing. Determine what the different clusters are used for, and why the clusters need to be separated. The separation of clusters might be ESS storage cluster, protocol cluster, or compute client cluster.

ESS allows users to have shared access to files in either the cluster where the file system was created or other ESS clusters. These configurations are described in the following figures. It is possible to export specific file systems with specific remote clusters, and also export them as readOnly and do root squashing.

Figure 4-1 on page 43 shows a set of application client cluster Nodes that are mounting file systems from remote ESS storage clusters.

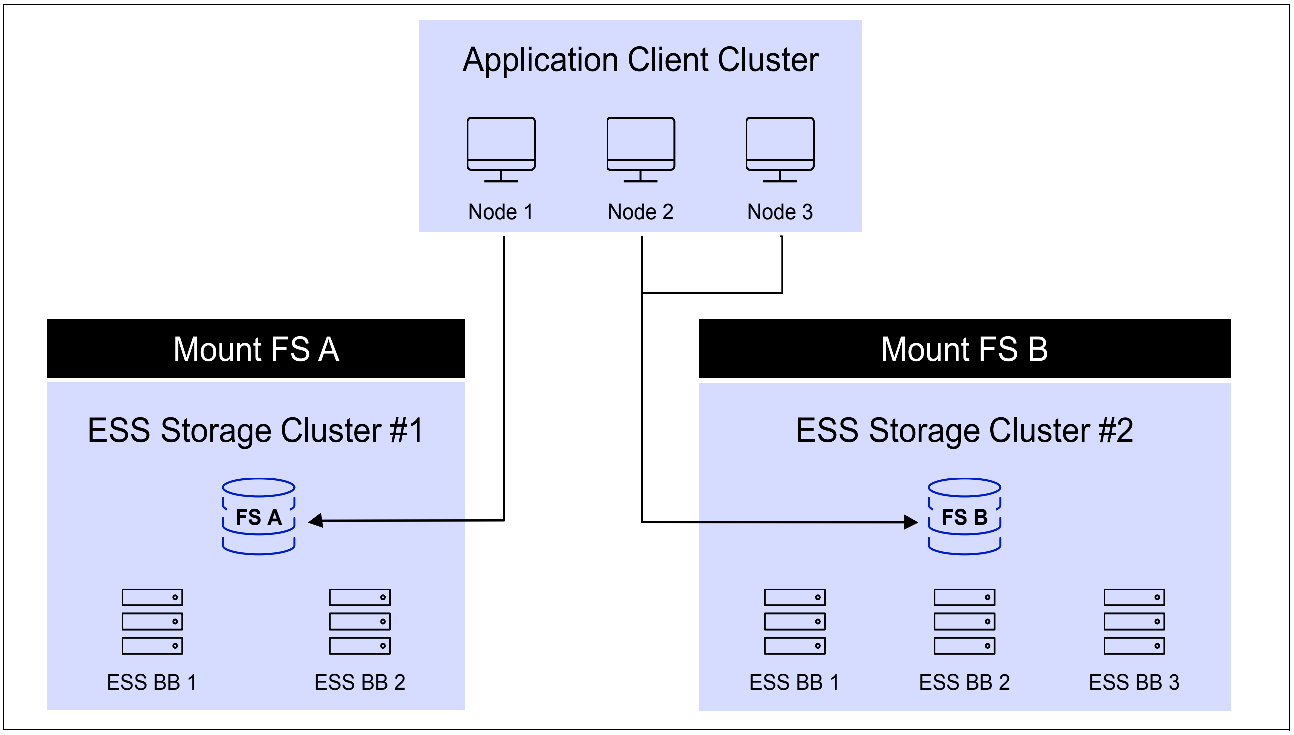


Figure 4-1 Application client cluster Nodes that are mounting file systems from remote ESS storage clusters

Figure 4-2 depicts the protocol cluster with an alternative authentication scheme.

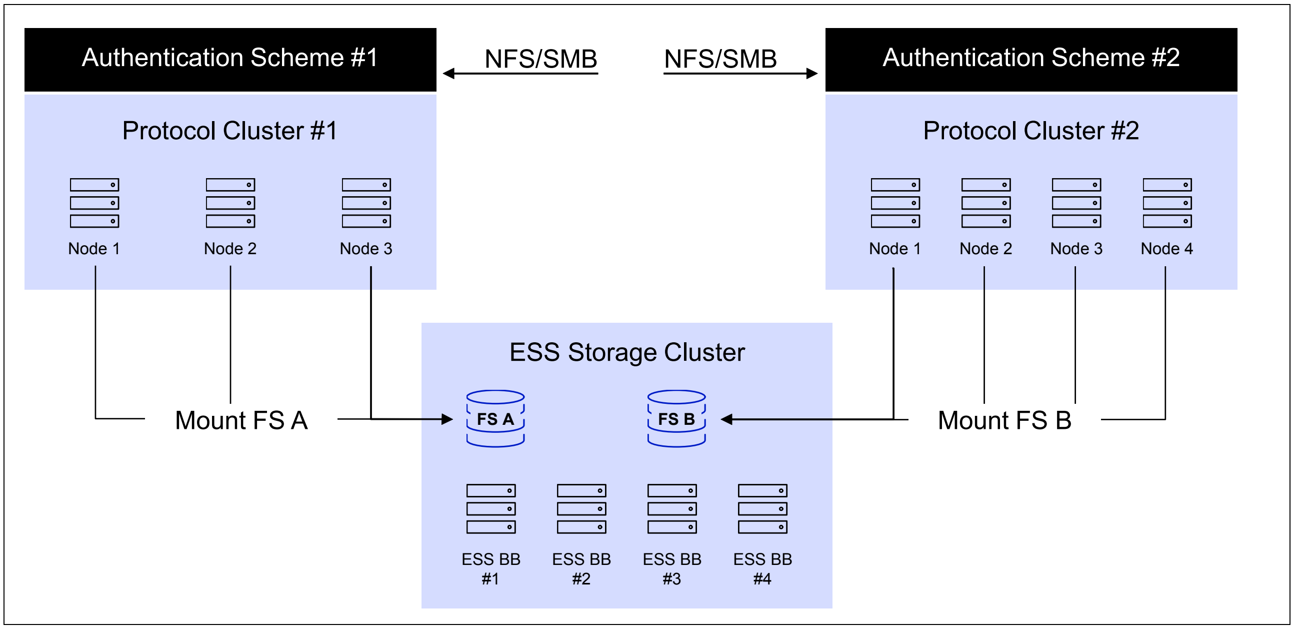


Figure 4-2 Protocol cluster with an alternative authentication scheme

Finally, Figure 4-3 on page 44 shows the separate protocol and client clusters that are mounting different file systems.

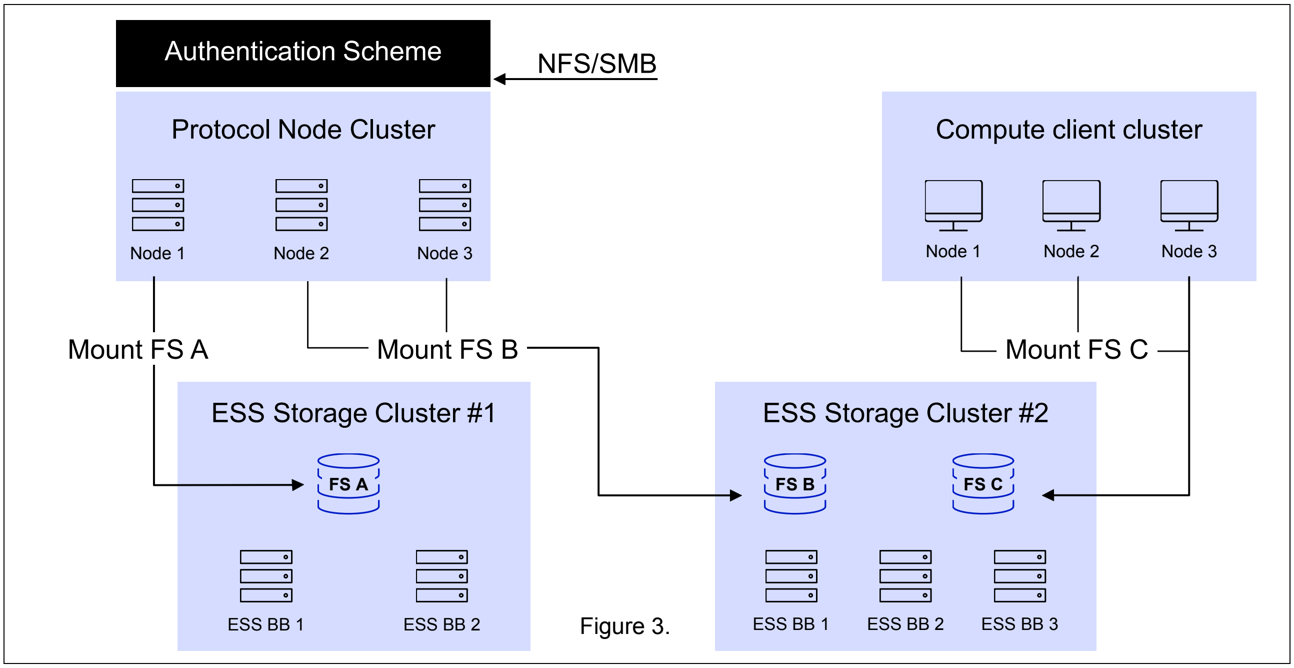


Figure 4-3 Separate protocol and client clusters that are mounting different file systems

While you plan for a multicluster installation, it is important to note the following points:

Each cluster is installed, managed, and upgraded independently. So, follow these guidelines:

- A file system is administered only by the cluster where the file system was created. Other clusters might be allowed to mount the file system. However, their administrators cannot add or delete disks, change characteristics of the file system, enable or disable quotas, run the **mmfsck** command, and so on. The only commands that other clusters can issue are list type commands, such as: **mmlsfs**, **mmlsdisk**, **mmlsmount**, and **mmdf**.
- Because each cluster is managed independently, there is no automatic coordination and propagation of changes between clusters. In contrast, automatic coordination happens between the nodes within a cluster. This implication follows:
  - If the administrator of cluster1 (the owner of file system gpfs1) decides to delete it or rename it, the information for gpfs1 in cluster2 becomes obsolete.
  - An attempt to mount gpfs1 from cluster2 fails.

  It is assumed that when such changes take place, the two administrators will inform each other. Then, the administrator of cluster2 can then use the update or delete options of the **mmremotefs** command to make the appropriate changes.
- Use the update option of the **mmremotecluster** command to reflect these changes to these values:
  - names of the contact nodes
  - name of the cluster
  - public key files
- Every node in a cluster needs to have a TCP/IP connection to every other node in the cluster. Similarly, every node in the cluster that requires access to another cluster's file system must be able to open a TCP/IP connection to every node in the other cluster.
- Each cluster requires its own GUI. The GUI might be installed onto the CES nodes, but performance must be considered.

- Each cluster has its own REST API.
- Each cluster has its own health monitoring. This means that error events that are raised in the one cluster are not visible in the other cluster and vice versa.
- Availability of certain performance metrics depends on the role of the cluster. That is, NFS metrics are available on protocol clusters only.

4.3 Guidelines

The following practices are recommended when you implement multicluster integration with ESS:

- Unless the cluster is small, it is recommended to separate the storage and the compute cluster.
- It is recommended that the storage cluster should include only 'storage-related nodes'. These types of nodes belong in their own cluster:
  - Protocol nodes (such as NFS, SMB, Object)
  - Backup nodes (such as Protect)
  - Archive nodes (IBM Linear Tape File System™ Enterprise Edition (LTFS-EE)
  - Active File Management (AFM) gateway nodes

  The storage cluster can be managed by the storage administrator.
- If a protocol cluster is installed into an environment with an existing storage cluster, the ESS version that is used should comply with the protocol cluster. The installation can be performed either manually or by using the installation toolkit.
- It is possible that different clusters are administered by different organizations. So, it is possible for each cluster to have a unique set of user accounts. For consistency of ownership and access control, a uniform user-identity namespace is preferred. Users must be known to the other cluster. You typically achieve this goal as follows:

  a. Create a user account in the other cluster.
  b. Give this account the same set of user and group IDs that the account has in the cluster where the file system was created.

  However, this approach might pose problems in some situations.

  GPFS helps to solve this problem by optionally performing user ID and group ID remapping internally, by using user-supplied helper applications. For a detailed description of the GPFS user ID remapping convention, see the IBM white paper entitled UID Mapping for GPFS in a Multi-cluster Environment in the IBM Knowledge Center.
- Access from a remote cluster by a root user presents a special case. It is often desirable to disallow root access from a remote cluster, while you allow regular user access. Such a restriction is commonly known as root squash. A root squash option is available when you make a file system available for mounting by other clusters by using the **`mmauth`** command. This option is similar to the NFS root squash option. When enabled, it causes GPFS to squash superuser authority on accesses to the affected file system on nodes that are located in remote clusters.
- The configuration limits need to be treated the same for nodes in all the clusters. For example, the max socket connections (`fsysctl net.core.somaxconn`) value should be the same.
- A cluster might own a file system has a *maxblocksize* configuration parameter that is different from the maxblocksize configuration parameter of the cluster that desires to mount a file system. In this case, a mismatch can occur and file system mount requests might fail with messages to this effect. Use the **`mmlsconfig`** command to check your

maxblocksize configuration parameters on both clusters. Correct any discrepancies with the **mmchconfig** command.

- Use the show option of the **mmremotecluster** and **mmremotefs** commands to display the current information about remote clusters and file systems.
- Cross-protocol change notifications do not work on remotely mounted file systems. For example, an NFS client might change a file. In this case, the system does not issue a *file change* notification to the SMB client that has asked for a notification.
- Each protocol cluster must use a dedicated file system. It is not allowed to share a file system between multiple protocol clusters.
- The storage cluster owns all of the exported ESS file systems. This ownership includes at least two file systems per protocol cluster (one CES shared root + one data file system).
- The protocol clusters cannot own any ESS file systems. Only remote mounts from the storage cluster are allowed.
- Any file system can be remotely mounted by exactly one protocol cluster. Sharing a file system between multiple protocol clusters might cause data inconsistencies.
- The primary use case for multi-cluster protocol is to allow multiple authentication configurations. Do not use the setup for these purposes:
  - extending the scalability of Cluster Export Services (CES)
  - working around defined limitations (for example, number of SMB connections)
- This setup provides some level of isolation between the clusters, but there is no strict isolation of administrative operations. Also, there is no guarantee that administrators on one cluster cannot see data from another cluster. Strict isolation is guaranteed through NFS or SMB access only.
- Storage and protocol clusters are in the same site/location. High network latencies between them can cause problems.

4.4 Networking

Each node in a cluster needs to have a TCP/IP connection to every other node in the cluster. Similarly, every node in the cluster that requires access to another cluster's file system must be able to open a TCP/IP connection to every node in the other cluster.

Nodes in two separate remote clusters that mount the same file system are not required to be able to open a TCP/IP connection to each other. Here is an example:

- A node in clusterA mounts a file system from clusterB.
- A node in clusterC desires to mount the same file system.
- Nodes in clusterA and clusterC do not have to communicate with each other.

4.4.1 Node roles

When a remote node mounts a local file system, it *joins* the local cluster. So, the cluster manager manages its leases, the token servers on the local cluster manages tokens, and so on. You must take that into account when you size and design the node designation.

You might have a protocol cluster that is separate from the ESS storage cluster. In this case, it is recommended that you have the quorum, cluster manager, and file system manager functions on the remote protocol cluster.

Be aware that the availability of certain performance metrics depends on the role of the cluster. For example, NFS metrics are available on protocol clusters only.

Due to the separation of duties (storage clusters own the file systems and protocol clusters own the NFS/SMB exports), certain management tasks must be done in the corresponding cluster:

- File system-related operations like creating file systems, filesets, or snapshots must be done in the storage cluster.
- Export-related operations like creating exports, managing CES IP addresses, and managing authentication must be done in the protocol cluster.

The resource cluster is unaware of the authentication setup and UID mapping. For this reason, all actions that require a user or a group name must be done in the corresponding protocol cluster. For example, you must generate quota reports and manage access control lists (ACLs) in that protocol cluster.

4.5 Initial setup

The procedure to set up remote file system access involves the generation and exchange of authorization keys between the two clusters. In addition, the administrator of the GPFS cluster that owns the file system must authorize the remote clusters that are to access it. In turn, the administrator of the GPFS cluster that seeks access to a remote file system must define to GPFS the remote cluster and file system whose access is desired.

The package gpfs.gskit must be installed on all the nodes of the owning cluster and the accessing cluster. For more information, see the installation chapter for your operating system, such as Installing IBM Spectrum Scale on Linux nodes and deploying protocols.

4.5.1 Setting up protocols over remote cluster mounts

Figure 4-3 on page 44 shows the separation of tasks that are performed by each cluster.

Storage cluster owns the file systems and the storage. Protocol clusters contain the protocol node that provides access to the remotely mounted file system, through NFS or SMB.

Here, the storage cluster owns a file system and the protocol cluster remotely mounts the file system. The protocol nodes (CES nodes) in the protocol cluster export the file system via SMB and NFS.

You can define one set of protocol nodes per cluster, by using multiple independent protocol clusters that remotely mount file systems. Protocol clusters can share access to a storage cluster but not to a file system. Each protocol cluster requires a dedicated file system. Each protocol cluster can have a different authentication configuration, thus allowing different authentication domains while you keep the data at a central location. Another benefit is the ability to access existing ESS-based file systems through NFS or SMB without adding nodes to the ESS cluster.

Configuring protocols on a separate cluster

The process for configuring protocols on a separate cluster is in many respects the same as for a single cluster. However, there are a few differences mainly in procedure order.

This procedure assumes an environment with the server, network, storage, and operating systems are installed and ready for ESS use. For more information, see Installing section of the *IBM Spectrum Scale documentation* at IBM Knowledge Center.

Perform the following steps:

1. Install ESS on all nodes that are in the storage and protocol clusters. If you install a protocol cluster into an environment with an existing ESS cluster, the ESS version that is used should comply with the protocol cluster. The installation can be performed either manually or by using the installation toolkit. Do not create clusters or file systems or Cluster Export Services yet.
2. Create the storage and protocol clusters. Proceed with cluster creation of the storage cluster and one or more protocol clusters. Ensure that the configuration parameter maxBlockSize is set to the same value on all clusters.
3. Create file systems on the storage cluster, taking the following points into consideration:
   - **CES shared root file system:** Each protocol cluster requires its own CES shared root file system. Having a shared root file system that is different from the file system that serves data eases the management of CES.
   - **Data file systems**: At least one file system is required for each protocol cluster that is configured for Cluster Export Services. A data file system can be exported only from a single protocol cluster.
4. Before you install and configure Cluster Export Services, consider the following points:
   - **Authentication:** Separate authentication schemes are supported for each CES cluster.
   - **ID mapping:** The ID mapping of users that are authenticating to each CES cluster. It is recommended to have unique ID mapping across clusters, but not mandatory.

You must judiciously determine the ID mapping requirements and prevent possible interference or security issues.

   - **GUI:** GUI support for remote clusters is limited. Each cluster should have its own GUI. The GUI may be installed onto CES nodes but performance must be considered.
   - **Object:** Object is not supported in multi-cluster configurations.

   For a list of limitations, see Limitations of protocols on remotely mounted file systems.

5.Configure clusters for remote mount. For more information, see Mounting a remote GPFS file system.

6.Install and configure Cluster Export Services by using the installation toolkit or manually. For more information, refer to the following links:

   - Installing IBM Spectrum Scale on Linux nodes with the installation toolkit
   - Manually installing the IBM Spectrum Scale software packages on Linux nodes

5. Use the remotely mounted CES-shared root file system. After SMB and/or NFS is enabled, new exports can be created on the remotely mounted data file system.

4.6 Ongoing maintenance (upgrades)

Before you schedule an upgrade of a cluster that contains ESS and protocol nodes, planning discussions must take place to determine the current cluster configuration and to understand which functions might face an outage.

https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.1/com.ibm.ess.v5r31.qdg.doc/bl8qdg_upgradeesscesmain.htm

Each cluster is installed, managed, and upgraded independently. There is no special process to upgrade clusters in a multi-cluster environment. Upgrades are performed on a cluster-boundary basis.

When you choose an IBM ESS version, the release should comply with release-level limitations.

After all clusters in the environment are upgraded, the release and the file system version should be changed. The release version might be changed concurrently. However, changing the file system version requires the file system to be unmounted. To view the differences between file system versions, see Listing file system attributes.

To change the IBM ESS release, issue the following command on each cluster:

```
mmchconfig release=LATEST
```

Note: Nodes that run an older version of ESS on the remote cluster can no longer mount the file system. A command fails if any nodes that are running an older version are mounted at time that the command is issued.

To change the file system version, issue the following command for each file system on the storage cluster:

```
mmchfs <fs> -V full
```

If your requirements call for it, issue the following command:

```
mmchfs <fs> -V compat
```

This enables only backward-compatible format changes.

5

Cluster Export Services (CES) integration with ESS

This chapter discusses the Cluster Export Services (CES) integration with ESS and contains the following sections:

5.1 Cluster Export Services (CES) and protocol nodes support in ESS

With the ESS 5.3.1.1 release, a protocol node feature code is introduced. This protocol node feature code allows the purchase of POWER8 nodes with a very specific hardware configuration -- tested and tuned by IBM -- for providing CES services.

> **Note:** In this document, we are going to talk about only the CES setup using the integrated protocol nodes that are ordered with an ESS solution. It is very likely that some of the recommendations might apply to protocol nodes that are ordered separately. However, that scenario is not in the scope of this document.

The protocol nodes that come with ESS are IBM Power 5148-22L with the following hardware characteristics:

- 2 x 10core 3.34 Ghz POWER8 processors
- 128 GB or greater memory
- Two 600 GB 10k RPM SAS HDDs in RAID10 mirror that uses the IPRaid adapter
- 1 GbE 4port network adapter in slot C12
- 1 Three x 16 or x8 network adapters in slots C5, C6, and C7
- 1 Four x 8 network adapters in slots C2, C3, C10, and C11, which are available by additional card orders

For more information about the hardware and software of the protocol nodes of the ESS solution, see the following web page regarding ESS support for CES and protocol nodes:

https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.1/com.ibm.ess.v5r31.qdg.doc/bl8qdg_protocolnodessupportoverview.htm

5.1.1 Guidelines

From a Spectrum Scale perspective, protocol nodes are Network Shared Disk (NSD) clients. Hence the NSD protocol is used to access the Spectrum Scale file systems that are being reexported through the protocol nodes.

The Spectrum Scale network for NSD protocol should be dedicated only for that purpose. *No CES traffic should be put through the same physical interfaces*. Protocol nodes come with multiple network ports to enable separation of the different networks.

For the CES network, the one that reexports the Spectrum Scale file systems with protocols other than NSD, consider the following points:

- The CES nodes on ESS setup should be dedicated to CES operations only. Do not add other workloads in the CES nodes.
- Use link aggregation (Link Aggregation Control Protocol (LACP) or IEEE 802.3ad) whenever possible. Keep in mind that one flow of data can use only one physical port at the same time.
- CES IP addresses should be able to be used on any protocol node. If different subnets are used, then all the CES IPs in a given CES group must be able to run on any node in that group.

- CES IPs are created as aliases on each CES node. Do not include the primary address of an adapter in the CES IP address pool.
- DNS or /etc/hosts must be able to resolve CES IPs.
- CES does not manage the subnet or netmask configuration.
- Whenever possible, use independent filesets to separate CES shares.
- When in doubt, use the defaults. Then, test and compare to the baseline. Change one setting and compare again. Repeat until the results fulfill your needs, which might actually be the defaults.
- Consider using jumbo frames. Unfortunately, the wider the network the more complex is enablement. Check whether a higher MTU of 1500 can be used in your setup.
- You should have an internal network that communicates with the ESS IO nodes and at least one “external” network for CES exports.

5.1.2 Sizing

The hardware sizing on the protocol nodes is already taken care of when you order the nodes that are integrated with ESS, currently the 5148-22L. You should start with at least two nodes to have some resilience.

Note: For the most recent information, check the IBM Spectrum Scale Frequently Asked Questions and Answers (FAQ) on the following web page:

https://www.ibm.com/support/knowledgecenter/STXKQY/gpfsclustersfaq.html

The hardware that you receive is already compliant with the requirements. So, you should focus on how many nodes you need to fulfill your needs, including an allowance for resilience. Network sizing is paramount on CES. All clients use the CES IP addresses to connect through the protocol (Server Message Block (SMB), Network File System (NFS)). The network part gets its own dedicated nodes. See 5.1.3, “Networking planning” on page 54.

This list can help you see how many connections you can have for your protocol nodes:

- A maximum of 3,000 SMB active connections per protocol node with a maximum of 20,000 SMB connections active per cluster.
- A maximum of 4,000 NFS connections per protocol node.
- A maximum 2,000 concurrent Swift Object requests per protocol node.

Refer to this list to determine the number of nodes that you need to fulfill your connections. You should also account for node failures and plan to at least have one extra node that can handle your workload.

Note: If you are using SMB in any combination with other protocols, you can configure up to 16 protocol nodes.

If only NFS and Swift Object (no SMB) are enabled, you can have 32 protocol nodes.

If you are planning to use the SMB protocol with CES, you should see the *SMB Best Practices web page* at the following web page:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/General%20Parallel%20File%20System%20%28GPFS%29/page/SMB%20Best%20Practices

This document prsents topics that are SMB-related and gives multiple suggestions on the configuration of CES.

5.1.3 Networking planning

As a rule of thumb for NFS and SMB on a 10-Gbit network, provide capabity for 800 MB/sec sequential throughput per port per client. Your performance might vary, so always check the conditions in your environment. For example, check switch interlinks and client settings, which have a significant impact on any performance. Also, remember that LACP gives aggregated performance to all the clients, not to a single one as shown in Figure 5-1.

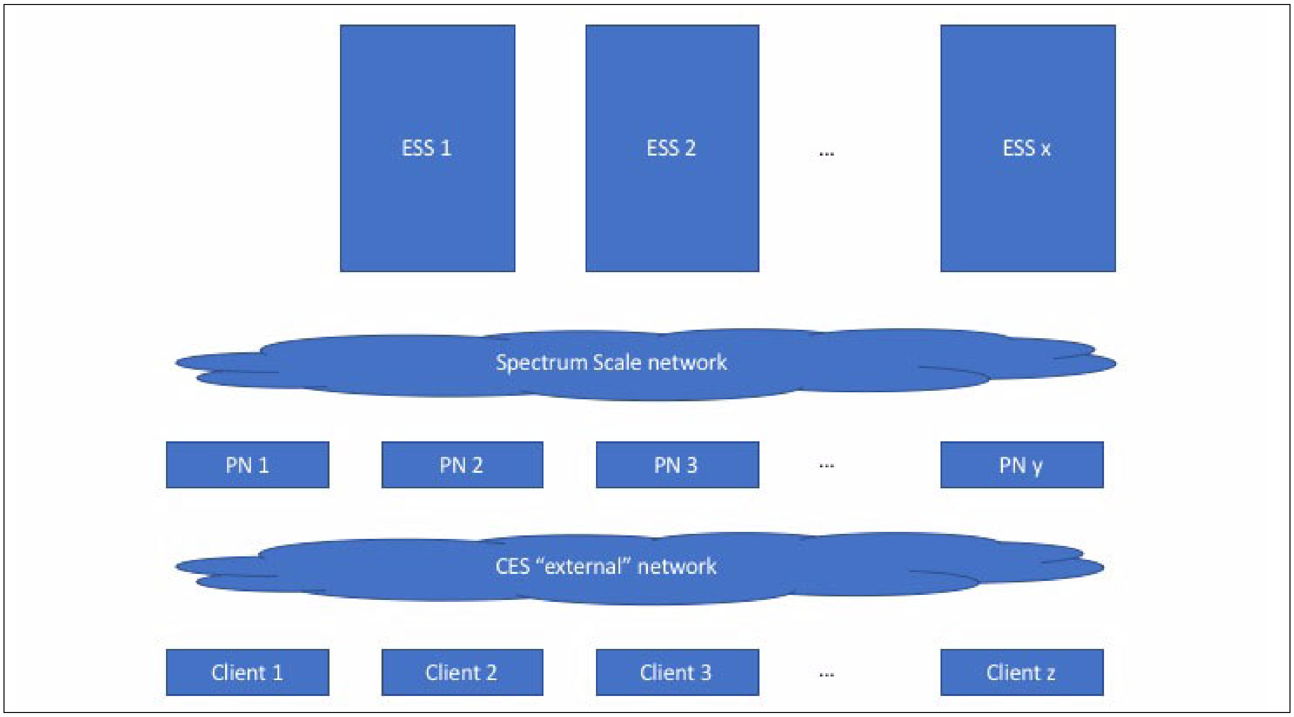


Figure 5-1 CES network

For details about how the CES IPs are assigned, see this web page:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.0/com.ibm.spectrum.scale.v5r00.doc/bl1adm_cesipaliasing.htm

CES IP addresses have the following characteristics:

- Shared between all CES protocols.
- Organized in an address pool (there can be fewer or more CES addresses than nodes).
- Hosted on the CES nodes (there can be CES nodes without CES addresses).
- Can move between CES nodes (triggered manually with a command or as part of a CES failover).
- Must not be used for General Parallel File System (GPFS) communication at the same time.

CES IP addresses have these restrictions:

- The network on CES nodes must be configured so that all CES IPs can run on any CES node. Typically, this configuration requires that all CES nodes have at least one NIC interface or VLAN-compatible interface with each CES IP network address. If different subnets are used, all the CES IPs in a given CES group must be able to run on any node in that group.

- CES IPs are created as aliases on each CES node. Do not include the primary address of an adapter in the CES IP address pool.
- DNS or /etc/hosts must be able to resolve CES IPs.
- CES does not manage the subnet or netmask configuration. Configuration must be done manually.

For details on how to configure the CES network, see this web page:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.0/com.ibm.spectrum.scale.v5r00.doc/bl1adm_cesipaliasing.htm

For operational instructions on creating shares, see this web page regarding CES features:

https://www.ibm.com/support/knowledgecenter/en/STXKQY_5.0.1/com.ibm.spectrum.scale.v5r01.doc/bl1adv_ces_features.htm

Note: A good reference for CES is the *Implementing IBM Spectrum Scale,* REDP5254 Redpaper, in particular 3.1.4 *Enabling Cluster Export Services and configuring CES IP* and 3.2 *Creating the SMB export*. See the following web page:

http://www.redbooks.ibm.com/abstracts/redp5254.html?Open

5.1.4 Tunable settings

Performance tuning is a journey. The settings that were optimal yesterday might not the best ones tomorrow. Performance is heavily influenced by factors like workload and software levels (OS, middleware, and so on). Initially for protocol nodes that are ordered with the ESS block, the installation takes care of the base settings. These settings cover the common cases, which are listed on the following web page:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.1/com.ibm.ess.v5r31.qdg.doc/bl8qdg_ostuningsppc64le.htm#bl8qdg_ostuningsppc64le

Note: Be aware that the scope of this document is only protocol nodes that are part of an ESS building block. In other words, they are part of the same order for at least one ESS, as integrated protocol nodes for ESS. Any protocol node that is ordered through other channels, such as IBM Power or other platforms, is not covered.

Those settings, and other settings and software levels, are set during the installation of the protocol nodes from the EMS (management server node), as explained on the following web page:

https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.1/com.ibm.ess.v5r31.qdg.doc/bl8qdg_protocolnodessupportoverview.htm

5.2 Initial setup

As protocol nodes that are part of an ESS building block, the installation flow is part of the ESS installation flow. That means that EMS does take care of firmware levels, OS levels, drivers levels (also referred as OFED) settings, and so on. Links to this information are provided in 5.1.3, “Networking planning” on page 54.

5.2.1 Installation

There are two possible configurations on protocol nodes that are part on an ESS building block:

- **Configuration 1:** 5148-22L protocol nodes that are ordered and racked with a new 5148 ESS (PPC64LE)
- **Configuration 2:** 5148-22L protocol nodes ordered stand-alone and added to an existing 5148 ESS (PPC64LE)

For Configuration 1, follow the steps that are described on the following web page:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.1/com.ibm.ess.v5r31.qdg.doc/bl8qdg_protoconodeswith5148ess.htm

For Configuration 2, follow the steps that are described on the following web page:

https://www.ibm.com/support/knowledgecenter/SSYSP8_5.3.1/com.ibm.ess.v5r31.qdg.doc/bl8qdg_protoconodesstandalone.htm

5.3 Ongoing maintenance (upgrades)

As the current ESS version includes CES building blocks, the updates are part of the ESS upgrade and update paths. That path would cover the firmware, OS, and driver and platform levels of the CES nodes. While that simplifies the operational part of the upgrade, proper planning is still required. To prepare for the upgrade, always refer to the documentation of the version that you are going to update.

> **Note:** For the ESS 5.2.0 the *Quick Deployment Guide* can be found on the following web page:
>
> https://www.ibm.com/support/knowledgecenter/SSYSP8_5.2.0/ess_qdg.pdf?view=kc

For a full list of possible impacts, always refer to the documentation. However, as starting point plan for the following issues:

- SMB: Requires quiescing all I/O for the duration of the upgrade. Due to the SMB clustering functionality, differing SMB levels cannot co-exist within a cluster at the same time. This requires a full outage of SMB during the upgrade.
- NFS: Recommended that you quiesce all I/O for the duration of the upgrade. NFS experiences I/O pauses during an upgrade. Depending upon the client, NFS also might experience mounts and disconnects.
- Object: Recommended that you quiesce all I/O for the duration of the upgrade. Object service is down or interrupted multiple times during the upgrade process. Clients might experience errors or they might be unable to connect during this time. They should retry as appropriate.

5.3.1 Process flow (EMS, IO Nodes, and CES)

The installation toolkit does perform the upgrade of the Protocol Nodes software and firmware. Before proceeding, you must check that the information about the cluster that the toolkit has is up to date. If it is not up to date, you can try to gather the information with the **spectrumscale** command, as shown in Example 5-1.

Example 5-1 Spectrumscale populate

```
# ./spectrumscale config populate --node EMSNode
```

For limitations of the populate option, see the S*pectrum Scale toolkit documentation*. Information on the following web page applies to Spectrum Scale 5.2.0 version:

https://www.ibm.com/support/knowledgecenter/STXKQY_5.0.2/com.ibm.spectrum.scale.v5r02.doc/bl1ins_configpoplimits.htm

Note: The toolkit can be used for Protocol Nodes that are part of the same building block as ESS, ordered as part of the ESS or as extension of the ESS as explained on 5.1, “Cluster Export Services (CES) and protocol nodes support in ESS” on page 52.

The populate operation might not fix your issues. And the installation toolkit might not have up-to-date information. In that case, you can try to add it manually. If at this point you do not have an up-to-date configuration on the toolkit, do not proceed with the upgrade and contact IBM Support.

The protocol nodes are updated before the IO nodes of the ESS cluster. As part of the planning, notice where your quorum and management nodes reside. That way, you always have enough quorum and management available to perform the online upgrade. The toolkit updates the OS, OFED drivers, and firmware node by node.

You can do an offline and faster upgrade path instead. Explore both options during your planning to see which one fits your needs better.

Keep in mind, the more complex your cluster, the more planning that you must do. You might have DMAPI, AFM, and NSD delivered by multiple types of storage outside of the ESS, and multiple policies. All the interactions with the automated toolkit need to be planned.

5.3.2 Software matrix support - FAQ

For software-level support, you should always refer to the release notes of the version you plan to install. For ESS Version 5.3.1, refer to the following web pages:

- *ESS Version 5.3.1 Release Notes:*

  https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.1/sts531_welcome.html

- The *IBM Spectrum Scale RAID Frequently Asked Questions and Answers*:

  https://www.ibm.com/support/knowledgecenter/SSYSP8/gnrfaq.html

- *IBM Spectrum Scale Frequently Asked Questions and Answers*:

  https://www.ibm.com/support/knowledgecenter/en/STXKQY/gpfsclustersfaq.html

If you have any doubts, you can always raise a support ticket through the normal IBM support channels for your organization.

6

Scenarios for disaster recovery: Stretch cluster and active file management

This chapter provides the information about configuration scenarios for disaster recovery. In these scenarios, you use the IBM Spectrum Scale active file management (called AFM in this document). AFM provides a basis for asynchronous disaster recovery and synchronous replication by using the stretch cluster.

The following topics are discussed in this chapter:

6.1 Introduction to disaster recovery with AFM

With the AFM feature, IBM Spectrum Scale expands its data protection configuration options into disaster recovery (DR) scenarios. AFM-based disaster recovery is referred to as *AFM async DR* in this document.

AFM async DR uses asynchronous data replication between two sites. The replication mechanism for DR is file-based AFM based on replication at the level of filesets. This DR capability is based on a strict one-to-one active-passive model. In this model, the sites are represented as being primary and secondary. The filesets that you replicated from a primary site are mapped one-to-one to the filesets at a secondary site. This is accomplished by establishing a DR relation at the fileset level. The files modified/created at the primary site are replicated to the secondary site asynchronously after the DR relation is established.

AFM masks wide area network (WAN) latencies and outages in the following ways:

- It uses IBM Spectrum Scale to cache massive data sets
- It provides asynchronous data movement between Cache and Home.

While AFM operates, the AFM async DR feature focuses on the accomplishment of business recovery objectives. Specifically, it targets disaster recovery goals by replicating all the data asynchronously from a primary site to a secondary site.

In AFM async DR, the applications that run on the primary site use read/write independent filesets whose data is replicated to the secondary site. The secondary site (the DR site) is read-only for direct writes under normal conditions, except for the data-replicated operations from the primary site.

The storage and network configurations of the primary and secondary filesets that you configure for asynchronous DR can be created independently. With AFM async DR, you use an existing or new fileset with the configured mode as primary (the replication data source). The target fileset that is on the secondary site should be an empty fileset. After the filesets are identified or created, you establish the AFM async DR relationship.

Figure 6-1 on page 61 shows the AFM async DR relationship.

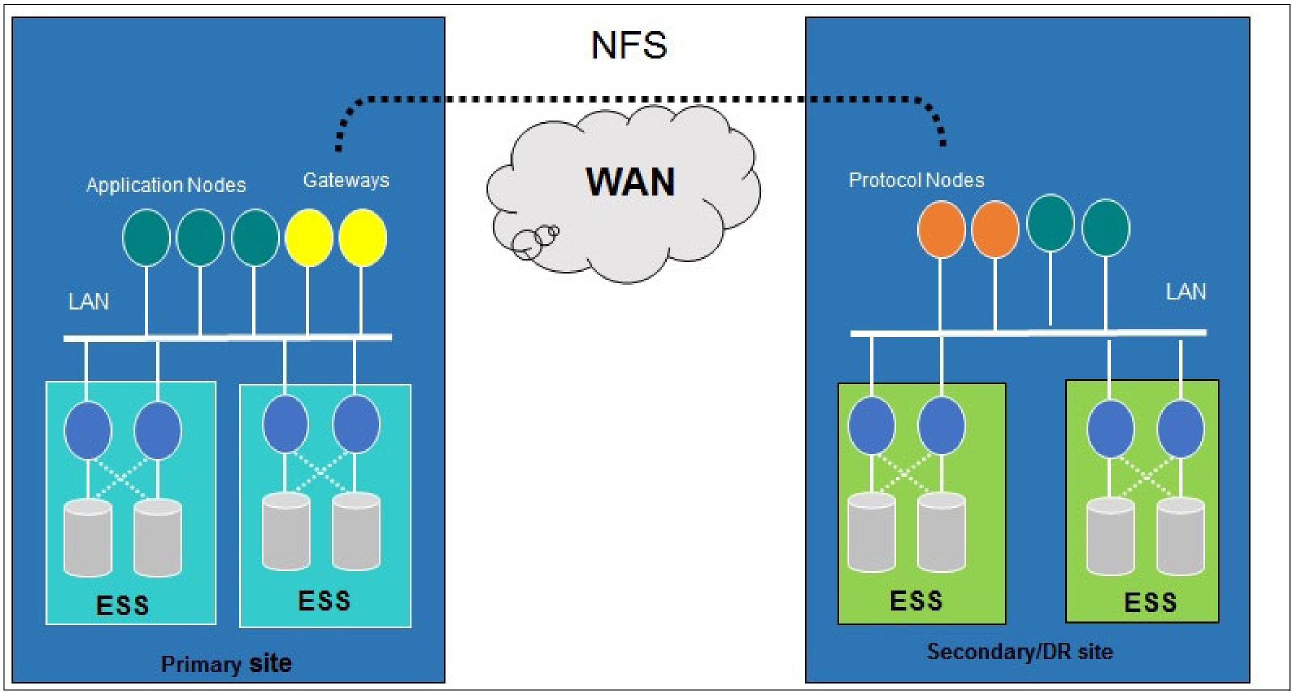


Figure 6-1 Active file management async DR (AFM async DR) relationship

Figure 6-1 shows how AFM async DR allows the primary and secondary sites to use single- or multiple-ESS systems that take the role of Network Shared Disk (NSD) servers that connect the primary and secondary clusters in Spectrum Scale.

6.2 How does it work?

The following are sequences of the high-level steps that are executed during the updating of the files of the independent fileset at primary site. This is the fileset for which the AFM async DR relationship is established.

1. The applications that run on the Spectrum Scale client nodes of the primary use NSD protocol to communicate with the NSD servers to store the data on storage media.
2. After data is written at the primary storage by using the Remote Procedure Call (RPC) calls, the gateway node of the corresponding fileset is informed about the data-write operation. The RPC does not contain any data but only the attributes that describe the write operation. This RPC might cause some extra latency for client’s write operations.
3. After RPC-to-gateway-node communication is successfully completed, the client is informed about the completion of the write (IO) operation.
4. The gateway node queues the new write operations at the end of the queue that is maintained for the fileset. For each fileset, separate queues are maintained at the gateway node. The gateway node applies various optimization techniques, for example, coalescing multiple write operations, canceling operations like create, and delete operations of same files.
5. The operations queued at the gateway node would not immediately be replicated to the secondary. Instead, they are kept in the queue for some time. This delay is configured as the a*sync delay* so that optimization operations can be applied to those operations.
6. After async delay is complete, the gateway node takes the following actions:

   a. Fetches the operations from the front of the queue.
   b. Reads the data corresponding to the operation from NSD servers.
   c. Replicates to the secondary using protocols, such as NFS.

7. The protocol server that runs at the secondary receives the data for the write operation. Then, it updates the corresponding file by writing the data to an NSD server (the ESS server).

8. In case of disaster at primary, the application can fail over to the secondary and use the secondary for business continuity. After a disaster occurs at primary, data replication to secondary might be interrupted abruptly, which might cause data inconsistency at secondary.

   For the data consistency, periodic peer snapshots are taken at primary and secondary based on RPO (Recovery Point Objective) interval that is configured. When disaster occurs at primary, the secondary is restored to most recent snapshot before applications failed over to the secondary. This behavior ensures data consistency at secondary.

6.2.1 What data is replicated

All file user data, metadata (including user extended attributes, except *inode number* and *atime*), hard links, renames, and clones from primary are replicated to the secondary. All file system/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 the primary and secondary sites is performed asynchronously, and only the modified blocks for data and metadata are transferred.

6.3 Planning

The general Spectrum Scale Planning guidelines should be followed for planning Spectrum Scale clusters, both at the primary and the secondary sites.

6.3.1 Get approval through the development approval process

The initial feedback from the field suggests that success of a disaster recovery solution depends on administration discipline, including careful design, configuration, and testing. Considering this, IBM chose to disable the AFM-based Asynchronous Disaster Recovery feature (AFM async DR) by default. Customers who want to deploy the AFM async DR feature must first review their deployments with IBM Spectrum Scale development. You should contact IBM Spectrum Scale Support at `scale@us.ibm.com` to have your use case reviewed. IBM helps you optimize your tuning parameters and enable the feature. Always include this message when you contact IBM Support.

6.3.2 Introduction to planning

AFM async DR is characterized by these two modes:

- The fileset in the primary cluster uses the primary mode.
- The secondary cluster uses the secondary mode.

As part of initial planning, the modes of the Spectrum Scale clusters must be defined.

The AFM async DR relation can be established between primary and secondary in three ways, based on availability of the initial data.

Status of the data	Impact on establishing the AFM async DR relation
This is initial data.	The AFM async DR relation can be established between new filesets that you create at primary and secondary.
The primary site exists and applications are using independent filesets.	You can create a new fileset at the secondary site. And the AFM async DR relation can be established between an independent fileset at primary and new fileset at secondary.
AFM single writer (SW) or independent writer (IW) relations already existed between the home and cache. And you want to convert the AFM relation to and AFM async DR relation.	The SW/IW relations can be converted to an AFM async DR relation. You cannot convert the other AFM relations -- Read-Only (ro), Local-Update (lu) -- to an AFM async DR relation.

A disaster recovery solution considers two important business-critical parameters:

- **Recovery time objective (RTO)**: Represents the amount of time that is 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. Also, the following factors can provide all the solution requirements to fulfill the business criteria:

- required system resources,
- the volume of data that is generated for a given time, and
- the network bandwidth.

6.3.3 Gateway nodes

The gateway plays a critical role in replicating the modified data of individual filesets to the corresponding filesets of the secondary site. The gateway node uses the protocol (NFS or GPFS), which is configured while you create the DR relationship, to replicate the data to the secondary site. As part of initial planning, users identify the filesets whose data must be replicated from primary to secondary. Users must also identify the corresponding filesets (DR filesets) at secondary to map one to one with the primary filesets.

The Gateway node maintains in-memory queues of data replication operations that are replicated after some delay (async delay). The async delay can be configurable. When the async delay is longer, the queue consumes more resources (memory). Similarly, each gateway node can support multiple filesets to replicate the modified data to the secondary site. When a gateway node supports more filesets, the node consumes more resources, too. The assignment of gateway nodes to individual filesets at the primary site happens automatically based on these factors:

- number gateway nodes configured
- number of independent filesets that are configured for AFM async DR

The gateway nodes need to have access to an extra network that is configured to communicate with the secondary site. The gateway nodes use this extra network to replicate the modified data to the secondary site. The gateway nodes serve as a client that replicates the modified data to the secondary site. It does so by communicating with protocol servers that run at the secondary site.

On the ESS environment, a separate node is configured as gateway node. The required resources connect to the extra network to communicate with the secondary site. While it replicates the data, the gateway node reads the modified data blocks or bytes from the ESS server and replicates that data to the secondary site.

6.3.4 Networking

Networking is one of the most important factors when you use Spectrum Scale and ESS. The performance of network plays an important role in performance and throughput delivery by ESS systems. Within Spectrum Scale, the ESS servers would be NSD servers that are configured as part of Spectrum Scale cluster. The Daemon network and Admin networks are used to communicate with ESS server within Spectrum Scale. The gateway node replicates the modified data asynchronously by reading the data from NSD servers and uses the Daemon Network.

Also, in the configuration of AFM async DR, you must define or configure an additional network. This network is used for replication of the data from primary site to secondary site. This network can be called a DR network.

The DR network

The Spectrum Scale DR Network for AFM async DR has these characteristics:

- It is used to replicate data between the primary and secondary site.
- It requires TCP/IP.
- A WAN can connect a primary and secondary that are far from each other.
- A LAN can connect a primary and secondary that are close to each other.
- In addition to TCP/IP, you can use the Remote direct memory access (RDMA) to replicate data between primary and secondary. They must be connected to each other over a LAN in which the latency is low.

It is important for the DR Network to be reliable and stable. The AFM async DR gateway node can recover the network connections between primary and secondary if there are small network outages. But an unstable or higher-latency network might cause the gateway node to consume more resources. This condition arises when many data replication operations are pending at the gateway node, which might lead to failure of the gateway node. The bandwidth and latency of the DR network should, more than enough to replicate the data that is generated (modified) at primary without any backlog.

6.3.5 Transport protocols

The AFM async DR network uses the following protocols to replicate the data from the primary site to the secondary site:

- **NFS protocol:** The gateway node replicates the modified data to the secondary site by sending the modified data based on the Network File System (NFS) protocol. The gateway node runs the NFS client and connects to NFS servers that are configured at secondary, and communicates with NFS servers to replicate the data. The NFS protocol can be used over an IBM WebSphere® Application Server that supports TCP/IP or over a LAN that supports TCP/IP.

- **GPFS protocol:** The gateway node can use the IBM General Parallel File System (GPFS) native protocol to replicate data from the primary to the secondary site. The GPFS protocol can be used over a low-latency LAN network that supports only TCP/IP or RDMA.

It is important for the DR Network to be reliable and stable. The AFM async DR gateway node can recover the network connections between primary and secondary if there are small network outages. But an unstable or higher-latency network might cause the gateway node to consume more resources. This condition arises when many data replication operations are pending at the gateway node, which might lead to failure of the gateway node. The bandwidth and latency of the DR network should more than enough to replicate the data that is generated (modified) at primary without any backlog.

6.3.6 AFR DR limitations

Multiple parameters are considered while the DR relation is established between primary and secondary filesets for DR, such as RPO intervals, network bandwidth, and latency. Some of these parameters enforce limitations during configuration of AFM async DR, between the primary and secondary. Also, some Spectrum Scale features like File Placement Optimizer are not supported with AFM async DR. But none of these limitations and restrictions are enforced by the ESS environment. More details of the AFM async DR limitations can be found at AFR DR Limitations.

6.4 Initial setup

This section describes the configuration and initial setup of a DR relation between the primary and secondary sites.

6.4.1 Prerequisites

The primary and secondary sites are two separate operational clusters that are separated from each other through LAN or WAN. The NFS or GPFS protocol can be used for communication. The gateway nodes and NFS servers must be planned on the primary and secondary clusters. It is preferable to configure gateway nodes and NFS servers in the primary and secondary clusters before you create filesets and start applications. If you are planning to use GPFS protocol, the primary file system must be remotely mounted on all the gateway nodes in the secondary cluster.

User IDs and group IDs must be managed the same way across the primary and secondary clusters. These issues are described in Installing and upgrading AFM-based Disaster Recovery section of the Spectrum Scale documentation.

You must set up the NFS server at secondary. You do this by exporting the filesets from primary that must have a one-to-one mapping when the DR relation is established.

6.4.2 Primary and secondary configurations

The AFM async DR relation can be established between two independent filesets of Spectrum Scale clusters of the primary and secondary sites. The following are different ways to create the DR relationship.

- **Create an AFM-based DR relationship between two new filesets that are created at primary and secondary:**

  The steps at Creating an AFM-based DR relationship from the Spectrum Scale documentation describe how to do this:

  Create AFM async DR relationship between two newly created independent filesets.

- **Create an AFM-based DR relationship by converting existing GPFS filesets to AFM async DR:**

  Sometimes a primary site already has an independent fileset that is being used by applications and you need to establish AFM async DR for that fileset. The steps at Converting GPFS filesets to AFM async DR from the Spectrum Scale documentation describe the steps followed to do this:

  Create AFM async DR relationship by converting existing independent GPFS filesets from primary site.

- **Create an AFM-based DR relationship by converting existing an AFM relationship to AFM async DR:**

  The steps at Converting AFM relationship to AFM async DR describe the steps followed to do this:

  Create AFM async DR relationship by converting existing independent GPFS filesets from primary site.

The AFM async DR relation between the primary and secondary sites can be configured or customized by changing various configuration parameters. The section Configuration parameters for AFM-based DR from the Spectrum Scale documentation describe various configuration parameters to configure the AFM async DR relation.

6.4.3 Tuning

The AFM async DR gateway node uses the protocols (NFS, GPFS) configured to replicate the data from primary to the secondary site. To improve performance, you must tune these protocols,

- at the client (AFM async DR gateway node) and
- at the protocol servers (secondary site) that host the export of DR filesets.

The parameters that must be tuned for the protocol are described at Tuning for kernel NFS backend on AFM and AFM async DR in Spectrum Scale documentation in the *Configuring* chapter.

It is recommended that you follow this guideline:

Increase the worker1threads to twice the number of AFM DR filesets that are using GPFS backend, than exist on all file systems in the secondary cluster.

More details about this recommendation can be found at Recommended worker1threads on primary cluster.

6.5 Ongoing maintenance

This section describes the upgrade of the primary and secondary sites. It also describes AFM async DR features that are used to fail over and fail back the application between secondary and primary if a disaster affects the primary site.

6.5.1 Upgrading AFM async DR

Before you upgrade to a newer version of IBM Spectrum Scale, consider the version from which you are upgrading. IBM Spectrum Scale supports a limited degree of backward compatibility between two adjacent releases. Coexistence and compatibility measures are required. For example, you temporarily can use IBM Spectrum Scale nodes that run on the newer version alongside nodes that run an earlier version. For details, see IBM Spectrum Scale supported upgrade paths.

The AFM async DR systems on the primary and secondary sites can be upgraded independently, in any order. They can be upgraded on a different schedule, too. The secondary site exports the filesets to primary. During the upgrade of secondary, access to the secondary is disconnected. In a disconnected state, the primary builds up the queue for application data modification operations. After secondary becomes available again, these operations are executed by replicating data to secondary. It is advised that you upgrade secondary when the workload on primary is low. That way, the upgrade does not overuse resources by monopolizing the queue on the gateway node.

At a primary site with a multiple-gateway environment, you can upgrade gateway nodes one-by-one. Consider this scenario:

- Filesets are associated with the gateway node that you must upgrade.
- At the same time, these filesets are failover for another gateway node.
- Any write-class operation triggers the recovery feature that builds the queue on the failover gateway node to continue replicating updated data to secondary.
- In this scenario, the primary-to-secondary connection does not end. However, some performance degradation can be seen due to another gateway node that is working for the connection for those failover filesets.

In heavy-load systems, the transfer of the filesets to another gateway node might have a performance impact. It is advisable to run such upgrades when the load on the system or the number of data transfers is minimal.

You can upgrade ESS servers independently from the gateway nodes. The servers can be in rolling-upgrade mode, in any order, but you must follow the Spectrum Scale upgrade guidelines. For more details for an AFM async DR upgrade, see the section Upgrading AFM and AFM async DR in the Spectrum Scale documentation.

6.5.2 Recovery

This section describes the AFM async DR features that recover business continuity if the primary site is affected by a disaster. The section also describes how to set up a new secondary if a disaster affects the original secondary site.

Failover to the secondary site

Under DR, any time that a disaster affects the primary site, an application fails over to secondary to ensure continuity. As part of failover to secondary, you can restore the latest snapshot data on the secondary site. By default, applications fail over to secondary without restoring the secondary to the latest snapshot. After failover, the secondary site becomes the acting primary site. Its mode changes to read/write and its status changes to ready to serve the applications.

For more information on failover to the secondary, see the Failover to the secondary site section in the Spectrum Scale documentation.

Failback to the old primary site

If the disaster that happens at primary does not destroy the system at the primary site, the primary site can be recovered. After a primary site is recovered, you can proceed as follows:

1. Run the failback procedure to restore failover for applications from the original primary site to secondary.
2. Reestablish the AFM async DR relation between the recovered primary and secondary sites.

For more details on failing back to the recovered original primary, follow the steps that are described in Failback to the old primary site section in the Spectrum Scale documentation.

Failback to a new primary site

If the disaster destroyed the primary site completely, a new primary site can be set up and populated with data. After the new primary site is set up, the applications' failover to secondary can be failed back to the new primary site. And the AFM async DR relation can be reestablished between the new primary and the secondary.

For more details on failing back to the new primary, follow the steps that are described in the Failing back to the new primary site section in the Spectrum Scale documentation.

Changing the secondary site

The secondary site might also be affected by a disaster, or secondary might fail to function. In either case, you must set up a new secondary site to achieve these goals:

- Support disaster recovery
- Reestablish the AFM async DR relation between the new secondary and primary.

In this scenario, the applications that run on primary continue without any disruption.

You can set up a new secondary and create the AFM async DR relation between the new secondary and primary. Follow the steps in the Changing the secondary site section in the Spectrum Scale documentation.

This chapter includes information about synchronous replication using the IBM Spectrum Scale stretch cluster for the synchronous disaster recovery feature and IBM Spectrum Scale integration on ESS systems.

6.6 Introduction to synchronous replication using the stretch cluster

You can support synchronous data replication between two sites for disaster recovery. For example, you can define a single Spectrum Scale cluster over three geographically separate sites: two production sites and a tiebreaker site. One or more file systems are created, mounted, and accessed concurrently from the two active production sites that are connected over a reliable network.

The data and metadata replication features of Spectrum Scale are used to maintain a secondary copy of each file system block. This setup relies on the concept of disk failure groups to control the physical placement of the individual copies:

In this sceanrio, you have two copies of the data in separate locations. If one site has an unrecoverable disaster, the data can be recovered from a single site with no data loss. Data from two separate sites can share a namespace and can be accessed by either site.

Figure 6-2 shows the Synchronous mirroring with Spectrum Scale replication.

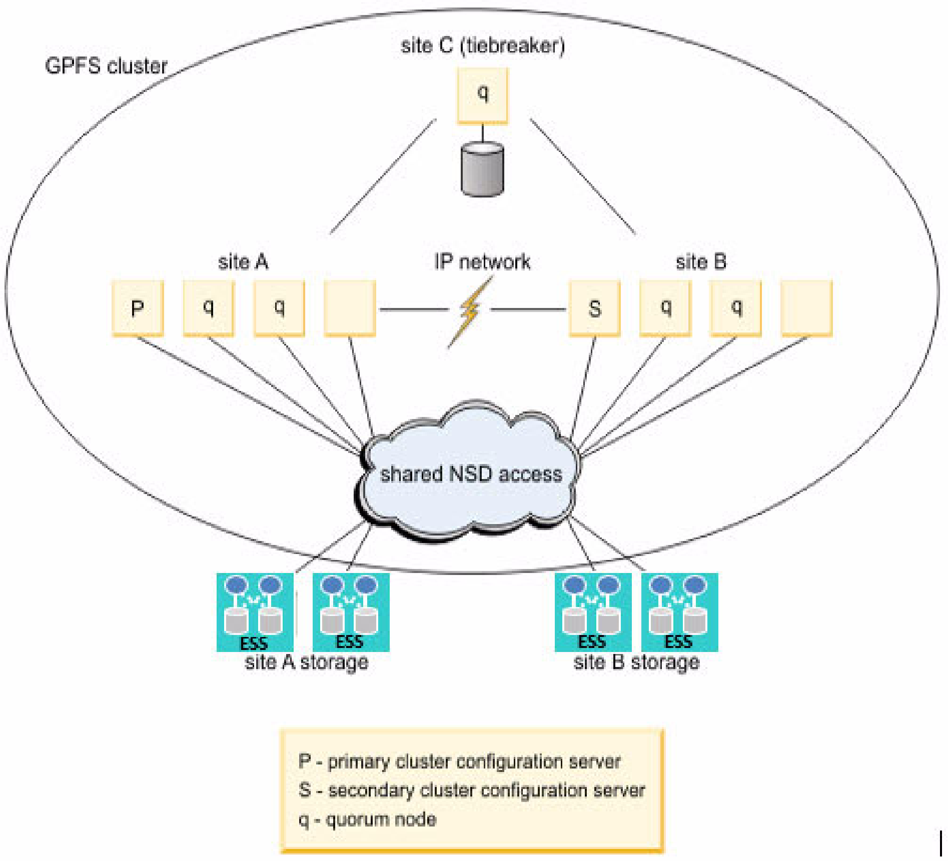


Figure 6-2   GPFS cluster

The stretch cluster for synchronous data replication between two production sites is configured as active-active. That means that clients can read and write data from either site of the cluster. For more information, see. Synchronous mirroring with GPFS replication.

6.6.1 How it works

In a stretch cluster configuration, the data- and metadata-replication features of Spectrum Scale maintain a secondary copy of each file system block. The system relies on the concept of disk failure groups to control the physical placement of the individual copies:

- Separate the set of available VDisk volumes into two failure groups, with one failure group defined at each of the active production sites.
- Create a replicated file system. Specify a replication factor of 2 for both data and metadata.

When it allocates new file-system blocks, GPFS always assigns replicas of the same block to distinct failure groups. This approach provides enough redundancy. It allows each site to continue operating independently should the other site fail.

The stretch cluster for synchronous data replication ensures that:

- The data is always available.
- The data can be read and written in both locations.
- Recovery actions from applications are not required, except for changing the IP address/ hostname.

Because data is synchronously replicated, the application receives consistent data and no data is lost during failover or failback.

6.6.2 What data is replicated

Based on the configuration, either the user data, the metadata, or both are replicated to production sites synchronously. But for disaster recovery support, you must mirror both data and metadata onto production sites.

6.7 Planning

You must follow the general Spectrum Scale Planning guidelines for planning Spectrum Scale clusters.

6.7.1 Introduction

In stretch cluster configuration that use Spectrum Scale replication, a single Spectrum Scale cluster is defined over three geographically separate sites. These sites consist of two production sites and a third tiebreaker site.

Spectrum Scale enforces a node quorum rule to prevent multiple nodes from assuming the role of the file system manager, in the event of a network partition. Thus, a majority of quorum nodes must remain active so that the cluster can sustain normal file system usage. Furthermore, Spectrum Scale uses a quorum replication algorithm to maintain the content of the file system descriptor (one of the central elements of the Spectrum Scale metadata). When formatting the file system, Spectrum Scale assigns some number of disks (usually three) as the descriptor replica holders. These holders are responsible for maintaining an up-to-date copy of the descriptor. Like the node quorum requirement, a majority of the replica holder disks must remain available at all times to sustain normal file system operations. This file system descriptor quorum is internally controlled by the Spectrum Scale daemon. However, a disk might fail due to a disaster. In this case, you must manually intervene to inform Spectrum Scale that the disk is no longer available and must be excluded from use.

In stretch cluster configuration, when it allocates new file system blocks, Spectrum Scale always assigns replicas of the same block to distinct failure groups. This approach provides a sufficient level of redundancy. Each site can continue operating independently should the other site fail. The individual production sites should have their own ESS servers that work as NSD servers to store the data and metadata. The VDisks of the production sites should be organized into distinct failure groups.

The synchronous replication of a stretch cluster system requires a reliable high-speed, low latency network to access shared NSDs (network shared disks) from sites. Failbacks affect performance on the application.

6.7.2 Networking

Networking is one of the most important factors when you use Spectrum Scale and ESS. The performance of the network plays an important role in performance and throughput that an ESS system delivers. Synchronous replication requires a reliable high-speed, low-latency network to access shared VDisk (NSDs) from production sites.

But there are no special networking requirements for stretch cluster configuration. For example:

- No need to create different subnets.
- No need to have Spectrum Scale nodes in the same network across the two production sites.
- Production sites can be on different virtual LANs (VLANs).

6.8 Initial setup

This section describes the configuration and initial setup of a DR Spectrum Scale stretch cluster across two data access sites and one tiebreaker site.

6.8.1 Prerequisites

In a configuration that uses Spectrum Scale replication, a single Spectrum Scale cluster is defined over three geographically separate sites. These sites consist of two production sites and a third tiebreaker site. The production sites are configured with one or more ESS servers as NSD servers. A third tiebreaker site has the tiebreaker role for the node and file system descriptor quorum decisions.

6.8.2 Configuration of Spectrum Scale stretch cluster

The Spectrum Scale stretch cluster feature is configured for mirroring the data and metadata between two sites (site A and site B). The feature requires a tiebreaker site (site C). The ESS storage servers are part of the two production sites (site A and site B) and are configured as NSD servers for the Spectrum Scale stretch cluster. The VDisk volumes that are configured on ESS servers are used to define two distinct failure groups, one at each production site.

The tiebreaker site plays the role of tiebreaker for the node and file system descriptor quorum decisions. The tiebreaker site consists of:

1. **A single quorum node**

The function of this node is to serve as a tiebreaker in Spectrum Scale quorum decisions. The node does not require normal file system access and SAN connectivity. To ignore disk access errors on the tiebreaker node, enable the **unmountOnDiskFail** configuration parameter through the mmchconfig command. When enabled, this parameter forces the tiebreaker node to treat the lack of disk connectivity as a local error. As a result, the system detects a failure to mount the file system, rather than reporting this condition to the file system manager as a disk failure.

2. **A single network shared disk**

The function of this disk is to provide an additional replica of the file system descriptor file. This replica is needed to sustain quorum should a disaster cripple one of the other descriptor replica disks. Create a network shared disk (NSD) over the tiebreaker node's internal disk defining these characteristics:

- The local node as an NSD server.
- The disk usage as descOnly.

  The *descOnly* option instructs Spectrum Scale to store only file system descriptor information on the disk.

For more information on configuring the Spectrum Scale stretch cluster along with tiebreaker site, see Setting up IBM Spectrum Scale synchronous replication.

6.9 Ongoing maintenance

This section describes the upgrade of Spectrum Scale stretch cluster and DR features to work as follows:

1. Fail over to surviving site.
2. Fail back to original state, after the failed sites are recovered.

6.9.1 Upgrading Spectrum Scale stretch cluster

Before you upgrade to a newer version of IBM Spectrum Scale, consider the version from which you are upgrading. IBM Spectrum Scale supports a limited degree of backward compatibility between two adjacent releases. Coexistence and compatibility measures are required. For example, you temporarily can use IBM Spectrum Scale nodes that run on the newer version alongside nodes that run an earlier version. For details, see IBM Spectrum Scale supported upgrade paths.

The Spectrum Scale stretch cluster is a single Spectrum Scale cluster. The general guide lines to upgrade Spectrum Scale should be followed to upgrade Spectrum Scale stretch cluster. For more information, see the *Upgrading* chapter in Spectrum Scale documentation.

6.9.2 Recovery

This section describes the features that synchronous DR supports, based on stretch cluster. These features work to recover business continuity if a site is affected by disaster.

The Spectrum Scale replication over stretch cluster allows for failover to the surviving site without disruption of service if the following condition is true:

Both the remaining site and the tiebreaker site remain functional.

The system remains in this state until a decision is made to restore the operation of the affected site by executing the failback procedure. If the tiebreaker site is also affected by the disaster and is no longer operational, GPFS quorum is broken. You must manually to restore file system access.

- **Failover to surviving site:** At any time, if disaster affects a site, the application can be failover to a surviving site to ensure continuity. The process for failover to a surviving site depends on whether the tiebreaker site is affected or not. If tiebreaker site is available, the proposed three-site configuration is resilient to a complete failure at any single hardware site. If all disk volumes in one of the failure groups become unavailable, Spectrum Scale performs a transparent failover to the remaining set of disks. Then, it continues serving the data through the surviving subset of nodes, without requiring administrative intervention.

If the tiebreaker site also fails along with any other site, you must manually intervene to achieve failover to a surviving site. For detailed steps to failover to a single surviving site, see Failover to the surviving site.

- **Failback procedures**: The failback procedure depends upon whether the nodes and disks at the affected site have been repaired or replaced as described below:
  - Here are the procedures for temporary outages:
    - If the configuration has not been altered, fail back to the original state. The detailed steps to failback are described in the Failback with temporary loss and no configuration changes section.
    - Consider the scenario where the configuration has been altered, configuration is server-based configuration, and primary and secondary configuration servers are in use. In this case, follow the procedure to fail back to the original state. This procedure is described in the Failback with temporary loss and configuration changes in server-based configuration section.
    - If the configuration has been altered and configuration is Clustered Configuration Repository (CCR) configuration, follow the procedure to fail back to the original state. This procedure is described in the `Failback with temporary loss using the Clustered Configuration Repository (CCR) configuration mechanism` section.
  - For a permanent outage, remove and replace the failed resources. Resume the operation of Spectrum Scale across the cluster by following the failback procedure that is described in the `Failback with permanent loss` section.

7

Adding ESS Generation 2 (PPC64LE) building blocks to ESS Generation 1

This chapter describes the planning and recommended practices for deploying a system with mixed generation 1 (Gen1) and generation 2 (Gen2) hardware. It has the following sections:

7.1 Overview

You deploy new ESS building blocks to extend the capacity of an existing ESS cluster. The storage in these new building blocks can be used to extend existing file systems, or to create new file systems. For details, see Section 1.3, “Building blocks” on page 5.

ESS Gen1 is based on the Power8 architecture and runs in big-endian (BE) mode (PPC64**BE**). For Gen2, ESS transitioned to the POWER8 architecture and runs in little-endian (LE) mode (PPC64**LE**). IBM Spectrum Scale seamlessly converts data between little-endian and big-endian. So, the storage from both ESS Gen1 and Gen2 can be used in the same Spectrum Scale cluster, and even in the same file system.

This section describes the planning and best practices for deploying a system with mixed Gen1and Gen2 hardware.

7.2 Network and hardware planning

Network connectivity is a critical aspect of integration for an ESS cluster. The following sections describe planning required for EMS nodes and network connectivity before you mix ESS Gen1 and Gen2 system.

7.2.1 EMS node requirements

An ESS Management Server (EMS) node can manage multiple ESS building blocks. In the case of Gen1 and Gen2 hardware in the same cluster, a separate EMS for each generation is required. At least one EMS must be dedicated to managing the PPC64BE architecture, and a new PPC64LE EMS is required to manage the Gen2 EMS.

In most deployments, the EMS is also responsible for running the management GUI for Spectrum Scale. The Gen1 EMS that is used to manage the Gen1 PPC64BE nodes runs the management GUI. It serves as a collector node for performance monitoring statistics. As a part of adding the new Gen2 hardware, a new Gen2 EMS node is needed. This new node runs the Spectrum Scale GUI and performance collector. The GUI and performance collection is shut down on the Gen1 EMS node and moved to on the Gen2 EMS. Existing nodes need to be reconfigured to send performance data to Gen2 EMS.

7.2.2 Network requirements

A typical ESS deployment requires four networks that are described in this section. The planning information here includes the requirements for adding Gen2 hardware to a Gen1 cluster.

Management and provisioning network

The management and provisioning network is sometimes called the XCat network, and runs on 1 Gb. It is used to provision systems during the initial install and also during some maintenance and during code upgrades.

When you add a Gen2 ESS to a Gen1 system, this network must remain a single, flat network with all systems accessible, and all nodes reachable and resolvable. Multiple subnets are not supported on this network, and each system must be directly accessible and attached to this network.

Clustering network

The clustering network is sometimes called the data network, and this network is used for the majority of node-to-node communication. This is typically a high-speed interconnect and can be either InfiniBand or Ethernet based.

Gen2 ESS building block can continue to use this network for cluster communications between the cluster nodes. In most cases, this involves connecting the new ESS cluster nodes to the same Ethernet or InfiniBand switch that the Gen1 nodes use.

Ensure that sufficient ports and IPs are available before you add the nodes in the system. In many cases, multiple Ethernet ports are bonded and Gen2 nodes can be bonded in the same way as the Gen1 nodes.

External public network

Administrators use the external public network to access the system GUI and also for general management of the system.

Gen2 systems require a new EMS node. For this reason, you must ensure that the EMS node is connected to this network and has an accessible and routable IP address assigned to it. This new EMS node runs the Spectrum Scale GUI. It is important to inform cluster administrators that the IP used to manage the Spectrum Scale cluster will change on this network.

Service network

The service network is used by the flexible service processor on the EMS and IO nodes. It manages the system hardware. On the Gen1 systems, there is an HMC attached to this network, too. For Gen2, an HMC is not required.

Gen2 ESS systems require a separate service network. This network must be logically isolated from the service network that is used to manage the Gen1 systems and the HMC.

7.3 Storage planning

This section describes strategies for using the storage in the new Gen2 building blocks that are added to the system.

There are typically three ways that new storage is used when it is added to a cluster:

- To replace existing storage devices that are no longer required. This might happen because of warranty issues, to replace slower storage with faster storage, or for other reasons.
- To expand the capacity of an existing file system.
- To create new file systems.

Creation of a new file system is the most straightforward option. It requires minimal configuration change for an existing file system.

The additional building blocks can be used to create VDisks and storage for the new file system, as required.

7.3.1 Expanding an existing file system

If the new building block is used to expand an existing file system, you must determine how the new storage will be used. Also, determine what storage pools will be used for the expansion. You must also be aware that a Spectrum Scale file system uses disk capacity for both data and metadata. Data and metadata can be split into different pools, with metadata being limited to a default pool known as the *system pool*.

Spectrum Scale best practices require that storage with different performance characteristics, or with different capacity should be located into different storage pools. Gen2 hardware has different amount of disk types, available disk capacities, enclosure size and controllers, and storage. For this reason, a Gen2 system differs in both capacity and in performance. It is preferable to place Gen2 VDisk NSDs (NSDs that are defined on the VDisks) into pools that are separate from Gen1 VDisk NSDs. If multiple types of Gen2 building blocks are being added (for example, a GSxS and GLxS or a hybrid GHxS containing both SSD and spinning disk storage) you should configure each drive type in a separate pool.

You can use the **mmdf** or **mmlspool** commands to determine how the existing storage is configured. Addition of new storage to existing file system depends on two factors.

- Current file system configuration
- Intended purpose of new storage capacity

You can configure file systems with either a single ‘system’ pool, or with ‘system’ and one or more ‘data’ pools. The additional capacity that you add can be used for either data or metadata. The following table shows several approaches to adding storage space.

Table 7-1 Scenarios for adding storage space

Scenario	Process for adding storage space
Add space for data to a file system that is configured with a single-system pool.	Consider the case of a file system with a single-system storage pool, used both for metadata and data. In this case, the best practice is to create a second pool to contain the storage from the Gen2 hardware. You can then use Spectrum Scale placement and ILM (information lifecycle management) policies to seamlessly store. Or you can move data from the existing system pool to the new storage pool. These policies can direct certain filesets to a specific pool. Also, policies can be triggered when the system pool reaches a specific capacity, at which time data moves from one pool to another.
Add space for data only to a file system that is configured with multiple pools.	Your system might have sufficient space for metadata in the system pool, and the new storage will be used to store data only. In this case, it is recommended that you add another data pool with the new storage in the Gen2 system. You can modify the existing ILM and placement policies to place or move data into this new pool as required.
Add space for metadata to a file system.	Typically this activity is required for adding faster storage for metadata into the system, or to replace Gen1 hardware that is near end of its service life. It is recommended that you completely migrate the system pool from Gen1 hardware, and move it completely to Gen2 hardware. You can either decommission the Gen1 hardware or you can repurpose it to extend the file system capacity in another data pool.

Consider these additional procedural points:

- You can add Gen2 hardware to the system pool if the existing system pool contains only metadata. Gen2 blocks that you add should have equal or more capacity than what exists on the system pool already.
- You can suspend Gen1 NSDs and migrate the data from Gen1 to Gen2 using the **mmrestripefs** command. Then, you can either remove Gen1 disks from the system pool or use them for a new data pool.
- If the existing system pool contains data and metadata, confirm whether there is enough Gen2 capacity to hold the full contents of the system pool.
  - If there is as much or more Gen2 capacity as Gen1 capacity, the process is the same as the metadata-only procedure above.
  - If there is only enough Gen2 capacity to hold metadata, you must create a new data pool. If there is not enough free capacity, then after you add the Gen2 storage, some Gen1 storage can be suspended. Data can be migrated off from the suspended Gen1 storage, which can then be used to create a new pool. Using policies, data can be moved to this new pool, and this process can be repeated until all Gen1 storage is removed.

7.4 Maintenance

Most maintenance and problem determination of Gen1 and Gen2 mixed systems can be handled through the ESS GUI, which runs on the Gen2 EMS node. Notable exceptions are code upgrades and hardware maintenance.

The HMC and Gen1 EMS are still required to maintain Gen1 hardware. Documented service actions can be performed on the appropriate system.

7.4.1 Upgrading building block version

You must do upgrades individually on the Gen1 and Gen2 hardware. Typically, the Gen2 hardware must be upgraded first, followed by the Gen1 hardware. The standard upgrade guides contain step-by-step instructions on upgrading each set of hardware.

7.4.2 Upgrading client nodes

You can do versioning for Spectrum Scale on cluster nodes in a mixed system indepependently from building block upgrades. The Spectrum Scale cluster can coexist with minor variations of version. However, it is recommended that you run same version on all the nodes in a cluster.

8

Other use cases

In this chapter, we present IBM Spectrum Scale use cases that use ESS and are already documented elsewhere. This chapter introduces such use cases and has the following sections:

- 8.1, “IBM Spectrum Scale with big data and analytics solutions” on page 82
- 8.2, “Recommended practices for Genomics Medicine workloads in IBM Spectrum Scale” on page 82
- 8.3, “IBM Spectrum Protect Blueprints (formerly Tivoli Storage Manager)” on page 83
- 8.4, “IBM Spectrum Archive EE (formerly Linear Tape File System)” on page 83
- 8.5, “IBM Cloud Object Store” on page 83
- 8.6, “IBM Cloud Private (ICP)” on page 84

8.1 IBM Spectrum Scale with big data and analytics solutions

IBM Spectrum Scale is flexible and scalable software-defined file storage for analytics workloads. Enterprises around the globe deploy IBM Spectrum Scale to form large data lakes and content repositories to perform *High Performance Computing (HPC)* and analytics workloads. It is known to scale performance and capacity without bottlenecks. *Hortonworks Data Platform (HDP)* is a leader in Hadoop and Spark distributions. HDP addresses the needs of data-at-rest, powers real-time customer applications, and delivers robust analytics that accelerate decision making and innovation.

IBM Spectrum Scale solves the challenge of explosive growth of unstructured data against a flat IT budget. IBM Spectrum Scale provides unified file and object software-defined storage for high-performance, large-scale workloads, and it can be deployed on-premises or in the cloud.

IBM Spectrum Scale is POSIX compatible, so it supports various applications and workloads. By using IBM Spectrum Scale HDFS Transparency Hadoop connector, you can analyze file and object data in place, with no data transfer or data movement. Traditional systems and analytics systems use and share data that is hosted on IBM Spectrum Scale file systems.

Hadoop and Spark services can use a storage system to save IT costs because no special-purpose storage is required to perform the analytics. IBM Spectrum Scale features a rich set of enterprise-level data management and protection features. These features include snapshots, information lifecycle management (ILM), compression, and encryption, which provide more value than traditional analytic systems do.

For more information, see the following document:

http://www.redbooks.ibm.com/abstracts/redp5397.html?Open

8.2 Recommended practices for Genomics Medicine workloads in IBM Spectrum Scale

IT administrators, physicians, data scientists, researchers, bioinformaticians, and other professionals who are involved in the genomics workflow need the right foundation to achieve their research objectives efficiently. At the same time, they want to improve patient care and outcomes. Thus, it is important to understand the different stages of the genomics workload and the key characteristics of it.

Advanced genomics medicine customers are outgrowing NAS storage. The move from a traditional NAS system or a modern scale-out NAS system to a parallel file system like IBM Spectrum Scale requires a new set of skills. Thus, the IBM Spectrum Scale Blueprint for Genomics Medicine Workloads must provide basic background information. It must also offer optional professional services to help customers successfully transition to the new infrastructure.

For more information, see the following document:

http://www.redbooks.ibm.com/abstracts/redp5479.html?Open

8.3 IBM Spectrum Protect Blueprints (formerly Tivoli Storage Manager)

The IBM Spectrum Protect Blueprints explains how to use ESS as a disk pool for IBM Spectrum Protect. This enables the backup server to have a different storage from the one it uses for backing up systems. It also enables high performing storage for both backup and restores at competitive prices.

For more information, see the following web page:

https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Storage%20Manager/page/IBM%20Spectrum%20Protect%20Blueprints

8.4 IBM Spectrum Archive EE (formerly Linear Tape File System)

More and more files are stored in file systems. Over time, many files are never or rarely accessed.

Deleting files sounds easy but requires additional decision processes and approvals in a business environment. Thus, most files are retained in the file storage system and there is constant data growth.

> **Note:** ESS is certified to be the IBM Spectrum Scale storage when IBM Spectrum Archive is used.
>
> https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?infotype=an&subtype=ca&appname=gpateam&supplier=897&letternum=ENUS217-239

Archiving with IBM Spectrum Scale and IBM Spectrum Archive EE is flexible and easy. The product archives and recalls files with no administrator operations, while it provides an optimized total cost of ownership.

For more information, see the following web page:

https://www-01.ibm.com/support/docview.wss?uid=tss1wp102504

8.5 IBM Cloud Object Store

You can combine IBM Cloud™ Object Storage with IBM Spectrum Scale through the Transparent Cloud Tier (TCT). When you use TCT, Cloud Object Store can be used as a colder tier, it can be an on-site Cloud Object Store, and a remote Cloud Object Store.

For more information, see the following web page:

https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WUS12361USEN

8.6 IBM Cloud Private (ICP)

You can use ESS to provide IBM Spectrum Scale storage in IBM Cloud Private, as explained on the following web page:

https://medium.com/ibm-cloud/ibm-spectrum-scale-with-ibm-cloud-private-8bf801796f19

For a more generic approach to IBM Spectrum Scale and Containers, you can look at the presentation on the following web page:

http://files.gpfsug.org/presentations/2017/NERSC/Dean-Ubiquity-usergroup2017(1).pdf

For information on persistent storage in containers, see the following web page:

https://www.ibm.com/blogs/systems/containers-persistent-storage-ibm/

You can also check the Ubiquity Storage Service for Containers Ecosystems on the following web page:

https://github.com/IBM/ubiquity

Related publications

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

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Some publications in this list might be available in softcopy only.

- Introduction Guide to the IBM Elastic Storage Server, REDP5253
- *Implementing IBM Spectrum Scale,* REDP5254
- *IBM Spectrum Scale: Big Data and Analytics Solution Brief*, REDP5397
- *IBM Spectrum Scale Best Practices for Genomics Medicine Workloads*, REDP5379

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:

- Elastic Storage Server (ESS) Version 5.3.1 IBM Knowledge Center:
  https://www.ibm.com/support/knowledgecenter/en/SSYSP8_5.3.1/sts531_welcome.html
- IBM Spectrum Scale Version 5.0.2 IBM Knowledge Center:
  https://www.ibm.com/support/knowledgecenter/en/STXKQY/ibmspectrumscale_welcome.html
- IBM Spectrum Protect Blueprints:
  https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Storage%20Manager/page/IBM%20Spectrum%20Protect%20Blueprints
- IBM Cloud Object Storage and IBM Spectrum Scale integration:
  https://www-01.ibm.com/common/ssi/cgi-bin/ssialias?htmlfid=WUS12361USEN
- IBM Cloud Private and Elastic Storage Server integration:
  https://medium.com/ibm-cloud/ibm-spectrum-scale-with-ibm-cloud-private-8bf801796f19
- A generic approach to IBM Spectrum Scale and Containers:
  http://files.gpfsug.org/presentations/2017/NERSC/Dean-Ubiquity-usergroup2017(1).pdf

Help from IBM

IBM Support and downloads
ibm.com/support

IBM Global Services
ibm.com/services

REDP-5487-00

ISBN 0738457418

Printed in U.S.A.