A Guide to the IBM Clustered Network File System

Discusses the technical architecture

Explains installation and configuration

Includes administration and operation information

Ira Chavis
Dave Coutts
Bob Demkowicz
Jay Huie
Shuzheng Liu
Sheryl Qualters
Daniel Turkenkopf

Redpaper

ibm.com/redbooks

International Technical Support Organization

A Guide to the IBM Clustered Network File System

November 2010

REDP-4400-01

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

Second Edition (November 2010)

This edition applies to the IBM Clustered Network File System package.

Contents

Notices

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
BladeCenter®
Calibrated Vectored Cooling™
DPI®
eServer™
Global Business Services®
GPFS™
IBM®
PowerExecutive™
PowerPC®
Redbooks®
Redpaper™
Redbooks (logo) ®
System p®
System Storage®
System x®
Tivoli®
TotalStorage®
WebSphere®

The following terms are trademarks of other companies:

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

InfiniBand, and the InfiniBand design marks are trademarks and/or service marks of the InfiniBand Trade Association.

AMD, the AMD Arrow logo, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

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

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

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

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

Preface

The Clustered Network File System (CNFS) is a capability based on IBM® General Parallel File System (GPFS™) running on Linux® which, when combined with System x® servers or BladeCenter® Servers, IBM TotalStorage® Disk Systems, and Storage Area Networks (SAN) components, provides a scalable file services environment. This capability enables customers to run a General Parallel File System (GPFS) data-serving cluster in which some or all of the nodes actively export the file system using NFS.

This IBM Redpaper™ publication shows how Cluster NFS file services are delivered and supported today through the configurable order process of the IBM Intelligent Cluster. The audience for this paper includes executive and consultant decision makers and technical administrators who want to know how to implement this solution.

The team that wrote this paper

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

Ira Chavis is a Certified Consulting IT Specialist in the Advanced Client Technology (A.C.T!) Centers in the IBM Systems and Technology Group (STG). Working in the STG Center for Solution Integration (CSI), he currently specializes in infrastructure architecture and solutions involving IBM server and storage technologies. He has over 28 years of diversified software engineering and IT experience. Prior to working at IBM, Ira worked at Digital Equipment Corporation in varying assignments. Ira holds certifications as an IBM eServer™ Certified Expert in System x, IBM Grid Technical Sales, Microsoft® Certified System Engineer (NT4), and Red Hat Certified Technician.

Dave Coutts is a Senior Technical Staff Member and the Systems and Technology Group Automotive/Industrial Chief Technology Officer (CTO). As CTO, Dave focuses on taking advantage of server and storage offerings from IBM into Selected Business Solutions for the industrial sector and on communicating requirements back to strategy and development organizations. Key to this role are the partnerships with SWG, GBS, GTS, and S&D peers around the globe. Prior to his CTO role, Dave was a Co-Director of the Server Systems Institute, a Core Team member of the Systems Technology Outlook, and he led many STG transformational activities based on his server development expertise. In 2003, Dave helped launch the Poughkeepsie affiliate of the IBM Academy of Technology as the initial chairperson, and in 2004 he was elected to the IBM Academy of Technology.

Bob Demkowicz is an ITIL® Certified Infrastructure Architect with the Center for Solution Integration team in Poughkeepsie, NY. He has 28 years of experience at IBM. This experience includes development from circuit design through system integration, device driver design, and test and various project management aspects of IBM business. Bob was the I/O Technical Chief Engineering Manger for IBM System p® cluster interconnect prior to his joining of the CSI team. His expertise includes distributed computing systems, Storage Area Networks (SANs), server I/O subsystems and adapters, RISC processor design, server I/O performance, and high-performance cluster interconnect. He has additional experience with establishing vendor programs with emphasis on inter-company alliances. He is now involved in designing and architecting business/infrastructure solutions and working with industry leaders to define reference architectures.

Jay Huie is an IT Infrastructure Specialist focused primarily on integrating Linux deployments into enterprise environments. During his career at IBM, he has worked with multiple product families and has had extensive experience designing, deploying, testing, and auditing enterprise architectures. His current technical work covers clusters and grids, and managing IT environments. Of particular importance is the integration of business requirements and IT capabilities, by assisting clients by making tactical and strategic recommendations for their IT infrastructure. Jay has a degree in Computer Engineering from Case Western Reserve University.

Shuzheng Liu is a Software Engineer in the GPFS team in the United States. He has eight years of experience in the functional verification testing field. His areas of expertise include GPFS, NFS, CNFS, and CIFS (Samba). He holds a Ph.D. in physics and an master's degree computer science from Southern Illinois University Carbondale.

Sheryl Qualters is a Project Manager and Certified PMP in the Advanced Client Technology (A.C.T!) Centers in the IBM Systems and Technology Group (STG). With over 20 years of experience at IBM, Sheryl is currently responsible for project management of solutions at the STG Center for Solution Integration (CSI) in Poughkeepsie, NY. Her previous assignments included working at the System p benchmark center in Poughkeepsie, NY, and various positions at IBM in Rochester, MN.

Daniel Turkenkopf is a Solutions Architect in the IBM Design Center for IT Optimization and Business Flexibility located in Poughkeepsie, NY. He leads collaborative design sessions with clients to effectively take advantage of IBM technology in their infrastructure. His areas of expertise include service-oriented architectures and systems management. Prior to this role, Dan was a J2EE Application Architect with IBM Global Business Services® Public Sector practice, where he was responsible for the user interface tier of a multi-billion dollar modernization effort for a federal agency. His product experience includes WebSphere® Application Server, WebSphere Portal Server, and Tivoli® Provisioning Manager. Dan holds a bachelor's degree in Mathematics and a bachelor's degree in Economics with a concentration in Management and Information Systems from the University of Pennsylvania.

Acknowledgements

The team who wrote this guide would like to thank the following people for their time and efforts:

Stuart Alexander, IBM Raleigh
Srihari Angaluri, IBM Raleigh
Bob Curran, IBM Poughkeepsie
Tomer Perry, IBM Poughkeepsie
Marc Eshel, IBM Almaden
Manoj Naik, IBM Almaden
Sven Oehme, IBM Germany
Art Wieboldt, IBM Dallas

Mike Ebbers
International Technical Support Organization, Poughkeepsie Center

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBM-Redbooks

- 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 Clustered Network File System

Creating, managing, retrieving, and protecting data is a critical requirement of every business today. For many businesses, however, data is both an asset and a liability as they struggle with the rapid growth in content.

IBM Global Technology Services offers the following analysis:

> The efficient management of data is an ongoing struggle between access and scalability. Providing access to file-level data—the files associated with individual documents, multimedia files, database and other applications—becomes more difficult as more users are provided access and more data is stored. Achieving the scalability needed to respond to data volume growth also typically results in higher hardware and software costs, and greater management challenges.[1]

This chapter provides an overview of the IBM Clustered Network File System (CNFS).

[1] http://www-935.ibm.com/services/us/its/pdf/sofs-wp-gtw00939-usen-01-102307.pdf

1.1 Overview

Network-attached storage (NAS) solutions provide simplicity, manageability, and access, but until now they have lacked scalability and cost-effective fault tolerance, which has kept them from serving a role beyond the departmental level.

The response to these challenges is an IBM offering based on proven clustered file-serving technology in the form of the General Parallel File System (GPFS). GPFS is a high-performance, shared-disk, clustered file system developed by IBM. GPFS distinguishes itself from other cluster file systems by providing concurrent high-speed file access to applications running on multiple nodes. In addition to providing file system storage capabilities, GPFS provides tools for management and administration of the GPFS cluster and allows for shared access to file systems from remote GPFS clusters. It also supports new information life-cycle management features to simplify data management and enhance administrative control.

Most data types and application areas are suited for this approach. The key benefit of this approach is enhanced scalability, performance, and availability for NFS clients. This approach is suited especially to the industry sectors, including automotive, financial, electronic design, communications, and pharmaceuticals.

1.2 IBM Clustered Network File System

Figure 1-1 illustrates a typical CNFS.

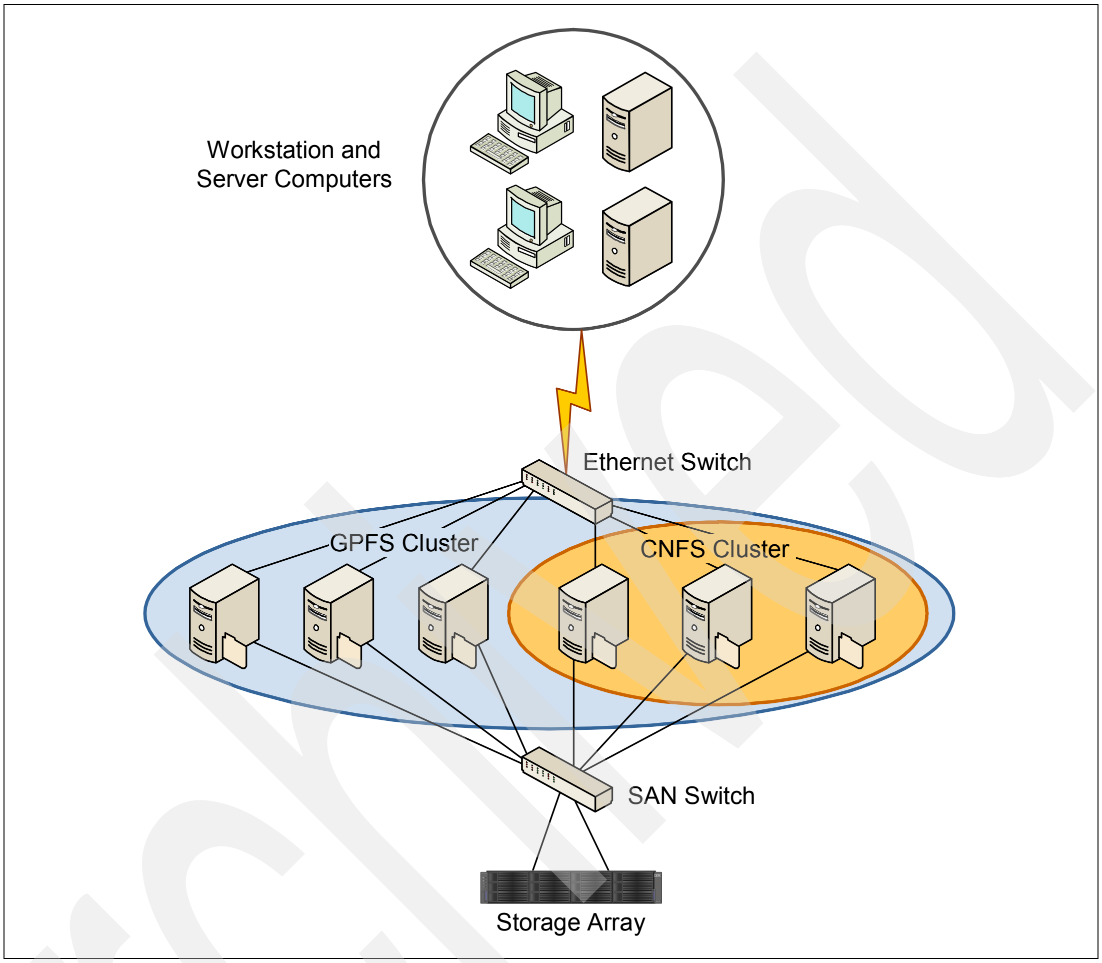

Figure 1-1 Clustered Network File Server

The CNFS provides a scalable file services environment with the following characteristics:

- Highly availability
- High performance and scalability
- Low-cost entry and incremental growth
- Compatibility with NFS clients

This capability enables customers to run a GPFS data serving cluster in which some or all of the nodes actively export the file system using NFS. All of the nodes in the cluster exporting the same file system to NFS clients collectively provide the CNFS. These nodes are a sub-cluster of the GPFS cluster and are referred to as CNFS *cluster members*.

Unlike other commercial file-serving applications, CNFS is designed to scale with both CPU and I/O capacities to address a range of customer file-serving applications. An overview of current business trends and challenges can help you understand why customers are interested in the CNFS.

1.3 Business trends and line of business requirements

Creating, managing, retrieving, and protecting data is a critical requirement of every business today. For many businesses, however, data is both an asset and a liability as they struggle with the rapid growth in content. While many factors contribute to data growth, three key trends are significant contributors:

- Digital representation of physical systems and processes
- Capture of digital content from physical systems and sources
- Deliveries of digital content to a global population

These trends are driven by the following kinds of applications:

- Product life cycle management (PLM) systems, which include product data management (PDM) systems and mechanical, electronic, and software design automation
- Service life cycle management (SLM) systems
- Information life cycle management (ILM), including email archiving
- Video on demand: Online, broadcast, and cable
- Digital video surveillance (DVS): Government and commercial
- Video animation rendering
- Seismic modeling and reservoir analysis
- Pharmaceutical design and drug analysis
- Digital health care systems
- Web 2.0 and service-oriented architecture

The following sections illustrate specific business examples, which provide additional insight.

1.3.1 Product life cycle management for the automotive industry

Automotive manufacturers today face significant challenges. Competitive pressures are leading to an increased number of new automobile model launches in shorter development times. Differentiation to reach new and younger buyers is often delivered through electronics and embedded software systems. Pressures to reduce development and capital expense mean that manufacturers will soon not be able to afford to verify the growing number of automotive models in the physical world. Additionally, lost revenue due to quality and warranty expense demands better processes.

In response to these challenges, manufacturers are turning to virtual system design and verification prior to testing the real vehicle in metal form. In the process, they must also deliver proof of analysis to meet regulatory requirements.

Accelerating the ability to explore and solve complex physical representations in the virtual (simulated) domain involves the following key systems:

- Data management (the metadata that captures the pedigree of information, that is, the *who*, *what*, *when*, and *where* data)
- Design representation (for example, geometry)
- Models for analysis (meshed and assembled models with load case applied for Computer Aided Engineering (CAE))
- Storing of analysis results (from the CAE solvers running on high-performance computing clusters)

While the first system in this list is largely a database topic, all four involve content stored in file systems. Users of this content can be:

- Designers using workstations
- Analysts using Windows® or Linux clients
- High-performance computing (HPC) clusters retrieving data for analysis or returning results
- Product data management or simulation data management systems supporting the design and analysis environment
- Product documentation tools

1.3.2 Financial services

There is a growing trend within financial services industries to develop grid-style systems where there is significant sharing of data among the nodes of the grid and throughout grids. The most prevalent applications for this type of computing are various forms of analytics in financial markets and insurance actuarial departments. The aggregate number of compute systems that require the data can be large, but the data requirements of any individual system might be relatively modest. It is more economical to use the NFS client that usually is provided with the operating system than it is to install a specialized client, such as GPFS, throughout the entire set of application nodes.

This approach serves well for applications that do not have a large write component in their storage I/O mix and do not have data performance needs that exceed the capabilities of NFS. Specific environments that might meet these needs within the financial sector include:

- Trading analytics that have a high compute to data ratio but need shared data across a grid.
- Insurance analytics that have similar compute and data characteristics to a trading program.
- Branch optimization environments that assign branch or contact center workers to available compute resources at a regional hub. This process can be done with various forms of client virtualization. You then can move any data resources to the compute resource that is assigned to the branch worker on a given day. The data requirements of a single worker can be quite modest, but the aggregate data rates can be quite high and can require fault tolerance characteristics.

1.3.3 Electronic design automation

Electronics and semiconductor companies are faced with the following ongoing design environment challenges to keep pace with consumer demand for more robust and sophisticated electronics devices:

- Integrating more electronic features into ever-shrinking silicon chips
- Improving engineering productivity and global design collaboration
- Improving design turnaround time and time-to-market, with typically less than six to nine months between product releases
- Keeping engineering and manufacturing costs low and under control

To meet these challenges, companies are experiencing significant and dramatic increases in electronic design automation (EDA) compute capacity (often doubling every 18 to 24 months), design team sizes, and data volumes. Global design teams need high capacity, high bandwidth, and highly available concurrent access to data that is spread throughout compute farms around the enterprise.

To meet the data sharing requirements, it is desirable and economical to use the standard Linux NFS client delivered in a scalable and highly available manner with a global namespace using CNFS and the underlying General Parallel File System. The implementation can be customized and tuned to optimize for diverse types of EDA workloads. For example:

- System (hardware and software) functional verification workloads typically involve a large number of small files, perhaps 100 MB files with an average size of 8 KB generated by thousands of engineers. Small file write performance and storage optimized for I/O processors (IOPs) performance are critical. Verification jobs are redefined continuously and reprioritized based on prior results with a heavy dependency on feeding the correct data to the correct compute resource at the correct time without bottlenecks.
- Physical design and tape-out workloads typically involve a high compute to I/O ratio and large files. Large file read/write performance and storage optimized for streaming I/O is important.

1.3.4 Communications and digital media

The amount of data and the number of objects that need to be stored are exploding with the move from analog to digital content. Today, most customers in this sector have the need to read and write data with bandwidths beyond 10 Gbps, and it is common that they have to achieve this read/write performance from or to a single file from multiple sources or destinations, respectively.

A most advanced single filer is not able to handle this workload well. Its single-file performance is limited by the number of disks available for striping data and ultimately by the number of data storage devices at the back-end system. To meet such performance requirements, a solution must be able to spread a single file across as many disks as possible and as many controllers or nodes as possible. A single device is not able to handle such high I/O workload.

Alternatively, customers in this sector might have the need to transfer tens of millions of small objects that could potentially fit on a single storage device. In such a scenario, a filer architecture shows fundamental bottlenecks, and several filers combined are needed to meet the throughput requirements. Unfortunately, such an approach introduces difficulties with respect to backup performance and maintainability of such a huge number of objects. In general, this huge number of objects is split across various namespaces on several filers, thereby introducing management overhead for the operations team and not optimally utilizing the available resources of the total system because of an unbalanced I/O demand across the filers.

1.3.5 Pharmaceuticals

Most large-scale pharmaceutical customers need to store a wide variety of data, sometimes for 30 years or longer, to comply with various government regulations. Today, the data is spread across many files across multiple filers. The reason for such a solution design is not a lack of individual filer capacity, but more the challenge of managing these numbers of files effectively in a single filer system, integration with long-term archival capabilities, and being prepared for disasters and other requirements such as data searches.

1.3.6 Other industries

Aerospace and defense, chemical and petroleum, pharmaceutical, health care, digital media systems, and other areas have similar requirements, and each area points to the rapid growth in file-based data. According to Yankee Group (January 9, 2007), file-based data is already 65 - 80% of all data and is growing 50 - 70% per year. At the 70% rate, 1 exabyte of file-based data would grow to 14 exabytes in five years.

1.4 Challenges and customer requirements

According the IBM Global Technology Services document *NAS systems scale out to meet growing storage demand*, the challenges of file data management are described as follows:

> The efficient management of data is an ongoing struggle between access and scalability. Providing access to file-level data—the files associated with individual documents, multimedia files, database and other applications—becomes more difficult as more users are provided access and more data is stored. Achieving the scalability needed to respond to data volume growth also typically results in higher hardware and software costs, and greater management challenges.[2]

NAS solutions provide simplicity, manageability, and access, but until now they have lacked the scalability and cost-effective fault tolerance that has kept them from serving a role beyond the departmental level.

Emerging NAS systems, and managed storage services that are powered by these technologies, offer compelling business value to organizations because of their broad spectrum of applicability, competitive price and performance, and scalability to meet growing and evolving data demands.

The problem is that current solutions fall short of requirements. NAS appliances and simple file servers have many limitations, which include:

- Current NAS solutions do not scale. In some cases file systems are limited in capacity, forcing customers to add capacity server by server. The primary issue is scalable delivery of data to consuming applications.
- Simply adding NAS appliances leads to data fragmentation, which occurs by assigning separate portions of the file name space to each NAS space. While automounters help in the steady state, finding the server that has the file you need, the problems arise with fragmented management across the servers. Capacity management becomes a problem

[2] http://www-935.ibm.com/services/us/its/pdf/sofs-wp-gtw00939-usen-01-102307.pdf

and doing anything that rebalances is difficult. This fragmentation complicates workflow, regulatory compliance, and ILM requirements. There is no way to apply policies across these independent data islands.

- Adding more and more NAS appliances leads to a management nightmare, as file systems have to be stitched together as though they were a unified whole.
- Collections of NAS appliances leads to hot spots and poor performance. NAS appliances with the most in-demand files are pegged while others are idle. There is no easy way to balance the performance load.
- Isolated direct-attached or SAN-based disk leads to underutilization, as disks allocated to one server or NAS appliance cannot be used by others. The resulting total utilization can be as low as 15%.

Today, customers want their NAS systems to:

- Grow capacity dynamically.
- Provide a single, global view into all content.
- Increase bandwidth to data and reduce latency where required.
- Manage data across the life cycle.
- Provide consistent access to data with minimal down time.

The result is a requirement on aggregate bandwidth and I/O rates that have been unique to HPC environments before. The challenge has been delivering this capability to the general user in a simplified way. Making file servers clustered is easy in theory, but hard to build and maintain because there is too much complexity. The need for this increases, however, as you consider other emerging factors such as:

- Migration, integration, or removal of storage for file services is difficult, so many traditional systems exist and need to be maintained.
- Backup windows are a big issue and get worse while the amount of data continuously grows.
- Integration of ILM functions into the file system becomes more important as the amount of terabytes radically increases.

1.5 Solution elements and key features

The response to the challenges that we discussed in the previous section is an offering based on proven clustered file serving technology in the form of IBM General Parallel File System (GPFS). GPFS is a high-performance, shared-disk, clustered file system developed by IBM. GPFS distinguishes itself from other cluster file systems by providing concurrent high-speed file access to applications running on multiple nodes. In addition to providing file system storage capabilities, GPFS provides tools for management and administration of the GPFS cluster and allows for shared access to file systems from remote GPFS clusters. It also supports new information life cycle management features to simplify data management and enhance administrative control. GPFS has been available on Linux since 2001 and was first offered on AIX® in 1998 for IBM Scalable Parallel systems.

Drawing upon years of scalable file serving for the HPC space, this technology forms the core of a CNFS file serving offering that is much easier to build, maintain, and deliver for the general user. In addition to the GPFS file management and locking capabilities, enhanced monitoring, failover, and load balancing have been added to bring scalable, highly available GPFS capabilities to the NFS user.

The primary elements of the IBM CNFS file system are as follows:

- IBM General Parallel File System and CNFS member nodes
- Linux distribution (Red Hat and SUSE)
- IBM BladeCenter and System x servers
- IBM TotalStorage Disk Systems
- SAN switches
- System management

Key features include:

- High availability
- High performance and scalability
- Low-cost entry and incremental growth
- Compatibility with NFS clients

See 3.2, “Recommended CNFS initial base configuration” on page 20, for more details about this configuration.

1.6 Targeted users

Most data types and application areas are suited for this approach. The key benefit of this is enhanced scalability, performance, and availability for NFS clients. This approach is especially suited to the industry sectors that we mention in 1.4, “Challenges and customer requirements” on page 7.

From a general application perspective, HPC applications require server processing power, but they also typically have high-volume, high-performance storage demands. Usually, these applications analyze data or model results, need a large amount of storage to run, and can generate huge data volumes quickly. However, the need for CNFS file services goes beyond HPC applications.

Providing centralized storage for files, a place where users can find data easily, is growing in importance when you consider the collaborative nature of many activities. Breaking down the barriers to secure access to data is an issue throughout organizations. CNFS offers the scale and performance to support more users and more and bigger files simply. Starting with GPFS 3.1, CNFS also enables enhanced Information Lifecycle Management capabilities.

Finally, many organizations are undertaking advanced initiatives, such as service-oriented architecture (SOA), to enable them to connect disparate systems into operational structures or to break apart inflexible system structures (often fixed application interconnects) into more flexible and diverse offerings. Often, the first step in these initiatives is the consolidation of dispersed resources to fewer, yet more scalable, systems. CNFS supports consolidation by centralizing data to a single file system and streamlining management tasks associated with saving, moving, and accessing files. It also allows applications dispatched on these infrastructures to find the data required via the global namespace of the single file system. Scalability and high availability are key advantages for SOA environments.

1.7 Required skills for the reader of this paper

This guide assumes that the reader has an understanding of the following products or technologies:

- Linux (Red Hat or SUSE), including NFS
- IBM System x and BladeCenter servers
- IBM General Parallel File System
- Storage concepts, including storage area networks
- Networking
- Systems management frameworks such as xCAT

2

Clustered NFS technical architecture

This chapter discusses the architecture of the Clustered Network File System (CNFS) solution and how it provides system availability.

2.1 Architecture

Traditionally, NFS storage architecture is client-server based with a single server that provides data services for a large number of distributed clients. The central server's storage capacity is usually located on a collection of internal disks or otherwise directly attached storage, and the availability features are built around hardware or procedural capabilities, such as RAID or tape backups. Figure 2-1 illustrates a classic NFS architecture.

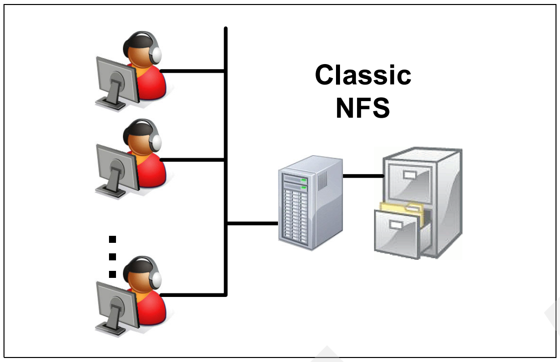

Figure 2-1 Classic NFS

This style of deployment provides a centralized point from which to manage and enforce an environment's data policies and affords an IT environment with the greatest initial capabilities for the least amount of complexity. Operationally, however, such a centralized approach might suffer from difficulty in avoiding constraints, primarily limited storage and network bandwidth. As the number of clients scales up, it can be difficult to maintain acceptable performance levels. Although this architecture provides centralized management, it often leaves an IT environment at the mercy of a single point of failure. In the event of a server outage, even if the data has not been corrupted, service cannot continue until this single access point has been restored.

Techniques that exist for increasing the redundancy and parallelism of this environment typically resolve around practices such as multiple I/O adapters and other more complicated data services. However, even with these components in place, without a corresponding increase in network capacity, a bandwidth bottleneck to the clients still exists. Furthermore, none of these practices solve any availability concerns that exist because of the reliance on a single server.

Because a continued focus and need have been placed on evolving data management practices, these techniques have fueled a better understanding and implementation of solutions to scale performance and availability. More advanced solutions to build clustered file systems, such as General Parallel File System (GPFS), parallelize and distribute the storage and availability requirements across multiple servers, or nodes, in an environment.

GPFS can abstract the underlying storage layer and provide a common, unified file system across multiple distributed nodes. In conjunction with accelerating I/O capability, this distributed environment also increases the availability and recoverability of an infrastructure's data because the data is not wholly dependent on the availability and storage capability of a single node. One drawback to such an approach is the requirement that all nodes that need access to the data utilize the GPFS product suite. Another is the complexity in managing the configuration for every node.

In our approach, a hybrid solution is utilized so that GPFS nodes can participate in a GPFS cluster but also be used to serve data in a more *client-server* style manner to distributed systems. These client systems do not need to participate in the GPFS cluster, but can instead utilize a more ubiquitous protocol such as NFS. To avoid falling prey to the same network bandwidth and availability weaknesses that a centralized model has, these CNFS member nodes can be used to load balance the I/O requests and provide failover capabilities between the nodes (Figure 2-2).

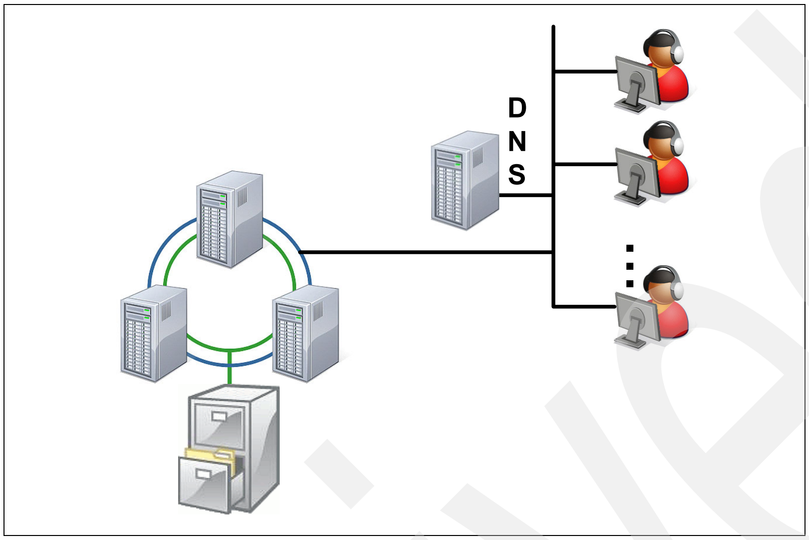

Figure 2-2 Clustered NFS

The CNFS features extend GPFS to keep the data between the CNFS nodes synchronized and also to manage lock protection and IP failovers for the NFS cluster. CNFS does this by utilizing and monitoring the standard protocol stacks and by providing features to manage these protocols.

By taking advantage of these standard components, IT administrators deploying a CNFS solution benefit from the long history of quality that these existing software stacks provide.

We depict the components and describe their interrelationship next.

To the NFS client, the components represent a single front that provides NFS capabilities. In reality, the many components communicate internally to provide this functionality. Figure 2-3 illustrates the components.

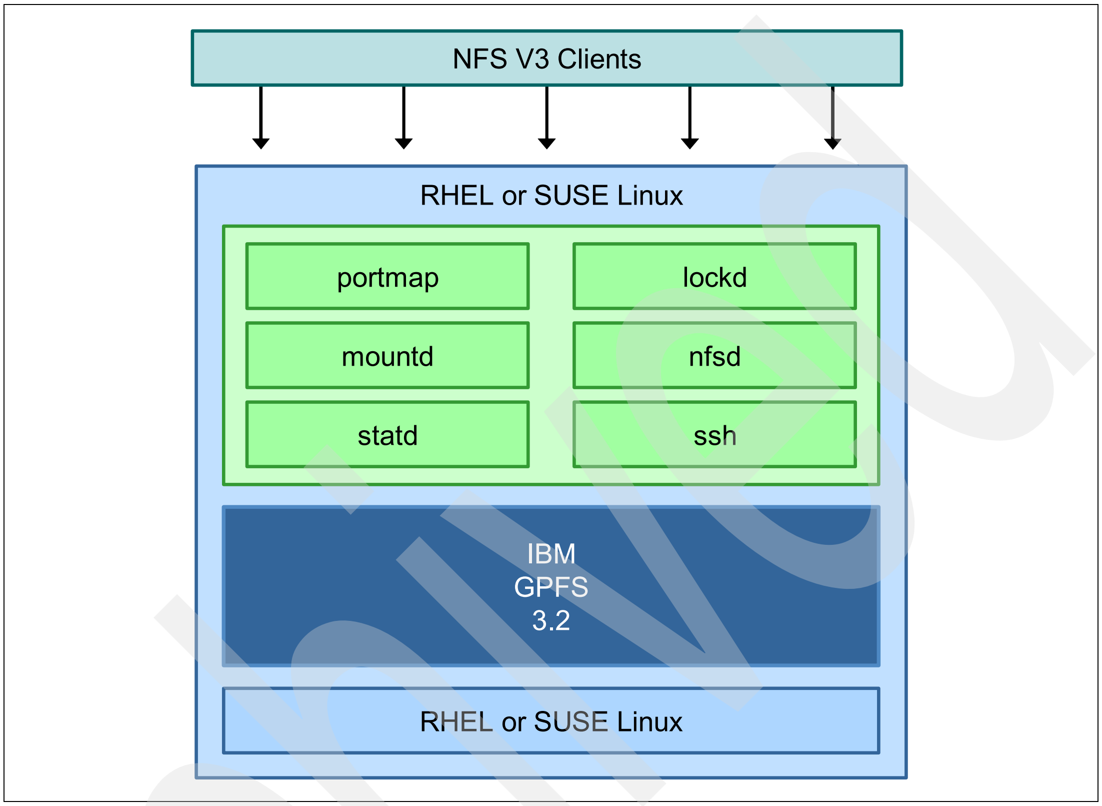

Figure 2-3 Component layers

The components include:

- Portmap
  - Maps between a remote procedure call and a standard DARPA port.
  - Required for any service that uses RPC (including support for existing NFS clients).
  - Each service that uses RPC must first register with the service, and then clients can get the IP port mapping by querying the portmapper.
- nfsd
  - Provides the actual transport service for NFS
  - Multi-threaded kernel process (using nfsd.ko module)
  - Invoked by **`rpc.nfsd`** <*`no. of thread`*>
- mountd
  - Implements the nfs mount protocol
  - Gets a mount/unmount request from a client and checks whether the request is valid
  - Provides the client with a *file handle* and records the activity in **`rmtab`**

- statd
  - Implements Network Status Monitor protocol
  - Works with lockd to implement locking
  - Implemented as kernel process in SUSE (adds sm-notify for lock reclamation)
  - By default, records clients with locks in `/var/lib/nfs/sm`
  - Sends lock reclaim requests to clients in case server reboots
- lockd
  - Also known as the Network Lock Manager (NLM)
  - Works with **`statd`** to manage NFS locks
  - Stores lock information in kernel memory
  - Notifies **`statd`** of locks, so **`statd`** can record the client on disk if needed

By taking advantage of the distributed power and availability of GPFS, along with the ubiquity of network file sharing protocols like NFS, the best-of-both-worlds approach of CNFS can be achieved.

2.2 Availability features

Maintaining availability of the exported file systems is one of the key features of CNFS over and above traditional NFS. Availability is demonstrated through two major capabilities:

- Load balancing, which spreads the traffic over multiple systems, allowing the cluster to scale to meet capacity requirements
- Failover, which allows the cluster to recover from the failure of a given node without affecting the client

2.2.1 Load balancing

While there are many options for load balancing network traffic, currently CNFS only supports round-robin DNS. Under a round-robin DNS scheme, a single host name is associated with multiple IP addresses. The first request that the DNS server receives for that host name is routed to the first IP address in the list. A second request returns the second IP address. This process continues for every request that the DNS server receives, looping back to the first address when the end has been reached. For more details about configuring a round-robin DNS setup using BIND, see 4.3.1, “Load balancing” on page 29.

2.2.2 Monitoring and failover

Every member in the NFS cluster has a utility that monitors the CNFS components on the node, including the core GPFS later and the network and rsh/ssh utilities. Upon failure detection, and based on customer configuration, the monitoring utility can invoke a failover command.

In the event of a node outage, the CNFS environment maintains availability by redirecting traffic from a failed node to another functioning node. The primary mechanism for this is TCP/IP address takeover with NFS lock recovery, and the impact to the client is minimal or nonexistent, but varies by application. There are three primary ways in which a cluster member can fail:

- The network communication between the client and the serving node drops.
- The NFS daemon fails.
- The GPFS subsystem itself fails.

The most likely outage is an NFS daemon error, and in this case a CNFS failover halts the GPFS daemon as though the subsystem has failed. This is done to use the existing GPFS clustering infrastructure. A GPFS failure, whether actual or induced, is managed by the overall GPFS cluster, including lock and state recovery. Then, as part of the GPFS recovery, the NFS cluster failover mechanism is invoked. The NFS cluster failover script transfers the NFS load that was served by the failing node to another node in the NFS cluster.

The NFS node failover steps are as follows:

1. The NFS monitoring utility detects an NFS-related failure.
2. The NFS monitoring utility stops NFS serving and fails (that is, kills) the GPFS daemon.
3. The GPFS cluster detects the GPFS node failure. All of the clustered NFS nodes enter a grace period to block all NFS client lock requests.
4. The GPFS cluster completes recovery, including the release of any locks held by the failing node.
5. The NFS cluster moves the NFS locks from the failing node to another node in the cluster and invokes NFS recovery.
6. The NFS cluster performs IP address takeover (including the sending of gratuitous ARPs).
7. The NFS cluster notifies all relevant NFS clients to start lock reclamation.
8. Clients reclaim locks according to NFS standards.
9. At the end of the grace period all operations return to normal.
10. In certain scenarios, a condition can exist where GFPS local locks issued during the grace period might not be recognized by NFS during the reclaim process.

In the event of a network failure between the client and the NFS server, standard NFS timeout mechanisms occur to free any locks held by the client. If an outage is transient, the client might not notice any effect, primarily depending on the manner in which the NFS export was mounted (for example, hard or soft).

3

Clustered NFS and the IBM Intelligent Cluster

A Clustered Network File System (CNFS) environment is based on an IBM Intelligent Cluster offering. CNFS is a part of the IBM GPFS software option of the IBM Intelligent Cluster. This chapter describes the IBM Intelligent Cluster hardware and software options and illustrates a starting configuration for a CNFS cluster that can be expanded and customized to accommodate client requirements. It also addresses how to order this configuration and what support options are available.

3.1 IBM Intelligent Cluster

An IBM Intelligent Cluster (Figure 3-1) is an IBM integrated cluster that is sold as a complete hardware and software system which is comprises pre-tested IBM and OEM components to meet the needs of customers with high-performance computing (HPC). The IBM Intelligent Cluster provides customers a low risk fully supported solution that translates into a single point of contact for all hardware, including third-party switches, networking, and terminal server components. Software components include system management and clustered file system support. An IBM Intelligent Cluster also reduces the cost of acquiring computing capability by delivering integrated hardware and software. This integration further optimizes deployment time and reduces complexity and management costs.

For more information about the IBM Intelligent Cluster, see:

http://www-03.ibm.com/systems/x/hardware/cluster/index.html

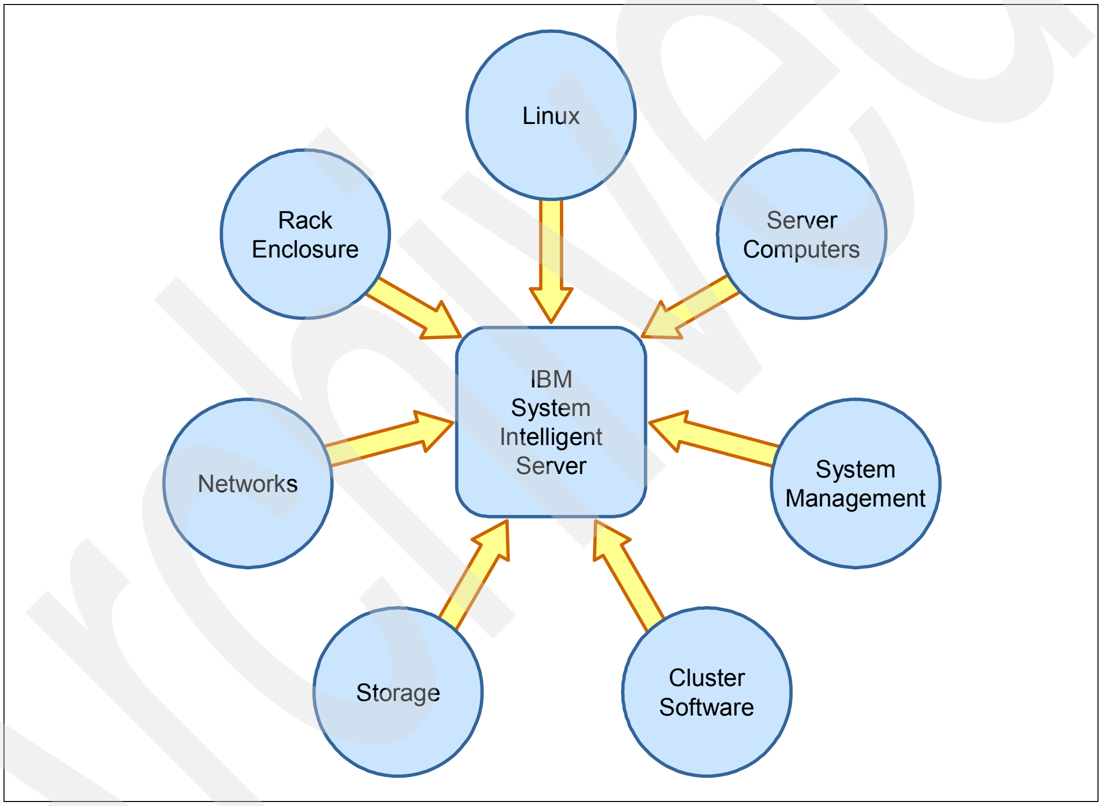

Figure 3-1 IBM Intelligent Cluster overview

IBM Intelligent Cluster features and components summary

The IBM Intelligent Cluster includes the following features and components:

- Rack optimized and BladeCenter servers based on the Intel® Xeon, AMD and PowerPC® processors
- High capacity IBM System Storage® System Storage and Storage Expansion Units
- Gigabit Ethernet cluster interconnect
- Terminal server and KVM switch
- Space-saving flat panel monitor and keyboard
- Support for Red Hat or SUSE Linux operating systems

- Hardware installed and integrated in 25U or 42U Enterprise racks
- Software options:
  - Robust cluster systems management such as Cluster Systems Manager and xCAT
  - Scalable parallel file system software such as General Parallel File System (GPFS)
- Scales up to 1024 cluster nodes (larger systems and additional configurations available—contact your IBM representative or IBM Business Partner)
- Optional installation and support services from IBM Global Services or an authorized partner or distributor
- Clients must obtain the version of the Linux operating system specified by IBM from IBM, the Linux Distributor, or an authorized reseller

For more information, consult Appendix B, “IBM Intelligent Cluster Software options” on page 59. In addition, you can find more detailed configuration and option information at:

http://www-03.ibm.com/systems/clusters/hardware/reports/factsfeatures.html

3.2 Recommended CNFS initial base configuration

The purpose of this reference configuration is to present the entry level configuration or a *starting point* that is required to implement CNFS and also to highlight the key hardware and software options of the IBM Intelligent Cluster. Figure 3-2 shows the CNFS initial base configuration.

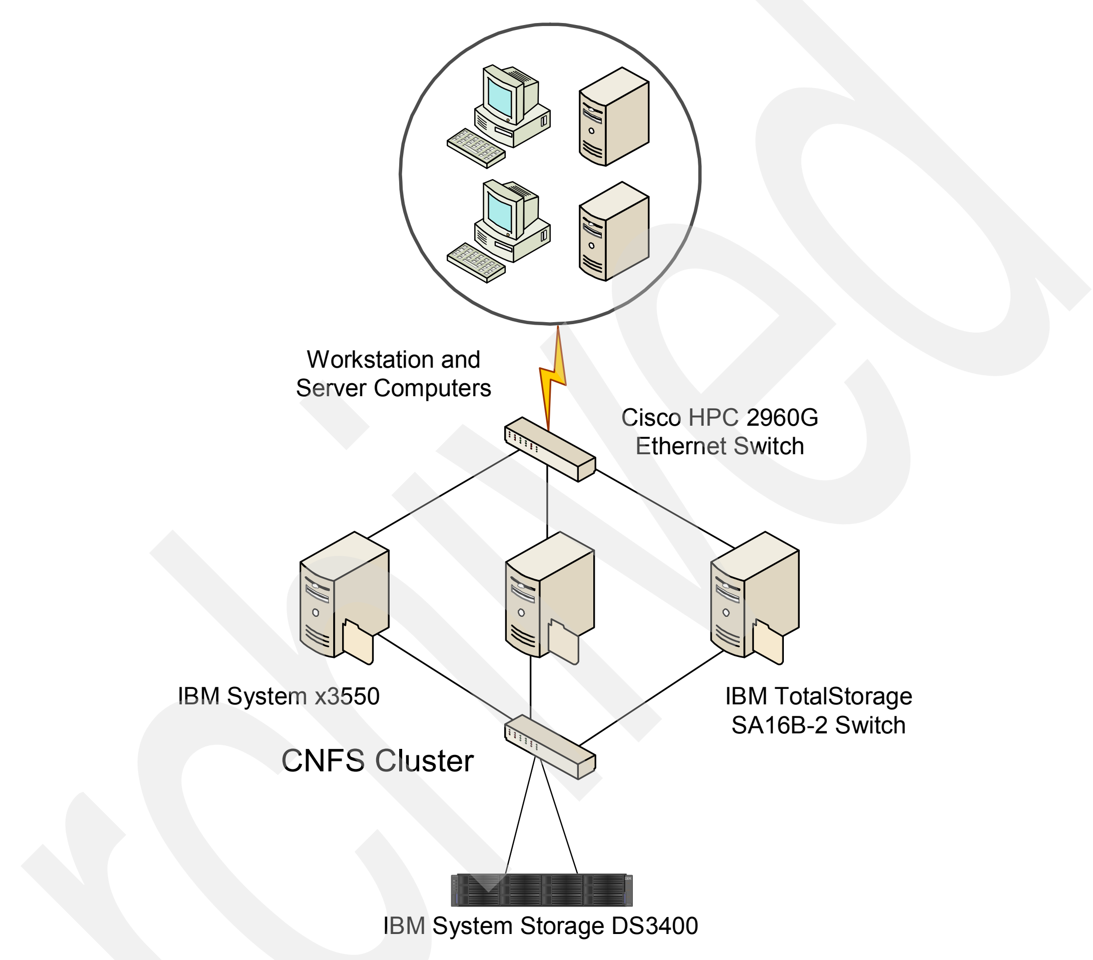

Figure 3-2 CNFS initial base configuration

3.2.1 Key features of the configuration

The key features of the recommended configuration include:

- CNFS tier featuring three IBM System x3550 CNFS cluster members for improved availability.
- Dual NICs in each storage nodes for teaming of network adapters for increased bandwidth and availability of network connections.
- Dual Fibre Channel host bus adapters in each storage node that provide multiple paths to the IBM Total Storage 3400 storage array in case of path failure.

Table 3-1 lists the details of the recommended CNFS initial base configuration.

Table 3-1 Recommended CNFS initial base configuration

Hardware	Intelligent Cluster 25U Rack Cabinet ► IBM Local 2x8 Console Manager ► LCD/Keyboard Console ► Power Distribution Units
	Three System x 3550 ► Quad-Core Intel Xeon® Processor E5345 (2.33 GHz 8 MB L2 1333 MHz 80 w) ► 5 GB Memory ► IBM ServeRAID-8k SAS Controller ► Two 146 GB RAID-1 Internal Disk ► Two Fibre Channel Ports for Availability ► PCIe Dual Port Ethernet Card
	IBM System Storage DS3400 Dual Controller Fibre Channel Storage Array ► Six 300 GB 15 K 3.5 Hot-Swap SAS HD ► 5P plus “Hot Spare”
	Cisco HPC-E 2960G 24-port Ethernet Switch Bundle
SAN	IBM TotalStorage SAN 16B-2
Software	Linux: Red Hat Enterprise Linux 5 or SUSE Linux Enterprise Server 10 and required kernel patches. See 4.1, “Prerequisites ” on page 26.
	IBM Global Parallel File System (GPFS) V3.2
	Systems Management (optional): xCat installed on one of the CNFS storage nodes

You can find a detailed configuration in Appendix C, “Clustered NFS initial base configuration bill of materials” on page 63. This configuration can be ordered and customized by your IBM Intelligent Cluster Sales Specialist to meet your infrastructure requirements.

3.2.2 CNFS: IBM Intelligent Cluster rack configuration

Figure 3-3 shows a view of the CNFS initial base configuration rack and suggests a possible placement of the hardware components.

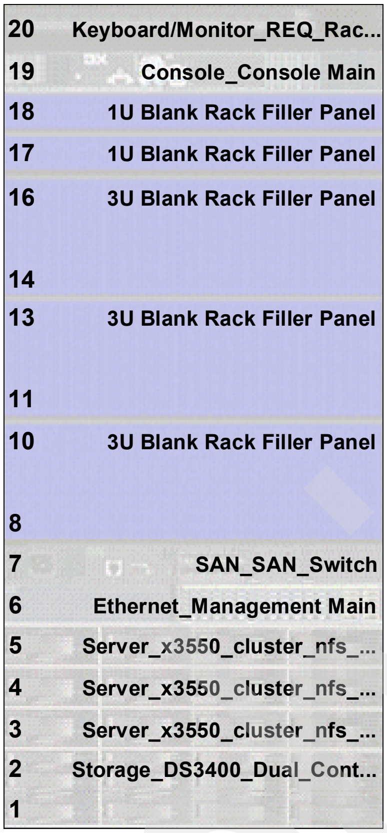
20 Keyboard/Monitor_REQ_Rac...
19 Console_Console Main
18 1U Blank Rack Filler Panel
17 1U Blank Rack Filler Panel
16 3U Blank Rack Filler Panel
14
13 3U Blank Rack Filler Panel
11
10 3U Blank Rack Filler Panel
8
7 SAN_SAN_Switch
6 Ethernet_Management Main
5 Server_x3550_cluster_nfs_...
4 Server_x3550_cluster_nfs_...
3 Server_x3550_cluster_nfs_...
2 Storage_DS3400_Dual_Cont...
1

Figure 3-3 Clustered NFS rack configuration

3.2.3 Configuration considerations

When evaluating the configuration of an IBM Intelligent Cluster, here are some items you might consider when planning your cluster:

- Scale up

  See 6.2.1, “IBM Intelligent Cluster CNFS advanced configurations” on page 44, for clusters that support scaling up to meet your infrastructure requirements.

- Multifunction IBM Intelligent Cluster

  Although the sample configuration is designed only for file serving, you can order extra storage or compute nodes for your cluster to partition your IBM Intelligent Cluster so it can provide both compute and CNFS functionality, depending on your requirements.

- Management node

  The IBM Intelligent Cluster configuration includes the option for a systems management node. See “Systems management” on page 60 for more information. For a small number of storage nodes, you can choose to host the systems management responsibility on one of your storage nodes. As the configuration grows, consideration should be given to adding a dedicated systems management node.

- SAN switches

  The initial configuration uses one SAN switch that can be divided into two fabric zones to support the use of dual controllers on the IBM TotalStorage DS3400 Storage arrays. To achieve a greater degree of availability, you can add a second SAN switch to support an independent fabric for each Fibre Channel path.

- Ethernet switch

  The initial configuration uses one Ethernet switch to support the network requirements of the CNFS cluster. For increased network availability and performance, you can add a second Ethernet switch to the IBM Intelligent Cluster configuration.

3.3 Ordering an IBM Intelligent Cluster and CNFS

You can order the IBM Intelligent Cluster by contacting your regional IBM Cluster Sales Specialist. See your local IBM sales team for more information.

3.4 Installation service and support

IBM offers a wide range of initial installation service and ongoing support options for the IBM Intelligent Cluster. The IBM Intelligent Cluster hardware installation support is included at no charge on 42U and 25U racks. Additionally, HPC cluster software services and Support Line for Linux Clusters are available as optional fee-based services. Cluster Installation Support Services are also available through the Cluster Enablement Team (CET) as optional fee-based services. Contact your IBM Cluster Sales Specialist for more information about support options.

4

Clustered NFS installation and configuration

This chapter provides a summary of the installation and configuration steps to deploy the Clustered Network File System (CNFS) components of General Parallel File System (GPFS). We assume that GPFS has been installed and configured before completing these steps.

4.1 Prerequisites

This section lists the prerequisites that must be met before you install and configure CNFS.

4.1.1 System prerequisites

The following system prerequisites must be met before you begin the installation and configuration:

- A Linux 2.6 kernel

  Distributions currently supported are Red Hat Enterprise Linux (RHEL) Versions 4 and 5 and SUSE Linux Enterprise Server (SLES) Versions 9 and 10.

- Operating system patches

  - If NLM locking is required, a kernel patch that updates the **`lockd`** daemon to propagate locks to the clustered file system must be applied. This patch is currently available at:

    `http://sourceforge.net/tracker/?atid=719124&group_id=130828&func=browse/`

    Depending on the version of SLES that you are using, this patch might exist partially. If this condition exists, you might need to resolve certain conflicts. Contact your support organization if necessary.

  - To permit NFS clients to reclaim their locks with a new server after failover, the reclaim message from **`statd`** must appear to come from the IP address of the failing node (and not the node that took over, which is the one that actually sends the message).

    On SUSE, **`statd`** runs in the kernel and does not implement the interface to support this requirement (notification only, -N option). Therefore, on SUSE, the common NFS utilities (**`sm-notify`** in the user space) are needed to implement this function.

    The patches required for the **`util-linux`** package are:

    - patch-10113: Supports **`statd`** notification by name

      `http://support.novell.com/techcenter/psdb/2c7941abcdf7a155ecb86b309245e468.html`

    - patch-10852: Specifies a host name for the **`-v`** option

      `http://support.novell.com/techcenter/psdb/e6a5a6d9614d9475759cc0cd033571e8.html`

    - patch-9617: Allows selection of IP source address on command line

      `http://support.novell.com/techcenter/psdb/c11e14914101b2debe30f242448e1f5d.html/`

      Note: SLES10 SP2 supports this requirement and does not need this patch.

  - For RHEL, use of nfs-utils 1.0.7 is required for rpc.statd fixes. See:

    `http://www.redhat.com/`

4.1.2 Network prerequisites

The following network prerequisites must be met before you begin the installation and configuration:

- A separate set of IP addresses for GPFS and NFS must be defined. These NFS addresses can be real or virtual (aliased). However, these addresses *must* be configured on the servers as static (not DHCP) and to not start at boot time. Addresses used for GPFS configuration (for example, for intra-GPFS communication) cannot be the same addresses used for NFS serving because those IP address cannot be failed over.
- All interfaces, except the public interfaces or virtual interfaces used for NFS serving, need to be configured to start automatically at network start. IP addresses used for NFS serving are configured, but not started. These addresses are started by the cluster scripts to ensure that the NFS services are ready for serving before requests are allowed.

4.2 GPFS configuration

We do not discuss GPFS configuration in this paper. However, you can find the GPFS V3.2 documents at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs23/bl1pdg10/bl1pdg1046

4.3 CNFS configuration

The components necessary for CNFS are installed by default during the GPFS installation. However, there are additional configuration steps that are required before starting the CNFS cluster. Follow these steps:

1. Create a separate GPFS location for the CNFS shared files. Configure this directory to be available (mounted) when CNFS starts. Register this directory with GPFS as the CNFS shared root directory using the following command:

   ```
   mmchconfig cnfsSharedRoot=<dir>
   ```

   The following example shows the configuration of the CNFS shared root on a GPFS directory:

   ```
   mmchconfig cnfsSharedRoot=/gpfsha
   ```

   Ideally, this location is on a file system that is separate from the one that contains the directories to be exported by NFS. If it is located on the same file system as the NFS exported directories, it is important to restrict access to the CNFS shared file directory.
2. Add all GPFS file systems that need to be exported to the `/etc/exports` file on every server. It is imperative that the file contents are the same for each server.

3. Add the nodes to the CNFS cluster.

   To add a single node, use the following command:

   ```
   mmchnode -N <node_name> --cnfs-enable –cnfs-interface=<nfs_ip>
   ```

   In this command, *node_name* is the node name for GPFS and *nfs_ip* is the node name that IP used for NFS.

   > **Note:** If *nfs_ip* is a virtual interface, then GPFS defines the virtual interface automatically. If it was defined previously, GPFS returns an error.

   The following example shows adding a node to the CNFS cluster:

   ```
   mmchnode -N c5n44 --cnfs-enable --cnfs-interface=c5n44ha
   ```

   To define several nodes together, use the following command:

   ```
   mmchnode -S mmchnode-input
   ```

   In this command, the **mmchnode-input** might look like:

   ```
   c5n33 --cnfs-enable --cnfs-interface=c5n33ha
   c5n41 --cnfs-enable --cnfs-interface=c5n41ha
   c5n42 --cnfs-enable --cnfs-interface=c5n42ha
   c5n43 --cnfs-enable --cnfs-interface=c5n43ha
   ```

4. Optional: Use **mmchconfig** to configure cluster-wide CNFS parameters using the following command:

   ```
   mmchconfig cnfsvip=<dns_name>,cnfsmountdport=<mountd_port>,
   cnfsnfsdprocs=<nfsd_procs>
   ```

   Where:

   dns_name Is a virtual name that represents a list of IP addresses for NFS servers, which allows clients to be distributed among the CNFS nodes using DNS round-robin. In case one of the servers goes down, another server in the list takes over and releases any locks held by the failed server. This is known as *lock reclaim mechanism*.

   mountd_port Is a port number to be used for rpc.mountd. For CNFS to work correctly with automounter, the rpc.mountd on the different nodes *must* be bound to the same port. Because there is no default value, a value must be provided.

   nfsd_procs Is the number of nfsd kernel threads (default is 32).

5. Optional: It is possible to create multiple failover groups by assigning a group ID to each node in the CNFS cluster using the following command:

   ```
   mmchnode -N nodename --cnfs-groupid=xx
   ```

   When a node fails, it first attempts to redirect traffic to another node with the same group ID. If a node with the same group ID cannot be found, traffic fails over to a node with a group ID with the same digit in the tens place. For example, if a node with a group ID of 25 fails, it first tries to fail over to another node with a group ID of 25. However, if there is no other available node with this group ID, the traffic can be moved to any node with a group ID of 20 through 29. If these nodes are also not available, then failover does not take place.

4.3.1 Load balancing

The current load balancing for CNFS is based on DNS round-robin. In this mode, each request to the NFS cluster is routed to a different node in sequential order. Although there are many options for a DNS server, this paper describes how to configure round-robin using the Berkeley Internet Name Domain (BIND) daemon.

Within a BIND zone file, you can define multiple address records with the same host name. Optionally, you can specify a time to live (TTL) value that indicates how long the results can be used without validation. With the first DNS request for a particular host name, BIND returns the first IP address that is associated with that host name and remembers which IP was provided. On the second request, BIND returns the next IP address in the list and will continue in this fashion, looping as needed.

The following sample shows a BIND configuration:

```
cnfs.ibm.com.  60  IN  A  192.168.1.1
cnfs.ibm.com.  60  IN  A  192.168.1.2
cnfs.ibm.com.  60  IN  A  192.168.1.3
cnfs.ibm.com.  60  IN  A  192.168.1.4
```

4.3.2 NFS exports file

Before a share can be accessed using NFS protocol, an administrator must declare it for export by registering it in the `/etc/exports` file. This registration must take place on every node in the NFS cluster, and it is critical for every server in a CNFS cluster to have identical copies of this exports file.

At its most basic, the `/etc/export` files includes a list of directories to be exported as NFS shares, as well as the IP addresses or host names of the systems that are allowed to connect to the share. CNFS also requires the use of the fsid option for correct operation. See 5.1, “Exporting share” on page 34, for more information about the fsid option.

Here is a sample `/etc/exports` file:

```
/usr/share       *cnfs.ibm.com(rw,fsid=745)
```

There are many additional options that you can set to add security controls and to affect performance. For details about configuring the NFS options, see:

http://ldp.linuxhelp.ca/HOWTO/NFS-HOWTO/server.html#CONFIG

4.4 NFS verification

This section explains NFS verification.

4.4.1 Viewing the CNFS configuration options

To view the CNFS configuration options, use the following command:

```
mmlsconfig |grep cnfs
```

The sample output from this command is as follows:

```
# mmlsconfig | grep cnfs
cnfsSharedRoot /gpfsha
cnfsReboot yes
cnfsDebug 2
```

4.4.2 Querying the CNFS cluster status

To query the CNFS cluster status, use the following command:

```
mmlscluster –cnfs
```

Example 4-1 shows a query of the CNFS cluster.

Example 4-1 The CNFS cluster

```
# mmlscluster -cnfs
GPFS cluster information
========================
 GPFS cluster name:         c5n33.ppd.pok.ibm.com
 GPFS cluster id:           680751772142215689
Cluster NFS global parameters
  Shared root directory:                /gpfsha
  Virtual IP address:                   (undefined)
  rpc.mountd port number:               (undefined)
  nfsd threads:                         32
  Reboot on failure enabled:            yes
  CNFS monitor enabled:                 yes
 Node  Daemon node name            IP address       CNFS state  group  CNFS IP address list
-------------------------------------------------------------------------------------------
   1   c5n33.ppd.pok.ibm.com       9.114.132.33     enabled        0   9.114.132.31
   5   c5n38.ppd.pok.ibm.com       9.114.132.38     enabled       31   9.114.132.36
   6   c5n44.ppd.pok.ibm.com       9.114.132.44     enabled       32   9.114.132.47
   8   c5n41.ppd.pok.ibm.com       9.114.132.41     enabled       33   9.114.132.54
  10   c5n42.ppd.pok.ibm.com       9.114.132.42     enabled       34   9.114.132.55
```

4.4.3 Querying the CNFS node status

To query the CNFS node status, follow these steps:

1. Open the GPFS log file and look for the text `mmnfsmonitor: monitor has started`, which shows that the CNFS is functioning correctly.
2. Issue the **`mmgetifconf`** or **`ifconfig`** command to see whether all Ethernet interfaces have been started by CNFS. In the case of a virtual IP, both commands show a pseudo interface for this address.

Example 4-2 shows output from querying the interface.

Example 4-2 The interface

```
# mmgetifconf
lo 127.0.0.1 255.0.0.0
eth0 9.114.132.33 255.255.255.128
eth0:0 9.114.132.31 255.255.255.128
# ifconfig
```

```
eth0      Link encap:Ethernet  HWaddr 00:03:47:24:81:4C
          inet addr:9.114.132.33  Bcast:9.114.132.127  Mask:255.255.255.128
          inet6 addr: fe80::203:47ff:fe24:814c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:11918959 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6434247 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:1951334078 (1860.9 Mb)  TX bytes:1567549753 (1494.9 Mb)
eth0:0    Link encap:Ethernet  HWaddr 00:03:47:24:81:4C
          inet addr:9.114.132.31  Bcast:9.255.255.255  Mask:255.255.255.128
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:358823 errors:0 dropped:0 overruns:0 frame:0
          TX packets:358823 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:156734785 (149.4 Mb)  TX bytes:156734785 (149.4 Mb)
```

5

Clustered NFS administration and operation

This chapter presents ways to interact with Clustered Network File System (CNFS) by describing a set of related tasks that illustrate the configuration and operations of a CNFS cluster. It also presents a set of best practices for securing your CNFS storage environment and describes how to extend the solution to meet the requirements of your infrastructure.

5.1 Exporting share

Before a share can be accessed through the NFS protocol, an administrator must declare it for export by registering it in the `/etc/exports` file. This registration must take place on every node in the NFS cluster.

At its most basic, the `/etc/export` file contains a list of directories to be exported as NFS shares, as well as the IP addresses or host names of the systems that are allowed to connect to the share.

This is a sample `/etc/exports` file:

```
/usr/share      *cnfs.ibm.com
```

An fsid value must be specified for each GPFS file system that is exported on NFS. The format of the entry in `/etc/exports` for the GPFS directory `/gpfs/dir1` looks like this:

```
/gpfs/dir1 cluster1(rw,fsid=745)
```

The administrator must assign fsid values subject to the following conditions:

- The values must be unique for each file system.
- The values must not change after reboots. Unexport the file system before any change is made to an already assigned fsid.
- Entries in the `/etc/exports` file are not necessarily file system roots. You can export multiple directories within a file system. In the case of different directories of the same file system, the **`fsid`** should be different. For example, in the GPFS file system `/gpfs`, if two directories are exported (dir1 and dir2), the entries might look like this:

  ```
  /gpfs/dir1 cluster1(rw,fsid=745)
  /gpfs/dir2 cluster1(rw,fsid=746)
  ```

- If a GPFS file system is exported from multiple nodes, each **`fsid`** should be the same on all nodes.

There are many additional options that can be set to add security controls and to affect performance. For details about configuring the NFS options, see:

http://ldp.linuxhelp.ca/HOWTO/NFS-HOWTO/server.html#CONFIG

After changing the entries in the `/etc/exports` file, the administrator needs to refresh the NFS daemon with the changes. Refer to the operating system documentation for the proper command.

5.2 Mounting NFS share

To use an NFS file system, a user must mount the share on the client machine. An NFS client can run any operating system, although certain operating systems require the installation of additional programs to act as an NFS client. This section discusses a client running a variant of the Linux operating system.

To begin the process of mounting the NFS share, issue the following command:

```
mkdir <dir_name>
```

The user then issues a **mount** command to connect an instance of the exported NFS share on a server in the cluster to the local operating system at the specified mount point. The **mount** command accepts various options based on application requirements, including:

- rsize=<BYTES>,wsize=<BYTES>

  These options represent the datagram size that is used by NFS for reading and writing. Setting these options causes the NFS client to attempt to negotiate the specified buffer sizes up to the sizes specified. Depending on an application’s workload, a larger buffer size typically improves performance up to a point. However, both the server and the client have to provide support. In the case where one of the participants does not support the size specified, the size negotiated is the largest that both support.

- timeo=n

  This option sets the time (in tenths of a second) that the NFS client waits for a request to complete. The default value is 7 (0.7 seconds). What happens after a timeout depends on whether you use the hard or soft option.

- hard

  This option explicitly marks this volume as hard-mounted, the default. This option causes the server to report a message to the console when a major timeout occurs and to continue retrying the failing operation indefinitely. Use of the hard option is suggested because it insulates the user from effects of a server mount failure.

- soft

  A soft-mount, as opposed to a hard-mount, of a share causes an I/O error to be reported to the process that is attempting a file operation when a major timeout occurs.

- intr

  This option allows signals to interrupt an NFS call and is useful for aborting a process when the server does not respond.

For more information about the **mount** command and options, see the *Linux Network Administrator's Guide*.

Successful completion of the mount process allows the user to navigate and use the remote NFS share at the specified directory as though it were a local directory. Possible errors include inability to connect to the host name or IP address that denotes the NFS cluster or invalid user permissions.

The following sample commands mount an NFS drive on a Linux system:

```
mkdir /mnt/nfs_share
mount –o hard -t nfs cnfs.ibm.com:/nfsshare /mnt/nfs_share
```

5.3 Starting CNFS

Because the CNFS service relies on GPFS, it is necessary to start the GPFS daemon on all related CNFS nodes. The administrator starts the daemon with the **mmstartup** command. You can find further details about this command in the *GPFS Administration and Programming Reference Guide*.

5.4 Adding a node

If the CNFS configuration does not contain enough capacity to meet the workload needs, it might become necessary to add one or more nodes to the cluster. You can add NFS server nodes either singularly or in batch. To add a single node to the CNFS cluster, use the following command:

```
mmchnode -N <node_name> --cnfs-enable –cnfs-interface=<nfs_ip>
```

Where *node_name* is the node name that is used for GPFS and *nfs_ip* is the node name that is used for NFS.

Note: If *nfs_ip* is a virtual interface, then GPFS defines the virtual interface automatically. If it has been defined previously, GPFS returns an error.

The following sample adds a single node:

```
mmchnode -N c5n44 --cnfs-enable --cnfs-interface=c5n44ha
```

To add several nodes together to the CNFS cluster, use the following command:

```
mmchnode -S mmchnode-input
```

In this case, the mmchnode-input might look like:

```
c5n33 --cnfs-enable --cnfs-interface=c5n33ha
c5n41 --cnfs-enable --cnfs-interface=c5n41ha
c5n42 --cnfs-enable --cnfs-interface=c5n42ha
c5n43 --cnfs-enable --cnfs-interface=c5n43ha
```

After adding a node to the CNFS cluster, it can also be added to the round-robin DNS configuration so that traffic begins to flow to that node. See 4.3.1, “Load balancing” on page 29, for additional information.

5.5 Removing a CNFS node from a cluster

If the NFS cluster contains excess capacity, an administrator can choose to remove nodes from the cluster and use them for alternative purposes. As with adding nodes, removal of NFS server nodes can be done singularly or in batch.

Note: Removing the node from the CNFS cluster might not stop the node from NFS sharing.

Before removing the node from the CNFS cluster, it is necessary to stop new traffic from being directed to that node by removing the node entry from the round-robin DNS setup. Refer to 4.3.1, “Load balancing” on page 29, for additional information.

After a node is removed from DNS, no new clients can connect to it, although existing clients might still be connected to this server. To remove a server node without disrupting the client access, it is necessary to initiate a failover of the node before removing it from the cluster. To fail over a node, stop the GPFS daemon for that node using the following command:

```
mmshutdown -N <node_name>
```

After all NFS traffic has been failed over to other nodes, proceed to remove the node permanently from the CNFS cluster using the following command:

```
mmchnode -N <node_name> --cnfs-interface=DELETE
```

This command removes all CNFS-related records of the node in `/var/mmfs/gen/mmsrdfs`.

The following example removes a node from the CNFS cluster:

```
e99c2rp1:~ # mmchnode -N e99c4rp1 --cnfs-interface=DELETE
Tue Nov 13 11:55:38 PST 2007: mmchnode: Processing node e99c4rp1.ppd.pok.ibm.com
mmchnode: Propagating the cluster configuration data to all affected nodes. This
is an asynchronous process.
```

5.6 Disabling a CNFS node

To temporarily remove a CNFS node from the cluster, perhaps to service the system, the administrator can disable the node but not remove it from the CNFS cluster.

Note: Removing the node from the CNFS cluster might not stop the node from NFS sharing.

Before disabling the node from the CNFS cluster, it is necessary to stop new traffic from being directed to that node. You can stop new traffic from being directed to the node by removing the node entry from the round-robin DNS setup. Refer to 4.3.1, “Load balancing” on page 29, for additional information. If you are removing the CNFS member node for a short period of time, it is not necessary to remove the node entry from the round-robin DNS setup.

Note: When you enable the node again, make sure to add the node back into the round-robin DNS setup so that traffic is directed to the node.

To avoid disrupting any clients that are connected the node, the node must be failed over before disabling it. To fail over the node, stop the GPFS daemon for that node using the following command:

```
mmshutdown -N <node_name>
```

After all NFS traffic is failed over to other nodes, proceed to disable the node using the following command:

```
mmchnode -N <node_name> --cnfs-disable
```

The following example disables a CNFS node:

```
mmchnode -N c5n44 --cnfs-disable
```

To disable several nodes at the same time, use the following command:

```
mmchnode -S mmchnode-input
```

In this case, the mmchnode-input might look like:

```
c5n33 --cnfs-disable --cnfs-interface=c5n33ha
c5n41 --cnfs-disable --cnfs-interface=c5n41ha
c5n42 --cnfs-disable --cnfs-interface=c5n42ha
c5n43 --cnfs-disable --cnfs-interface=c5n43ha
```

5.7 Failing over a node

Before removing a node from the CNFS cluster, it is important to fail over the node so that any clients that are connected to the node can be rerouted to another node in the cluster. To force a node to fail over, the administrator needs to shut down the GPFS daemon on the node using the following command:

```
mmshutdown -N <node_name>
```

Note: Removing the node from the CNFS cluster might not stop the node from NFS sharing.

5.8 Failing back a node

A CNFS node can be returned to service by restarting the GPFS daemon on the node using the following command:

```
mmstartup -N <node_name>
```

Starting the GPFS daemon adds the node back into the CNFS cluster and ensures that the NFS daemon is started.

5.9 Adding storage

In the event that more disk space is needed as part of the NFS cluster, you can add disks to the existing GPFS file systems in a variety of ways. For more information, see the *GPFS Administration and Programming Reference Guide*, available at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs32.basicadm.doc/bl1adm_adddisk.html

5.10 Removing storage

If a given disk or set of disks is no longer needed, you can remove it from the GPFS cluster. For details on the removal of disks from the GPFS cluster, refer to the *GPFS Administration and Programming Reference Guide*, available at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs32.basicadm.doc/bl1adm_deldisk.html

5.11 Monitoring the cluster

Monitoring the state of the CNFS cluster and debugging any issues typically involves examining the GPFS log file. You typically find the source of any issues in that file. Under certain conditions, you might want to examine the Linux system log file and the GPFS trace file.

This section provides examples of the contents of these important diagnostic aids. For more detailed information about the GPFS logging environment, see the *GPFS Problem Determination Guide*, which is available at:

http://publib.boulder.ibm.com/epubs/pdf/bl1pdg12.pdf

5.11.1 GPFS log file

Example 5-1 shows an example of CNFS monitoring in GPFS log file.

Example 5-1 CNFS monitoring in the GPFS log file

```
Mon Nov 26 15:30:23 EST 2007: cp -dp /var/mmfs/tmp/statd/sm.bak/* --target-directory /var/mmfs/tmp/statd/sm
Mon Nov 26 15:30:23 EST 2007: mmnfsup: SM_NOTIFY for 9.114.132.36
Mon Nov 26 15:30:23 EST 2007: mmnfsup: Notify for host c5n38ha
Mon Nov 26 15:30:23 EST 2007: mmnfsup: notify /var/mmfs/tmp/statd c5n38ha
Mon Nov 26 15:30:23 EST 2007: cp -dp /var/mmfs/tmp/statd/sm/* --target-directory
/gpfsha/.ha/nfs/9.114.132.38/statd/sm
Mon Nov 26 15:30:24 EST 2007: rpc.statd -N -n c5n38ha
Mon Nov 26 15:30:24 EST 2007: cp -dp /var/mmfs/tmp/statd/sm.bak/* --target-directory
/gpfsha/.ha/nfs/9.114.132.38/statd/sm
Mon Nov 26 15:30:24 EST 2007: mkdir -m 0755 -p /var/mmfs/tmp/statd/sm /var/mmfs/tmp/statd/sm.bak
Mon Nov 26 15:30:24 EST 2007: mmnfsup: Saving current boot time 1194893682 in /var/mmfs/tmp/statd
Mon Nov 26 15:30:24 EST 2007: mmnfsmonitor: monitor has started.
Mon Nov 26 15:30:40 EST 2007: mmnfsmonitor: ifUp 9.114.132.60
Mon Nov 26 15:30:42 EST 2007: ifconfig eth0:2 9.114.132.60 netmask 255.255.255.128
Mon Nov 26 15:30:42 EST 2007: mmnfsmonitor: Enabled aliased IP 9.114.132.60 on eth0:2
Mon Nov 26 15:30:42 EST 2007: arping -q -c 3 -A -I eth0 9.114.132.60
Mon Nov 26 15:30:44 EST 2007: mmnfsmonitor: Sent gratuitous ARP on eth0 for 9.114.132.60
Mon Nov 26 15:30:44 EST 2007: mmnfsmonitor: monitor detected 9.114.132.60 was down and restarted.
Mon Nov 26 15:30:44 EST 2007: mmnfsmonitor: ifUp 9.114.132.54
Mon Nov 26 15:30:46 EST 2007: ifconfig eth0:3 9.114.132.54 netmask 255.255.255.128
Mon Nov 26 15:30:46 EST 2007: mmnfsmonitor: Enabled aliased IP 9.114.132.54 on eth0:3
Mon Nov 26 15:30:46 EST 2007: arping -q -c 3 -A -I eth0 9.114.132.54
Mon Nov 26 15:30:48 EST 2007: mmnfsmonitor: Sent gratuitous ARP on eth0 for 9.114.132.54
Mon Nov 26 15:30:48 EST 2007: mmnfsmonitor: monitor detected 9.114.132.54 was down and restarted.
```

5.11.2 The var/log/messages/ or dmsg

Example 5-2 provides an example of CNFS activities in `/var/log/messages`.

Example 5-2 CNFS activities

```
Nov 26 15:29:48 c5n38 kernel: nfsd: last server has exited
Nov 26 15:29:50 c5n38 CNFS: Enabling interface(s) for IP address(es) 9.114.132.36
Nov 26 15:29:55 c5n38 CNFS: Dynamic enabling of grace period not supported in the is kernel,
lockd will be restarted.
Nov 26 15:29:55 c5n38 CNFS: Starting NFS services
Nov 26 15:29:56 c5n38 nfs: Starting NFS services: succeeded
Nov 26 15:29:57 c5n38 nfs: rpc.rquotad startup succeeded
Nov 26 15:29:58 c5n38 nfs: rpc.nfsd startup succeeded
Nov 26 15:29:58 c5n38 nfs: rpc.mountd startup succeeded
Nov 26 15:30:09 c5n38 CNFS: Initiating IP takeover of 9.114.132.44 due to node failure/recovery
Nov 26 15:30:23 c5n38 CNFS: Reclaim of NLM locks initiated for node 9.114.132.44
Nov 26 15:30:23 c5n38 rpc.statd[25178]: Version 1.0.7 Starting
Nov 26 15:30:23 c5n38 rpc.statd[25178]: Flags: Notify-Only
Nov 26 15:30:23 c5n38 CNFS: Reclaim of NLM locks initiated for node 9.114.132.38
Nov 26 15:30:24 c5n38 CNFS: monitor has started.
Nov 26 15:30:44 c5n38 CNFS: monitor detected 9.114.132.60 was down and restarted.
Nov 26 15:30:48 c5n38 CNFS: monitor detected 9.114.132.54 was down and restarted.
```

6

Best practices

This chapter discusses the best practices for the IBM Clustered Network File System (CNFS) solution, including security, expandability, and integration.

6.1 NFS security considerations

The Network File System (NFS) protocol was created by Sun MicroSystems to allow systems to share and access remote file systems over a network environment in as transparent a manner as possible. Transparency was achieved at the application level by supporting standards such as POSIX file semantics. NFS shares were also intended to be file system independent, allowing servers to export any file system or directory that is available to the remote system. The stated goal was simplicity and effectiveness. The initial versions of NFS were not designed to support file locking, although this feature was added in later versions.

To understand NFS security, there are three primary relationships to consider:

- How to secure the NFS server, specifically from a trusted client
- How to protect an NFS server from untrusted clients
- How an NFS client can protect against a mounted share

In this section, we discuss NFS from traditional security perspectives. The latest version, NFS V4, provides for enhanced security solutions such as key authentication, which needs to be deployed in most environments but that we do not cover here because CNFS currently only supports NFS V3. However, you need to consider the availability of NFS V4 clients for the appropriate platforms, as backward compatibility of NFS has limited adoption of some of these more advanced features.

6.1.1 Securing an NFS server from a trusted client

The primary focus of NFS security design is the relationship between trusted clients and servers. This expectation of trust is accomplished through three mechanisms defined in the `/etc/exports` configuration file. (For an example of the `/etc/exports` file contents, refer to 5.1, “Exporting share” on page 34.)

- The first consideration is the *type of share* (that is, read-only or read-write). The principle guiding this decision is simple: Do not share anything read-write that can be made read-only. If both types of data exist, then it is best to take advantage of the transparency of NFS and deploy a layered set of shares with these distinctions explicitly specified. In a layered environment, individual directories can be shared according to whether they are read-only or read-write. If this type of share is too complex for individual clients, the server can mount the directories, preserving access types, to a new mount point, and then simply export this single aggregated share to the necessary clients.
- The second, more flexible, mechanism for securing a share is through *system filtering*. Exports can be shared globally to any client, but specifying only those systems that should access a share is the best way to limit exposure. Systems can be defined by host name or by IP address. Even IP blocks can be specified. When considering the global access list (that is, using an asterisk (*)), remember that it is unlikely that every system in the world needs to access your share. Shares can even be made read-only to one IP address and read-write to another (for example, allowing every system to write its logs to a system-specific location but allowing every system read-only access to all logs.)
- The third mechanism for securing a server from trusted clients is through *user IDs*. When a file is created on an NFS share, it is associated with a user ID (for example, 500.). This corresponds to the standard POSIX UID and GID identifiers. It is important to realize that the NFS client passes the ID number rather than the associated name, and these IDs can vary by system. So, if user *Alice* has ID number *500* on Client *A* and creates a file, the file is associated with ID number 500. If Alice then moves to a different client where her ID number is 600, she will not be able to access that same file. Therefore, most systems in an NFS environment utilize a common user repository to ensure that IDs are synchronized

throughout all systems. Traditionally, this role has been fulfilled by a service such as NIS. However, Kerberos, LDAP, or other enterprise user management software can be used.

- The last consideration in securing a client is to use the root_squash option, which prevents root from writing to the exported file system

Figure 6-1 illustrates these three techniques.

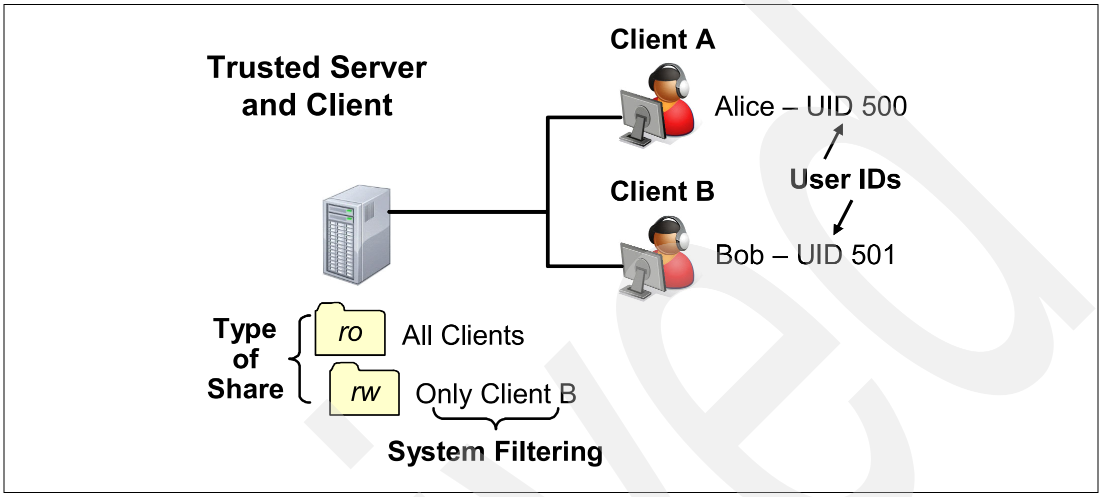

Figure 6-1 Access control in an NFS environment

6.1.2 Protecting an NFS server from untrusted clients

As with most security considerations, protecting a server from an untrusted or explicitly malicious client can become intricate. Ensuring that the *type of share* is correct and that trusted clients cannot corrupt data accidentally (that is read-only versus read-write) is the first step to protecting against malicious clients.

However, more complex considerations must be made at the *system filtering* level. There are two important considerations necessary when securing at a system level. Specifying a host name introduces a dependency on an external service, in this case DNS, which can introduce a potential attack vector for a malicious client. While specifying an IP address is more secure, consider that traditional transport mechanism for NFS, UDP, is vulnerable to packet spoofing, and the more popular TCP/IP has the potential to be exploited as well. Consider disabling UDP connectivity and deploying network auditing utilities to monitor traffic to an NFS server.

The *user ID* mechanism of NFS provides the simplest method for a client to compromise a NFS server. Proper network filtering and subnetting are the primary ways to prevent an attacker from even being able to access a server. However, with NFS versions, including V3 and earlier, if a malicious client can access the server, there are no granular methods to ensure that a malicious client cannot impersonate a given user ID, although by default NFS shares are shared as root_squash, which ensures that an attacker cannot create a file that is owned by root. An administrator can also use mechanisms outside of NFS, such as explicitly specifying an assigned user ID when mounting shares on the NFS server.

6.1.3 Protecting a client from an NFS server

Although NFS security is typically considered server-centric, exploiting this inherently trusted relationship might allow an attacker to gain access to a number of clients. Therefore, it is important to understand how a client can maintain a level of certainty when mounting a file system from a server. A traditional NFS environment does not provide protection from *man in the middle* attacks. Clients should consider mounting an NFS share *nosuid*, which prevents *set uid* binaries that can run with escalated privileges or that can even limit all executable privileges through a *noexec* option. If performance is not a priority, then a tunneled environment which both protects NFS traffic and provides a strong key-pair handshake can provide further levels of protection.

When assessing your security requirements, understanding the client side of the exchange is critical for managing security at an enterprise level.

Administrators with existing GPFS environments might be familiar with the advanced security and life cycle management features that GPFS provides. When sharing any data through NFS, many considerations must be made. Beyond the technology practices, GPFS technical practitioners should also consider any specific compliance and regulatory restrictions. If any such restrictions exist, take care to evaluate and validate appropriate methods of prevention, detection, and auditability.

For more information about NFS, see “Related publications” on page 67.

6.2 Expanding the solution

CNFS architecture is based on a scalable layered approach on the IBM Intelligent Cluster platform. CNFS can be a stand alone or integrated into the enterprise infrastructure. The implementation allows for expansion and integration into your environment to meet the required workloads. The areas of expandability and integration include:

- IBM Intelligent Cluster Servers with a variety of processor, memory, and I/O characteristics
- Communication networks for management of hardware resources, private CNFS operations, and for public access of the NFS file systems
- Storage subsystems with controllers supporting a different number of host interfaces, disk drives technologies offering a range of capacity and performance, and number of disk drives
- Information Lifecycle Management (ILM)
- Backup

6.2.1 IBM Intelligent Cluster CNFS advanced configurations

This section presents a set of sample IBM Intelligent Cluster CNFS configurations ranging from four to 64 CNFS cluster servers.

In Chapter 5, “Clustered NFS administration and operation” on page 33, we discussed the entry-level configuration of a three-node cluster. These configurations are intended to provide enterprise capacity and scalability to meet your infrastructure requirements. Because these configurations are recommendations, contact your local IBM Cluster Sales Specialist for more information about advanced IBM Intelligent Cluster configurations.

CNFS medium configuration with four servers

Table 6-1 provides information for IBM Intelligent Cluster CNFS for a medium configuration.

Table 6-1 IBM Intelligent Cluster CNFS medium configuration

	Base configuration	Max configuration	Max capacity
Servers		24	24
SAN switches	IBM SAN 32-B (eight ports installed)	IBM SAN 32-B (32 ports installed)	IBM SAN 32-B (32 ports installed)
Ethernet switches	3 gigabit Ethernet	3 gigabit Ethernet	3 gigabit Ethernet
Storage subsystem	Controller: One DS4800 with eight host ports Disk drawers: 8 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5	Controller: Three DS4800 with eight host ports Disk drawers: 48 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5	Controller: Eight DS4800 with eight host ports Disk drawers: 128 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5
Capacity	6.5 TB to 26 TB	39 TB to 160 TB	104 TB to 429 TB
I/O rate (estimate)	~800 MBps	~4.8 GBps	~4.8 GBps

Figure 6-2 shows a CNFS medium system configuration with four servers.

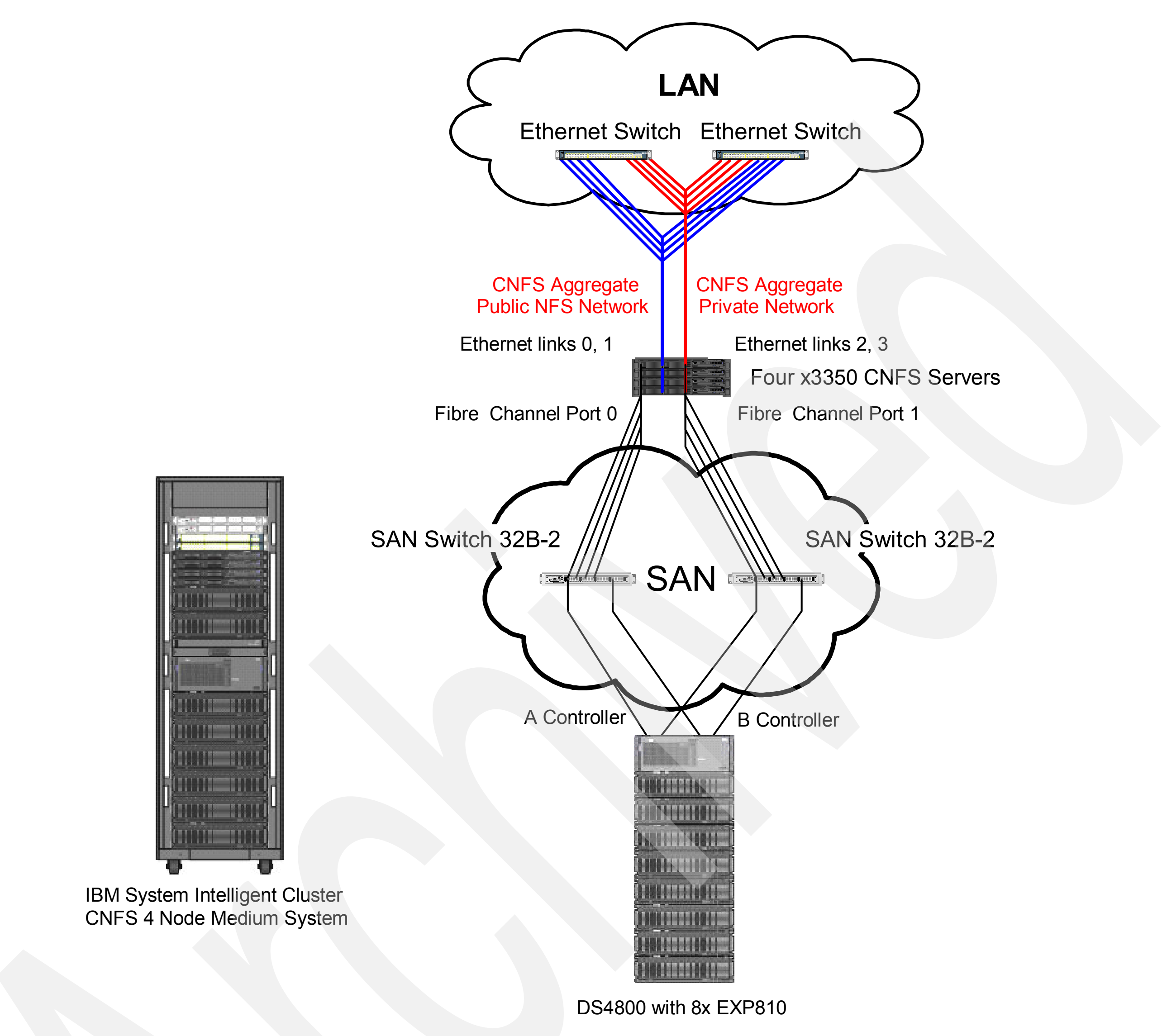

Figure 6-2 CNFS medium system configuration with four servers

CNFS large configuration with 24 servers

Table 6-2 provides information for IBM Intelligent Cluster CNFS for a large configuration.

Table 6-2 IBM Intelligent Cluster CNFS large configuration

	Base configuration	Max configuration	Max capacity
Servers	24	40	40
SAN switches	IBM System Storage SAN64B-2 Two 64 port (32 ports installed)	IBM System Storage SAN64B-2 Two 64 port (48 ports installed)	IBM System Storage SAN64B-2 Two 64 port (48 ports installed)
Ethernet switches	3 gigabit Ethernet	3 gigabit Ethernet	3 gigabit Ethernet
Storage subsystem	Controller: Three DS4800 with eight host ports Disk drawers: 48 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5	Controller: Five DS4800 with eight host ports Disk drawers: 80 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5	Controller: Eight DS4800 with eight host ports Disk drawers: 128 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5
Capacity	39 TB to 160 TB	65 TB to 268 TB	104 TB to 429 TB
I/O rate (estimate)	~4.8 GBps	~8 GBps	~8 GBps

Figure 6-3 shows a CNFS large-system configuration with 24 servers.

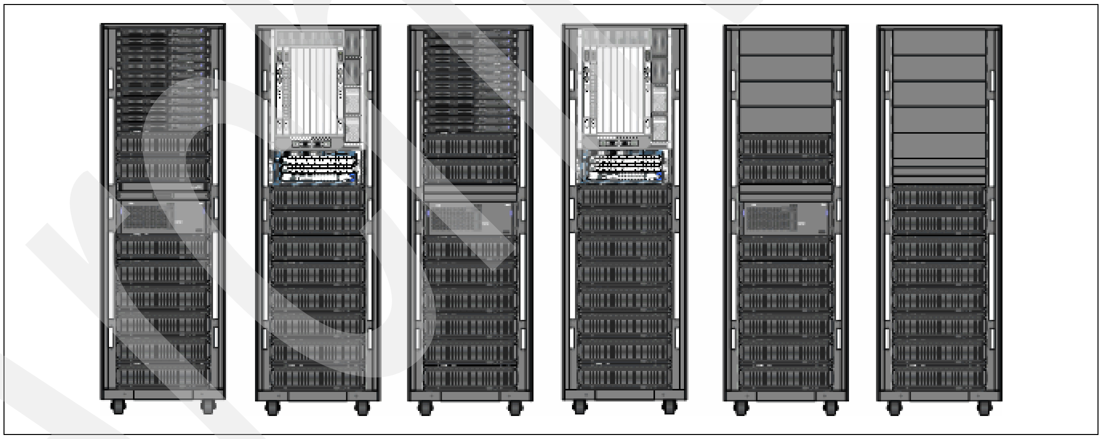

Figure 6-3 CNFS large-system configuration with 24 servers

CNFS extra large configuration

Table 6-3 provides information for IBM Intelligent Cluster CNFS for an extra large configuration.

Table 6-3 IBM Intelligent Cluster CNFS extra large configuration

	Base configuration	Max configuration	Max capacity
Servers	40	64	64
SAN switches	IBM System Storage SAN64B-2 Four 64 port (32 ports installed)	IBM System Storage SAN64B-2 Four 64 port (48 ports installed)	IBM System Storage SAN64B-2 Four 64 port (48 ports installed)
Ethernet switches	3 gigabit Ethernet	3 gigabit Ethernet	3 gigabit Ethernet
Storage subsystem	Controller: Five DS4800 with eight host ports Disk drawers: 80 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5	Controller: Eight DS4800 with eight host ports Disk drawers: 128 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5	Controller: Forty-eight DS4800 with eight host ports Disk drawers: 768 EXP810 with 16 drives distributed across all the DS4800 LUN: 4 + Parity RAID 5
Capacity	65 TB to 268 TB	65 TB to 268 TB	626 TB to 2.58 PB
I/O rate (estimate)	~8 GBps	~12 GBps	~12 GBps

For general considerations regarding CNFS storage subsystem design for maximum performance, see the GPFS FAQs, available at:

`http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/index.jsp?topic=/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfs_faqs.html`

6.2.2 IBM Intelligent Cluster networking requirements

The CNFS cluster based on the IBM Intelligent Cluster requires three types of networks:

- The *management network* is the Ethernet network where the System x and BladeCenter Servers, BladeCenter chassis, SAN switches, Storage controllers, and the Ethernet switches are managed.
- The *private network* is a CNFS cluster network in which the internal cluster communications occur between cluster nodes.
- The *public Network* is a CNFS cluster network that provides the interface into the Enterprise network for NFS clients to access the CNFS file systems.

Best practices

This section lists the best practices for providing optimum operation for high availability and throughput.

Management network

The System x and BladeCenter servers need the appropriate adapters that permit remote management capability. The number of ports that are required for a CNFS system can be estimated as follows:

- One port per server connected into the management port, Remote Supervisor Adapter 2 (RSA-2) or Base Management Controller (BMC)
- One port per SAN switch (Enterprise class director might require two ports)
- Two ports per dual-controller storage subsystems like the IBM System Storage DS4800
- One port per Ethernet switch used for CNFS private and public networks (Enterprise switch might require two ports)

Private network

The network needs to be configured with high-performance aggregate links in to redundant switch fabrics. The aggregate links need to be configured with link-level failover. All CNFS servers need to be in the same subnet.

Public network

The network needs to be configured with high-performance aggregate links in to redundant switch fabrics. The aggregate links should be configured with link-level failover. The network operations center (NOC) should verify that the design meets its policies. The CNFS server in an associated failover group needs to be in the same subnet.

Ethernet link aggregation

Your CNFS cluster can be expanded by aggregating your network adapters into a link that provides both increased throughput and availability. This aggregation technique has many names, such as link aggregation, EtherChannel (Cisco-specific implementation), network teaming, or network bonding. More specifically, there are two specific implementations:

- Cisco EtherChannel
- IEEE 802.3ad Link Aggregation

Both implementations allow the sum of all network adapters connected between server and network switch components to be combined to increase bandwidth. If an adapter fails, network traffic is automatically sent on to the next available adapter without disruption to existing user connections. The adapter is returned automatically to service on the EtherChannel or link aggregation when it recovers (Table 6-4).

Table 6-4 Differences between EtherChannel and IEEE 802.3ad link aggregation

EtherChannel	IEEE 802.3ad
Requires switch configuration	Little, if any, configuration of switch required to form aggregation. Initial setup of the switch might be required.
Supports various packet distribution modes	Supports only standard distribution mode.

The Linux bonding driver provides a method for aggregating multiple network interfaces into a single logical bonded interface. The behavior of the bonded interfaces depends on the mode. Generally, modes provide either hot standby or load-balancing services.

Mode 0 (Balance-rr) is a round-robin policy that provides load balancing and fault tolerance. The link needs to terminate in the same Ethernet fabric that could be a single point of failure.

Mode 1 (active-backup) is an active backup policy providing fault tolerance. The link can be split across two Ethernet fabrics for fault tolerance.

The initial base configuration for the CNFS IBM Intelligent Cluster described in 3.2, “Recommended CNFS initial base configuration” on page 20 has an additional Ethernet adapter that is part of the configuration and should be used to create an aggregated link for the NFS public network. You can find further details at:

http://www.linux-foundation.org/en/Net:Bonding

For more information about link aggregation and Ethernet bonding, also see:

http://en.wikipedia.org/wiki/Link_aggregation
http://linux-ip.net/html/ether-bonding.html

Ethernet communications can also be virtualized over InfiniBand. Performance of TCP/IP over IB is substantially faster than teamed or aggregated Ethernet links. Similar to Ethernet aggregation, InfiniBand links can be aggregated to achieve higher bandwidth. Your IBM Intelligent Cluster can be ordered or upgraded to support InfiniBand communications.

6.2.3 Storage expansion

Each CNFS cluster member requires connections to elements of the storage area network. The best practices for storage expansion include:

- Each CNFS cluster member should have a connection to each storage array in the cluster.
- Dual SAN fabric is suggested to avoid issues that appear when connections are changed.
- Each CNFS cluster member requires two connections to the SAN.
- When using an IBM System Storage DS4800, each CNFS cluster member requires two paths to the DS4800, one per controller, and that the DS4800 have eight ports, four per controller.

A LUN on the SAN is the basic building block for CNFS storage. Generally, a LUN consisting of a 4+P RAID 5 physical disk is sufficient to handle the block size of NFS. 2 + 2 RAID 10 can be useful for metadata as the file count within CNFS approaches 100 million files. A spare drive is suggested in each enclosure to maximize availability and minimize performance changes as a drive fails in a given drawer.

For the configurations that we describe here, we use the definition *optimum operation*, which is assumed to be the minimum number of DS4800s with the minimum EXP810s for the maximum I/O rate.

Storage capacity can be scaled up with the addition of more drawers under a controller. Throughput can be scaled out by adding more controllers. The overall CNFS generalized guidelines for storage are:

- Approximately 200 MBps per System x rack optimized servers with dual 4 Gbps Fibre Channel HBA
- Approximately 1600 MBps per DS4800 limited by number of drawers
  - Approximately 100 MBps per EXP810 Storage drawer
  - 8.1 TB per EXP810 Storage drawer with 16 x 73 GB Fibre Channel Drives using RAID 5 (4+P)

- 16.3 TB per EXP810 Storage drawer with 16 x 146 GB Fibre Channel Drives using RAID 5 (4+P) plus hot spare
- 33.5 TB per EXP810 Storage drawer with 16 x 300 GB Fibre Channel Drives using RAID 5 (4+P) plus hot spare

Table 6-5 shows CNFS cluster member nodes configured for optimum operation. Capacity can increase by maximizing the number of EXP810 per DS4800 and utilizing spare Fibre Channel Ports.

Table 6-5 CNFS storage capacity configured for optimum operation

CNFS servers	3	4	8	16	24	32	40	48	56	64
TB capacity (73 GB drive)	4.1	5.7	11.4	22.8	34.3	45.7	57.1	68.5	76.7	91.4
TB capacity (146 GB drive)	8.2	11.4	22.8	45.7	68.5	91.4	114.2	137.1	153.4	182.7
TB capacity (300 GB drive)	16.8	23.5	46.9	93.9	140.8	187.8	234.7	281.6	315.2	375.5

Table 6-6 lists the number of DS4800 and EXP810 to reach optimum operation.

Note: Adding applications that require file system space outside NFS, such and TSM backup, requires additional storage.

Table 6-6 Storage controller and storage drawers for optimum capability per CNFS configuration

# DS480s	1	1	1	2	3	4	5	6	8	8
# EXP810 's (3x(4+P)+S)	5	7	14	28	42	56	70	84	94	112
CNFS servers	3	4	8	16	24	32	40	48	56	64

Table 6-7 shows an estimate of the total number of Fibre Channel ports that are required for CNFS optimum operation. These ports need to be distributed across the two fabrics.

Table 6-7 SAN Switch Port requirements per CNFS configuration

Server ports	6	8	16	32	48	64	80	96	112	128
Storage ports	8	8	8	16	24	32	40	48	48	56
Total ports	14	16	24	48	72	96	120	144	160	184
CNFS servers	3	4	8	16	24	32	40	48	56	64

6.3 Integration into other environments

This section describes how to back up and archive your CNFS data.

6.3.1 Backup

CNFS has routines that support information management of large file systems by performing the following operations:

- Scanning an entire file system for inode changes
- Creating a list of files that have changed
- Parceling out the list of files for backup

An example of how CNFS cluster members can be backed up is by installing IBM Tivoli Storage Manager (TSM) for storage area networks (SANs). It features TSM, which enables LAN-free backup. TSM for SAN can be installed on one of your CNFS cluster members or on another node in your IBM Intelligent Cluster.

TSM for SAN allows the client system to read and write data directly to and from storage devices that are attached to a SAN. Instead of passing or receiving the information over the LAN, data movement is off-loaded from the server where TSM is installed, making network bandwidth available for other uses. For instance, using the SAN for client data movement decreases the load on the TSM server and allows it to support a greater number of concurrent client connections.

Figure 6-4 shows an overview of all the TSM components and illustrates the SAN data movement. The TSM Administrative Console includes the Integrated Solution Console (ISC) with the TSM Administration Center. The dashed lines indicate movement of control information and metadata. The solid lines indicate data movement. The solid blue line represents TSM using SAN storage for the TSM database. The TSM library manager manages the client's data, which is stored on a tape library. A TSM server library client manages a client's data, which is stored on a tape library.

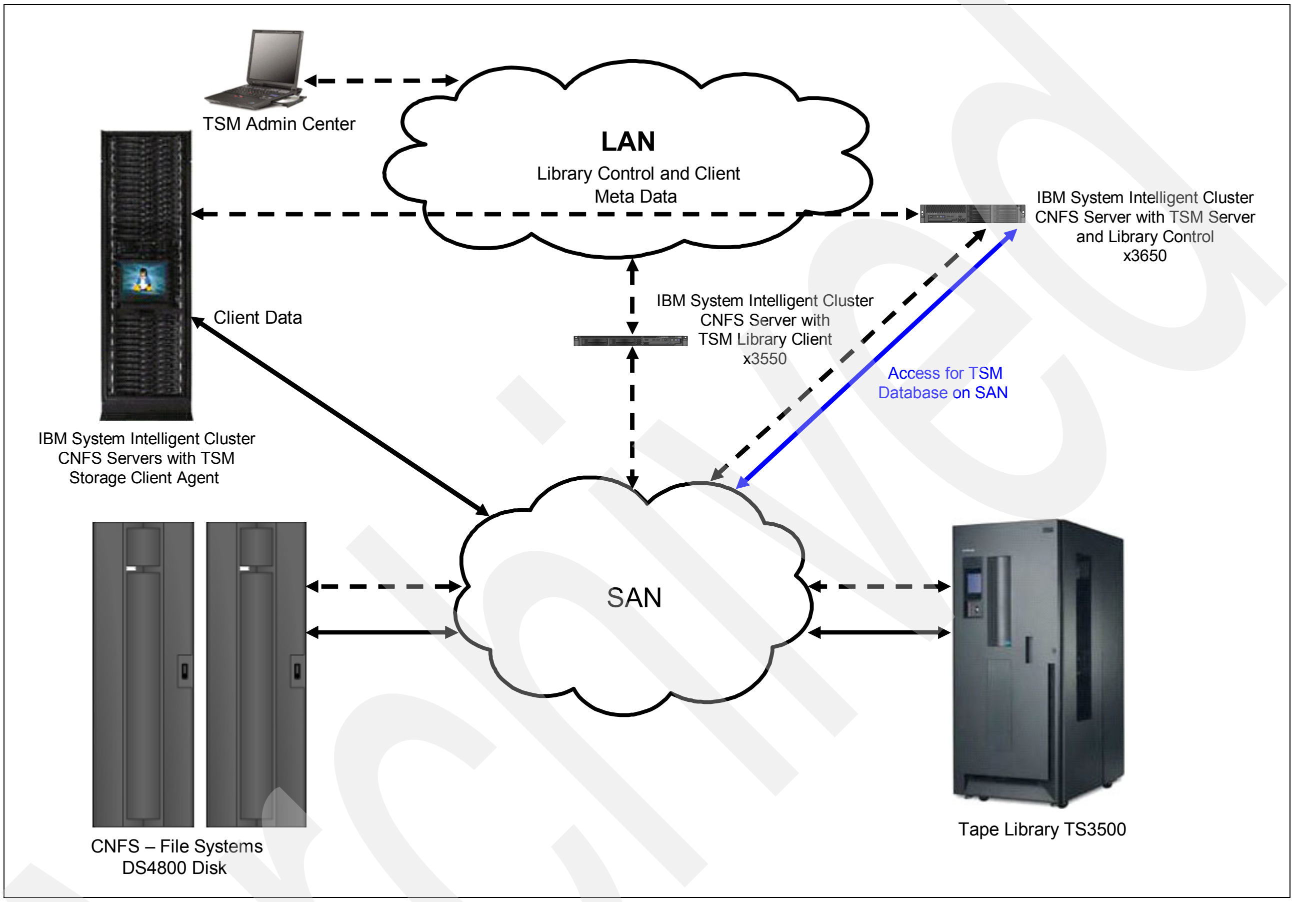

Figure 6-4 CNFS data movement for LAN-free backup

Backups of multiple nodes that share storage can be consolidated to a common target node name on the TSM server (Figure 6-5). This capability is useful in a cluster that might have different servers responsible for performing the backup over time. The asnodename option also allows data to be restored from a different system than the one that performed the backup. An *agent node* is a client node that has been granted authority to perform client operations on behalf of a target node. A *target node* is a client node that grants authority to one or more agent nodes to perform client operations on its behalf.

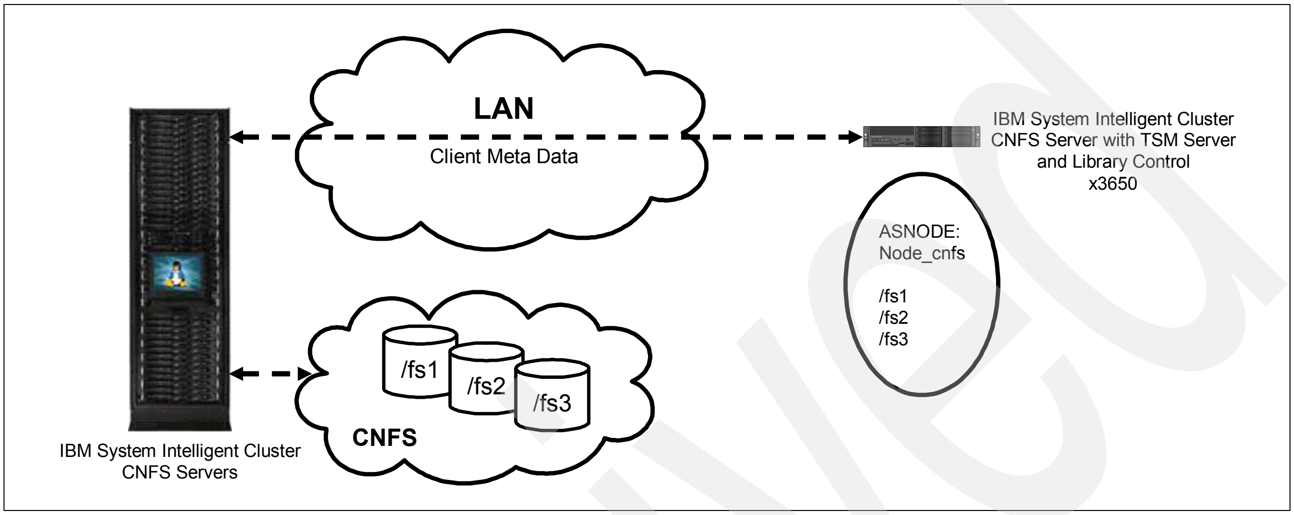

Figure 6-5 Backup of CNFS

For more information, see “Related publications” on page 67.

6.3.2 Hierarchical Storage Manager Information Lifecycle Management

CNFS addresses Information Lifecycle Management (ILM) migration of information between active data and other hierarchies of storage by policy. The following abstractions of ILM are addressed:

- Storage pool: Group of LUNs
- Fileset: Subtree of a file system
- Policy: Rules for assigning files to storage pools

CNFS utilizes GPFS tight integration of ILM functionality into a file system that can support many files shared across multiple systems.

For more information, see *Implementing a Scalable Information Lifecycle Solution using GPFS*, at:

http://www-03.ibm.com/systems/clusters/software/whitepapers/gpfs_ilm.html/

Implementing a Scalable Information Lifecycle Solution using GPFS reviews various approaches to ILM and describes ILM tools that are available starting with GPFS 3.1. It also helps provide an understanding of the benefits of the GPFS tools for ILM by examining the following approaches:

- A layered software solution that sits on top on existing file systems
- A network protocol approach that uses NFS/CIFS communications

A

IBM Intelligent Cluster hardware options

This appendix lists IBM Intelligent Cluster hardware options, including servers, storage, networking, and racks.

System x and Blade Server options

You can configure IBM Intelligent Cluster to use one or more of the following System x or BladeCenter servers:

- x3650: Dual core up to 3.0 GHz, quad core up to 2.66
- x3550: Dual core up to 3.0 GHz, quad core up to 2.66
- x3455: Dual core up to 2.8 GHz
- x3655: Dual core up to 2.6 GHz
- x3755: Dual core up to 2.8 GHz
- HS21: Dual core up to 3.0 GHz, quad core up to 2.66
- HS21 XM: Dual core up to 3.0 GHz, quad core up to 2.33
- JS21: 2.7/2.6 GHz, 2.5/2.3 GHz
- LS21: Dual core up to 2.6 GHz
- LS41: Dual core up to 2.6 GHz

Storage options

You can order the IBM Intelligent Cluster with a wide range of disk arrays and SAN interconnect options to meet your data capacity, performance, cost, and technology requirements. IBM offers direct-attached, Fibre Channel, and Serial Attached SCSI (SAS) disk technologies that can be included in your IBM Intelligent Cluster. When using larger Fibre Channel configurations, SAN switches are also orderable to build a SAN.

The external storage options for the IBM Intelligent Cluster include:

- Storage servers: System Storage DS4800 (FC), DS4700 (FC), DS3400 (FC), DS3200 (SAS)
- Storage expansion: EXP810 Storage Expansion Unit, EXP3000 direct-attach storage expansion
- SAN switches: IBM System Storage SAN 16B-2, SAN 32B-2 SAN Switches

Networking options

Both standard Ethernet and low-latency networking options are available for your IBM Intelligent Cluster. Depending on whether you are using rack-mounted or BladeCenter-based servers, there are a multitude of choices to meet your cluster requirements. The available networking options for your IBM Intelligent Cluster include the following Ethernet options:

- BladeCenter Network Switch Modules
  - Cisco
  - Nortel
  - Copper
  - Optical passthru
- Rack mounted
  - Cisco
  - Force10
  - Nortel
  - SMC

In addition to standard Ethernet, you can equip your IBM Intelligent Cluster with low-latency networking hardware for parallel MPI and HPC applications. The low-latency options for both rack-optimized and BladeCenter-based IBM Intelligent Cluster configurations include:

- InfiniBand
  - Cisco
  - Voltaire
- Myrinet: Myrinet

Rack and enclosure options

You can order the IBM Intelligent Cluster with a standard 42 U rack or with a smaller 25 U rack, depending on the required footprint and capacity of your cluster.

The available IBM Intelligent Cluster rack options are:

- 42U primary/expansion rack: 79.5" H x 25.2" W x 43.3" D (2020 mm x 640 mm x 1100 mm), 574.2 lbs (261 kg)
- 25U rack: 49.0" H x 23.8" W x 39.4" D (1344 mm x 605 mm x 1001 mm), 221 lbs (100.2 kg)

Each IBM System x server and BladeCenter chassis that is included in your IBM Intelligent Cluster includes industry power and cooling technologies, such as Calibrated Vectored Cooling™, energy-efficient power supplies, low-voltage processors, thermal diagnostics, and IBM Power Configuration at no extra charge. Along with IBM PowerExecutive™, which is a downloadable software package, you can monitor and proactively manage the power consumption of your IBM Intelligent Cluster.

B

IBM Intelligent Cluster Software options

In addition to the hardware options of the IBM Intelligent Cluster Software, there are many software options that you can configure based on your cluster requirements. Software can be ordered from the following categories:

- Operating systems
- Clustered file systems
- System management

Operating systems

The IBM Intelligent Cluster supports the following operating system distributions and versions:

- Red Hat Enterprise Linux (RHEL) 4**,** 5
- SUSE Linux Enterprise Server (SLES) 9, 10

To use Clustered Network File System (CNFS) and IBM General Parallel File System (GPFS) under these operating systems, you might need to apply patches so that CNFS operates properly. See Chapter 4, “Clustered NFS installation and configuration” on page 25, for more information.

GPFS 3.2 for Linux and CNFS

The GPFS is a high-performance shared-disk file management solution that provides fast, reliable access to a common set of file data from two computers to hundreds of systems. GPFS integrates into your environment by bringing together mixed server and storage components to provide a common view to enterprise file data. GPFS provides online storage management, scalable access, and integrated information life-cycle tools capable of managing petabytes of data and billions of files.

The proven GPFS file management infrastructure provides the foundation for optimizing the use of your computing resources with the following features:

- Scalable, high-performance shared disk file system for AIX and Linux systems
- Capable of supporting multi-petabytes of storage and thousands of disks within a single file system
- High reliability and availability through redundant paths and automatic recovery from node and disk failures
- Information life cycle management (ILM) tools that simplify data management and enhance administrative control
- Powers many of the world’s largest supercomputers

CNFS is integrated into GPFS V3.2 and does not require any additional options to take advantage of it. Just follow the configuration steps in Chapter 4, “Clustered NFS installation and configuration” on page 25, and you are on your way to having an industry-leading CNFS solution that is built into GPFS at no extra cost.

Systems management

Your IBM Intelligent Cluster offers several options for the System Management Clustered NFS environment, which we discuss in this section.

Cluster systems manager

Cluster systems manager (CSM) is designed to minimize the cost and complexity of administering clustered and partitioned systems by enabling comprehensive management and monitoring of the entire environment from a single point of control. CSM provides:

- Software distribution, installation, and update (operating system and applications)
- Comprehensive system monitoring with customizable automated responses

- Distributed command execution
- Hardware control
- Diagnostic tools
- Management by group
- Both a graphical interface and a fully scriptable command-line interface

In addition to providing all the key functions for administration and maintenance of distributed systems, CSM is designed to deliver the parallel execution required to manage clustered computing environments effectively. CSM supports homogeneous or mixed environments of IBM servers running AIX or Linux. CSM is a fully supported IBM Intelligent Cluster software option. You can obtain support for design and implementation of a CSM-based IBM Intelligent Cluster.

Extreme cluster administration toolkit

Extreme Cluster Administration Toolkit (xCAT) is a toolkit that you can use for the deployment and administration of Linux clusters. Its features are based on user requirements, and many of its features take advantage of IBM System x and BladeCenter hardware.

xCAT makes simple clusters easy and complex clusters possible by making it easy to:

- Install a Linux cluster with utilities for installing many machines in parallel.
- Manage a Linux cluster with tools for management and parallel operation.
- Set up a high-performance computing software stack, including software for batch job submission, parallel libraries, and other software that is useful on a cluster.
- Create and manage diskless clusters.

xCAT works well with the following cluster types:

- HPC: High-performance computing physics, seismic, computational fluid dynamics, finite element analysis, weather, and other simulations, and bio-informatic work.
- HS: Horizontal scaling web farms, and so forth.
- Administrative: Not a traditional cluster, but a convenient platform for installing and administering a number of Linux machines.
- Windows or other operating systems: With xCAT's cloning and imaging support, it can be used to rapidly deploy and conveniently manage clusters with compute nodes that run Windows or other operating systems.
- Other: xCAT's modular tool kit approach makes it easily adjustable for building any type of cluster.

xCAT is available as a download from the IBM AlphaWorks website:

http://www.alphaworks.ibm.com/tech/xCAT

It is best suited for your own cluster provisioning and system management for your IBM Intelligent Cluster.

C

Clustered NFS initial base configuration bill of materials

This appendix lists the hardware and software, along with their feature numbers, for the initial base configuration of the Clustered Network File System (CNFS) solution.

Initial base configuration

Table C-1 lists the CNFS initial base configuration Bill of Materials.

Table C-1 Clustered NFS initial base configuration bill of materials

PN	Description	Quantity
	CLUSTER TOTAL	
	Server_x3550_cluster_nfs_server	**3**
7978C2U	(Server_x3550_cluster_nfs_server) x3550 Base Model (2.33 Ghz/1333 MHz Clovertown, 8 MB L2, 3.5" HS SAS, (2x) 1 GB PC2-5300, Open Bay)	3
31P4310	Rack Installation of 1U Component	3
39Y6126	-SB- PRO/1000 PT Dual Port Server Adapter by Intel	3
40K1044	146 GB 15 K 3.5" Hot-Swap SAS HDD	6
40K1272	Quad-Core Intel Xeon Processor E5345 (2.33 GHz 8 MB L2 1333 MHz 80w)	3
25R8064	IBM ServeRAID-8k SAS Controller	3
32R2815	x3550 670w Redundant AC Power Supply	3
39R6525	IBM 4 Gbps FC Single-Port PCI-E HBA	3
39M5791	4 GB Kit (2x2 GB) DDR2 (PC2-5300) FBDIMM, 667 MHz	6
21P2073	7978 3 YR IOR 24x7 4 Hour	3
58P8665	Install 4 or more OBI/3rd Party Options on System x	3
	Storage_DS3400_Dual_Controller	**1**
172642X	(Storage_DS3400_Dual_Controller) DS3400 FC-SAS, Dual Controller	1
06P7515	Rack installation of > 1U component	1
39Y7932	4.3 m, 10A/100-250V, C13 to IEC 320-C14 Rack Power Cable	1
39M5696	1 m LC-LC Fibre Channel Cable (Storage)	4
39R6475	IBM 4 Gbps SW SFP Transceiver	4
44J8073	1726-4 3 YR IOR 24x7 4 Hour	1
43X0802	300 GB 15 K 3.5" Hot-Swap SAS HDD	5
39Y8951	DPI® Universal Rack PDU w/Nema L5-20P and L6-20P U.S. Line Cord	1
	SAN_SAN_Switch	**1**
200516B	(SAN_SAN_Switch) IBM TotalStorage SAN16B-2	1
31P4310	Rack Installation of 1U Component	1
22R4901	B16 4-Port Activation	2
39Y7932	4.3 m, 10A/100-250V, C13 to IEC 320-C14 Rack Power Cable	1
41E9144	2005B16 3 YR IOR 24X7 4 Hour	1
22R4897	4 Gbps SW SFP Transceiver 4-Pack	3
39M5696	1 m LC-LC Fibre Channel Cable (Storage)	12
	B16 Full Fabric - Plant	1

	Ethernet_Management Main	**1**
4670030	(Ethernet_Management Main) Cisco HPC-E 2960G 24-port Ethernet Switch Bundle	1
31P4310	Rack installation of 1U component	1
39Y7932	4.3 m, 10A/100-250V, C13 to IEC 320-C14 Rack Power Cable	1
	Ethernet_Compute Main	**1**
4670016	(Ethernet_Compute Main) Cisco HPC-E 4506 Enclosure Bundle	1
06P7515	Rack installation of > 1U component	1
40K9152	Cisco 4506 Air Baffle and Rack Mount Kit	1
40K9117	Cisco HPC-E 4548 48-port Gb Ethernet Line Card	1
40K5607	Cisco 1000BASE-T GBIC Module	1
39Y7916	2.5 m, 16A/100-250V, C19/C20 Rack Power Cable	2
	Console_Console Main	**1**
17351GX	(Console_Console Main) IBM Local 2x8 Console Manager	1
31P4310	Rack installation of 1U component	1
	Keyboard/Monitor_REQ_Rack_Cluster_NFS_Main_Rack	**1**
17231RX	(Keyboard/Monitor_REQ_Rack_Cluster_NFS_Main_Rack) 1U 17 in Console Kit Base w/o Keyboard	1
31P4310	Rack installation of 1U component	1
40K5372	UltraNav Keyboard - USB (US English)	1
	Rack_Cluster_NFS_Main_Rack	**1**
14102RX	Intelligent Cluster 25U Rack Cabinet	1
02R2149	Rack Assembly - 25U Rack	1
26K7477	Cluster Hardware & Fabric Verification - 25U Rack	1
41E8810	1410 - 3-year onsite repair 24x7x4 hour	1
39Y8939	DPI 30amp/250V Front-end PDU with NEMA L6-30P (208v) Line Cord	1
39Y8951	DPI Universal Rack PDU w/Nema L5-20P and L6-20P U.S. Line Cord	2
	Rack_AUTO_85	**1**
14104RX	Intelligent Cluster 42U Rack Cabinet	1
06P7514	Rack Assembly - 42U Rack	1
58P8609	Cluster Hardware & Fabric Verification - Subsequent 42U Rack	1
39Y8939	DPI 30amp/250V Front-end PDU with NEMA L6-30P (208v) Line Cord	1
	Cables	
40K8951	1.5 m Yellow CAT5e Cable	3
40K8933	0.6 m Yellow CAT5e Cable	3
40K8927	10 m Blue CAT5e Cable	6
	Software	
4815M4U	Red Hat 5 Compute	1
48157HU	RHEL HPCC 4-Pack 1-2 Socket Basic Red Hat Support 3 YR	1
39M2895	IBM USB Conversion Option (4-pack)	1

25R5560	IBM 3U Quick Install Filler Panel Kit	3
25R5559	IBM 1U Quick Install Filler Panel Kit	2
	GPFS V3.2	3

Related publications

We consider the publications that we list in this section particularly suitable for a more detailed discussion of the topics that we cover in this paper.

IBM Redbooks publications

For information about ordering these publications, see “How to get IBM Redbooks publications” on page 67.

- *Linux Clustering with CSM and GPFS*, SG24-6601
- *Building a Linux HPC Cluster with xCAT*, SG24-6623
- *IBM Tivoli Storage Manager Version 5.3 Technical Workshop Presentation Guide*, SG24-6774

Other publications

The following publications also provide additional information:

- *AIX Network File System Security*

  http://publib.boulder.ibm.com/infocenter/pseries/v5r3/index.jsp?topic=/com.ibm.aix.security/doc/security/nfs_authentication.htm

  This website provides general guidelines and AIX-specific information. It also covers more advanced security features included with NFSv4.

- *Security and NFS*

  http://tldp.org/HOWTO/NFS-HOWTO/security.html

  This website is Linux-specific and does not cover newer NFS versions. However, it provides a strong reference for general NFS security and system-level techniques, outside of NFS, that can provide further protection.

How to get IBM Redbooks publications

You can search for, view, or download Redbooks publications, Redpapers publications, Technotes, draft publications, and additional materials, as well as order hardcopy Redbooks publications, at this website:

ibm.com/redbooks

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

A Guide to the IBM Clustered Network File System

Discusses the technical architecture

Explains installation and configuration

Includes administration and operation information

The Clustered Network File System (CNFS) is a capability based on IBM General Parallel File System (GPFS) running on Linux, which, when combined with System x servers or BladeCenter Servers, IBM TotalStorage Disk Systems, and Storage Area Networks (SAN) components, provides a scalable file services environment. This enables customers to run a GPFS data serving cluster in which some or all of the nodes actively export the file system using NFS.

This IBM Redpaper publication shows how Cluster NFS file services are delivered and supported today through the configurable order process of the IBM Intelligent Cluster. The audience for this paper includes executive and consultant decision makers, and technical administrators who want to know how to implement this solution.

REDP-4400-01