IBM High Performance Computing
July 2006

An Introduction to GPFS

gpfsintro072506.doc

Contents

Overview

This paper provides an overview of IBM General Parallel File System (GPFS) Version 3, Release 1 for AIX 5L™ and Linux®. It includes concepts key to understanding, at a high level, available features and functionality.

This paper covers core GPFS concepts including the high-performance file system, direct storage area network (SAN) access, network based block I/O and new features Information Life Cycle (ILM) management, Network File System (NFS) V4 improvements and increased scalability with distributed token management.

The goal of this paper is to provide an introduction to GPFS features and terminology. For a more detailed description of any of these topics you should reference the product documentation See: *GPFS V3.1 documentation*.

This paper is based on the latest release of GPFS though much of the information applies to prior releases as well. It is assumed that the reader has a basic knowledge of clustering and storage networks.

What is GPFS?

IBM General Parallel File System (GPFS) is a high-performance shared-disk cluster file system. GPFS distinguishes itself from other cluster file systems by providing concurrent high-speed file access to applications executing on multiple nodes of an AIX 5L cluster, a Linux cluster, or a heterogeneous cluster of AIX 5L and Linux 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.

GPFS provides scalable high-performance data access from a single node to 2,000 nodes or more. Up to 512 Linux nodes or 128 AIX 5L nodes with access to one or more file systems are supported as a general statement and larger configurations exist by special arrangements with IBM. The largest existing configurations exceed 2,000 nodes. GPFS has been available on AIX® since 1998 and Linux since 2001. It has proven time and again on some of the world's most powerful supercomputers[1] to provide efficient use of disk bandwidth.

GPFS was designed from the beginning to support high performance computing (HPC) and has been proven very effective for a variety of applications. It is installed in clusters supporting relational databases, digital media and scalable file services. Very demanding large environments have made GPFS a solid solution for any size application.

GPFS supports various system types including IBM System p5™ and machines based on Intel or AMD processors such as IBM System x™ environment. Supported operating systems for GPFS Version 3.1 include AIX 5L V5.3 and selected versions of Red Hat and SUSE Linux distributions.

This paper introduces a number of GPFS features and describes core concepts. This includes the file system, high availability features, information life cycle management (ILM) support and various cluster architectures.

The file system

A GPFS file system is built from a collection of disks which contain the file system data and metadata. A file system can be built from a single disk or contain thousands of disks, each up to 2 Terabytes in size, storing Petabytes of data. A GPFS cluster can contain up to 32 mounted file systems. There is no limit placed upon the number of simultaneously opened files within a single file system.

---

[1] Four clusters in the top 10 as of July 7, 2006 - Source: Top 500 Super Computer Sites: http://www.top500.org/

Application interfaces

Applications can access files through standard UNIX® file system interfaces or through enhanced interfaces available for parallel programs. Parallel and distributed applications can be scheduled on GPFS clusters to take advantage of the shared access architecture. Parallel applications can concurrently read or update a common file from multiple nodes in the cluster. GPFS maintains the coherency and consistency of the file system via sophisticated byte level locking, token (lock) management and logging.

GPFS provides a unique set of extended interfaces which can be used to provide high performance for applications with demanding data access patterns. These extended interfaces are more efficient for traversing a file system, for example, and provide more features than the standard POSIX interfaces.

Performance and scalability

GPFS provides unparalleled performance especially for larger data objects and excellent performance for large aggregates of smaller objects. GPFS achieves high performance I/O by:

- Striping data across multiple disks attached to multiple nodes.
- Efficient client side caching.
- Supporting a large block size, configurable by the administrator, to fit I/O requirements.
- Utilizing advanced algorithms that improve read-ahead and write-behind file functions.
- Using block level locking based on a very sophisticated token management system to provide data consistency while allowing multiple application nodes concurrent access to the files.

GPFS recognizes typical access patterns like sequential, reverse sequential and random and optimizes I/O access for these patterns.

GPFS token (lock) management coordinates access to files or shared disks ensuring the consistency of file system data and metadata when different nodes access the same file. New in GPFS V3.1 is the ability for multiple nodes to act as token managers for a single file system. This allows greater scalability for high transaction workloads.

Along with distributed token management, GPFS provides scalable metadata management by allowing all nodes of the cluster accessing the file system to perform

file metadata operations. This key and unique feature distinguishes GPFS from other cluster file systems which typically have a centralized metadata server handling fixed regions of the file namespace. A centralized metadata server can often become a performance bottleneck for metadata intensive operations and can represent a single point of failure. GPFS solves this problem by managing metadata at the node which is using the file or in the case of parallel access to the file, at a dynamically selected node which is using the file.

Administration

GPFS provides an administration model that is consistent with standard AIX 5L and Linux file system administration while providing extensions for the clustering aspects of GPFS. These functions support cluster management and other standard file system administration functions such as quotas, snapshots, and extended access control lists.

GPFS provides functions that simplify cluster-wide tasks. A single GPFS command can perform a file system function across the entire cluster and most can be issued from any node in the cluster. These commands are typically extensions to the usual AIX 5L and Linux file system commands. GPFS provides support for the Data Management API (DMAPI) interface which is IBM's implementation of the X/Open data storage management API. This DMAPI interface allows vendors of storage management applications such as IBM Tivoli® Storage Manager (TSM) to provide Hierarchical Storage Management (HSM) support for GPFS.

Quotas enable the administrator to control and monitor file system usage by users and groups across the cluster. GPFS provides commands to generate quota reports including user, group and fileset inode and data block usage.

A snapshot of an entire GPFS file system may be created to preserve the file system's contents at a single point in time. A snapshot contains a copy of only the file system data that has been changed since the snapshot was created, using a copy-on-write technique. The snapshot function allows a backup or mirror program to run concurrently with user updates and still obtain a consistent copy of the file system as of the time that the snapshot was created. Snapshots provide an online backup capability that allows easy recovery from common problems such as accidental deletion of a file, and comparison with older versions of a file.

GPFS enhanced access control protects directories and files by providing a means of specifying who should be granted access. On AIX 5L, GPFS supports NFS V4 access control lists (ACLs) in addition to traditional ACL support. Traditional GPFS ACLs are based on the POSIX model. Access control lists (ACLs) extend the base permissions, or standard file access modes, of read (r), write (w), and execute (x) beyond the three categories of file owner, file group, and other users, to allow the definition of additional users and user groups. In addition, GPFS introduces a fourth access mode, control (c), which can be used to govern who can manage the ACL itself.

In addition to providing application file service, for example, GPFS data may be exported to clients outside the cluster through NFS or Samba including the capability of exporting the same data from multiple nodes. This allows a cluster to provide scalable file service by providing simultaneous access to a common set of data from multiple nodes. Data availability is provided by allowing access to a file from another node in the cluster, when one or more nodes are inoperable.

Data availability

GPFS is fault tolerant and can be configured for continued access to data even if cluster nodes or storage systems fail. This is accomplished though robust clustering features and support for data replication.

GPFS continuously monitors the health of the file system components. When failures are detected appropriate recovery action is taken automatically. Extensive logging and recovery capabilities are provided which maintain metadata consistency when application nodes holding locks or performing services fail. Data replication is available for journal logs, metadata and data. Replication allows for continuous operation even if a path to a disk or a disk itself fails.

Using these features along with a high availability infrastructure ensures a reliable enterprise storage solution.

Information Lifecycle Management (ILM)

GPFS is designed to help you to achieve data lifecycle management efficiencies through policy-driven automation and tiered storage management. GPFS V3.1 introduces support for information lifecycle management (ILM). The use of storage pools, filesets and user-defined policies provide the ability to better match the cost of your storage resources to the value of your data.

Storage pools allow you to create groups of disks within a file system. This is an enhancement to existing GPFS file system storage management capabilities. You can create tiers of storage by grouping your disks based on performance, locality or reliability characteristics. For example, one pool could be high performance fibre channel disks and another more economical SATA storage.

A fileset is a sub-tree of the file system namespace and provides a way to partition the namespace into smaller, more manageable units. Filesets provide an administrative boundary that can be used to set quotas and be specified in a policy to control initial data placement or data migration. Data in a single fileset can reside in one or more storage pools. Where the file data resides and how it is migrated is based on a set of rules in a user defined policy.

There are two types of user defined policies in GPFS: File placement and File management. File placement policies direct file data as files are created to the appropriate storage pool. File placement rules are determined by attributes such as file name, the user name or the fileset. File management policies allow you to move,

replicate or delete files. You can use file management policies to move data from one pool to another without changing the files location in the directory structure. They can be used to change the replication status of a file, allowing more granular control over space used for data availability. In addition, they allow you to prune the file system, deleting files as defined by policy rules. File management policies are determined by file attributes such as last access time, path name or size of the file.

Cluster configurations

GPFS supports a variety of cluster configurations independent of which file system features you require. Cluster configuration options can be characterized into three categories:

- Shared disk
- Network block I/O
- Sharing data between clusters.

Shared disk

A shared disk cluster is the most basic environment. In this configuration, the storage is SAN attached to all machines in the cluster as shown in Figure 1.

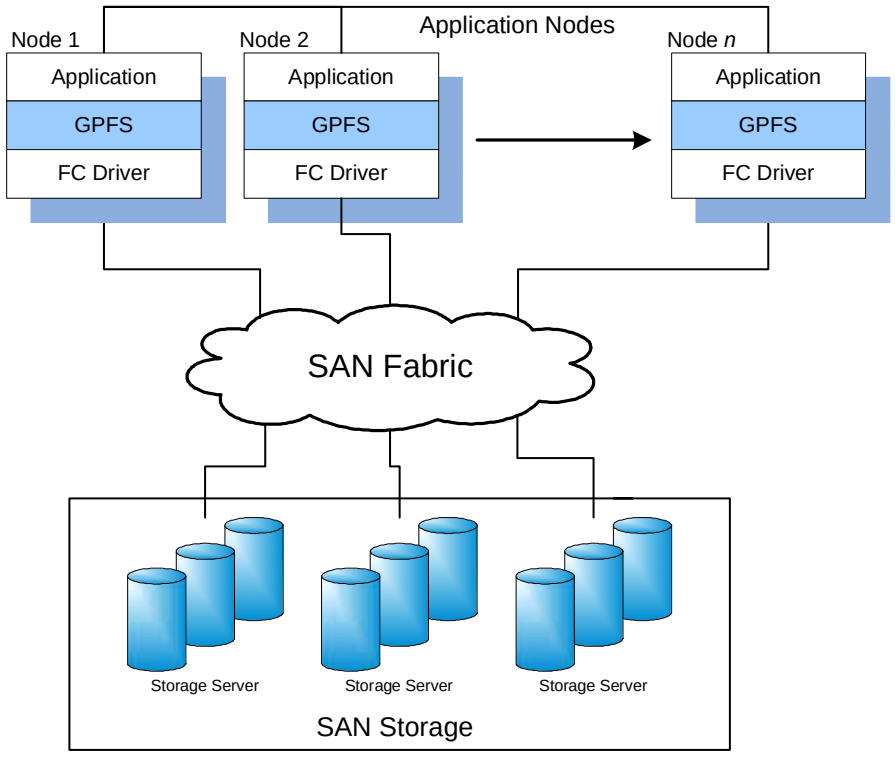

Figure 1: SAN Attached Storage

Figure 1 illustrates a fibre channel SAN. The nodes are connected to the storage via the SAN and to each other using a LAN. Data used by applications flows over the SAN and control information flows among the GPFS instances on the cluster via the LAN.

This configuration is optimal when all nodes in the cluster need the highest performance access to the data. For example, this is a good configuration for providing network file service to client systems using NFS or Samba or high-speed data access for digital media applications.

Network-based block IO

In some environments, where every node in the cluster cannot be attached to the SAN, GPFS makes use of an IBM provided network block device capability. GPFS provides a block level interface over the network called Network Shared Disk (NSD). Whether using NSD or a direct attachment to the SAN the mounted file system looks the same to the application, GPFS transparently handles I/O requests.

GPFS clusters use NSD to provide high speed data access to applications running on LAN attached nodes. Data is served to these client nodes from an NSD server, called the I/O server. In this configuration, disks are SAN attached only to the I/O servers. Each I/O server is attached to all or a portion of the disk collection. It is recommended that multiple I/O servers serve each disk to avoid a single point of failure.

GPFS uses a communications interface for the transfer of control information and data to NSD clients. These communication interfaces need not be dedicated to GPFS; but need to provide sufficient bandwidth to meet your GPFS performance expectations and for applications which share the bandwidth. New in GPFS V3.1 is the ability to designate separate IP interfaces for intra-cluster communication and the public network. This provides for a more clearly defined separation of communication traffic. To enable high speed communication GPFS supports 1Gbit and 10 Gbit Ethernet, IBM eServer™ High Performance Switch (HPS), InfiniBand and Myrinet for control and data communications.

An example of the I/O server model is shown in Figure 2.

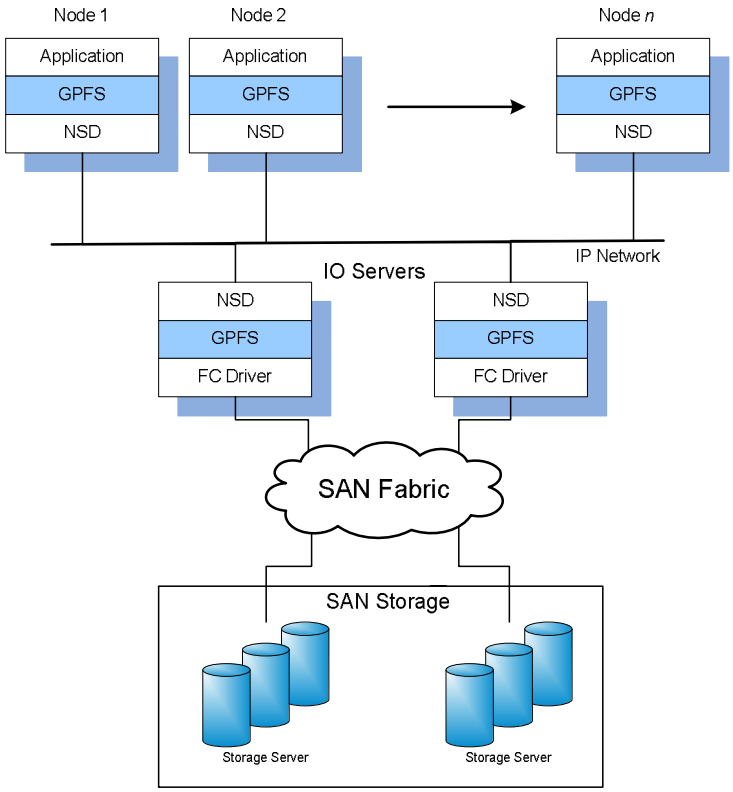

Figure 2: Network block IO

In this configuration, a subset of the total node population is defined as I/O server nodes. The I/O Server is responsible for the abstraction of disk data blocks across an IP-based network. The fact that I/O is remote is transparent to the application. Figure 2 shows an example of a configuration where a set of compute nodes are connected to a set of I/O servers via a high-speed interconnect or an IP based network such as Ethernet. In this example, data to the I/O servers flows over the SAN and both data and control information to the clients flow across the LAN.

The choice of how many nodes to configure as I/O servers is based on individual performance requirements and the capabilities of the storage subsystem. High bandwidth LAN connections should be used for clusters requiring significant data transfer. This can include 1Gbit, 10 Gbit, the use of link aggregation (etherchannel or bonding) or higher performance networks such as the HPS or InfiniBand.

The choice between SAN attachment and network block I/O is a performance and economic one. In general, using a SAN provides the highest performance; but the cost and management complexity of SANs for large clusters is often prohibitive. In these cases network block I/O provides an option.

Network block I/O is well suited to grid computing and clusters with sufficient network bandwidth between the I/O servers and the clients. For example, a grid is

effective for statistical applications like financial fraud detection, supply chain management or data mining.

Sharing data between clusters

GPFS allows you to share data across clusters. You can allow other clusters to access one or more of your file systems and you can mount file systems that belong to other GPFS clusters for which you have been authorized. A multi-cluster environment allows the administrator to permit access to specific file systems from another GPFS cluster. This feature is intended to allow clusters to share data at higher performance levels than file sharing technologies like NFS or Samba. It is not intended to replace such file sharing technologies which are tuned for desktop access or for access across unreliable network links. A multi-cluster environment requires a trusted kernel at both the owning and sharing clusters.

Multi-cluster capability is useful for sharing across multiple clusters within a physical location or across locations. Clusters are most often attached using a LAN, but in addition the cluster connection could include a SAN. Figure 3 illustrates a multi-cluster configuration with both LAN and mixed LAN and SAN connections.

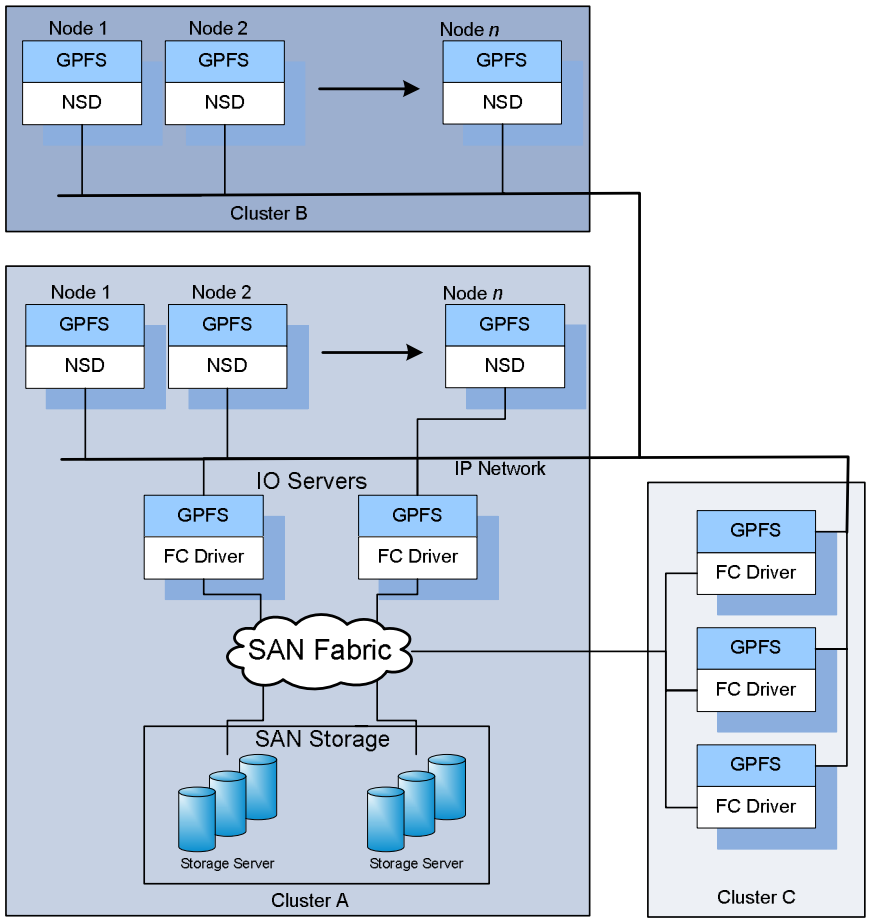

Figure 3: Multi-cluster

In Figure 3, Cluster B and Cluster C need to access the data from Cluster A. Cluster A owns the storage and manages the file system. It may grant access to file systems which it manages to remote clusters such as Cluster B and Cluster C. In this example, Cluster B and Cluster C do not have any storage but that is not always true. They could own file systems which may or may not be accessible outside their cluster. Commonly in the case where a cluster does not own storage, the nodes are grouped into clusters for ease of management. When the remote clusters need access to the data, they mount the file system by contacting the owning cluster and passing required security checks. Cluster B accesses the data through an extension of the NSD network utilizing NSD protocols. Cluster C accesses data through an extension of the storage network and controls flow through an IP network shown in Figure 3. Both types of configurations are possible.

Summary

With unparalleled scalability and performance, GPFS is the file storage solution for demanding I/O environments such as digital media with support for high bandwidth streaming data. It is also a cornerstone of grid applications such as market research, financial analytics, data mining and other large statistical workloads. Scalable file services for enterprise wide user file storage using NFS, FTP and Samba are also well suited. Lastly, numerous GPFS high-availability features provide a solid infrastructure for relational database applications and clustered web or application services.

You can get details on any of these features in the GPFS V3.1 documentation available at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfsbooks.html

See the GPFS FAQ for a current list of tested machines and Linux distribution levels and supported interconnects at:

http://publib.boulder.ibm.com/infocenter/clresctr/vxrx/topic/com.ibm.cluster.gpfs.doc/gpfs_faqs/gpfsclustersfaq.html

For more information on IBM General Parallel File System, visit ibm.com/servers/eserver/clusters/software/gpfs.html or contact your IBM representative.

NOTICES AND DISCLAIMERS

Product information and data has been reviewed for accuracy as of the date of initial publication. Product information and data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or programs(s) described herein at any time without notice.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which they operate or do business. Consult your local IBM representative or IBM authorized reseller for information about the product and services available in your area.

Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe on IBM's respective intellectually property rights, may be used instead. It is the user's responsibility to evaluate and verify the operation of any non-IBM product, program or service.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted according to the terms and conditions of the agreements under which they are provided. Neither party is responsible for the performance or interoperability the other products discussed herein or of any third-party's products.

The responsibility for use of this information or the implementation of any of these techniques is a client responsibility and depends on the customer's or user's ability to evaluate and integrate them into their operating environment. Customers or users attempting to adapt these techniques to their own environments do so at their own risk. IN NO EVENT SHALL IBM BE LIABLE FOR ANY DAMAGE ARISING FROM THE USE OF THIS INFORMATION, INCLUDING BUT NOT LIMITED TO, LOSS OF DATA, BUSINESS INTERRUPTION, LOSS OF PROFIT OR LOSS OF OPPORTUNITY.

Trademarks

IBM, the IBM logo, AIX 5L, eServer, System p5, System p5 and Tivoli are trademarks or registered trademarks of International Business Machines Corporation in the United States or other countries or both. A full list of U.S. trademarks owned by IBM may be found at: http://www.ibm.com/legal/copytrade.shtml.

UNIX is a registered trademark of The Open Group in the United States, other countries or both.

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

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

The IBM home page on the Internet can be found at: http://www.ibm.com.