Enhancing High-Performance Computing Clusters with Parallel File Systems

Large-scale scientific computation often requires significant computational power and involves large quantities of data. However, the performance of I/O subsystems within high-performance computing (HPC) clusters has not kept pace with processing and communications capabilities. Inadequate I/O capability can severely degrade overall cluster performance, particularly in the case of multi-teraflop clusters. This article discusses the advantages that parallel file systems can offer HPC clusters and how these specialized file systems can help enhance overall scalability of HPC clusters.

BY AMINA SAIFY; GARIMA KOCHHAR; JENWEI HSIEH, PH. D.; AND ONUR CELEBIOGLU

Related Categories:

File systems

High performance

High-performance computing (HPC)

IBRIX file system

Lustre file system

Parallel file systems

Parallel systems

Parallel Virtual File System (PVFS)

Performance

RAID

Visit www.dell.com/powersolutions for the complete category index to all articles published in this issue.

When building a high-performance computing (HPC) cluster, the system architect can choose among three main categories of file systems: the Network File System (NFS); storage area network (SAN) file systems, and parallel file systems. NFS is generally considered easy to use, and most system administrators working with the Linux® OS are familiar with it. However, the NFS server can be a single point of failure and NFS does not scale well across large clusters. SAN file systems can provide extremely high data throughput and are usually implemented with Fibre Channel storage. The main drawback is that a SAN file system can be costly, and each node connected to the SAN must have a Fibre Channel host bus adapter (HBA) to connect to the Fibre Channel network.

In parallel file systems, a few nodes connected to the storage—known as I/O nodes—serve data to the rest of the cluster. The main advantages a parallel file system can provide include a global name space, scalability, and the capability to distribute large files across multiple nodes. In a cluster environment, large files are shared across multiple nodes, making a parallel file system well suited for I/O subsystems. Generally, a parallel file system includes a metadata server (MDS), which contains information about the data on the I/O nodes. Metadata is the information about a file—for example, its name, location, and owner. Some parallel file systems use a dedicated server for the MDS, while other parallel file systems distribute the functionality of the MDS across the I/O nodes. This article examines three parallel file systems for HPC clusters: Parallel Virtual File System, the Lustre™ file system, and the IBRIX Fusion™ file system.

Parallel Virtual File System

Jointly developed by the Parallel Architecture Research Laboratory at Clemson University and the Mathematics and Computer Science Division at Argonne National Laboratory, Parallel Virtual File System (PVFS) is an open source parallel file system for Linux-based clusters. PVFS is very easy to install and compatible with existing binaries.

The metadata server in PVFS can be a dedicated node or one of the I/O nodes or clients. The node serving as the MDS runs a daemon called mgr, which manages

Reprinted from *Dell Power Solutions,* May 2005. Copyright © 2005 Dell Inc. All rights reserved.

the metadata for the files in the file system. In PVFS, data is distributed across multiple I/O nodes. The MDS provides information about how this data is distributed and maintains the locks on the distributed files for shared access. I/O nodes run a daemon called iod, which stores and retrieves files on local disks of the I/O nodes. PVFS creates files over the underlying file system on I/O nodes using traditional read, write, and map commands. RAID can be used on I/O nodes to provide data protection.

PVFS offers three interfaces through which PVFS files can be accessed: the native PVFS application programming interface (API), the Linux kernel interface, and the ROMIO interface. The native PVFS API allows applications to obtain metadata from the MDS. Once metadata is received, the data transfer occurs directly between the client and the I/O nodes. The PVFS Linux kernel includes a loadable module and a daemon called pvfsd, which sends the data request to the file system on behalf of applications. The pvfsd daemon uses functions from the PVFS library (libpvfs) to perform these operations. The ROMIO interface uses Message Passing Interface (MPI) to access PVFS files through the MPI I/O (MPI-IO) interface.

Lustre

Designed, developed, and maintained by Cluster File Systems, Inc., Lustre is an open source parallel file system for Linux clusters. Lustre stores file system metadata on a cluster of MDSs and stores file data as objects on object storage targets (OSTs), which directly interface with object-based disks (OBDs). The MDSs maintain a transactional record of high-level file and file system changes. They support all file system namespace operations such as file lookups, file creation, and file and directory attribute manipulation—directing the actual I/O requests to OSTs, which manage storage that is physically located on underlying OBDs. The MDS cluster is highly available and does not have a single point of failure.

In a Lustre-based cluster, clients contact the MDSs to access a file to determine which objects on particular storage controllers store which part of the file and to determine the striping pattern. After obtaining information about the location of data on the OSTs, clients establish direct connections to the OSTs that contain sections of the desired file, and then perform logical reads and writes to the OSTs. Periodically, the OSTs update the MDSs with new file sizes. OSTs are responsible for actual file system I/O and for interfacing with the underlying physical storage (the OBDs).

The interaction between an OST and the actual storage device occurs through a device driver, which enables Lustre to leverage existing Linux file systems and storage devices. For example, Lustre currently provides OBD device drivers that support Lustre data storage within journaling Linux file systems such as ext3, journaled file system (JFS), ReiserFS, and XFS. Lustre can also be used with specialized third-party OSTs like those provided by BlueArc. As part of handling I/O to the physical storage, OSTs manage locking, which allows concurrent access to the objects. File locking is distributed across the OSTs that constitute the file system, and each OST handles the locks for the objects that it stores.

Lustre also can work with object-based devices that follow the Object-based Storage Devices (OSD, formerly called OBSD) specification, as well as with block-based SCSI and IDE devices. The OSD specification handles objects or node-level data instead of byte-level data. Because most hardware vendors do not support OSD, the Lustre distribution comes with Linux device drivers that emulate OSD.

IBRIX Fusion

IBRIX Fusion is a commercial parallel file system developed by IBRIX, Inc. In traditional parallel computing terms, the architecture of IBRIX Fusion can be described as a loosely-coupled approach to distributing the metadata of the file system. IBRIX Fusion has a segmented architecture that is based on the divide-and-conquer principle. The file system is divided into multiple segments that are owned by I/O nodes known as segment servers. Each segment server maintains the metadata and locking of the files that are stored in its segments. Figure 1 shows a typical configuration using the IBRIX Fusion file system.

An allocation policy determines where—that is, in which segment—files and directories are placed, and this placement occurs dynamically when each file and directory is created. System administrators set allocation policies in accordance with the anticipated access patterns and specific criteria relevant to each installation, such as performance or manageability. Individual files can be striped across multiple segments to provide high throughput.

Segments can be migrated from one server to another while the file system is actively in use to provide load balancing. Additional storage can be added to the file system without increasing the number of servers by creating segments and distributing them

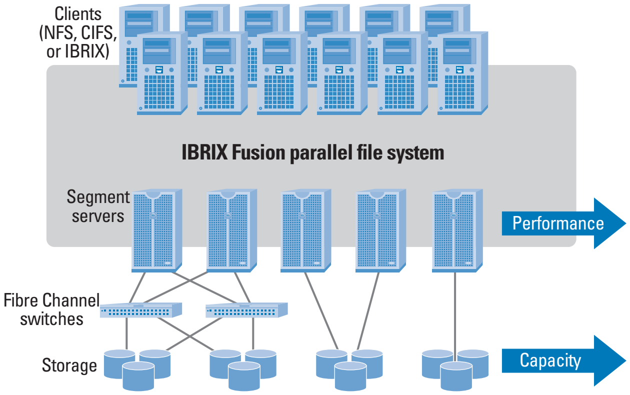

Figure 1. A typical IBRIX Fusion configuration

Reprinted from *Dell Power Solutions,* May 2005. Copyright © 2005 Dell Inc. All rights reserved.

among existing servers. Segment servers can be configured with failover protection so that designated standby servers can automatically take control of a server's segments and network identity in the event of a failure.

Clients can access the file system either as a locally mounted cluster file system using the IBRIX® driver or using network attached storage (NAS) protocols such as NFS and Common Internet File System (CIFS). A client using the IBRIX driver is aware of the segmented architecture of the file system and, based on the file and directory being accessed, can route requests directly to the segment server that owns the file it is requesting. Clients using NAS protocols must mount the file system from one (or more) of the segment servers. All requests are sent to the mounting server, which performs the required routing.

IBRIX Fusion can be deployed with segment servers configured to provide data over a SAN connection or from local storage. When a data request is made to a segment server, if the file resides on a segment owned by that server, the data is transferred directly to the client. If the file resides on a segment owned by a different server but is accessible over the SAN, then the requesting server obtains the metadata from the owning server and completes the I/O request. If the file is not accessible over the SAN, then the I/O request is completed by the owning server over the IP network.

Loss of access to one or more segments does not render the entire namespace inaccessible. Individual segments can be taken offline temporarily for maintenance operations and then returned to the file system.

Performance and scalability benefits of parallel file systems in HPC cluster environments

Parallel file systems such as PVFS, Lustre, and IBRIX Fusion are well suited for HPC cluster environments and have capabilities that fulfill critical I/O subsystem requirements. These file systems are designed to provide cluster nodes with shared access to data in parallel. They enable high performance by allowing system architects to use various storage technologies and high-speed interconnects. Parallel file systems also can scale well as an organization's storage needs grow. And by providing multiple paths to storage, parallel file systems can provide high availability for HPC clusters.

References

The Parallel Virtual File System Project. *The Parallel Virtual File System home page.* www.parl.clemson.edu/pvfs.

Cluster File Systems, Inc. *Lustre home page.* www.lustre.org.

Kashyap, Monica; Jenwei Hsieh, Ph.D.; Christopher Stanton; and Rizwan Ali. "The Parallel Virtual File System for High-Performance Computing Clusters." *Dell Power Solutions,* November 2002.

Saify, Amina; Ramesh Radhakrishnan, Ph.D.; Sudhir Srinivasan, Ph.D.; and Onur Celebioglu. "Achieving Scalable I/O Performance in High-Performance Computing Environments." *Dell Power Solutions,* February 2005.

Amina Saify is a member of the Scalable Systems Group at Dell. Amina has a bachelor's degree in Computer Science from Devi Ahilya University (DAVV) in India and a master's degree in Computer and Information Science from The Ohio State University.

Garima Kochhar is a systems engineer in the Scalable Systems Group at Dell. She has a B.S. in Computer Science and Physics from Birla Institute of Technology and Science (BITS) in Pilani, India, and an M.S. from The Ohio State University, where she worked in the area of job scheduling.

Jenwei Hsieh, Ph.D., is an engineering manager in the Scalable Systems Group at Dell, where he is responsible for developing high-performance clusters. His work in the areas of multimedia computing and communications, high-speed networking, serial storage interfaces, and distributed network computing has been published extensively. Jenwei has a B.E. from Tamkang University in Taiwan and a Ph.D. in Computer Science from the University of Minnesota.

Onur Celebioglu is an engineering manager in the Scalable Systems Group at Dell and is responsible for developing HPC clustering products. His current areas of focus are networking and HPC interconnects. Onur has an M.S. in Electrical and Computer Engineering from Carnegie Mellon University.

FOR MORE INFORMATION

IBRIX:
www.ibrix.com

Linux NFS:
nfs.sourceforge.net

Common Internet File System:
samba.org/cifs

Fibre Channel Industry Association:
www.fibrechannel.org

Dell™ NAS products:
www1.us.dell.com/content/products/compare.aspx/nasto?c=us&cs=04&l=en&s=bsd

LSI Logic adapters:
www.lsilogic.com/products/fibre_channel_hbas/index.html

BlueArc:
www.bluearc.com

Longoria, Gerald. "Resolving Data Storage Demands with SANs." *Dell Power Solutions,* Issue 1, 2000.

Reprinted from *Dell Power Solutions,* May 2005. Copyright © 2005 Dell Inc. All rights reserved.