LUSTRE™ FILE SYSTEM

High-Performance Storage Architecture and
Scalable Cluster File System
White Paper
December 2007

Abstract

This paper provides basic information about the Lustre™ file system. Chapter 1 discusses general characteristics and markets in which the Lustre file system is strong. Chapter 2 describes a typical Lustre file system configuration. Chapter 3 provides an overview of Lustre networking (LNET). Chapter 4 introduces Lustre capabilities that support high availability and rolling upgrades. Chapter 5 discusses file storage in a Lustre file system. Chapter 6 describes some additional features of the Lustre file system. And Chapter 7 provides information about a how a Lustre file system compares to other shared file systems.

Table of Contents

Chapter 1

Introducing the Lustre File System

Lustre is a storage architecture for clusters. The central component is the Lustre file system, a shared file system for clusters. The Lustre file system is currently available for Linux and provides a POSIX-compliant UNIX® file system interface. A complimenting Solaris version is planned for 2008.

The Lustre architecture is used for many different kinds of clusters. It is best known for powering seven of the ten largest high-performance computing (HPC) clusters in the world, with tens of thousands of client systems, petabytes (PB) of storage and hundreds of gigabytes per second (GB/sec) of I/O throughput. Many HPC sites use Lustre as a site-wide global file system, servicing dozens of clusters on an unprecedented scale. IDC lists Lustre as the file system with the largest market share in HPC. (Source: *IDC's HPC User Forum Survey, 2007* HPC Storage and Data Management: User/Vendor Perspectives and Survey Results)

The scalability offered by Lustre deployments has made them popular in the oil and gas, manufacturing, rich media, and finance sectors. Most interestingly, a Lustre file system is used as a general-purpose, datacenter back-end file system at a variety of sites, from Internet service providers (ISPs) to large financial institutions. With upcoming enhancements to wide-area support in Lustre networking (LNET) and storage software, the deployments in these market segments should become even more important.

The scalability of a Lustre file system reduces the need to deploy many separate file systems, such as one for each cluster or, even worse, one for each NFS file server. This leads to profound storage management advantages, for example, avoiding the maintenance of multiple copies of data staged on multiple file systems. Indeed, major HPC datacenters claim that for this reason they require much less aggregate storage with a Lustre file system than with other solutions. Hand in hand with aggregating file system capacity with many servers, I/O throughput is also aggregated and scales with additional servers. Moreover, throughput or capacity can be easily adjusted after the cluster is installed by adding servers dynamically.

Because Lustre software is open source software, it has been adopted by a number of other computing companies and integrated with their offerings. Both Red Hat and Novell (SUSE) offer kernels with Lustre patches for easy deployment. Some 10,000 downloads of Lustre software occur every month. Hundreds of clusters are supported Lustre software, with probably many more unsupported installations.

The Lustre architecture was first developed at Carnegie Mellon University as a research project in 1999. In 2003, Lustre 1.0 was released and was immediately used on many large production clusters with groundbreaking I/O performance. This performance was due in large part to enhancements to the Linux ext3 file system with high-performance enterprise features.

Chapter 2

Lustre Clusters

Lustre clusters contain three kinds of systems:

- File system clients, which can be used to access the file system
- Object storage servers (OSS), which provide file I/O service
- Metadata servers (MDS), which manage the names and directories in the file system

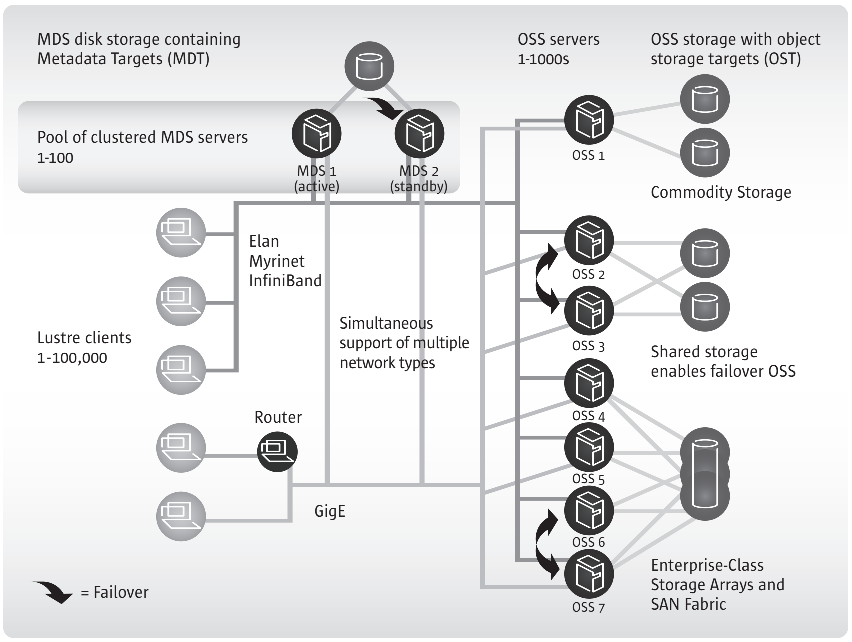

Figure 1. Systems in a Lustre cluster

The following table shows the characteristics associated with each of the three types of systems.

	Typical number of systems	Performance	Required attached storage	Desirable hardware characteristics
Clients	1–100,000	1 GB/sec I/O, 1000 metadata ops	None	None
OSS	1–1000	500 — 2.5 GB/sec	File system capacity/OSS count	Good bus bandwidth
MDS	2 (in the future 2–100)	3000–15,000 metadata ops/sec (operations)	1–2% of file system capacity	Adequate CPU power, plenty of memory

The storage attached to the servers is partitioned, optionally organized with logical volume management (LVM) and formatted as file systems. The Lustre OSS and MDS servers read, write, and modify data in the format imposed by these file systems. Each OSS can be responsible for multiple object storage targets (OSTs), one for each volume and I/O traffic is load balanced against servers and targets. Depending on the server's hardware, an OSS typically serves between 2 and 25 targets, each target up to 8 terabytes (TB) in size. The capacity of a Lustre file system is the sum of the capacities provided by the targets. An OSS should also balance the network bandwidth between the system network and the attached storage to prevent any network bottlenecks.

For example, 64 OSS servers, each with two 8-TB targets, provide a file system with a capacity of nearly 1 PB[1]. If this system uses 16 1-TB SATA disks, it may be possible to get 50 MB/sec from each drive, providing up to 800 MB/sec of disk bandwidth. If this system is used as a storage back-end with a system network such as InfiniBand, which supports a similar bandwidth, then each OSS could provide 800 MB/sec of end-to-end I/O throughput. It is important to note that the OSS must provide inbound and outbound bus throughput of 800 MB/sec simultaneously. The cluster could see aggregate I/O bandwidth of 64x800, or about 50 GB/sec. The architectural constraints described here are simple, however, in practice, extremely careful hardware selection, benchmarking and integration are required to obtain such results, which are tasks best left to experts.

1. Future Lustre file systems may feature server network striping (SNS). At that time, the file system capacity will be reduced according to the underlying RAID pattern used by SNS in that deployment.

Often OSS servers do not use internal drives, but instead use a storage array attached over Fibre Channel (FC) or Serial Attached SCSI (SAS) connections. In each case, hardware or software RAID is desirable with RAID 5 or RAID 6 striping patterns. OSS memory is used for caching read-only files and, in some cases, dirty data from writes. The CPU utilization of the OSS is currently minimal when Remote Direct Memory Access (RDMA)-capable networks are used (all networks are RDMA capable except TCP/IP). In the future, CPU utilization will increase as hardening of the disk file system is implemented. Software RAID 5 consumes about one processor core for every 300 MB/sec.

In a Lustre file system, storage is only attached to the server nodes, not to the client nodes. If failover capability is desired, storage must be attached to multiple servers. In all cases, the use of storage area networks (SANs) with expensive switches can be avoided because point-to-point connections between the servers and the storage arrays will normally provide the simplest and best attachments.

For the MDS nodes, the same considerations hold. Storage must be attached for Lustre metadata, for which 1–2 percent of the file system capacity is adequate. However, the data access pattern for MDS storage is quite different from the OSS storage: the former is a metadata access pattern with many seeks and read-and-write operations of small amounts of data, while the latter is an I/O access pattern, which typically involves large data transfers. High throughput to the MDS storage is not important. Therefore, it is recommended that a different type of storage be used for the MDS, for example FC or SAS drives, which provide much lower seek times. Moreover, for low levels of I/O, the RAID 5 and 6 patterns are very nonoptimal, and a RAID 0+1 pattern yields much better results. Lustre uses journaling file system technology on the targets, and for a MDS, an approximately 20-percent performance gain can sometimes be obtained by placing the journal on a separate device. The MDS typically requires CPU power, and at least four processor cores are recommended.

Lustre file systems are easy to configure. First, the Lustre software is installed, then the MDT and OST partitions are formatted using the standard UNIX mkfs command. Next, the volumes carrying the Lustre file system targets are mounted on the server nodes as local file systems. Finally, the Lustre client systems are mounted in a way very similar to NFS mounts. Figure 2 shows the configuration commands for the cluster shown in Figure 3.

On the MDS mds.your.org@tcp0:

```
mkfs.lustre --mdt --mgs --fsname=large-fs /dev/sdamount -t
lustre /dev/sda /mnt/mdt
```

On OSS1:

```
mkfs.lustre --ost --fsname=large-fs --mgsnode=mds.your.
org@tcp0 /dev/sdb mount -t lustre /dev/sdb /mnt/ost1
```

On OSS2:

```
mkfs.lustre --ost --fsname=large-fs --mgsnode=mds.your.
org@tcp0 /dev/sdc mount -t lustre /dev/sdc /mnt/ost2
```

On clients:

```
mount -t lustre mds.your.org:/large-fs /mnt/lustre-client
```

Figure 2. Lustre configuration commands

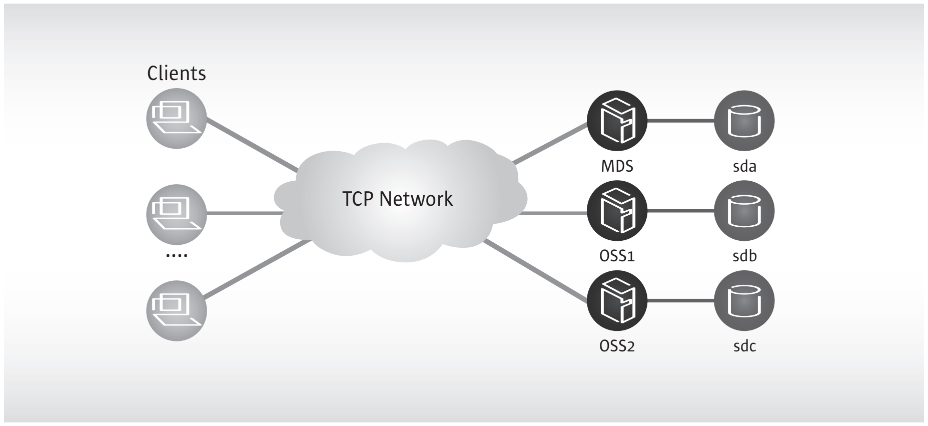

Figure 3. A simple Lustre cluster

Chapter 3

Lustre Networking

In a cluster with a Lustre file system, the system network is the network connecting the servers and the clients. The disk storage behind the MDS and OSS servers in a Lustre file system is connected to these servers using traditional SAN technologies, but this SAN does not extend to the Lustre client systems and typically does not require SAN switches. LNET is only used over the system network, where it provides all communication infrastructure required by the Lustre file system.

Key features of LNET include:

- RDMA, when supported by underlying networks such as Elan, Myrinet, and InfiniBand
- Support for many commonly used network types such as InfiniBand and IP
- High-availability and recovery features enabling transparent recovery in conjunction with failover servers
- Simultaneous availability of multiple network types with routing between them

The performance of LNET is extremely high. It is common to see end-to-end throughput over GigE networks in excess of 110 MB/sec, InfiniBand double data rate (DDR) links reach bandwidths up to 1.5 GB/sec, and 10GigE interfaces provide end-to-end bandwidth of over 1 GB/sec.

LNET has numerous other features offering rich choices for deployments. These are discussed in the *Lustre Networking* white paper.

Chapter 4
High Availability and Rolling Upgrades

Servers in a cluster are often equipped with an enormous number of storage devices and serve anywhere from dozens to tens of thousands of clients. A cluster file system should handle server reboots or failures transparently through a high-availability mechanism such as failover. When a server fails, applications should merely perceive a delay in the execution of system calls accessing the file system.

The absence of a robust failover mechanism can lead to hanging or failed jobs, requiring restarts and cluster reboots, which are extremely undesirable. The Lustre failover mechanism delivers call completion that is completely application transparent.

A robust failover mechanism, in conjunction with software that offers interoperability between versions, is needed to support rolling upgrades of file system software on active clusters. The Lustre recovery feature allows servers to be upgraded without the need to take the system down. The server is simply taken offline, upgraded, and restarted (or failed over to a standby server prepared with the new software). All active jobs continue to run without failures, merely experiencing a delay.

Lustre MDS servers are configured as an active/passive pair, while OSS servers are typically deployed in an active/active configuration that provides redundancy without extra overhead, as shown in Figure 4. Often, the standby MDS is the active MDS for another Lustre file system, so there are no nodes idle in the cluster.

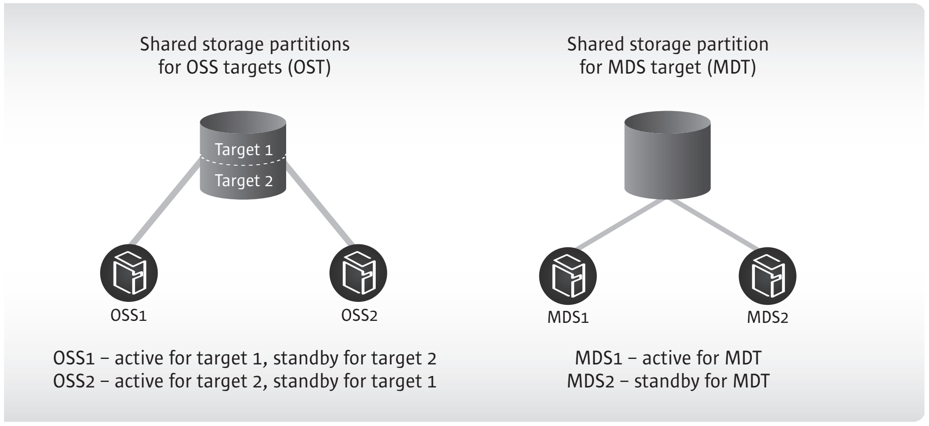
Shared storage partitions for OSS targets (OST)
Shared storage partition for MDS target (MDT)
Target 1
Target 2
OSS1
OSS2
MDS1
MDS2
OSS1 – active for target 1, standby for target 2
OSS2 – active for target 2, standby for target 1
MDS1 – active for MDT
MDS2 – standby for MDT

Figure 4. Lustre failover configurations for OSS and MDS servers

Although a file system checking tool `(lfsck)` is provided for disaster recovery, journaling and sophisticated protocols resynchronize the cluster within seconds, without the need for a lengthy fsck. Sun guarantees version interoperability between successive minor versions of the Lustre software. As a result, the Lustre failover capability is now regularly used to upgrade the software without cluster downtime.

Chapter 5
Where Are the Files?

Traditional UNIX disk file systems use inodes, which contain lists of block numbers where the file data for the inode is stored. Similarly, one inode exists on the MDT for each file in the Lustre file system. However, in the Lustre file system, the inode on the MDT does not point to data blocks, but instead points to one or more objects associated with the files. This is illustrated in Figure 5.

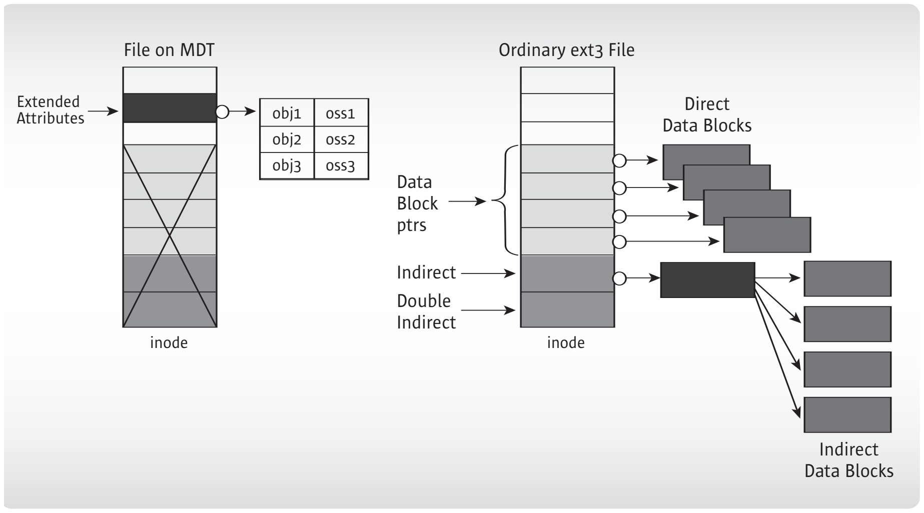

Figure 5. MDS inodes point to objects; ext3 inodes point to data

These objects are implemented as files on the OST file systems and contain file data. Figure 6 shows how a file open operation transfers the object pointers from the MDS to the client when a client opens the file, and how the client uses this information to perform I/O on the file, directly interacting with the OSS nodes where the objects are stored.

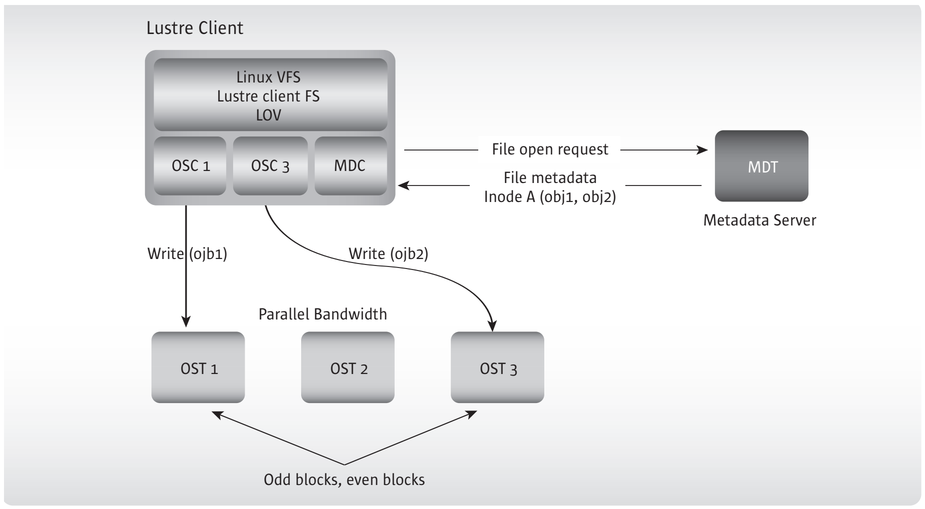

Figure 6. File open and file I/O in the Lustre file system

If only one object is associated with an MDS inode, that object contains all the data in that Lustre file. When more than one object is associated with a file, data in the file is "striped" across the objects.

Before exploring striping, some benefits from this arrangement are already clear. The capacity of a Lustre file system equals the sum of the capacities of the storage targets. The aggregate bandwidth available in the file system equals the aggregate bandwidth offered by the OSS servers to the targets. Both capacity and aggregate I/O bandwidth scale simply with the number of OSS servers.

Striping allows parts of files to be stored on different OSTs as shown in Figure 7. A RAID 0 pattern, in which data is striped across a certain number of objects, is used; the number of objects is called the stripe_count. Each object contains chunks of data. When the chunk being written to a particular object exceeds the stripe_size, the next chunk of data in the file is stored on the next target.

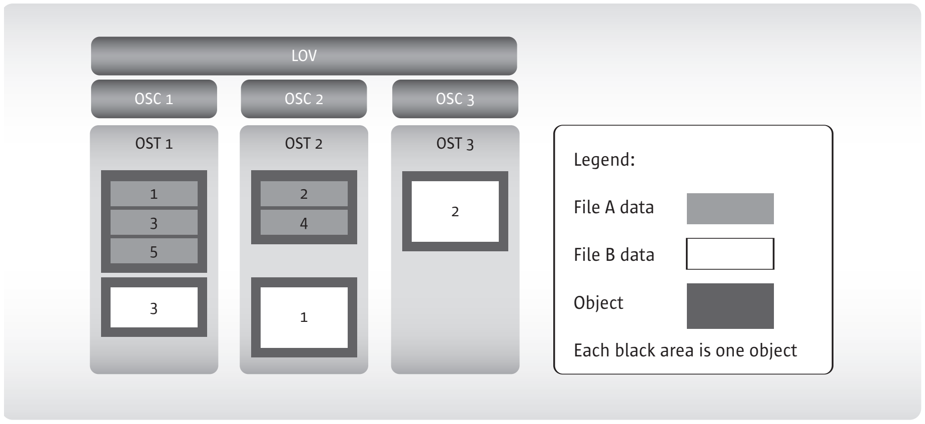

Figure 7. Files striped with a stripe count of 2 and 3 with different stripe sizes

Working with stripe objects leads to interesting behavior. For example, Figure 8 shows a rendering application in which each client node renders one frame. The application uses a shared file model where the rendered frames of the movie are written into one file. The file that is written can contain interesting patterns, such as objects without any data. Objects can also have sparse sections into which client 6 has written data, as shown in the third object in Figure 8.

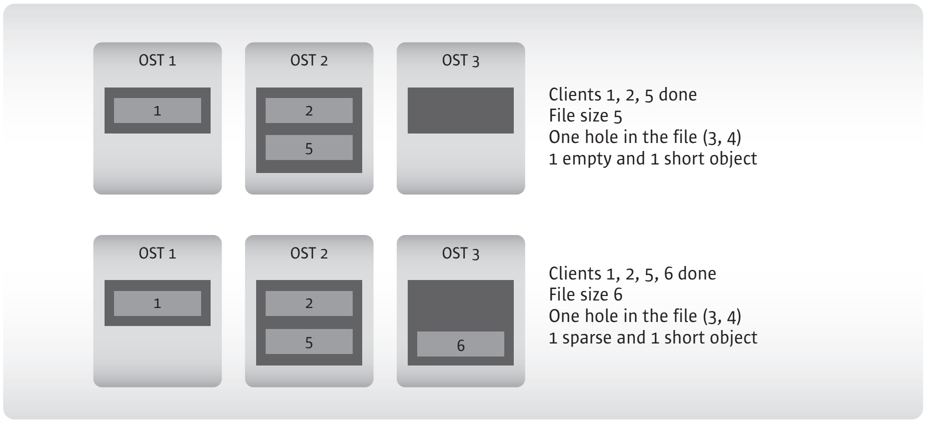
OST 1
OST 2
OST 3
1
2
5
Clients 1, 2, 5 done
File size 5
One hole in the file (3, 4)
1 empty and 1 short object
OST 1
OST 2
OST 3
1
2
5
6
Clients 1, 2, 5, 6 done
File size 6
One hole in the file (3, 4)
1 sparse and 1 short object

Figure 8. A compute application rendering a movie file with three objects

Striping of files presents several benefits. For example, maximum file size is not limited by the size of a single target. In fact, Lustre software can stripe files over up to 160 targets, and each target can support a maximum disk use of 8 TB by a file. This leads to a maximum disk use of 1.48 PB by a file in a Lustre file system. While this may seem enormous, some customers have applications that write 100-TB files. The maximum file size is much larger (2^64 bytes), but the file cannot have more than 1.48 PB of allocated data; hence, a file larger than 1.48 PB must have many sparse sections. While a single file can only be striped over 160 targets, Lustre file systems have been built with almost 5000 targets, which is enough to support a 40-PB file system.

Another benefit of striped files is that the I/O bandwidth to a single file is the aggregate I/O bandwidth to the objects in a file, and this can be as much as the bandwidth of up to 160 servers.

Chapter 6
Additional Features

This section discusses some additional features of the Lustre file system.

Interoperability — The Lustre file system runs on many CPU architectures (Intel Architecture 32-bit/64-bit, x86/x64, and PowerPC), and clients and servers are interoperable between these platforms. Moreover, Lustre software strives to provide interoperability between adjacent software releases. Versions 1.4.X (X>7) and version 1.6.0 can interoperate when clients and servers are mixed. However, future Lustre releases may require “server first” or “all nodes at once” upgrade scenarios.

Access control list (ACL) — The Lustre security model is currently that of a UNIX file system, enhanced with POSIX ACLs. A few additional noteworthy features available today include root squash and connecting from privileged ports only.

Quota — User and group quotas are available for the Lustre system.

OSS addition — The capacity of a Lustre file system and the aggregate cluster bandwidth can be increased without interrupting any operations by adding a new OSS with OSTs to the cluster.

Controlled striping — The default stripe count and stripe size can be controlled in various ways. The file system has a default setting that is determined at format time. Directories can be given an attribute so that all files under that directory (and recursively under any subdirectory) have a striping pattern determined by the attribute. Finally, utilities and application libraries are provided to control the striping of an individual file at creation time.

Snapshots — Ultimately, the Lustre file servers use volumes attached to the server nodes. Lustre software comes with a utility to create a snapshot of all volumes using LVM snapshot technology and to group the snapshots together in a snapshot file system that can be mounted with the Lustre system.

Backup tools — The Lustre 1.6 file system comes with two tools to support backups. One is a file scanner that can scan file systems very fast to find files modified from a certain point in time. The utility provides a list of path names of modified files that can be processed in parallel by other utilities, such as rsync, using multiple clients. The second utility is a modified version of the star utility, which can back up and restore Lustre stripe information.

Many current and future features are described in the Lustre roadmap and documented in the Lustre Operations Manual. Of these features, some are planned to be delivered during the coming year, while others are further out. For more details, visit www.sun.com/lustre.

Chapter 7

The Lustre File System Compared to Other Shared File Systems

The performance of a Lustre file system compares favorably to other shared file system solutions, including shared-disk file systems, solutions in which shared file systems are exported using NFS protocol, and object architecture-based systems.

7.1 Overview of solutions

Shared disk file systems were introduced predominantly by Digital VAX/VMS clusters in the early 1980s. They rely on a SAN based on Fibre Channel, iSCSI, or InfiniBand technology. The IBM General Parallel File System (GPFS), PolyServe storage solutions, Silicon Graphics clustered file system (CxFS), Red Hat Global File System (GFS), and TerraScale Technologies TerraFS all fall into this category. The architecture of these file systems mirrors that of local disk file systems, and performance for a single client is extremely good. Although concurrent behavior suffers from an architecture that is not optimized for scalability, these systems offer failover with varying degrees of robustness. GPFS has been very successful for clusters of up to a few hundred nodes. Typically, SAN performance on Fibre Channel is reasonable, but it cannot compete with clients that use InfiniBand, Quadrics, or Myricom networks with native protocols.

To limit the scalability problems encountered by shared disk file systems, systems such as GPFS, CxFS, GFS, and PolyServe Matrix Server are often used on an I/O sub-cluster that exports NFS. Isilon offers an appliance for this purpose. Each of the I/O nodes then exports the file system through NFS version 2 or 3. For NFS version 4, such exports are more complex due to the requirement for managing shared state among the NFS servers. While the scalability of NFS improves, the layering introduces further performance degradation, and NFS failover is rarely completely transparent to applications. NFS offers neither POSIX semantics or good performance.

A well-tuned Lustre cluster will normally outperform a NFS protocol-based cluster. Figure 9 and Figure 10 compare Luster file system performance with that of other file systems for parallel writes and for creation of metadata files in a single directory.[2]

Several systems offer novel architectures to address scalability and performance problems. Ibrix offers a symmetric solution, but little is publicly known about its architecture, semantics, and scalability. Panasas offers a server hardware solution combined with client file system software. It makes use of smart object iSCSI storage devices and a metadata server that can serve multiple file sets. Good scaling and security are achievable, even though all file locking is done by a single metadata server. The Panasas system uses TCP/IP networking. Lustre's architecture is similar, but is an open source, software-only solution running on commodity hardware.

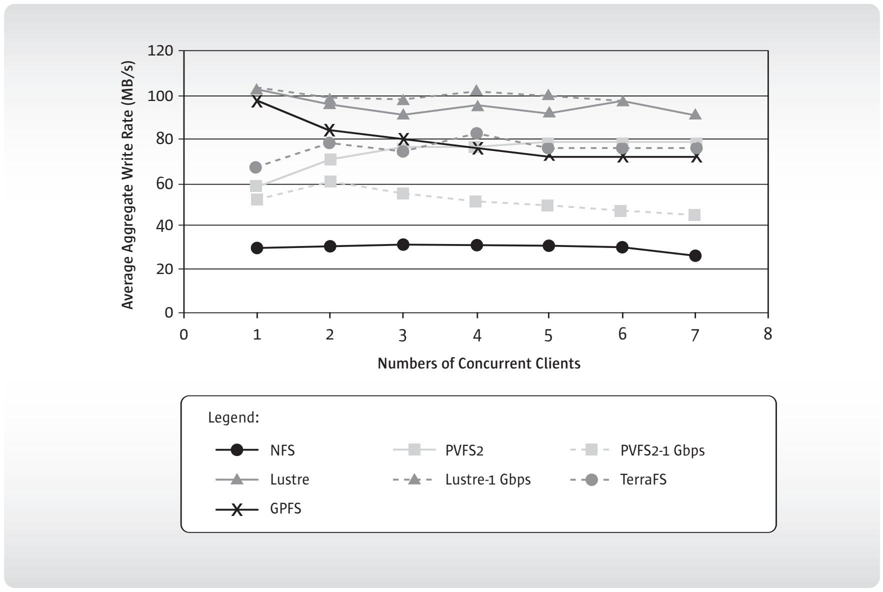

Figure 9. Xeon cluster average aggregate write bandwidth by number of clients

2. Figure 9 and Figure 10 taken from Shared Parallel Filesystems in Heterogeneious Linux Multi-Cluster Environments by Jason Cope*, Michael Oberg*, Henry M. Tufo*†, and Matthew Woitaszek*, (*University of Colorado, Boulder, †National Center for Atmospheric Research).

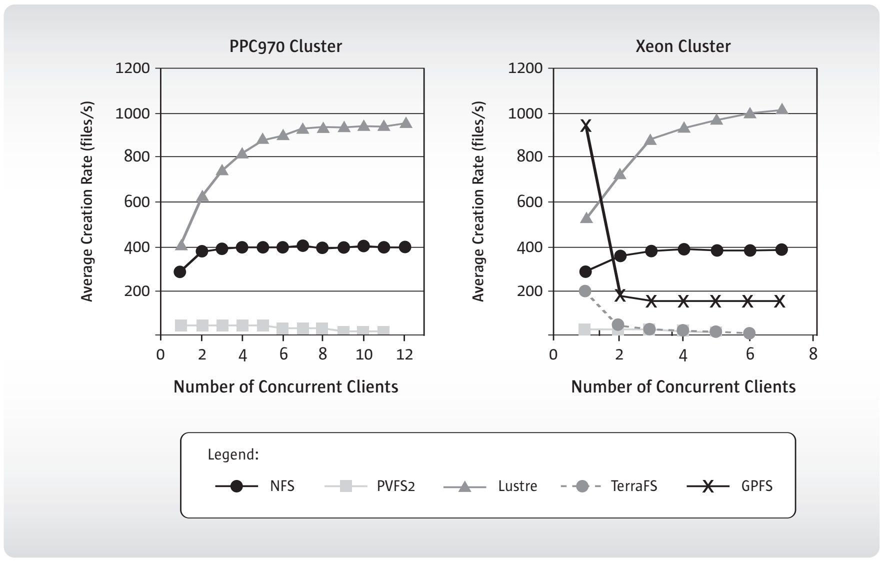

Figure 10. Average aggregate file creation in a single directory by number of clients

7.2 Shared file systems compared

The following table shows how the Lustre file system is differentiated from other shared file systems.

Aspect/FS	Lustre	GPFS	Panasas	StoreNext	Ibrix	QFS	NFS
License	Open Source	Proprietary	Proprietary	Proprietary	Proprietary	Proprietary	Clients open, most servers proprietary
Type of Solution	Software	Generally bundled with IBM hardware	Storage blades with disks	Software	Software	Software	Software and hardware
Availability	Numerous partners: Cray, Dell, HP, , DDN, Hitachi, Terascala, Red Hat, and SUSE	IBM	Panasas	Quantum	Ibrix	Sun	Widely available
Scalability (number of clients)	>25,000	1000	Hundreds	Dozens	Hundreds	256	Dozens
Networks supported	Most networks	IP, InfiniBand, and Federation	IP	SAN	IP	FC SAN	IP
Architecture	Object storage architecture	Traditional VAX cluster file system architecture	Object storage architecture with central lock service	Traditional VAX cluster file system architecture	Unknown	Distributed file system	Not a cluster file system, but a well-known standard
Modifiable	Integrated with numer-ous new networks and storage devices	Unknown	Only offered with Panasas hardware	No	Unknown	No	No

Chapter 8

About Sun

A singular vision, The Network is the Computer™, drives Sun in delivering industry-leading technologies that focus on the whole system — where computers, software, storage, and services combine. With a proven history of sharing, building communities, and innovation, Sun solutions create opportunities, both social and economic, around the world. You can learn more about Sun at sun.com.

Sun Microsystems, Inc. 4150 Network Circle, Santa Clara, CA 95054 USA **Phone** 1-650-960-1300 or 1-800-555-9SUN (9786) **Web** sun.com

SunWIN #522292 Lit. #SYWP13813-0 12/07