Sardar Patel Institute of Technology

Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

Distributed Computing ISE 1

Topic: Sun Network Distributed File System

Submitted by:

Shubham Golwal: 2020300015

Shubham Ghuge: 2020300012

Vyom Doshi: 2020300009

INTRODUCTION:

Lustre is an open-source file system that was developed in 1999 and released to general production in December 2003. It was developed under the direction of the Department of Energy (DOE) and has been widely deployed in national laboratories and large supercomputing centers.

Lustre is a parallel file system, which is a type of distributed file system. However, the difference lies in how data and metadata are stored. Distributed file systems support a single global namespace, as do parallel file systems, but a parallel file system chunks up files into data blocks and spreads file data across multiple storage servers, which can be written-to and read-from in parallel. Metadata is typically stored on a separate metadata server for more efficient file look up. In contrast a distributed file system uses standard network file access and the entire file data and metadata are managed by a single storage controller. For bandwidth intensive workloads, this single point of access becomes a bottleneck for performance. The Lustre parallel file system does not suffer this single controller bottleneck but the architecture required to deliver parallel access is relatively complex, which has limited Lustre deployments to niche applications.

ARCHITECTURE:

The Lustre file system architecture separates out metadata services and data services to deliver parallel file access and improve performance. The architecture consists of a series of I/O servers called Object Storage Servers (OSSs) and persistent storage targets where the physical data resides called Object Storage Targets (OSTs), typically based on spinning disk drives. In addition, Lustre has separate metadata services and file metadata is managed by a Metadata Server (MDS) while the actual metadata is persisted on a metadata target (MDT).

The OSSs are responsible for managing the OSTs, handling all I/O requests to the OST and. A single OSS typically manages between two and eight OSTs, after which an additional OST is required to maintain performance. The OSS requires a local file system to manage file placement on the OST, typically this requires an XFS or ZFS file system.

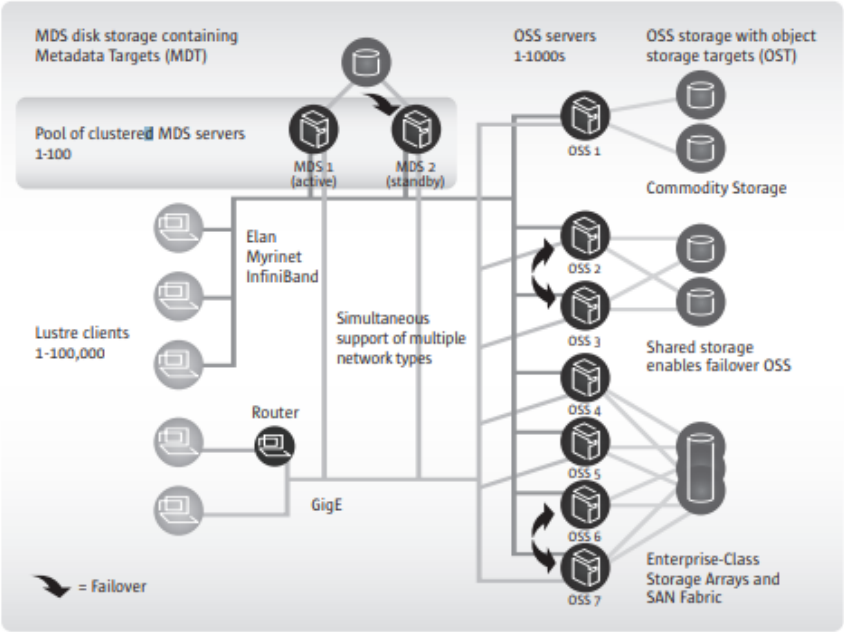
For metadata, the MDS provides metadata services managing all physical storage locations associated with each file so that I/O requests are directed to the correct set of OSSs and associated OSTs. The metadata server is never in the I/O path, a key difference from traditional NAS and clustered file systems. The MDS also requires a local file system to manage metadata placement on the MDT.

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



Systems in a Luster FS **credits to wiki.com*

THE LUSTRE INFRASTRUCTURE:

When a Lustre client wants to access a file, it sends a request to the MDS, which in turn accesses the associated MDT. The location of the data is returned to the client, which then directly accesses the OSS and associated OST. In real production environments the infrastructure is significantly more complex as the storage hardware used for MDTs and OSTs typically require

additional hardware RAID devices or the addition of ZFS for the back end file system, which provides software-based RAID protection on JBOD (Just a bunch of disks) storage.

The following diagram (figure 1) provides a simplified view of the physical configuration of a Lustre environment. It is important to note that installing Lustre is not like installing a NAS appliance, it is composed of several different services that are deployed on separate server infrastructure.

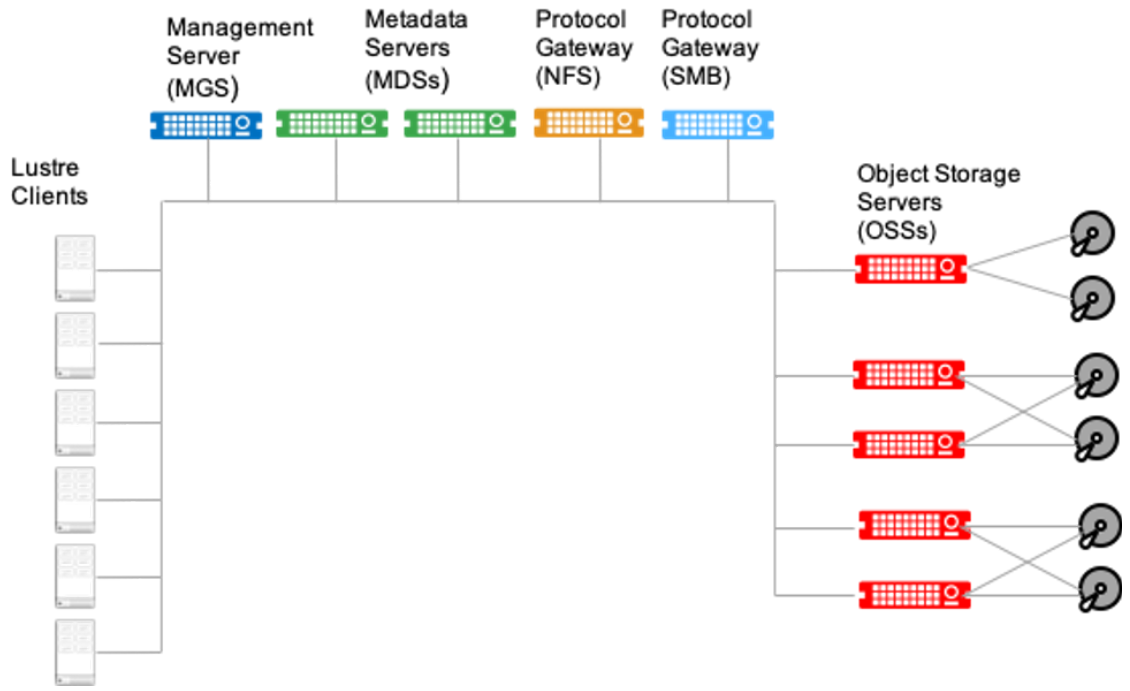
Following the diagram below, the environment requires a Lustre management service (MGS) server whose job is to support the multiple file systems running in the Lustre environment. Clients and servers have to connect to the MGS on startup to establish the file system configuration and get notifications of changes to the file systems. While not compute or storage resource intensive, the MGS can pose a single point of failure because if the host fails all the client and server services will be unavailable while the MGS is offline. In order to alleviate this, the MGS must be deployed in a high-availability configuration. Again, just like other services in Lustre, the MGS must have a local file system deployed to manage the physical data storage on the management service target (MGT).

The second set of server infrastructure is composed of metadata servers (MDSs). As described earlier, these servers manage the namespace for the Lustre file system. MDS servers are typically configured in a RAID 1 (mirror) to ensure the MGS server can access the metadata server in the event of a server failure. It requires careful consideration of how many metadata servers are required per file system and how much memory is required to support the number of Lustre clients accessing the MDS. For I/O intensive applications, getting the ratios correct can be a huge administration challenge and it is recommended to reference the Lustre tuning and sizing guide.

The third major component of server infrastructure is the object storage servers that house the file data. Typically the OSS make up the majority of the infrastructure as they manage the data services for individual files. Each OSS communicates with the OSTs that persist the physical data blocks. OSS nodes are connected over multiple paths to the physical media, protected with RAID controllers or protected using the ZFS software RAID for file protection. Each RAID or

ZFS stripe is an OST. Several strategies are utilized to manage storage services including layering file services on top of SAN infrastructure for more storage scale.

The final component of a Lustre environment is gateway servers to provide Linux and Windows user access via gateway services. These services are optional and may not be found in many HPC environments but are required if individual user access to the file system data is required.



*credits to wiki.com

LUSTRE VS WEKA PARALLEL FILE SYSTEM:

Lustre is now a 21 year old file system that has continued to be deployed in high performance environments because legacy shared file systems based on Network File System (NFS) or SMB just cannot scale to the demands of modern workloads in AI and machine learning. That said, although it delivers much better parallelism, its inherent large file optimized design limits its ability to effectively work for mixed workload and metadata intensive applications. WekaFS is a modern parallel file system which was released in 2017 to address I/O intensive workloads that require massively parallel access and huge metadata performance demands. WekaFS has been built from scratch to leverage flash based technology, delivering ultra low latency and excellent

small file performance. Unlike Lustre, which has been optimized for hard disk based storage and prefers 1MB file size, the Weka file system strips every file into 4KB chunks and serves them back in parallel. In essence, every file for Weka is a small file problem, hence the reason it can so effectively manage tiny file performance.

WekaFS has also solved the metadata choke point which too often cripples the Lustre file system. While the early design of separating data and metadata alleviated many of the problems of legacy NAS, it created a new bottleneck for modern workloads. WekaFS has solved the metadata performance issues by equally spreading data and metadata across all the nodes in the storage cluster. Each WekaFS node provides metadata services, data services, NFS and SMB access and internal tiering to S3 based object storage for massive scalability. The infrastructure is greatly simplified while the performance on small file and metadata is orders of magnitude better than the Lustre design. Performance scales linearly as more nodes are added and there is no requirement for complex inode setup to ensure the file system will scale to the size required by the application.

#			information							lo500	
	list	institution	system	storage vendor	filesystem	client	client	data	score	bw	md
	id				type	nodes	total procs			GiB/s	kIOP/s
1	sc19	WekaIO	WekaIO on AWS	WekaIO	WekaIO Matrix	345	8625	zip	938.95	174.74	5045.33
2	sc19	National Supercomputing Center in Changsha	Tianhe-2E	National University of Defense Technology	Lustre	480	5280	zip	453.68	209.43	982.78

USAGE:

High Performance Computing (HPC) clusters are created to provide extreme computational power to large scale applications. This computational power often results in the creation of very large amounts of data as well as very large individual files. For quite some time, the speed of processors and memory have risen sharply, but the performance of I/O systems has lagged behind.

While processors and memory have improved in cost/performance exponentially over the last 20 years, disk drives still essentially spin at the same speeds, and drive access times are still measured in numbers of milliseconds. As such, poor I/O performance can severely degrade the overall performance of even the fastest clusters. This is especially true of today's multi-petabyte clusters.

The Lustre file system is a parallel file system used in a wide range of HPC environments, small to large, such as AI/ML, oil and gas, seismic processing, the movie industry, and scientific research to address a common problem they all have and that is the ever increasing large amounts of data being created and needing to be processed in a timely manner. In fact it is the most widely used file system by the world's Top 500 HPC sites.

When designing a High Performance Computing (HPC) cluster, the HPC architect has three common file system options for providing access to the storage hardware. Perhaps the most common is the Network File System (NFS). NFS is a standard component in cloud computing environments. NFS is commonly included in what is known as NAS, or Network Attached Storage architectures. A second option available to choose is SAN file systems, or Storage Area Network file systems. And last, but not least, are parallel file systems. Lustre is the most widely used file system on the top 500 fastest computers in the world. Lustre is the file system of choice on 7 out of the top 10 fastest computers in the world today, over 70% of the top 100, and also for over 60% of the top 500.

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. 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 data centers 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.

PROTOCOLS OF LFS:

Lustre Networking (LNet): LNet is the high-speed data network protocol that clients use to access the file system. LNet is designed to meet the needs of large-scale compute clusters and is optimized for very large node counts and high throughput. LNet supports Ethernet, InfiniBand, Intel Omni-Path Architecture (OPA), and specific compute fabrics such as Cray Gemini.

LNet abstracts network details from the file system itself. LNet allows for full RDMA throughput and zero copy communications when available.

LNet supports routing, which provides maximum flexibility for connecting different network topologies. LNet routing provides an efficient protocol for bridging different networks, or employing different fabric technologies, such as Intel® OPA and InfiniBand. In larger file systems, with very large client counts, LNet can also support multihoming to improve

performance and reliability. For more details and guidance, see the document Configuring LNet Routers for File Systems based on Intel* EE for Lustre* Software.

Metadata Server (MDS): The MDS manages all name space operations for a Lustre file system. A file system's directory hierarchy and file information are contained on storage devices referred to as Metadata Targets (MDT), and the MDS provides the logical interface to this storage. A Lustre file system will always have at least one MDS and corresponding MDT, and more can be added to meet the scaling requirements of a particular environment. The MDS controls the allocation of storage objects on the Object Storage Servers for the file content when a file is created, and manages the opening and closing of files, file deletions and renames, and other namespace operations.

An MDT stores namespace metadata, such as filenames, directories, access permissions, and file layout, effectively providing the index for the data held on the file system. The ability to have multiple MDTs in a single file system allows directory subtrees to reside on the secondary MDTs, which is useful for isolating workloads that are especially metadata-intensive onto dedicated hardware (one could allocate an MDT for a specific set of projects, for example). Large, single directories can be distributed across multiple MDTs as well, providing scalability for applications that generate large numbers of files in a flat directory hierarchy.

Object Storage Server (OSS): OSSs provide bulk storage for the contents of files in a Lustre file system. One or more object storage servers (OSS) store file data on one or more object storage targets (OST), and a single Lustre file system can scale to hundreds of OSSs. A single OSS typically serves between two and eight OSTs (although more are possible), with the OSTs stored on direct-attached storage. The capacity of a Lustre file system is the sum of the capacities provided by the OSTs across all of the OSS hosts.

OSSs are usually configured in pairs, with each pair connected to a shared external storage enclosure that stores the OSTs. The OSTs in the enclosure are accessible via the two servers in an active-passive high availability failover configuration, to provide service continuity in the event of a server or component failure. The OSTs are mounted on only one server at a time, and are

usually evenly distributed across the OSS hosts to balance performance and maximize throughput.

Management Server (MGS): The MGS stores configuration information for all the Lustre file systems in a cluster and provides this information to other Lustre hosts. Servers and clients connect to the MGS on startup in order to retrieve the configuration log for the file system. Notification of changes to a file system's configuration, including server restarts, are distributed by the MGS.

Persistent configuration information for all Lustre nodes is recorded by the MGS onto a storage device called a Management Target (MGT). The MGS can be paired with an MDS in a high-availability configuration, with each server connected to shared storage. Multiple Lustre file systems can be managed by a single MGS.

Clients: Applications access and use file system data by interfacing with Lustre clients. A Lustre client is represented as a file system mount point on a host and presents applications with a unified namespace for all of the files and data in the file system, using standard POSIX semantics. A Lustre file system mounted on the client operating system looks much like any other POSIX file system; each Lustre instance is presented as a separate mount point on the client's operating system, and each client can mount several different Lustre file system instances concurrently.

SECURITY IN LUSTRE:

File System security is a very important aspect of a distributed file system. The standard aspects of security are authentication, authorization, and encryption. While SANs are largely unprotected, Lustre provides the OSTs with secure network attached disk (NASD) features. Rather than selecting and integrating a specific authentication service, Lustre can easily be integrated with existing authentication mechanisms using the Generic Security Service

Application Programming Interface (GSSAPI), an open standard that provides secure session communications supporting authentication, data integrity, and data confidentiality. Lustre authentication will support Kerberos 5 and PKI mechanisms as a backend for authentication. Lustre intends to provide authorization using access control lists that follow the POSIX ACL semantics. The flexibility and additional capabilities provided by ACLs are especially important in clusters that may support thousands of nodes and user accounts. Data privacy is expected to be ensured using an encryption mechanism such as that provided by the StorageTek/University of Minnesota SFS file system, where data can actually be automatically encrypted and decrypted on the client based on a shared key protocol which makes file sharing by project a natural operation. The OSTs are protected with a very efficient capability-based security mechanism which provides very significant optimizations over the original NASD protocol.

HIGH AVAILABILITY:

Lustre file system high availability features include a robust failover and recovery mechanism, making server failures and reboots transparent. Version interoperability between successive minor versions of the Lustre software enables a server to be upgraded by taking it offline (or failing it over to a standby server), performing the upgrade, and restarting it, while all active jobs continue to run, experiencing a delay while the backup server takes over the storage.

Lustre MDSes are configured as an active/passive pair exporting a single MDT, or one or more active/active MDS pairs with DNE exporting two or more separate MDTs, while OSSes are typically deployed in an active/active configuration exporting separate OSTs to provide redundancy without extra system overhead. In single-MDT filesystems, the standby MDS for one filesystem is the MGS and/or monitoring node, or the active MDS for another file system, so no nodes are idle in the cluster.

COMMUNICATION CONCEPT:

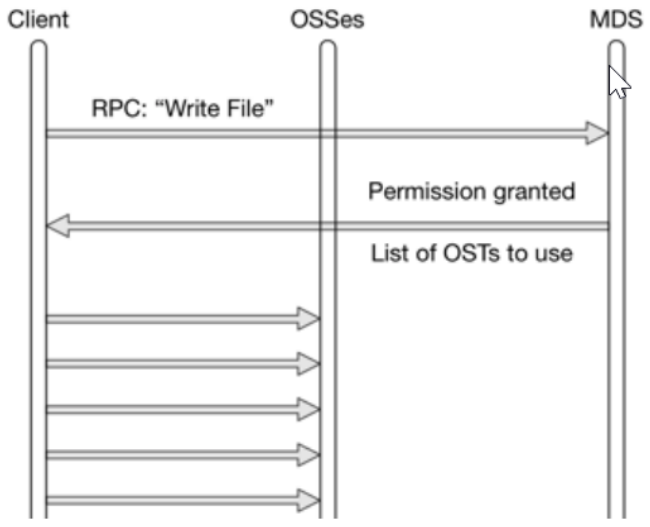
Lustre is actually a very simple concept. When a client requests to write a file to the file system, it sends a write request to the metadata server (MDS). The MDS verifies the user's identity and the file's intended location. The MDS returns a list of Object Storage Targets (OSTs) that the client can use to write the file based on the directory or file system settings. Once that response is

sent, the client only interacts with the assigned OSTs and does not need to communicate with the MDS. This is true for any file regardless of size, whether it's a few bytes or a few terabytes. And as this communication (if using InfiniBand) will be done exclusively over Remote Direct Memory Access (RDMA), the performance is exceptional and the latency is minimal (see below). The actual distribution is defined by the specific striping settings of a file, directory or file system. If the specific file write command does not have a predefined set of stripe settings, it will inherit the settings of the directory or file system to which it is written.

Data Flow The mechanism employed by Lustre to manage a write or read operation can be simplified using the following examples (note that RDMA assumes InfiniBand-based networks):

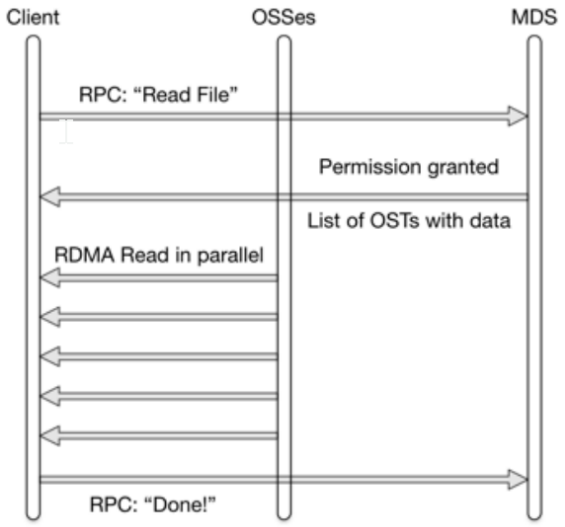
Write:

1. Client "asks" the MDS for permission to write a file. 2. MDS checks access rights, file properties, etc., and returns a list of OSTs to use. 3. The clients communicate directly with each OST, which writes the data in parallel until done (this communication continues until the entire file is written regardless of size, from KBs to TBs), and does not involve the MDS further.



Read:

1. Client “asks” the MDS for permission to read a file. 2. MDS checks access rights and file location, and returns a list of OSTs where the different stripes of the file are located. 3. The clients communicate directly with each OST, which reads the data in parallel until done reading the parts of the file the clients need. 4. Once the client is finished, it sends a single “done” to the metadata server to make the file accessible to other clients.



CONCLUDING POINTS:

- Lustre file system is an open source file system which provides significant performance as a distributed file system in HPC environments
- Lustre uses an object storage model for file I/O, and storage management to provide a substantially more efficient division of labor between computing and storage resources.
- Replicated, failover metadata Servers (MDSs) maintain a transactional record of high-level file and file system changes. Distributed Object Storage Targets (OSTs) are responsible for actual file system I/O and for interfacing with local or networked storage devices known as Object-Based Disks (OBDs).

REFERENCES:

1. https://www.researchgate.net/publication/241229733_Lustre_The_intergalactic_file_system

2. https://www.parallels.com/blogs/ras/storage-protocols/#:~:text=The%20five%20most%20commonly%20used,Server%20Message%20Block%20(SMB).
3. https://cse.buffalo.edu/faculty/tkosar/cse710/papers/lustre-whitepaper.pdf
4. https://wiki.lustre.org/images/7/77/LUG2019-Lustre_Security-Buisson.pdf
5. https://insidehpc.com/2015/04/introduction-to-the-lustre-file-system/