LUSTRE HPC FILE SYSTEM DEPLOYOMENT METHOD

Upanshu Singhal
Consultant Engineer
EMC Corporation
upanshu.singhal@emc.com

Tarun Mathur
Principal Program Manager
EMC Corporation
tarun.mathur@emc.com

EMC²

Table of Contents

Overview

With the mass adoption of clusters and explosive growth of data storage needs, I/O bandwidth challenges are becoming common in a variety of public and private sector environments. The Lustre file system is a natural fit for these places where traditional shared file systems, such as NFS, do not scale to the required aggregate throughput requirements of these clusters.
The Lustre file system is an open source shared file system designed to address the I/O needs of compute clusters spreading up to thousands of nodes. It is best known for powering the 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/s) of I/O throughput.

The main component of Lustre is the Lustre File System which is supported on a Linux Operating System and provides a POSIX- based UNIX file system interface. The ability of Lustre file system to scale capacity and performance for any need reduces the need to deploy many separate file systems such as one for each compute cluster.

Lustre aggregates and scales the storage capacity and I/O throughput of many servers and can be easily increased by adding servers dynamically. Lustre can be scaled up or down with respect to the number of client nodes, disk storage, and bandwidth. It can be deployed in a wide variety of configurations.

Installation and configuration of Lustre file system with numbers of Lustre data nodes and Lustre Clients is not just about executing RPMs; it is a complex and time consuming task. This article takes a step-by-step approach of Lustre file system installation and configuration.

This article will detail:

- Introduction to Lustre file system
- Different components of Lustre file system
- System requirement for Lustre installation
- Step-by-step approach to build Lustre components from the sources
- Step-by-step approach to install Lustre components on Linux hosts
- Step-by-step approach on configuring Lustre file system in an enterprise environment
- Introduction to basic tool set and command lines of Lustre file system
- Introduction to generating IO load on Lustre file system
- Introduction to performance measurement of Lustre file system using various tools

Benefits of this article:

- Helps understand HPC File System, its components, and usage
- Helps Professional Services to install and configure Lustre file system in an EMC-provided environment
- Single point of reference to Lustre file system and its installation and configuration
- Self-learning with minimal effort and time required from others
- Document is expandable. Any new learning will be consolidated here, no need for everybody to change/modify/update/recreate their own documents

The intended audiences for this article are engineering or professional services personnel. However, the article may also be used to publish a document for end users like a “Quick user guide” or “Lustre Cook Book” for quick installation and configuration of a HPC File System.

1. Introduction to Lustre File System

Lustre is a GNU General Public licensed, open-source distributed parallel file system originally developed by Sun Microsystem and now developed and maintained by an open source community, WhamCloud.

Lustre is a storage architecture for clusters. The main component of Lustre Clustre is the Lustre file system which is supported on a Linux Operating System and provides a POSIX-based UNIX File System interface. The ability of Lustre file system to scale capacity and performance for any need reduces the need to deploy many separate file systems such as one for each compute cluster.

Lustre aggregates and scales the storage capacity and I/O throughput of many servers and can be easily increased by adding servers dynamically. Lustre can be scaled up or down with respect to:

- Lustre Object Storage Servers
- Lustre Client nodes
- Disk storage
- Bandwidth

Lustre can be deployed in wide variety of configurations, e.g.

- Lustre Server: RedHat Linux (RHEL) and OEL x86_64
- Lustre Client: RedHat Linux (RHEL), OEL, SUSE, Scientific Linux, Fedora

Due to the extremely scalable architecture of the Lustre file system, Lustre deployments are popular in scientific supercomputing as well as in the oil and gas, manufacturing, rich media, and finance sectors. As per information available on the Internet, 15 of the 30 fastest supercomputers in the world use Lustre file system for high performance computing.

Providing the complete functionality of a file system, some of the features of Lustre file system are:

- Performance-enhanced ext4 file system
- POSIX Compliance
- High Performance heterogeneous networking

- High Availability
- Security
- ACL extended attributes
- Interoperability
- Object-based architecture
- Byte granular file and fine-grained metadata locking
- Quotas
- OSS addition
- Controlled stripping
- Network data integrity protection
- MPI I/O
- NFS and CIFS export
- Disaster Recovery Tool
- Internal monitoring and instrumentation interfaces

2. Lustre Architecture and its components

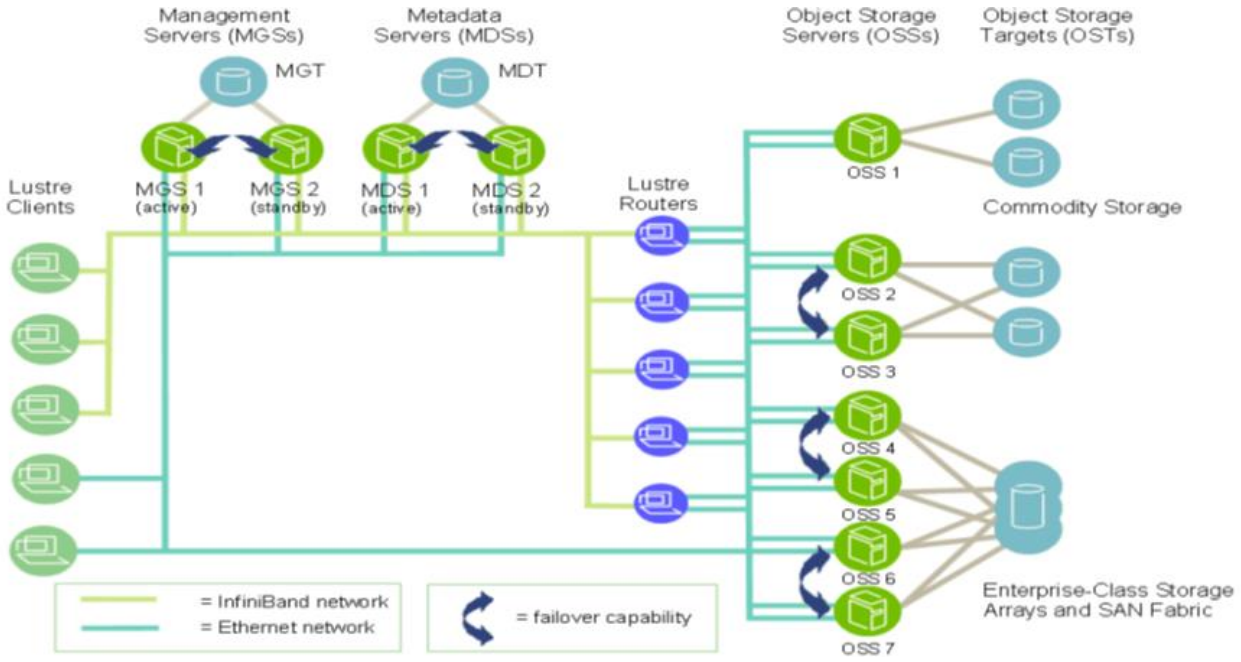


Figure 1: A Failover Lustre configuration

Lustre file system has 3 major components –

- Metadata Server
- Object Storage Server
- Lustre client.

In addition to this, a Management Server is also required which can be hosted on Metadata Server.

2.1 Lustre File System Components

- Metadata Server (MDS) – Manages names and directories in the Lustre file system
- Metadata Target (MDT) – Stores metadata such as filenames, directories, file layout, permissions, etc.
- Object Storage Server (OSS) – Provide File I/O Service and network request handling to OSTs
- Object Storage Target (OST) – User file data stores in one or more objects, each object on a separate OST
- Lustre Client – Computation, visualization, or desktop nodes running Lustre client software

2.2 Lustre Networking (LNET)

Lustre Networking (LNET) is a custom networking API that provides the communication infrastructure that handles metadata and file I/O data for the Lustre file system servers and clients.

2.3 Lustre Cluster

At scale, the Lustre cluster can include hundreds of OSSes and thousands of clients. More than one type of network can be used in a Lustre cluster. Shared storage between OSSes enables failover capability.

3. System configuration for Lustre Cluster and file system

Lustre file system is currently supported on various flavors of Linux system, commodity computer hardware or server class machines, local disk storage, FC / iSCSI based storage, Ethernet or Infiniband networking, etc. Memory and computing power is based on the load

of the I/Os. Memory requirement and CPU requirement is out of scope of this article and can be referred to in the Lustre Manual.

For our test bed, we used VMware-based virtual machines (VMs) for all the components of Lustre file system, Ethernet for networking, and EMC Symmetrix® storage system for storage devices. Configuration and system architecture of the test bed is:

Component	OS	Memory	CPU	Disk Storage
2 ESXi Servers	VMware ESXi 5.1	98 GB	12 Cores	1.3 TB Local Storage, VMAX® LUNs for Lustre file system
Lustre Meta Data Server & Management Server (**MDS / MGS**)	RHEL 6.4, Lustre 2.4	4GB	4 Cores	128 GB RAID 6 Thin LUN on VMAX
Lustre Object Storage Servers (**OSS**)	RHEL 6.4, Lustre 2.4	4GB	4 Cores	128 GB RAID 6 Thin LUNs on VMAX
Lustre Clients	RHEL 6.4, Lustre 2.4	4GB	4 Cores	Lustre OSTs
VMAX Performance Analyzer	Windows 2008 R2 with VMAX Performance Analyzer	4GB	2 Cores	Local Storage

Table 1: Test bed Lustre file system configuration

4. Lustre file system building from sources

When we started creating Lustre file system, we began with Lustre pre-compiled RPMs. However, it was very challenging to install and configure pre-compiled RPMs due to several dependencies on other software and libraries, system configuration, etc. It was also a challenge to find the right combination of dependent operating system, software, and

libraries. We were not able to bring up the complete Lustre file system with precompiled RPMs. While a lot of information is available about installation and configuration of Lustre, gaps were always found while bringing up the Lustre Clustre and file system.

Thus, after spending about a couple of weeks without much success, we decided to build the Lustre sources to create our Lustre Clustre and file system. While creating our test bed, Lustre 2.4 was the latest available software supported on RHEL 6.4.

This section describes the steps to compile and build RPMs for Lustre Server and Client software on RHEL 6.4

4.1 Lustre Source build environment

- **Install EPL 5** - This is required as it contains quilt
    - rpm -ivh http://download.fedoraproject.org/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm
- **Install quilt**
    - yum -y install quilt
- **Download following build tools**
    - asciidoc-8.4.5-4.1.el6.noarch.rpm
    - newt-devel-0.52.11-3.el6.x86_64.rpm
    - slang-devel-2.2.1-1.el6.x86_64.rpm
    - xmlto-0.0.23-3.el6.x86_64.rpm
- **Install the build tools**
    - Change directory to /home/<directory>/build_tools_rpm and install the RPMs
        - yum --nogpgcheck localinstall ./newt-devel-0.52.11-3.el6.x86_64.rpm ./slang-devel-2.2.1-1.el6.x86_64.rpm ./asciidoc-8.4.5-4.1.el6.noarch.rpm ./xmlto-0.0.23-3.el6.x86_64.rpm
- **Download and install Lustre 2.4 sources**
    - Server: "wget http://downloads.whamcloud.com/public/lustre/lustre-2.4.0/el6/server/RPMS/x86_64/lustre-source-2.4.0-2.6.32_358.6.2.el6_lustre.g230b174.x86_64_gd3f91c4.x86_64.rpm"
    - rpm -ivh lustre-source-2.4.0-2.6.32_358.6.2.el6_lustre.g230b174.x86_64_gd3f91c4.x86_64.rpm

        - This will install server sources in /usr/src/lustre-2.4.0
    - Client: "wget http://downloads.whamcloud.com/public/lustre/lustre-2.4.0/el6/client/RPMS/x86_64/lustre-client-source-2.4.0-2.6.32_358.6.2.el6.x86_64_gd3f91c4.x86_64.rpm"
        - rpm -ivh lustre-client-source-2.4.0-2.6.32_358.6.2.el6.x86_64_gd3f91c4.x86_64.rpm
            - This will install client sources in /usr/src/lustre-2.4.0

4.2 Build Lustre Kernel Sources RPMs

- **Create the directory structure for Kernel sources**
    - cd $HOME
    - mkdir -p kernel/rpmbuild{BUILD,RPMS,SOURCES,SPECS,SRPMS}
    - cd kernel
    - echo '%_topdir %(echo $HOME)/kernel/rpmbuild' > ~/.rpmmacros
- **Install the Kernel sources**
    - rpm –ivh

      http://ftp.redhat.com/pub/redhat/linux/enterprise/6Server/en/os/SRPMS/kernel-2.6.32-358.6.2.el6.src.rpm 2>&1 | grep -v mockb
    - Check for the required kernel version in : /usr/src/lustre-2.4.0/lustre/kernel_patches/which_patch
- **Prepare the Kernel sources using rpmbuild**
    - cd ~/kernel/rpmbuild
    - Execute command "rngd -r /dev/urandom &" to run it in the background
    - Execute command "rpmbuild -bp --target=`uname -m` ./SPECS/kernel.spec" to prepare the sources for RPM build
- We must have the kernel sources available now with RHEL patches applied in following location:
    - ~/kernel/rpmbuild/BUILD/kernel-2.6.32-358.el6/linux-2.6.32-358.el6.x86_64/
- **Modify the Makefile in for Kernel version**
    - ~/kernel/rpmbuild/BUILD/kernel-2.6.32-358.el6/linux-2.6.32-358.el6.x86_64/Makefile

        - EXTRAVERSION=.lustremaster
- Change directory to ~/kernel/rpmbuild/BUILD/kernel-2.6.32-358.el6/linux-2.6.32-358.el6.x86_64/
- **Overwrite the .config file**
    - cp /usr/src/lustre-2.4.0/lustre/kernel_patches/kernel_configs/kernel-2.6.32-2.6-rhel6-x86_64.config ./.config
- **Link the Lustre series file**
    - ln -s /usr/src/lustre-2.4.0/lustre/kernel_patches/series/2.6-rhel6.series series
- **Link the Luster patches file**
    - ln -s /usr/src/lustre-2.4.0/lustre/kernel_patches/patches patches
- **Apply the patches to kernel source using quilt**
    - quilt push -av
- **Build the new Kernel as RPM**
    - cd ~/kernel/rpmbuild/BUILD/~/kernel/rpmbuild/BUILD/kernel-2.6.32-358.el6/linux-2.6.32-358.el6.x86_64
    - make oldconfig || make menuconfig
    - make include/asm
    - make include/linux/version.h
    - make SUBDIRS=scripts
    - make include/linux/utsrelease.h
    - make rpm
- Kernel RPM must have been built and it can be found in ~/kernel/rpmbuild/RPMS/x86_64/ directory.

4.3 Build Lustre Server and Lustre Client RPMs

- **Change directory to /usr/src/lustre-2.4.0** (Wherever your Lustre sources are)
    - cd /usr/src/lustre-2.4.0
- **Configure Lustre sources for server**
    - ./configure --with-linux=/root/build/kernel/rpmbuild/BUILD/kernel-2.6.32.lustremaster/

- **Configure Lustre sources for client**
  - ./configure --disable-server --with-linux=/root/build/kernel/rpmbuild/BUILD/kernel-2.6.32.lustremaster/
- **Run make command after configuration completes**
  - make
- **Create RPMs after make completes**
  - make rpms

You should now have all the server and client RPMs in /root/kernel/rpmbuild/RPMS/x86_64

5. Install and configure Lustre file system Servers and Clients

We now have the entire Lustre server and client rpms built for our environment. We can use these rpms to install and configure Lustre file system on RHEL 6.4. For our test bed, we installed 1 MGS and MDS servers on one machine, four Lustre Object Storage Servers spread on both ESXi servers and eight Lustre Clients spread across both ESXi servers. Below is the system architecture of the test bed:

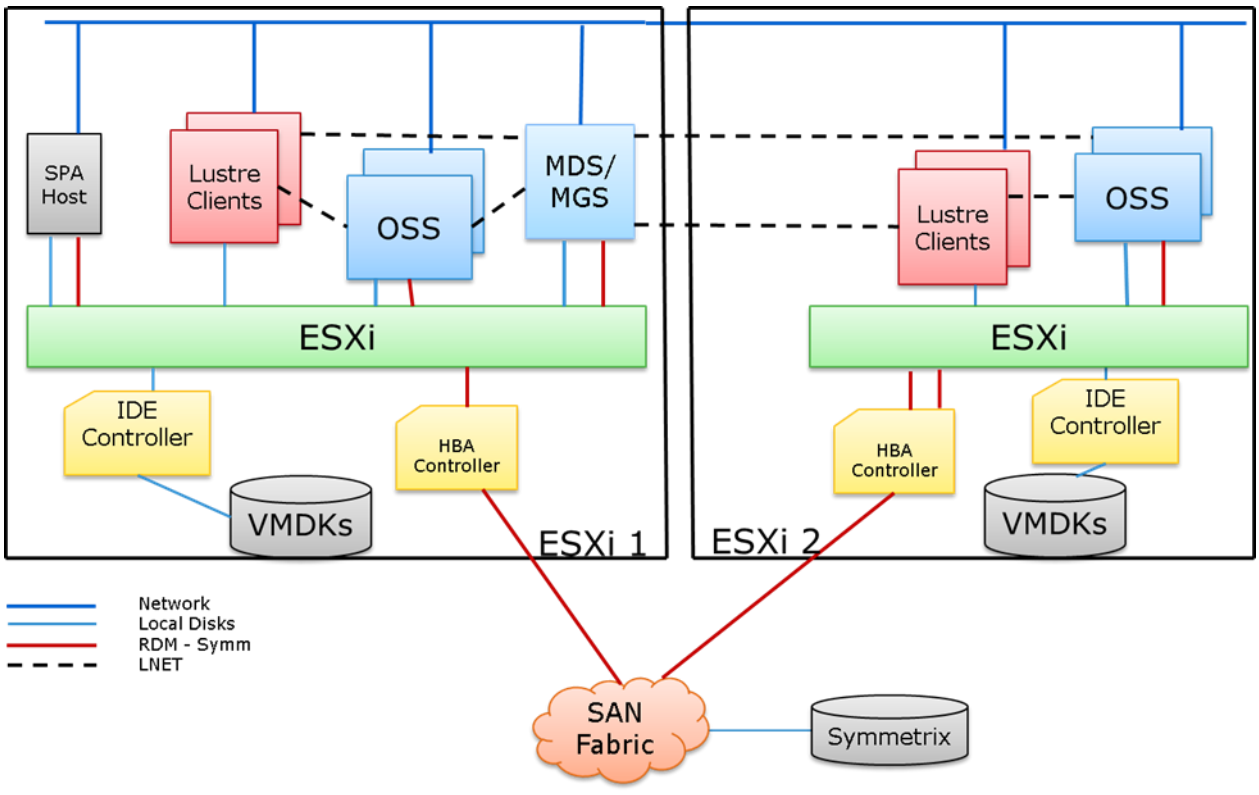


Figure 2: Lustre test bed system architecture

Configuration above is described as

- ESXi – VMware ESXi Server
- MDS / MGS – Lustre Meta Data Server and Lustre Management Server hosted on one Virtual Machine
- OSS – 4 Lustre Object Storage Servers, 2 each hosted on 2 different ESXi servers
- Lustre Clients – 8 Lustre clients, 4 each hosted on 2 different ESXi servers
- SPA Host – Performance Analyzer host
- Storage
    - OST and MDT storage is provisioned from SAN-based Symmetrix Storage
    - Lustre software is installed on local VMDKs of Linux Virtual Machines

Lustre Network (LNET) – A complete setup of Lustre file system enables the Lustre network and it is running on Ethernet

5.1 Install RHEL 6.4 Operating System on all target virtual machines with all updated packages. This should be the same as the Lustre source build machine.

- Configure the static IP address on all target machines
- Modify the /etc/hosts file for all target machines to have the IP address and machine name of all machines involved in Lustre file system

5.2 Install the Lustre kernel rpms

The first step is to install Lustre kernel rpms on each machine targeted for Lustre file system servers and clients.

- **Install kernel rpm**
    - rpm -ivh ~build/kernel/rpmbuild/RPMS/x86_64/kernel-2.6.32.lustremaster-1.x86_64.rpm
- **Create initrd using dracut**
    - /sbin/new-kernel-pkg --package kernel --mkinitrd --dracut --depmod --install 2.6.32.lustremaster
- **Reboot the Machine**

5.3 Install Lustre Servers – MGS, MDS, OSS

Lustre MGS / MDS and OSS server have the same method and steps for installation. Follow the steps below for each MGS/MDS and OSS server to install.

- **Turn on Lustre services and specify net for lnet**
  - chkconfig lustre on
- **Create /etc/modprobe.d/lustre.conf file**
  - vi /etc/modprobe.d/lustre.conf
- **Add following line to the lustre.conf file**
  - options lnet networks=tcp0
- **Modify /etc/selinux/config file to disable SELINUX**
  - vi /etc/selinux/config
  - Change SELINUX=Disabled
- **Add a device to the machine for Meta Data disk**
  - There are different methods for Physical Machine and Virtual Machine to add a drive, e.g. in Virtual Machine
    - Add a Symmetrix disk using RDM or create a new virtual disk
- **Reboot the machine**
- **Install the e2fsprogs utility**
  - Download e2fsprogs-1.42.7.wc1-7.tar file
    - Untar e2fsprogs-1.42.7.wc1-7.tar
      - tar -xvf e2fsprogs-1.42.7.wc1-7.tar
    - Change directory to e2fsprogs-1.42.7.wc1-7
    - Install all the rpms belongs to e2fsprogs
      - sudo rpm -Uvh e2fsprogs-?.* e2fsprogs-libs-?.* lib*
- **Install the Server RPMs**
  - **Change directory to RPMs**
    - cd /root/kernel/rpmbuild/RPMs/x86_64
  - **Install server RPMs in the following order:**
    - rpm -ivh lustre-ldiskfs-4.1.0-2.6.32.lustremaster_gd3f91c4.x86_64.rpm

- rpm -ivh lustre-osd-ldiskfs-2.4.0-2.6.32.lustremaster_gd3f91c4.x86_64.rpm
- rpm -ivh lustre-modules-2.4.0-2.6.32.lustremaster_gd3f91c4.x86_64.rpm
- rpm -ivh lustre-2.4.0-2.6.32.lustremaster_gd3f91c4.x86_64.rpm

5.4 Install Lustre client software

- **Turn on Lustre services and specify net for lnet**
  - chkconfig lustre on
- **Create /etc/modprobe.d/lustre.conf file**
  - vi /etc/modprobe.d/lustre.conf
- **Add following line to the lustre.conf file**
  - options lnet networks=tcp0
- **Modify /etc/selinux/config file to disable SELINUX**
  - vi /etc/selinux/config
  - Change SELINUX=Disabled
- **Reboot the machine**
- **Install the e2fsprogs utility**
  - Download e2fsprogs-1.42.7.wc1-7.tar file
    - Untar e2fsprogs-1.42.7.wc1-7.tar
      - tar -xvf e2fsprogs-1.42.7.wc1-7.tar
    - Change directory to e2fsprogs-1.42.7.wc1-7
    - Install all the rpms belongs to e2fsprogs
    - sudo rpm -Uvh e2fsprogs-?.* e2fsprogs-libs-?.* lib*
- **Install the Client RPMs**
  - **Change directory to RPMs**
    - cd /root/kernel/rpmbuild/RPMs/x86_64
  - **Install client RPMs in the following order:**
    - rpm -ivh lustre-client-modules-2.4.0-2.6.32.lustremaster_gd3f91c4.x86_64.rpm
    - rpm lustre-client-2.4.0-2.6.32.lustremaster_gd3f91c4.x86_64.rpm

5.5 Configure Lustre MGS / MDS server

- **Provision a new disk to the server for MDT device.**
  - To add the disk, use ESXi server “RDM” method to add the Symmetrix LUN to the MDS VM
- **Configure the MGS Server**
  - **Create a combined MGS and MDT File System on a block device**
    - mkfs.lustre --fsname=fsname --mgs --mdt --index=0 /dev/block_device
      - e.g. - mkfs.lustre --fsname=lustre --mgs --mdt --index=0 /dev/sdb
  - **Mount the combined MGS and MDT file system on the block device**
    - mount -t lustre /dev/block_device /mnt_point
      - e.g. mount -t lustre /dev/sdb /mnt

5.6 Configure Lustre OSS servers

- **Provision a new disk to the server for OST device**
  - To add the disk, use ESXi server “RDM” method to add the Symmetrix LUN to the OSS VM
- **Create an OST File System on a block device**
  - mkfs.lustre --fsname=fsname --mgsnode=<node_ip@tcp0> --ost --index=OST_index /dev/block_device
    - mkfs.lustre --fsname=lustre --mgsnode=10.xxx.xxx.xxx@tcp --ost --index=0 /dev/sdb
- **Mount the OST file system on the block device**
  - mount -t lustre /dev/block_device /mnt_point
    - e.g. mount -t lustre /dev/sdb /mnt

5.7 Configure Lustre clients

To configure Lustre client, we need not add any disk to Lustre client VM since the Lustre file system will be mounted on the client.

- **Mount the Lustre File system on the client node**
    - mount -t lustre MGS_node:/fsname /mount_point
        - e.g. mount -t lustre 10.xxx.xxx.xxx:lustre /mnt

5.8 Verifying the Lustre Cluster

We have configured an 8-node Lustre cluster. Use the following commands on Lustre client to verify that the Lustre cluster is running in a good state:

- Log in to Lustre client
- List all MDT devices in the Lustre cluster – **"lfs mdts"**
- List all OST devices in the Lustre cluster – **"lfs osts"**
- Show space usage per MDT and OSTs -**"lfs df -h"**
- Show all files residing on Lustre file system on /mnt - **"ls -lsah /mnt**
- Run dd command "**dd if=/dev/zero of=/mnt/zero1.dat bs=400M count =5**" to write some 2 GB of data on Lustre FS
- Show the newly created file Lustre file system on /mnt - "**ls -lsah /mnt"**
- Show the usage of Lustre file system -**"lfs df -h"**
    - This will show percentage and amount of data written on each OST in Lustre file system

5.9 Configure additional Lustre OSS servers

- Repeat configuration steps given in section 5.6
- Make sure to add an additional drive for OST device for each OSS server
    - Additional device can be added using a Symmetrix RDM device or virtual disk
- **Create Lustre file system on the new disk**
    - mkfs.lustre --fsname=fsname --mgsnode=<node_ip@tcp0> --ost --index=OST_index /dev/block_device
        - mkfs.lustre --fsname=lustre --mgsnode=10.xxx.xxx.xxx@tcp --ost --index=0 /dev/sdb
- **Mount the OST file system on the block device**
    - mount -t lustre /dev/block_device /mnt_point
        - e.g. mount -t lustre /dev/sdb /mnt

5.10 Configure additional Lustre OST device

- Add a new disk to the OSS server using RDM method or virtual disk
- **Create lustre file system on the new disk**
    - mkfs.lustre --fsname=fsname --mgsnode=<node_ip@tcp0> --ost --index=OST_index /dev/block_device
        - mkfs.lustre --fsname=lustre --mgsnode=10.xxx.xxx.xxx@tcp --ost --**index=1 /dev/sdc**
- **Mount the OST file system on the block device**
    - mount -t lustre /dev/block_device /mnt_point
        - e.g. mount -t lustre /dev/sdc /mnt

5.11 Configure additional Lustre Client

- Repeat step 5.7 for each new Lustre client

6. Lustre Benchmarking and Performance

There are several tools available to benchmark the parallel file systems and Lustre. These tools benchmark the raw device performance, file system for storage performance, network performance, CPU utilization, etc. Some of the tools we analyzed listed below:

Benchmark Tool	**Purpose**
Lustre I/O kit	- Validate Lustre Hardware - Validate Performance of H/W and S/W layers - Troubleshoot issues
I/O Meter	- Tool to measure and characterize single and clustered systems
I/O Zone	- Tool generates and measures a variety of file operations
IOR	- Performance benchmarking of Parallel File System
Bonnie++	- Hard drive - File System - % of CPU time - Amount of work done per second

VMAX Performance Analyzer	- Performance benchmark for Symmetrix storage system
	  - LUNs
	  - Host ports
	  - Disk ports

Table 2: Lustre Cluster and file system benchmarking tools

In this article, we will provide more details on Lustre IO tool kit, some sample performance data using IO Meter, and storage system performance using Symmetrix Performance Analyzer. The following picture shows the tools deployed in the test bed for benchmarking:

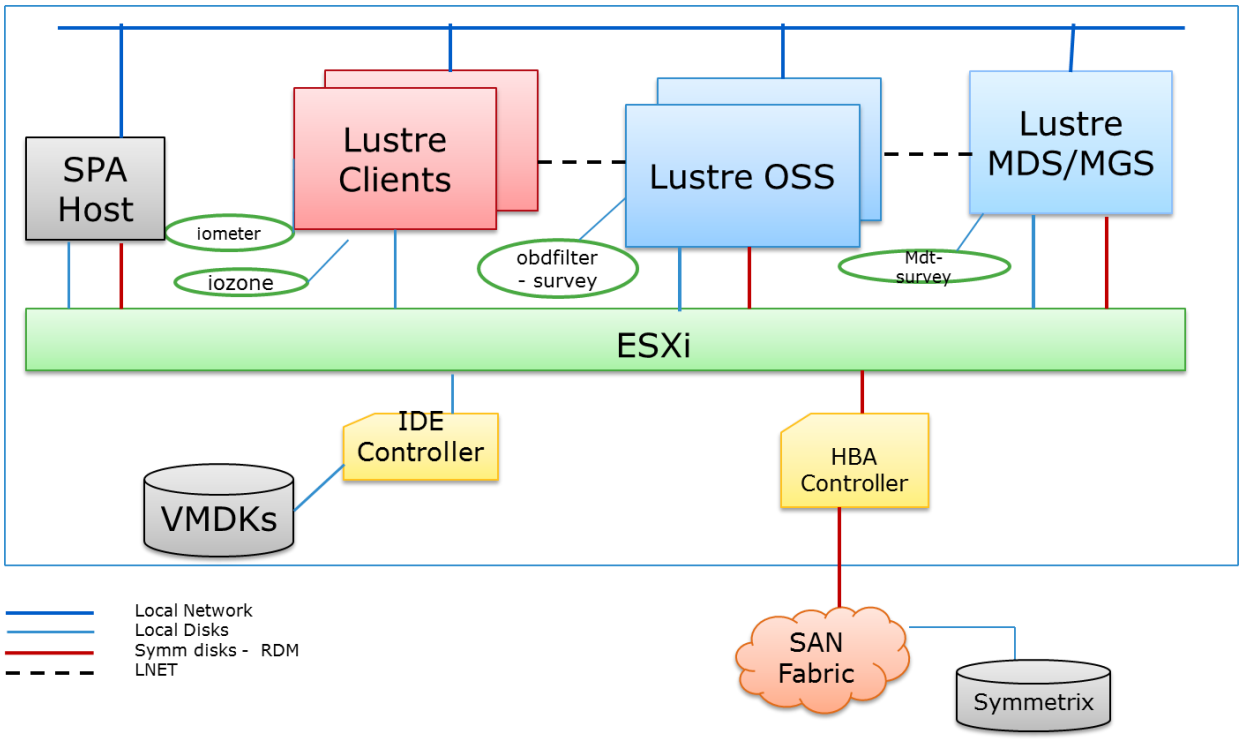


Figure 3: Benchmark tools configured in Lustre Clustre

6.1 Lustre I/O toolkit

Lustre I/O tool kit is part of the Lustre package and can be used on Lustre MDS, Lustre OSS, and Lustre clients.

- Lustre I/O toolkit is used to benchmark Lustre hardware to:
  - Validate that the hardware is working as expected
  - Validate the performance of the various hardware and software layers
  - Find and troubleshoot I/O issues
- Lustre I/O toolkit contains four tools
  - ***sgpdd-survey*** – Measure basic bare metal performance of devices
    - Example command:

      size=8388608 rszlo=1024 rszhi=4096 crglo=1 crghi=16 thrlo=1 thrhi=64 scsidevs="/dev/sdb" /usr/bin/sgpdd_survey

      - size – Total data size
      - Record Size – Lo or Hi

        - crg – Number of concurrent Regions Low and hi written or read
        - thr – Number of OSS threads Low and Hi
        - scsidev – scsi device name
    - ***obdfilter-survey*** – Measure performance of OST on OSS node or through client
        - Example command
            - ***noobjlo=2 noobjhi=16 thrlo=2 thrhi=64 size=24576 targets=lustre-OST0000 case=disk /usr/bin/obdfilter-survey***
                - ost - Total number of OSTs being tested
                - sz - Total amount of data read or written in KB
                - rsz - Record size in KB
                - obj - Total number of objects over all OSTs
                - thr - Total number of threads over all OST and objects
                - write /read/rewrite - Test name
    - ***ost-survey*** – Perform I/O against OST to allow performance comparison to detect any OST performance issue
        - Example command:
            - ***/usr/bin/ost-survey -s 300 (size in KB) /mnt/lustre***
    - ***mdt-survey*** – Performs test for local MDS server for Meta Data disk performance
        - Example command
            - ***thrhi=64 file_count=200000 /usr/bin/mds-survey***
                - Thr – Number of Thread Lo and Hi
                - targets – Number of MDT instances
                - file_count – Numbers of files per thread
                - dir_count – Total number of directories to test
                - stripe_count – Number of stripes on OST objects

- test_str - Test operation - Must have at least create and destroy
- start_number – base number for each thread
- layer - MDS stack's layer to be tested

6.2 /O Meter

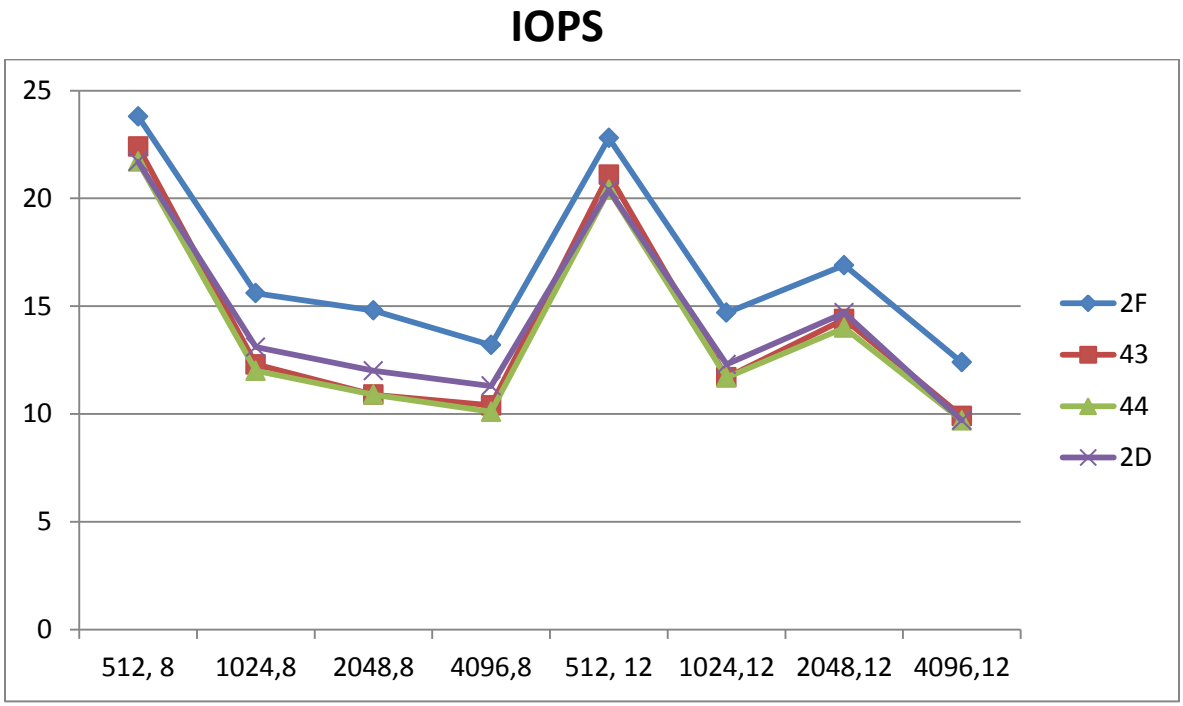
I/O Meter can be found at www.iometer.org. It has two components available for Windows and Linux.

- I/O Meter GUI runs on Windows host. I/O Meter is used to configure the work load, reporting of IOPS and throughput, start and stop the test, etc.
- Dynamo runs on the client, e.g. a Lustre client, and it generates load on the clients as per the load configured using I/O Meter GUI.
    - It is advised to build dynamo binaries on the target machine with the sources provided on iometer.org. This will avoid the need for any additional libraries to execute it.
    - Command line for dynamo is:
        - Dynamo –i <windows host ip> -m <dynamo host ip>

6.3 Symmetrix Performance Analyzer

Symmetrix Performance Analyzer (SPA) is part of Symmetrix Unisphere® GUI and can be installed on a Window host. SPA can be used to measure performance of a Symmetrix storage system with respect to LUNs, Front End host ports, and Back End disk ports. It can also be used to troubleshoot issues related to performance. It has a very rich functionality which is out of scope of this article.

6.4 Sample benchmark results using Lustre I/O Kit, I/O Meter, and SPA



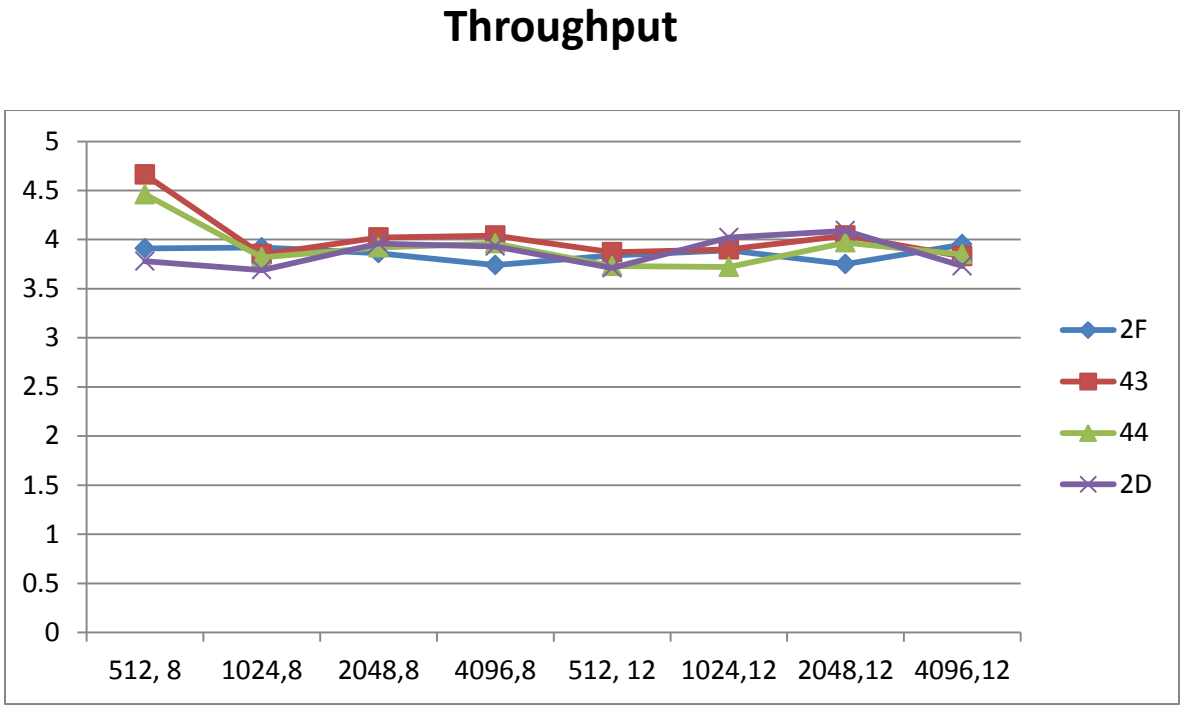


Figure 4: Graphs for Lustre file system benchmark

Above readings were done with following configuration and workload:

- I/O Meter
- 4 Lustre OSS Server with 1 OST Symmetrix device each
- 8 Lustre Clients
- 8-12 parallel worker thread on each client
- 512K – 4M data size
- Run workload for 30 minutes each

Above graphs show performance of Symmetrix back end disks with Lustre:

- IOPS trends: IOPS increase with number of clients/threads but decrease with block size
    - **Maximum IOPS**
        - Volumes – 89.6 IO/s
        - Clients – 58.11 IO/s
- Throughput trend: We have seen a saturation with block size 512KB and more, but we still need to investigate the bottleneck
    - **Maximum Throughput**
        - Volumes – 16.81 MB/s
        - Clients – 29.92 MB/s

7. Lustre Best Practices

7.1 Get the latest Lustre sources from www.whamcloud.com. You can find Lustre-related information on www.lustre.org as well.

7.2 Ensure that you use the same version of Linux Operating System and Kernel as mentioned on WhamCloud for the Lustre version you plan to deploy in your environment.

7.3 Ensure that you have installed all the required build tools and libraries for the Lustre version you plan to deploy.

7.3.1 Though the list of all the required build tools is given in this article, this could change based on the Lustre version you plan to deploy.

7.4 While compiling the Lustre sources, follow the same order described in this article or steps given on WhamCloud, otherwise you will not be able to compile the Lustre and will not get good installable Lustre RPM packages.

7.5 Once you build the Lustre RPMs, keep them on a shared host and use these RPMs for all Lustre server and client deployments. There is no need to compile Lustre source for each host.

7.6 All Lustre servers and Lustre clients must be running the same Operating System and Kernel version.

7.7 While configuring Lustre cluster and file system, ensure all Lustre servers and Lustre clients can access each other on the network with proper network configuration, or update the /etc/hosts file.

7.8 While configuring Lustre cluster, ensure that SELINUX is disabled.

7.9 Always install and configure a separate MGS / MDS server and do not merge it with OSS server. This will prevent any latency during Read / Write I/Os on Lustre file system.

7.10 For long term data integrity, Lustre filesystem devices (OST/MDT) must be RAID protected.

7.10.1 In this paper, we have used RAID 6 on EMC Symmetrix Storage system. This provides high availability and very high performance and protection of the data.

7.11 Use Lustre “stripping” feature to write files on Lustre filesystem in parallel in a multi-node Lustre filesystem.

7.12 For Higher availability of Lustre cluster, configure high availability using Linux clustering for each Lustre Server – OSS, MDS / MGS.

7.13 Refer Lustre Manual 2.x for further details on Lustre, using other Lustre features, troubleshooting Lustre etc.

7.14 Subscribe to Lustre mailing list hpdd-discuss@lists.01.org.Tthis is very useful for posting any query or getting help from the entire Lustre community which includes developers, users etc.

8. Conclusion

What we have learned:

- Lustre Clustre architecture
- Lustre file system and its components
- Installation and configuration requirements for Lustre
- Steps to build Lustre RPMs from sources
- Steps to install and configure each Lustre component
- Lustre file system benchmarking tools

This article is based on the theme of “Knowledge Sharing” where anybody can gain from/contribute to it at any time. Installation and configuration of Lustre cluster and file system requires a steep learning curve and often requires users to refer to several documents, websites, manuals, etc. Each component of Lustre has its own set of pre-requisites, recommendations, and limitations.

This article can be used as a quick cheat sheet for anybody dealing with Lustre deployment and using Lustre, and shall be able to reduce turnaround time and learning time and, most importantly, improve TCE (Total Customer Experience). The article is the first step to simplify/consolidate and reduce the start-up time to deploy/configure or modify any Lustre cluster.

Going forward, we ask subject matter experts on Lustre Cluster and EMC Storage tuning to review and update the article to keep it current. Again, the article is not owned by an individual or group but by EMC in general. Anybody can add value to it by contributing with the content, thereby helping the Lustre community at large.

References

[1] www.whamcloud.com

[2] http://wiki.lustre.org/index.php/Main_Page - Lustre Wiki

[3] http://wiki.lustre.org/index.php/Main_Page

[4] Sean Cochrane, Ken Kutzer, Lawrence McIntosh,(2009) “Solving The HPC I/O Bottleneck: SUN LUSTRE Storage System”, (Revision 2.0) (November)

[5] Lustre Operations Manual (Version 2.0), Sun ORACLE, (821-2076-10).

[6] Website: HowTo: iometer http://greg.porter.name/wiki/HowTo:iometer

[7] Website: Ramesh Natarajan,(2011) “10 Iozone Examples For Disk I/O Performance Measurement On Linux”
http://www.thegeekstuff.com/2011/05/iozone-examples/

[8] Xyratex (2012) “Lustre IO Benchmarking Methodology using IOR” (Version 1.3), (February).