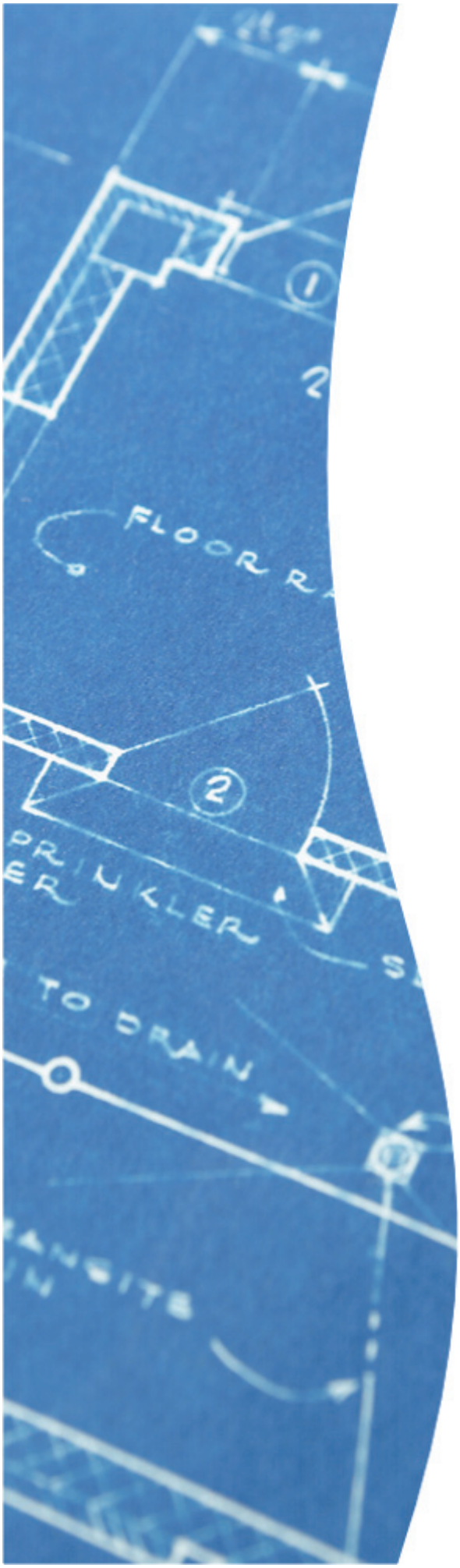

LUSTRE™ FILE SYSTEM: DEMO QUICK START GUIDE

Torben Kling-Petersen, Ph.D., Lustre Group

Sun BluePrints™ Online

Part No 820-7390-10
Revision 1.1, 4/6/09

Table of Contents

Lustre™ File System: Demo Quick Start Guide

The Lustre™ file system is a scalable, secure, robust, and highly-available cluster file system that addresses the I/O needs, such as low latency and extreme performance, of large computing clusters. Designed, developed, and maintained by Sun Microsystems, the Lustre file system is intended for environments where traditional shared file systems, such as NFS, do not scale to the required aggregate throughput or large number of nodes.

While the Lustre file system has been around for a number of years in the Open Source arena and there are a large number of installations worldwide, getting started without reading a 500+ page manual—and having to be a Linux expert as well—is difficult. This paper provides a simple cookbook for non-Linux experts on how to set up a Linux-based Lustre file system using small servers, workstations, PCs, or other available hardware for demonstration purposes.

This paper addresses the following topics:

- "Lustre file system overview" on page 1 provides a brief overview of the file system design.
- "Preliminary setup" on page 3 covers the preliminary installation and setup steps.
- "Configuration overview" on page 3 describes the configuration used in this article.
- "Lustre file system configuration" on page 10 provides step-by-step directions for configuring the Lustre file system.
- "Managing the file system" on page 12 describes common administrative tasks.

This paper assumes the readier is familiar with basic system administration tasks, but does not assume any explicit Linux or Lustre file system administration experience.

Lustre file system overview

The Lustre file system is a software-only architecture that allows a number of different hardware implementations. The main components of a Lustre architecture are the Lustre file system clients (Lustre clients), The Metadata Servers (MDS), and Object Storage Servers (OSS). Metadata Servers and Object Storage Servers implement the file system and communicate with the Lustre clients. Lustre clients access the Lustre file system via a dedicated network, such InfiniBand, Ethernet, or other network connections. For reasons of simplicity, only Ethernet-based connectivity is covered in this paper.

The Lustre file system uses an object-based storage model, and provides several abstractions designed to improve both performance and scalability. At the file system level, the Lustre technology treats files as objects that are located through Metadata Servers (MDS). Metadata Servers support all file system name space operations, such

as file lookups, file creation, and file and directory attribute manipulation. File data is stored in objects on the OSSs. The MDS directs actual file I/O requests from Lustre clients to OSSs, which manage the storage that is physically located on underlying storage devices. Once the MDS identifies the storage location of a file, all subsequent file I/O is performed directly between the client and the OSSs.

This design divides file system updates into two distinct types of operations: file system metadata updates on the MDSs, and actual file data updates on the OSSs. Separating file system metadata operations from actual file data operations not only improves immediate performance, but also improves long-term aspects of the file system such as recoverability and availability.

As shown in Figure 1, the Lustre technology can support a variety of configuration options including a choice of interconnects, single or dual Metadata Servers, and different storage attachment methods for the Object Storage Servers.

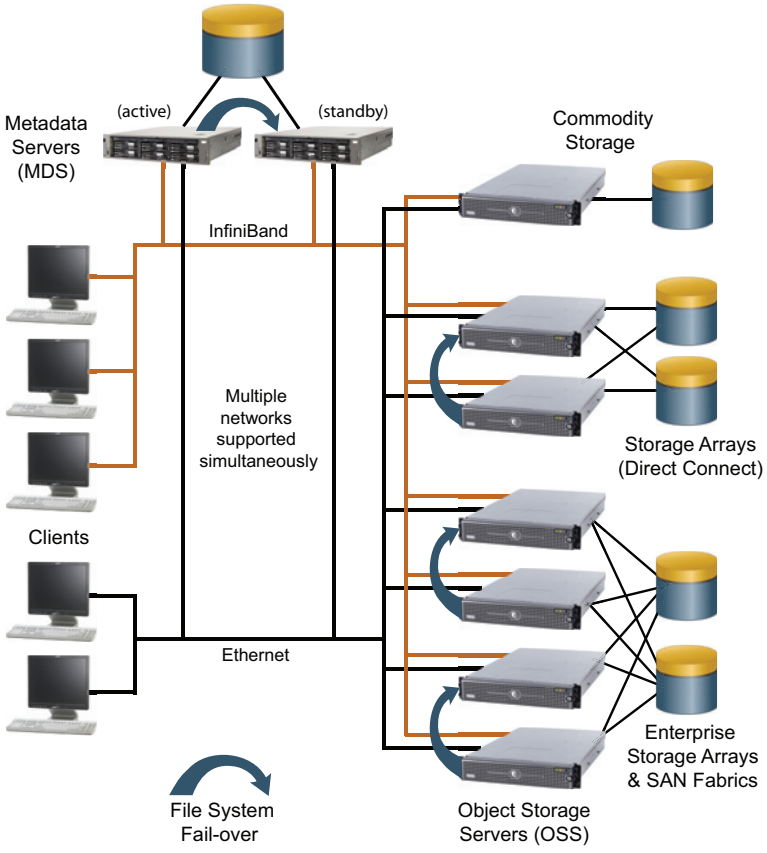

Figure 1. Lustre file system high-level architecture.

Further information on the Lustre file system can be found in the white paper, *Lustre File System: High-Performance Storage Architecture and Scalable Cluster File System*, found at:

`http://www.sun.com/offers/details/LustreFileSystem.html`.

The remainder of this Sun BluePrints article explains the steps necessary to set up a working Lustre file system for testing, development and evaluation purposes.

Configuration overview

The primary goal of this paper is to provide a basic configuration that can be used to demonstrate the Lustre file system. While the basic demo configuration can consist of virtually any system that boots Linux, the configuration described in this paper is based on three Sun Fire™ x64 two-socket servers: one used as a non-redundant MDS and two used as OSS in a non-HA configuration. Each Sun Fire x64 server was configured with two single-core AMD Opteron™ processors, 4 GB of RAM, and one 73-GB disk drive.Two Sun single-socket x64 workstations were used as clients. All systems were connected using Gigabit Ethernet.

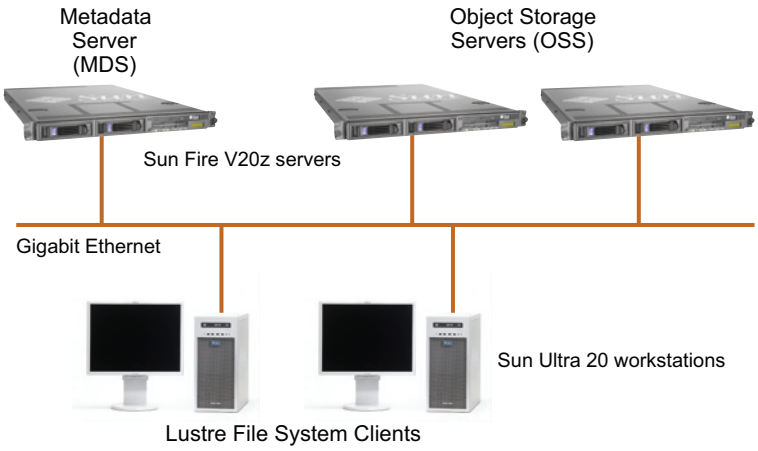

Figure 2. System configuration used for testing.

It is important to understand that while the example setup is fully functional from a parallel file system point of view, it is not optimized for performance, reliability or capacity and should not be used in any kind of HPC production system. It will, however, constitute an excellent demo, lab, or proof-of-concept system.

Preliminary setup

Preliminary setup steps include:

- Installing the Linux operating system on all servers and clients (see page 4)
- Creating the virtual volumes (see page 5)
- Installing the Lustre stack (see page 6)

These steps are described in the following sections.

Installing the Linux operating system

While any mainstream Linux OS can be used[1], the current selection fell on CentOS 5.2 as this is essentially the same binary as Red Hat Enterprise Linux 5.2 (but free). More information on CentOS is available at `http://centos.org/`.

Installing CentOS on the servers and clients is reasonably intuitive and does not require any specific Linux skills. The only part of the installation that requires any afterthought is formatting of the disk. As the system has only one disk, it needs to be partitioned to accommodate both the boot partition and the metadata target (MDT) or object store target (OST) partition.

In the present system, the disk was split into three partitions:

- Boot partition – 20 GB (enough to house the OS and some applications)
- Swap partition – 2 GB (not strictly necessary, but just in case)
- Logical volume partition – 50 GB (using a large LVM partition is critical to permit the creation of multiple virtual volumes used for the MDT/OST later in the configuration)

The systems are configured as follows:

Table 1. System configuration.

System	Name	IP	Functions
Sun Fire V20z #1	mds	192.168.0.10	Metadata Server
Sun Fire V20z #2	oss01	192.168.0.11	Object Store Server #1
Sun Fire V20z #3	oss02	192.168.0.12	Object Store Server #2
Sun Ultra 20 #1	client1	192.168.0.5	Client #1 (2nd Ethernet port on external network)
Sun Ultra 20 #2	client2	192.168.0.6	Client #2

After the install and reboot, primary setup is performed (and is very intuitive). To avoid potential problems, all internal firewalls are disabled. All communication from client to Lustre servers are done via `ssh`.

Note – The Lustre file system does *not* work with SELinux.

1. Refer to `http://www.sun.com/software/products/lustre/specs.xml` for currently supported OS releases.

Creating the virtual volumes

Virtual volumes need to be created on both Object Store Servers and on the Metadata Server. The MDS requires only one virtual volume; six virtual volumes are created on each OSS in this example configuration.

Object Server #1:

1. Listing the partitions on the first OSS reveals the following:

```
[root@LustreClient01 ~]# ssh root@192.168.0.11
root@192.168.0.11's password:
--------------
[root@oss01 ~]# fdisk -l

Disk /dev/sda: 73.4 GB, 73407868928 bytes
255 heads, 63 sectors/track, 8924 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes

   Device Boot      Start         End      Blocks   Id  System
/dev/sda1   *           1        2550    20482843+  83  Linux
/dev/sda2            8664        8924     2096482+  82  Linux swap /
Solaris
/dev/sda3            2551        8663    49102672+  8e  Linux LVM
```

2. Use the Linux command `pvcreate` to create a physical volume on `/dev/sda3`:

```
[root@oss01 ~]# pvcreate /dev/sda3
  Physical volume "/dev/sda3" successfully created
```

3. Next, create a volume group:

```
[root@oss01 ~]# vgcreate vg00 /dev/sda3
  Volume group "vg00" successfully created
```

4. Once the volume group is created, create a number of Object Store Targets (OSTs) in this volume group. This example creates six OSTs, named `ost1` through `ost6`:

```
[root@oss01 dev]# lvcreate --name vg00/ost1 --size 2G
  Logical volume "ost1" created
[root@oss01 dev]# lvcreate --name vg00/ost2 --size 2G
  Logical volume "ost2" created
[root@oss01 dev]# lvcreate --name vg00/ost3 --size 2G
  Logical volume "ost3" created
[root@oss01 dev]# lvcreate --name vg00/ost4 --size 2G
  Logical volume "ost4" created
[root@oss01 dev]# lvcreate --name vg00/ost5 --size 2G
  Logical volume "ost5" created
[root@oss01 dev]# lvcreate --name vg00/ost6 --size 2G
  Logical volume "ost6" created
```

Note – In this example, each OST is 2 GB in size for test purposes. The maximum size of an OST is 8 TB. In a typical production system, each OST would consist of several disks in a RAID 6 configuration.

---

5. Use the `ls` command to list all logical volumes:

```
[root@oss01 dev]# ls -l /dev/vg00
total 0
lrwxrwxrwx 1 root root 21 Nov  6 14:05 ost1 -> /dev/mapper/vg00-ost1
lrwxrwxrwx 1 root root 21 Nov  6 14:06 ost2 -> /dev/mapper/vg00-ost2
lrwxrwxrwx 1 root root 21 Nov  6 14:06 ost3 -> /dev/mapper/vg00-ost3
lrwxrwxrwx 1 root root 21 Nov  6 14:06 ost4 -> /dev/mapper/vg00-ost4
lrwxrwxrwx 1 root root 21 Nov  6 14:06 ost5 -> /dev/mapper/vg00-ost5
lrwxrwxrwx 1 root root 21 Nov  6 14:06 ost6 -> /dev/mapper/vg00-ost6
```

Object Server #2:

6. Login as root to the second OSS, and repeat Steps 1 to 5 to create six OSTs for the second OSS.

Metadata Server:

7. Login as root to the MDS, and repeat Steps 1 to 5 to create a single logical volume (MDT) for the MDS. As each file written to the Lustre file system requires 4 KB of storage space in the MDS, a 1 GB logical volume for the MDT is more than sufficient for most typical deployments.

Tip – Before downloading and installing the RPMs, test connectivity by confirming that `ping` works between all systems.

Installing the Lustre stack

The Lustre packages can be downloaded free of charge from the following location:

`http://www.sun.com/software/products/lustre/get.jsp`

Tip – It is recommended to always download the latest version of the packages.

---

1. A download account with Sun is required to access the Lustre file system. If needed, register for a download account.
2. Login to the download account, and select the appropriate platform. In this example, since the system is AMD Opteron processor-based and uses the open source version of Red Hat Enterprise Linux (RHEL), the Red Hat Enterprise Linux 5, x86_64 version was selected (see Figure 3).

Figure 3. Lustre file system download menu.

3. Once logged in, two groups of files are displayed: required files and optional files. Download all required files.

   In the release used in this study, the following files are listed as required:

   - lustre-*<version>*
   - kernel-lustre-*<version>*
   - lustre-ldiskfs-*<version>*
   - lustre-modules-*<version>*
   - e2fsprogs-*<version>*
   - lustre-client-*<version>*
   - lustre-client-modules-*<version>*

   The last two files listed above (`lustre-client-<version>` and `lustre-client-modules-<version>`) aren't strictly required, unless patchless clients are used. If patchless clients aren't being used, these two files can be omitted.

Note – Patchless clients do not require the kernel to be rebuilt, and therefore simplify deployment of client systems.

4. Once downloaded, the directory should contain:

```
[root@LustreClient01 Desktop]# ls Lustre_1.6.6_RPMs
e2fsprogs-1.40.11.sun1-0redhat.x86_64.rpm
kernel-lustre-smp-2.6.18-92.1.10.el5_lustre.1.6.6.x86_64.rpm
lustre-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
lustre-ldiskfs-3.0.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
lustre-modules-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
```

5. Copy the RPMs to the OSS:

```
[root@oss01 /]# scp -r
root@192.168.0.6:/root/Desktop/Lustre_1.6.6_RPMs/*.rpm /Lustre_RPMs
root@192.168.0.6's password:
e2fsprogs-1.40.11.sun1-0redhat.x86_64.rpm
100%  848KB 848.2KB/s   00:00
kernel-lustre-smp-2.6.18-92.1.10.el5_lustre.1.6.6.x86_64.rpm
100%  142MB  28.4MB/s   00:05
lustre-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
100% 4101KB   4.0MB/s   00:00
lustre-ldiskfs-3.0.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
100%  907KB 907.4KB/s   00:00
lustre-modules-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
100%   16MB  15.9MB/s   00:01
```

6. Change to the RPM directory and install all RPMs. Some Linux distributions ship with one or more Lustre RPMs, such as `e2fsprogs`, already installed. These RPMs need to be updated using the `rpm -U` command. All other RPMs are installed using the `rpm -i` command. Multiple RPMs can be installed using a single command statement, with each RPM separated with a space.

```
[root@oss01 Lustre_RPMs]# rpm -Uhv e2fsprogs-1.40.11.sun1-
0redhat.x86_64.rpm
Preparing...           ########################################[100%]
   1:e2fsprogs         ########################################[100%]
[root@oss01 Lustre_RPMs]# rpm -ihv
lustre-ldiskfs-3.0.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
kernel-lustre-smp-2.6.18-92.1.10.el5_lustre.1.6.6.x86_64.rpm
lustre-modules-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
lustre-1.6.6-2.6.18_92.1.10.el5_lustre.1.6.6smp.x86_64.rpm
Preparing...           ########################################## [100%]
   1:lustre            ########################################## [ 25%]
   2:lustre-ldiskfs    ########################################## [ 50%]
   3:kernel-lustre-smp ########################################## [ 75%]
   4:lustre-modules    ########################################## [100%]
WARNING: /lib/modules/2.6.18-
92.1.10.el5_lustre.1.6.6smp/kernel/net/lustre/ko2iblnd.ko needs
unknown symbol ib_create_cq
WARNING: /li ....      [the rest of the Warnings omited for brevity

Congratulations on finishing your Lustre installation!  To register
your copy of Lustre and find out more about Lustre Support, Service,
and Training offerings please visit
http://www.sun.com/software/products/lustre/lustre_reg.jsp
```

Note – This installation will yield a number of warnings, primarily due to the lack of the InfiniBand components. As InfiniBand components are not needed for Ethernet-based Lustre file systems, these warnings can be ignored.

7. Reboot the system.

8. When the system is back up, log in as root and check the currently running kernel version:

```
[root@oss01]# uname -a
Linux oss01 2.6.18-92.el5 #1 SMP Tue Jun 10 18:51:06 EDT 2008 x86_64
x86_64 x86_64 GNU/Linux
```

9. Next, change the default booting kernel in the `/boot/grub/grub.config` file (that is, change `default=1` to `default=0`):

```
[root@oss01 grub]# cat grub.conf
# grub.conf generated by anaconda
#
# Note that you do not have to rerun grub after making changes to this
file
# NOTICE:  You do not have a /boot partition.  This means that
#          all kernel and initrd paths are relative to /, eg.
#          root (hd0,0)
#          kernel /boot/vmlinuz-version ro root=/dev/sda1
#          initrd /boot/initrd-version.img
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/boot/grub/splash.xpm.gz
hiddenmenu
title CentOS (2.6.18-92.1.10.el5_lustre.1.6.6smp)
      root (hd0,0)
      kernel /boot/vmlinuz-2.6.18-92.1.10.el5_lustre.1.6.6smp ro
root=LABEL=/ rhgb quiet
      initrd /boot/initrd-2.6.18-92.1.10.el5_lustre.1.6.6smp.img
title CentOS (2.6.18-92.el5)
      root (hd0,0)
      kernel /boot/vmlinuz-2.6.18-92.el5 ro root=LABEL=/ rhgb quiet
      initrd /boot/initrd-2.6.18-92.el5.img
```

10. After modifying the `grub.config` file, reboot the system.
11. When the system is back up, log in as root and check the currently running kernel version:

```
[root@mds ~]# uname -a
Linux mds.lustre 2.6.18-92.1.10.el5_lustre.1.6.6smp #1 SMP Tue Aug 26
12:16:17 EDT 2008 x86_64 x86_64 x86_64 GNU/Linux
```

Repeat this procedure for the MDS, additional OSS, and client systems. The system is now ready to have the Lustre file system configured on it.

Lustre file system configuration

The Lustre file system configuration steps should always be performed in the following order:

- Metadata Server
- Object Store Servers
- Client

The configuration also initializes the file system and makes the file system usable once the client has been configured. While all steps below are done by hand, the Lustre file system configuration is fully scriptable and lends itself to simple deployment once the basic hardware setup is done. However, to fully understand the basics of each step, the following somewhat over-simplified description will create a base for the necessary script steps. Most of the system responses are included for comparative reasons.

Metadata Server

The following steps describe the Lustre file system configuration on the Metadata Server (MDS).

1. The baseline for the MDS is creating a file system named `lustre` on the server including the metadata target (`--mdt`) and the management server (`--mgs`).

```
[root@mds /]# mkfs.lustre --fsname lustre --mdt --mgs /dev/vg00/mdt

   Permanent disk data:
Target:     lustre-MDTffff
Index:      unassigned
Lustre FS:  lustre
Mount type: ldiskfs
Flags:      0x75
              (MDT MGS needs_index first_time update )
Persistent mount opts: errors=remount-ro,iopen_nopriv,user_xattr
Parameters: mdt.group_upcall=/usr/sbin/l_getgroups

checking for existing Lustre data: not found
device size = 5120MB
WARNING: The e2fsprogs package currently installed on your system does
not support "uninit_bg" feature.
Please install the latest version of e2fsprogs from
http://downloads.lustre.org/public/tools/e2fsprogs/
to enable this feature.
Feature will not be enabled until e2fsprogs is updated and 'tune2fs -O
uninit_bg %{device}' is run.

2 6 18
formatting backing filesystem ldiskfs on /dev/vg00/mdt
        target name  lustre-MDTffff
        4k blocks     0
        options        -J size=204 -i 4096 -I 512 -q -O dir_index -F
mkfs_cmd = mkfs.ext2 -j -b 4096 -L lustre-MDTffff  -J size=204 -i 4096
-I 512 -q -O dir_index -F /dev/vg00/mdt
Writing CONFIGS/mountdata
```

2. Create a mount point:

```
[root@mds /]# mkdir /mdt
```

3. Start the MDS node:

```
[root@mds /]# mount -t lustre /dev/vg00/mdt /mdt
```

This completes the MDS configuration for the Lustre file system.

Object Store Servers

The following steps describe the Lustre file system configuration on the Object Store Servers (OSSs).

1. Create mount points for the OSTs; this example uses six named `ost1` through `ost6`:

```
[root@oss01 ~] mkdir /mnt/ost1
...
[root@oss01 ~] mkdir /mnt/ost6
```

2. Use the `mkfs.lustre` command to create the Lustre file systems. For example:

```
[root@oss01 ~]# mkfs.lustre --fsname lustre --ost
  --mgsnode=192.168.0.10@tcp0 /dev/vg00/ost1

   Permanent disk data:
Target:     lustre-OSTffff
Index:      unassigned
Lustre FS:  lustre
Mount type: ldiskfs
Flags:      0x72
              (OST needs_index first_time update )
Persistent mount opts: errors=remount-ro,extents,mballoc
Parameters: mgsnode=192.168.0.10@tcp

checking for existing Lustre data: not found
device size = 2048MB
2 6 18
formatting backing filesystem ldiskfs on /dev/vg00/ost1
        target name  lustre-OSTffff
        4k blocks     0
        options  -J size=80 -i 16384 -I 256 -q -O
dir_index,uninit_groups -F
mkfs_cmd = mkfs.ext2 -j -b 4096 -L lustre-OSTffff  -J size=80 -i 16384
-I 256 -q -O dir_index,uninit_groups -F /dev/vg00/ost1
Writing CONFIGS/mountdata
```

Note – The management node can be addressed as mgs@tcp0 as well but using the IP address may often be simpler to debug.

3. Start the OSS by mounting the OSTs to the corresponding mount point. For example:

```
[root@oss01 ~]# mount -t lustre /dev/vg00/ost1 /mnt/ost1
```

4. Repeat the process for every OST. Once done, the related devices would look like this:

```
[root@oss01 ~]# cat /proc/fs/lustre/devices
  0 UP mgc MGC192.168.0.10@tcp 39e2ebb5-b3c1-e6d1-fa3d-23937c706dd1 5
  1 UP ost OSS OSS_uuid 3
  2 UP obdfilter lustre-OST0000 lustre-OST0000_UUID 7
  3 UP obdfilter lustre-OST0001 lustre-OST0001_UUID 7
  4 UP obdfilter lustre-OST0002 lustre-OST0002_UUID 7
  5 UP obdfilter lustre-OST0003 lustre-OST0003_UUID 7
  6 UP obdfilter lustre-OST0004 lustre-OST0004_UUID 7
  7 UP obdfilter lustre-OST0005 lustre-OST0005_UUID 7
```

5. Repeat this procedure on the second OSS.

Client

Setting up the client is very straightforward. All clients mount the same file system identified by the MDS. Use the following commands, specifying the IP address of the MDS server:

```
[root@LustreClient01 ~]# mkdir /mnt/lustre
[root@LustreClient01 ~]# mount -t lustre 192.168.0.10@tcp0:/lustre
  /mnt/lustre
```

Once the mount has been completed, the Lustre file system is ready to use.

Managing the file system

While this blueprint is not intended as a replacement for the manual, there are some tools included with the Lustre file system that are quite useful.

Note – The Lustre file system manual can be downloaded from:
`http://manual.lustre.org/index.php?title=Main_Page`

Using a prefix of `lfs` followed by a command, information about the entire file system can be obtained from the client. For instance, the basic command `df` preceded by the `lfs` trigger yields:

```
[root@LustreClient01 lustre]# lfs df -h
UUID                    bytes      Used Available  Use% Mounted on
lustre-MDT0000_UUID      4.4G    214.5M      3.9G    4% /mnt/lustre[MDT:0]
lustre-OST0000_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:0]
lustre-OST0001_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:1]
lustre-OST0002_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:2]
lustre-OST0003_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:3]
lustre-OST0004_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:4]
lustre-OST0005_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:5]

filesystem summary:      11.8G    499.7M     10.7G    4% /mnt/lustre
```

Using stripes

One of the main reasons for Lustre file system's performance is the striping of data blocks over multiple OSTs. The stripe count can be set on a file system, directory or file level.

To see the current stripe size, use the command `lfs getstripe [file, dir, fs]`. On the current system this will produce the following output:

```
root@LustreClient01 lustre]# lfs getstripe /mnt/lustre
OBDS:
0: lustre-OST0000_UUID ACTIVE
1: lustre-OST0001_UUID ACTIVE
2: lustre-OST0002_UUID ACTIVE
3: lustre-OST0003_UUID ACTIVE
4: lustre-OST0004_UUID ACTIVE
5: lustre-OST0005_UUID ACTIVE
/mnt/lustre
(Default) stripe_count: 2 stripe_size: 4M stripe_offset: 0
```

As can be seen, the default stripe count is 2 (that is, striping over two OSTs), default stripe size is 4 MB (can be set in K, M or G), and all writes start from the first OST.

Note – When setting the stripe, the offset is set before the stripe count.

Setting a new stripe pattern on the file system can look like this:

```
[root@LustreClient01 lustre]# lfs setstripe /mnt/lustre 4M 0 1
```

This example sets the stripe of `/mnt/lustre` to 4 MB blocks starting at OST0 and spanning over one OST. If a new file is created with these settings, the following results are seen:

```
[root@LustreClient01 lustre]# dd if=/dev/zero of=/mnt/lustre/test1 bs=10M
  count=100

root@LustreClient01 lustre]# lfs df -h
UUID                    bytes      Used Available  Use% Mounted on
lustre-MDT0000_UUID      4.4G    214.5M      3.9G    4% /mnt/lustre[MDT:0]
lustre-OST0000_UUID       2.0G      1.1G    830.1M   53%
/mnt/lustre[OST:0]
lustre-OST0001_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:1]
lustre-OST0002_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:2]
lustre-OST0003_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:3]
lustre-OST0004_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:4]
lustre-OST0005_UUID      2.0G     83.3M      1.8G    4% /mnt/lustre[OST:5]

filesystem summary:      11.8G      1.5G      9.7G   12% /mnt/lustre
```

As can be seen, the entire file was written to the first OST, and there is a very uneven distribution of data blocks.

Continuing with this example, the file is removed and the stripe count is changed to a value of `-1`, which means "stripe over all available OSTs."

```
[root@LustreClient01 lustre]# lfs setstripe /mnt/lustre 4M 0 -1
```

Now, when a file is created, the new stripe setting evenly distributes the data over all available OSTs:

```
[root@LustreClient01 lustre]# dd if=/dev/zero of=/mnt/lustre/test1 bs=10M
  count=100
100+0 records in
100+0 records out
1048576000 bytes (1.0 GB) copied, 20.2589 seconds, 51.8 MB/s

[root@LustreClient01 lustre]# lfs df -h
UUID                    bytes      Used Available  Use% Mounted on
lustre-MDT0000_UUID      4.4G    214.5M      3.9G    4% /mnt/lustre[MDT:0]
lustre-OST0000_UUID      2.0G    251.3M      1.6G   12%
/mnt/lustre[OST:0]
lustre-OST0001_UUID      2.0G    251.3M      1.6G   12%
/mnt/lustre[OST:1]
lustre-OST0002_UUID      2.0G    251.3M      1.6G   12%
/mnt/lustre[OST:2]
lustre-OST0003_UUID      2.0G    251.3M      1.6G   12%
/mnt/lustre[OST:3]
lustre-OST0004_UUID      2.0G    247.3M      1.6G   12%
/mnt/lustre[OST:4]
lustre-OST0005_UUID      2.0G    247.3M      1.6G   12%
/mnt/lustre[OST:5]

filesystem summary:     11.8G      1.5G      9.7G   12% /mnt/lustre
```

Determining stripe information for a file

The `lfs getstripe` command can be used to display information that shows over which OSTs a file is distributed. For example, the output from the following command (the multiple `obdidx` entries) indicates that the file `test1` is striped over all six active OSTs in the configuration:

```
[root@LustreClient01 ~]# lfs getstripe /mnt/lustre/test1
OBDS:
0: lustre-OST0000_UUID ACTIVE
1: lustre-OST0001_UUID ACTIVE
2: lustre-OST0002_UUID ACTIVE
3: lustre-OST0003_UUID ACTIVE
4: lustre-OST0004_UUID ACTIVE
5: lustre-OST0005_UUID ACTIVE
/mnt/lustre/test1
        obdidx         objid           objid           group
          0              8              0x8              0
          1              4              0x4              0
          2              5              0x5              0
          3              5              0x5              0
          4              4              0x4              0
          5              2              0x2              0
```

In contrast, the output from the following command, which lists just a single `obdidx` entry, indicates that the file `test2` is contained on a single OST:

```
[root@LustreClient01 ~]# lfs getstripe /mnt/lustre/test_2
OBDS:
0: lustre-OST0000_UUID ACTIVE
1: lustre-OST0001_UUID ACTIVE
2: lustre-OST0002_UUID ACTIVE
3: lustre-OST0003_UUID ACTIVE
4: lustre-OST0004_UUID ACTIVE
5: lustre-OST0005_UUID ACTIVE
/mnt/lustre/test_2
      obdidx           objid           objid           group
           2               8             0x8               0
```

Handling full OSTs

Sometimes the file system gets unbalanced, often due to changed stripe settings. If an OST gets filled up and one tries to write more information to the file system involving said OST, an error occurs.

The example below shows an unbalanced file system:

```
root@LustreClient01 ~]# lfs df -h
UUID                       bytes      Used Available  Use% Mounted on
lustre-MDT0000_UUID        4.4G    214.5M      3.9G    4% /mnt/lustre[MDT:0]
lustre-OST0000_UUID        2.0G    751.3M      1.1G   37%
/mnt/lustre[OST:0]
lustre-OST0001_UUID        2.0G    755.3M      1.1G   37%
/mnt/lustre[OST:1]
lustre-OST0002_UUID        2.0G      1.7G    155.1M   86%
/mnt/lustre[OST:2]  <-
lustre-OST0003_UUID        2.0G    751.3M      1.1G   37%
/mnt/lustre[OST:3]
lustre-OST0004_UUID        2.0G    747.3M      1.1G   37%
/mnt/lustre[OST:4]
lustre-OST0005_UUID        2.0G    743.3M      1.1G   36%
/mnt/lustre[OST:5]

filesystem summary:       11.8G      5.4G      5.8G   45% /mnt/lustre
```

In this case, OST:2 is almost full and when one tries to write additional information to the file system (even with uniform striping over all the OSTs), the write command fails as follows:

```
[root@LustreClient01 ~]# lfs setstripe /mnt/lustre 4M 0 -1
[root@LustreClient01 ~]# dd if=/dev/zero of=/mnt/lustre/test_3 bs=10M
  count=100
dd: writing `/mnt/lustre/test_3': No space left on device
98+0 records in
97+0 records out
1017192448 bytes (1.0 GB) copied, 23.2411 seconds, 43.8 MB/s
```

To enable continued use of the file system, the full OST has to be taken offline or, more specifically, rendered read-only. This can be accomplished using the `lctl` command.

Note – This action has to be done on the MDS, since this is the server that allocates space for writing:

---

1. Log in to the MDS server:

```
[root@LustreClient01 ~]# ssh root@192.168.0.10
root@192.168.0.10's password:
Last login: Wed Nov 26 13:35:12 2008 from 192.168.0.6
```

2. Use the `lctl dl` command to show the status of all file system components:

```
[root@mds ~]# lctl dl
  0 UP mgs MGS MGS 9
  1 UP mgc MGC192.168.0.10@tcp e384bb0e-680b-ce25-7bc9-81655dd1e813 5
  2 UP mdt MDS MDS_uuid 3
  3 UP lov lustre-mdtlov lustre-mdtlov_UUID 4
  4 UP mds lustre-MDT0000 lustre-MDT0000_UUID 5
  5 UP osc lustre-OST0000-osc lustre-mdtlov_UUID 5
  6 UP osc lustre-OST0001-osc lustre-mdtlov_UUID 5
  7 UP osc lustre-OST0002-osc lustre-mdtlov_UUID 5
  8 UP osc lustre-OST0003-osc lustre-mdtlov_UUID 5
  9 UP osc lustre-OST0004-osc lustre-mdtlov_UUID 5
 10 UP osc lustre-OST0005-osc lustre-mdtlov_UUID 5
```

3. Use the `lctl deactive` command to take the full OST offline:

```
[root@mds ~]# lctl --device 7 deactivate
```

4. Again, display the status of the file system components:

```
[root@mds ~]# lctl dl
  0 UP mgs MGS MGS 9
  1 UP mgc MGC192.168.0.10@tcp e384bb0e-680b-ce25-7bc9-81655dd1e813 5
  2 UP mdt MDS MDS_uuid 3
  3 UP lov lustre-mdtlov lustre-mdtlov_UUID 4
  4 UP mds lustre-MDT0000 lustre-MDT0000_UUID 5
  5 UP osc lustre-OST0000-osc lustre-mdtlov_UUID 5
  6 UP osc lustre-OST0001-osc lustre-mdtlov_UUID 5
  7 IN osc lustre-OST0002-osc lustre-mdtlov_UUID 5
  8 UP osc lustre-OST0003-osc lustre-mdtlov_UUID 5
  9 UP osc lustre-OST0004-osc lustre-mdtlov_UUID 5
 10 UP osc lustre-OST0005-osc lustre-mdtlov_UUID 5
```

As can be seen from the device list, OST2 is now inactive. If a new file is now written to the file system, the write will be successful as the stripes are allocated across all the other active OSTs.

Migrating data within a file system

As there's no way of moving stripes around within the file system, data must be migrated manually using the rather crude approach:

- Copy file `originalname` → `newname`
- Remove `originalname`
- Rename `newname` → `originalname`

However, first is it necessary to identify which file(s) need to be moved.

1. Identify the file to be moved. In the following example, output from the `getstripe` command indicates that the file `test_2` is located entirely on OST2:

```
[root@LustreClient01 ~]# lfs getstripe /mnt/lustre/test_2
OBDS:
0: lustre-OST0000_UUID ACTIVE
1: lustre-OST0001_UUID ACTIVE
2: lustre-OST0002_UUID ACTIVE
3: lustre-OST0003_UUID ACTIVE
4: lustre-OST0004_UUID ACTIVE
5: lustre-OST0005_UUID ACTIVE
/mnt/lustre/test_2
      obdidx           objid           objid           group
           2               8             0x8               0
```

2. Once the file(s) have been identified, they can be moved:

```
[root@LustreClient01 ~]# cp /mnt/lustre/test_2 /mnt/lustre/test_2.tmp
[root@LustreClient01 ~]# rm /mnt/lustre/test_2
rm: remove regular file `/mnt/lustre/test_2'? Y
```

3. A quick look at the file system indicates a much more even spread (compare the `df` output in this step with the `df` output in "Handling full OSTs" on page 15). Therefore, the file can be changed the file back to the original name (so all clients can "find" it):

```
[root@LustreClient01 ~]# lfs df -h
UUID                    bytes      Used Available  Use% Mounted on
lustre-MDT0000_UUID       4.4G    214.5M      3.9G    4%
/mnt/lustre[MDT:0]
lustre-OST0000_UUID       2.0G      1.3G    598.1M   65%
/mnt/lustre[OST:0]
lustre-OST0001_UUID       2.0G      1.3G    594.1M   65%
/mnt/lustre[OST:1]
lustre-OST0002_UUID       2.0G    913.4M   1000.0M   45%
/mnt/lustre[OST:2]
lustre-OST0003_UUID       2.0G      1.3G    602.1M   65%
/mnt/lustre[OST:3]
lustre-OST0004_UUID       2.0G      1.3G    606.1M   64%
/mnt/lustre[OST:4]
lustre-OST0005_UUID       2.0G      1.3G    610.1M   64%
/mnt/lustre[OST:5]

filesystem summary:      11.8G      7.3G      3.9G   61% /mnt/lustre

[root@LustreClient01 ~]# mv test2.tmp test2
[root@LustreClient01 ~]# ls /mnt/lustre
test1  test_2  test3  test_3  test4  test_4  test_x
```

4. Once done, the OST can be reactivated for further writes:

```
[root@mds ~]# lctl --device 7 activate

[root@mds ~]# lctl dl
  0 UP mgs MGS MGS 9
  1 UP mgc MGC192.168.0.10@tcp e384bb0e-680b-ce25-7bc9-81655dd1e813 5
  2 UP mdt MDS MDS_uuid 3
  3 UP lov lustre-mdtlov lustre-mdtlov_UUID 4
  4 UP mds lustre-MDT0000 lustre-MDT0000_UUID 5
  5 UP osc lustre-OST0000-osc lustre-mdtlov_UUID 5
  6 UP osc lustre-OST0001-osc lustre-mdtlov_UUID 5
  7 UP osc lustre-OST0002-osc lustre-mdtlov_UUID 5
  8 UP osc lustre-OST0003-osc lustre-mdtlov_UUID 5
  9 UP osc lustre-OST0004-osc lustre-mdtlov_UUID 5
 10 UP osc lustre-OST0005-osc lustre-mdtlov_UUID 5
```

Note – Remember that activation is done on the MDS.

Summary

This paper, intended for non-Linux experts, describes the process of installing and configuring the Lustre file system on a basic Linux-based hardware configuration. While this basic configuration is not optimized for performance or reliability, it implements a fully functional parallel file system suitable for a lab, demo, or proof-of-concept system.

Step-by-step directions are included for the preliminary setup tasks of installing the Linux operating system, creating virtual volumes, and installing the Lustre stack; as well as configuring the Lustre file system, including configuring the Metadata Server (MDS), Object Store Servers (OSS), and clients. In addition, the document includes procedures for common management tasks such as using stripes, handling full Object Store Targets (OSTs), and migrating data within a file system.

Setting up a production system uses the same techniques and software components. However, production systems also add complexity by requiring additional features such as installation of high availability (HA) functionality for the MDS and optionally for the OSS, optimal RAID configuration of disk arrays, and InfiniBand connectivity.

While never intended to be a thorough explanation of all Lustre file system configuration or administration, this document provides ample direction for setting up a functional Lustre file system using a minimal Linux-based hardware configuration.

About the author

Torben Kling-Petersen has worked with high performance computing in one form or another since 1994 and is currently working as a Senior Technical Specialist for HPC in Sun's Lustre Group. Over the years, he has worked in a number of capacities such as lead architect for enterprise datacenter infrastructure, technical research lead and product specialist for high-end visualization, to mention a few. In his present capacity, Torben works in a global role providing technical evangelism and solution architectures on petaflop-scale HPC projects.

References

As with all open source projects, the Internet contains a wealth of information and tips regarding the Lustre file system. The following sites are recommended starting points:

Lustre file systems:

`http://www.sun.com/software/products/lustre/`

Lustre file system Wiki:

`http://wiki.lustre.org/index.php?title=Main_Page`

Lustre file system discussion groups:

`http://groups.google.com/group/lustre-discuss-list?hl=en`

Lustre 1.6 Operations Manual:
`http://manual.lustre.org/manual/LustreManual16_HTML/`

Ordering Sun documents

The SunDocs℠ program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals through this program.

Accessing Sun documentation online

The `docs.sun.com` web site enables you to access Sun technical documentation online. You can browse the `docs.sun.com` archive or search for a specific book title or subject. The URL is
`http://docs.sun.com/`

To reference Sun BluePrints Online articles, visit the Sun BluePrints Online Web site at:
`http://www.sun.com/blueprints/online.html`

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

Printed in USA 04/09