The Abdus Salam
International Centre for Theoretical Physics

1967-27

Advanced School in High Performance and GRID Computing

3 - 14 November 2008

Introduction to the Lustre Parallel File System

CALUCCI Piero

S.I.S.S.A.
International School for Advanced Studies
Via Beirut 2-4
34014 Trieste
ITALY

Parallel File Systems for HPC

Introduction to Lustre

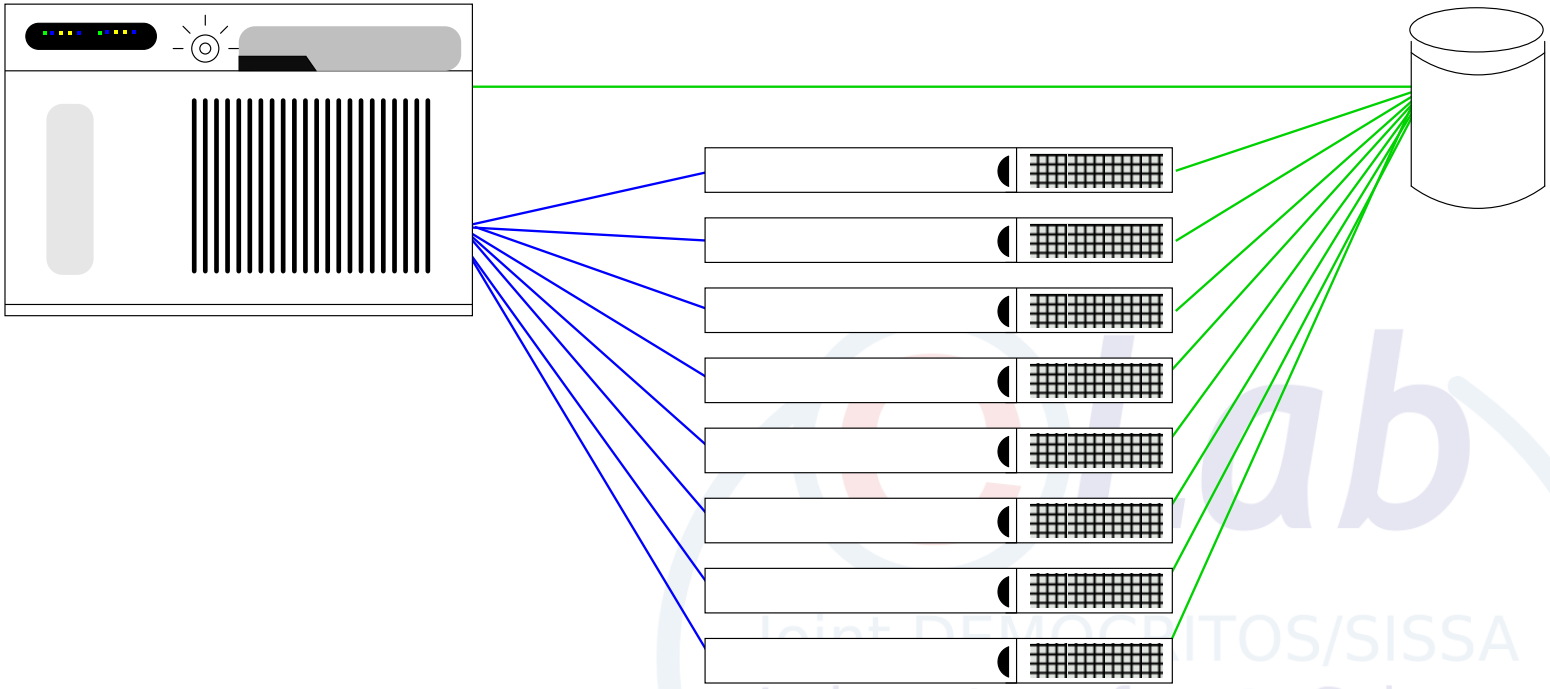
Piero Calucci

Scuola Internazionale Superiore di Studi Avanzati
Trieste

November 2008
Advanced School
in High Performance and Grid Computing

Outline

Cluster & Storage



A typical cluster setup with a Master node, several computing nodes and shared storage.
Nodes have little or no local storage.

Cluster & Storage

The Old-Style Solution

- a single storage server quickly becomes a bottleneck
- if the cluster grows in time (quite typical for initially small installations) storage requirements also grow, sometimes at a higher rate
  - adding space (disk) is usually easy
  - adding speed (both bandwidth and IOpS) is hard and usually involves expensive upgrade of existing hardware
- e.g. you start with an NFS box with a certain amount of disk space, memory and processor power, then add disks to the same box

Cluster & Storage

The Old-Style Solution /2

- e.g. you start with an NFS box with a certain amount of disk space, memory and processor power
  - adding space is just a matter of plugging in some more disks, ot ar worst adding a new controller with an external port to connect external disks
  - but unless you planned for excess network bandwidth, you cannot benefit from increased disk bandwidth
  - the available memory is now shared among a larger number of disks
  - the availble processor power has now to sustain a larger number of operations (this is usually less of an issue with current multicore, high-GHz processors, so you still probably benfit of the larger number of available spindles)

Cluster & Storage

The Parallel Solution

In a parallel solution

- you add storage servers, not only disk space
- each added storage server brings in more memory, more processor power, more bandwidth
- software complexity however is much higher
- there is no «easy» solution, like NFS or CIFS

Cluster & Storage

The Parallel Solution /2

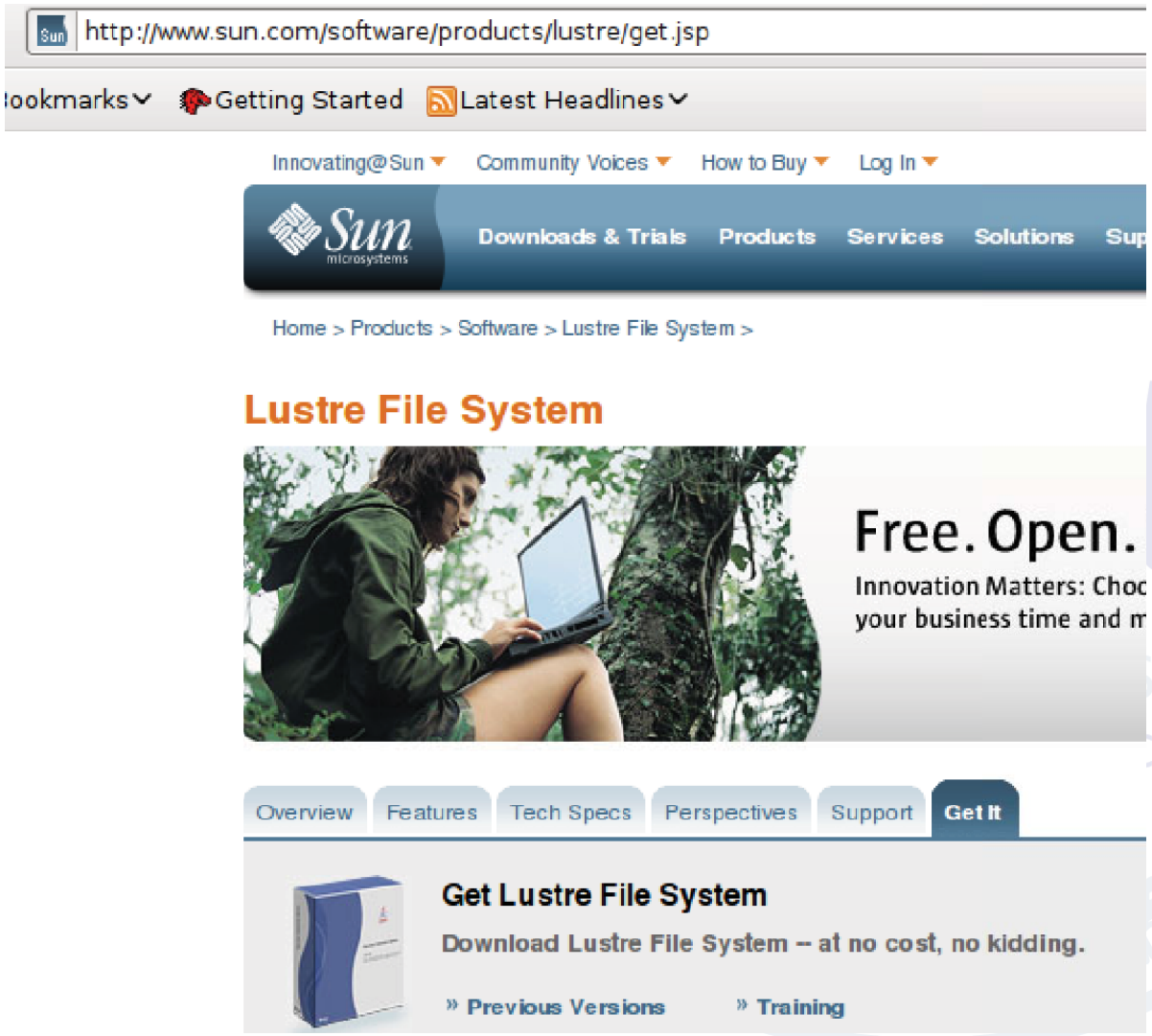
A parallel solution usually is made of

- several Storage Servers that hold the actual filesystem data
- one or more Metadata Servers that help clients to make sense of data stored in the file system

and optionally

- monitoring software that ensures continuous availablity of all needed components
- a redundancy layer that replicates in some way information in the storage cluster, so that the file system can survive the loss of some component server

Getting Lustre



Lustre Components

- Single or dual Metadata Server (MDS) with attached Metadata Target (MDT)
- multiple (up to 100s) Object Storage Server (OSS) with attached Object Storage Targets (OST)
- clients

MDS

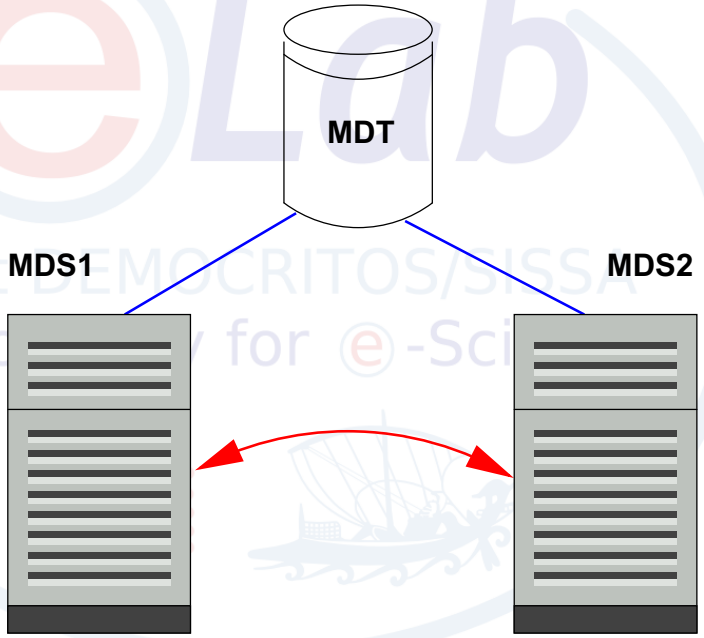
The Metadata Server manages all metadata operations on the file system (file names and directories).

A single MDS is needed for file system operation.
Multiple MDSs are possible in an active/standby configuration (all of them attached to a shared MDT storage)

MDS

The Metadata Server manages all metadata operations on the file system (file names and directories).

A single MDS is needed for file system operation.
Multiple MDSs are possible in an active/standby configuration (all of them attached to a shared MDT storage)



OSS

Object Storage Servers provide the actual I/O service, connecting to Object Storage Targets.

OSSs can be almost anything from local disks to shared storage to high-end SAN fabric.

Each OSS can serve one to dozen OSTs, and each OST can be up to 8TB in size.

The capacity of a lustre file system is the sum of the capacities provided by OSTs, while the aggregated bandwidth can approach the sum of bandwidths provides by OSSs.

Lustre Networking

All communication among lustre servers and clients is managed by LNET.

Key features of LNET include:

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

Lustre Networking /2

In the Real World LNET can deliver

- more than 100MB/s on GB ethernet (with PCIe, server-class NICs and a low- to midrange GB switch)
- more than 400MB/s on 4x SDR infiniband without any special tuning

(these are actual numbers measured on our small lustre cluster in the worst case of a single OSS; with multiple OSSs aggregated bandwidth will be much larger)

A Small Lustre Cluster

4 Supermicro 3U servers, each with

- 2 dual-core opteron 2216
- 16GB memory
- infiniband card
- 16-port Areca SATA RAID controller
- 16 500GB SATA disks
  - single RAID6 + 1 hot spare array (6.5TB available space)

Each OSS uses the local array as OST; one of the servers acts both as OSS and MDS.

Clients connect on the infiniband network, a separate ethernet connection is used for management only.

A Small Lustre Cluster /2

- initial file system creation is very simple with provided `lustre_config` script – it takes time however, and machines are basically unusable until `mkfs` is over
- in our setup the RAID controller is the bottleneck – maximum bandwidth per OSS decreases to around 300MB/s once cache is full
- single MDS provides no redundancy at all (and if you loose the MDT, the file system is gone)
- MDT and one OST share physical disks – suboptimal performance expected, especially on metadata intensive workloads

A Small Lustre Cluster /3

Even with all said limitation we observe a performance level that is «good enough»:

- write bandwidth around 1GB/s with a small number of concurrent clients («small» meaning more or less equal to the number of servers)
- write bandwidth around 750MB/s with a moderate number of concurrent clients («moderate» meaning no more than 10 times the number of servers)
- beyond that point write bandwidth decreases slowly with an increasing number of clients
- on the downside shared hardware between the MDT and one OST really hurts performance – we regularly observe the affected OSS finishing its work late

A Small Lustre Cluster /4

- read bandwidth in the 4–8GB/s range when data are already cached on servers (OSSs have 16GB main memory each)
- non-cached reads performance starts at 1.5GB/s and decreases with the same pattern as the write performance

Striping

The default configuration for a newly-created lustre filesystem is that each file is assigned to an OST.

File striping can be configured – and the great thing is that it can be configured on a per-file or per-directory base.

When striping is enabled

- file size can grow beyond OST available space
- bandwidth from several OSSs can be aggregated
- cache from several OSSs can be aggregated

For each file a different striping pattern can be defined (different stripe width and number of stripes).

Striping /2

In our experience, striping can make a huge difference when a single client is working with huge files (here «huge» means larger than combined caches on client and OSS).

However, in the more typical workload of many concurrent clients it makes no difference at best and can even hurt performance due to increased metadata access.

GPFS by IBM

General Parallel File System

- rock-solid, 10-years history
- available for AIX, Linux and Windows Server 2003
- proprietary license
- tightly integrated with IBM cluster management tools

OCFS2

Oracle Cluster File System

- once proprietary, now GPL
- available in Linux vanilla kernel
- not widely used outside the database world (as fa as I know)
- 16TiB limit (to be confirmed)

PVFS

Parallel Virtual File System

- open source, easy to install
- userspace-only server, kernel module required only on clients
- optimized for MPI-IO
- POSIX compatibility layer performance is sub-optimal

pNFS

Parallel NFS

- extension of NFSv4, under development
- some proprietary solutions available (Panasas)
- should put together benefits of parallel IO with those of using a standard solution (NFS)

"That's all Folks!"

<calucci@sissa.it>