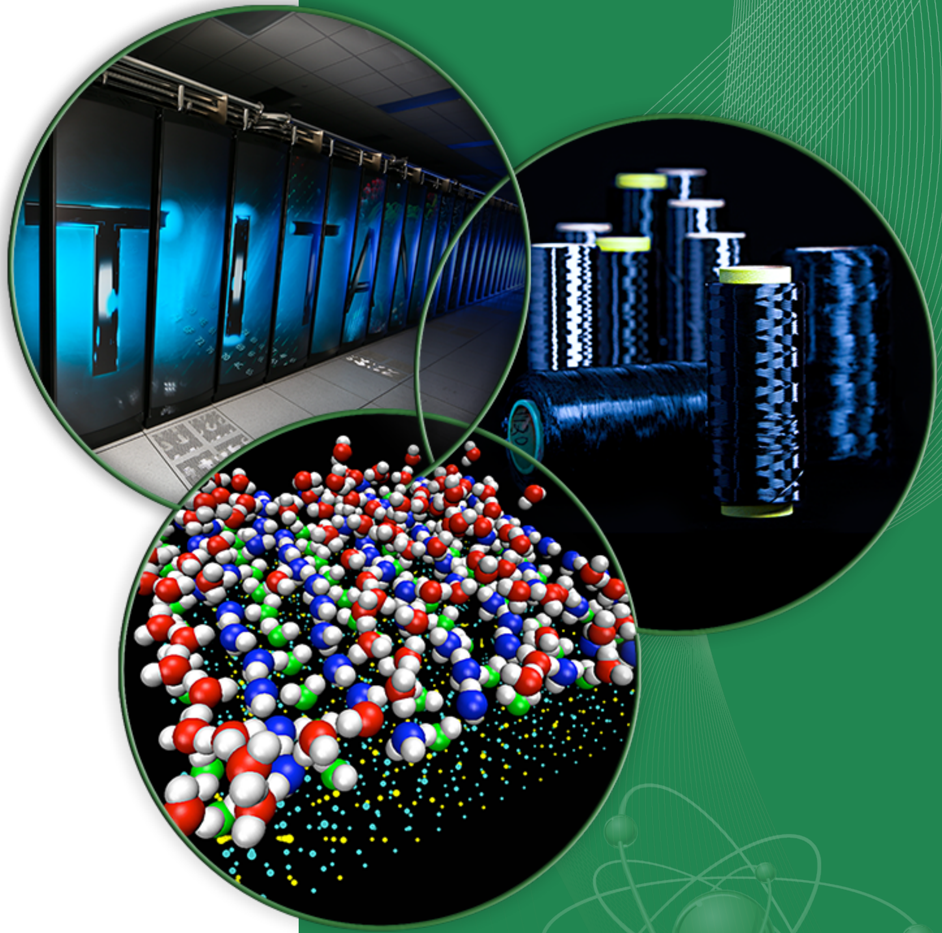
Oak Ridge National Laboratory

Computing and Computational Sciences Directorate



Introduction to Lustre

Rick Mohr
Jeffrey Rossiter
Sarp Oral
Michael Brim
Jason Hill
Joel Reed
Neena Imam

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE
National Laboratory

Outline of Topics

- What is Lustre?
- Lustre features
- Lustre architecture overview
- LNET transport layer
- Example Lustre setups
- File striping concepts
- I/O optimization for Lustre

The Need for Parallel File Systems

- High Performance Computing (HPC) has outgrown the ability of any single host
- The same holds true for Big Data problems:
  - (data set sizes) > (drive capacities)
  - Single server bandwidth is not sufficient to support access to all data from thousands of clients
- Need a parallel file system that can:
  - Scale capacity/bandwidth
  - Support large numbers of clients
- Lustre is a popular choice to meet these needs

What is Lustre?

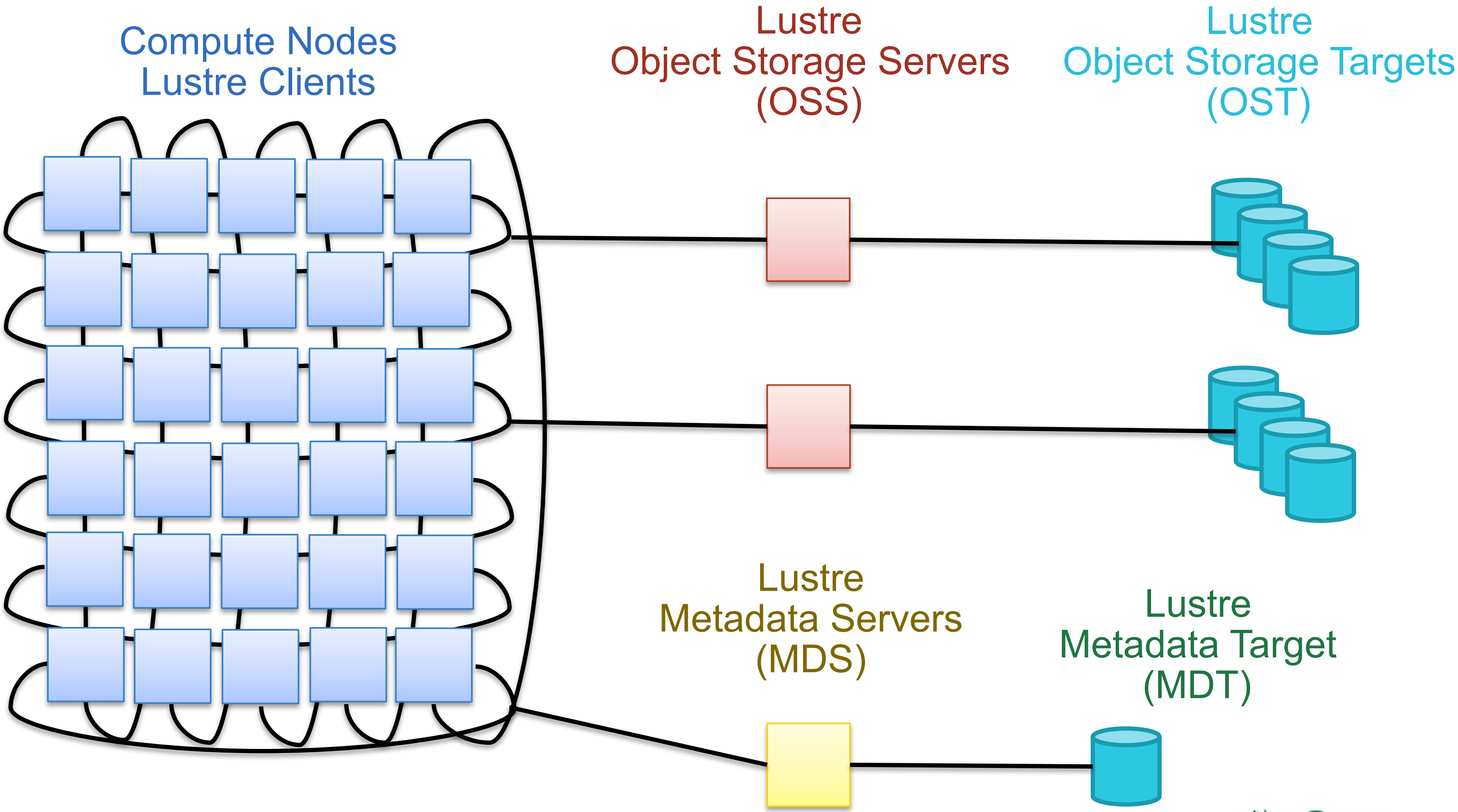
- Lustre is a massively parallel distributed file system that supports:
  - Thousands of clients
  - Large capacities (55 PB at LLNL)
  - High bandwidths (1.4 TB/s at ORNL)
  - POSIX semantics for I/O access
- Lustre is Open Source under GPLv2
- Used by many of the TOP500 supercomputers
- Not just for HPC (e.g., PayPal)

Lustre Features

- File striping across disks and servers
- Multiple metadata servers
- Online file system checking
- HSM integration
- Ability to add servers to existing file system
- User and group quotas
- Pluggable Network Request Scheduler

- RDMA support
- High availability
- I/O routing between networks
- Multiple backend storage formats (ldiskfs and ZFS)
- Storage pools
- CPU partitions
- Recovery features

Lustre Architecture



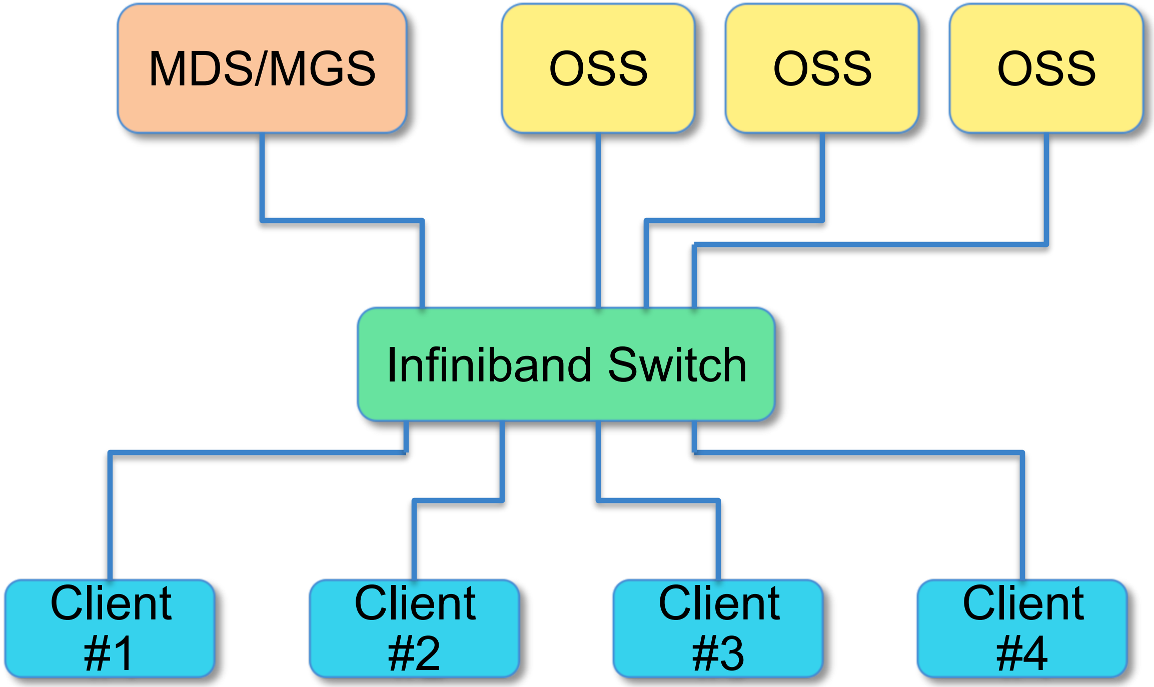
Lustre Components

- MDS – Manages filenames and directories, file stripe locations, locking, ACLs, etc.
- MDT – Block device used by MDS to store metadata information
- OSS – Handles I/O requests for file data
- OST – Block device used by OSS to store file data. Each OSS usually serves multiple OSTs.
- MGS – Management server. Stores configuration information for one or more Lustre file systems.
- MGT - Block device used by MGS for data storage

LNET Transport Layer

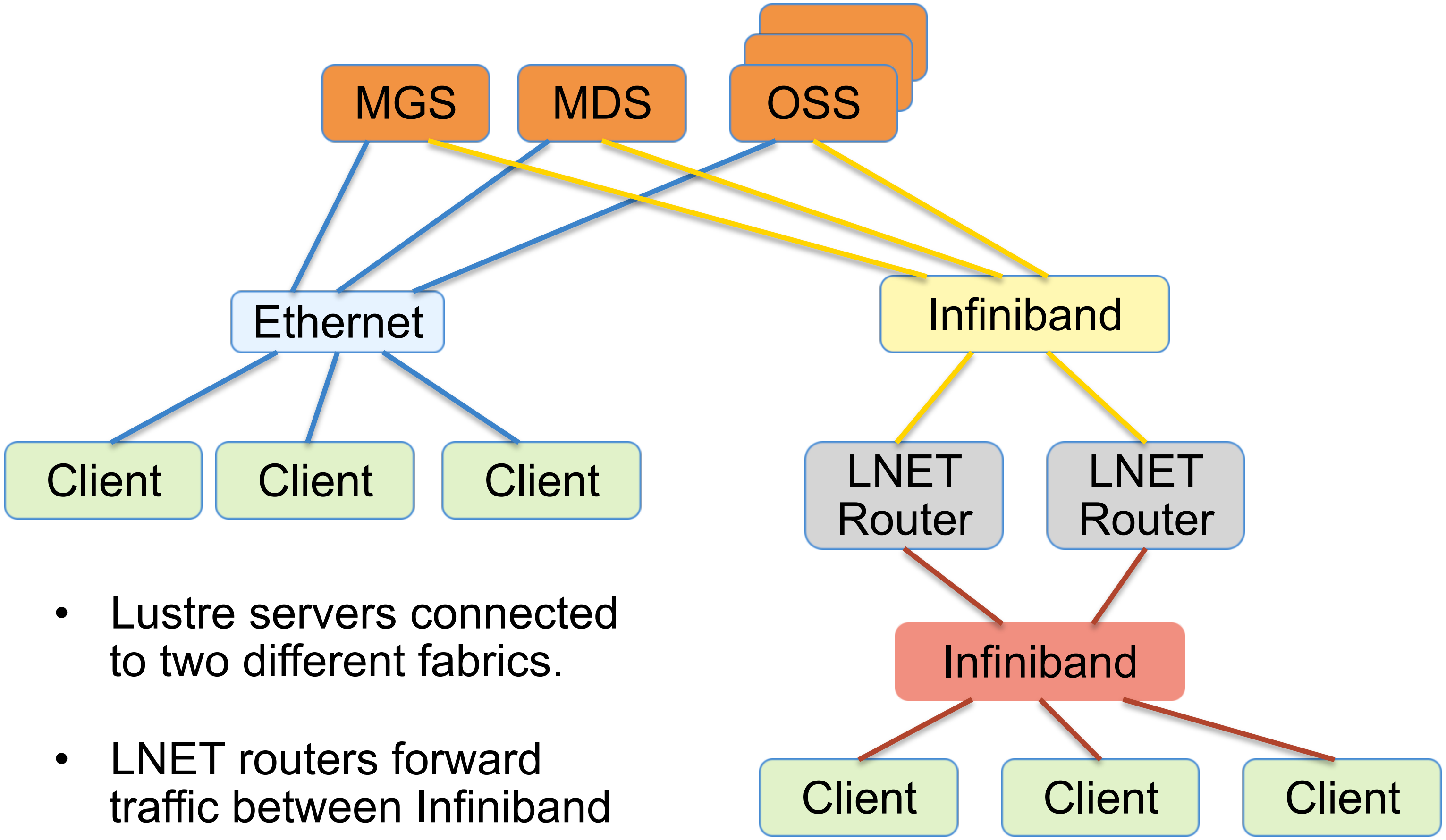
- Lustre Networking (LNET) provides the underlying communication infrastructure
- LNET is an abstraction for underlying network type
- Supported network types include:
  - TCP/IP
  - Infiniband
  - Cray high-speed interconnects (Gemini, Aries)
- LNET routing capabilities allow fine-grained control of data flow

Example: Simple Lustre Setup



- Combined MDS/MGS
- All hosts directly attached to the same Infiniband fabric (no routing)

Example: Complex Lustre Setup

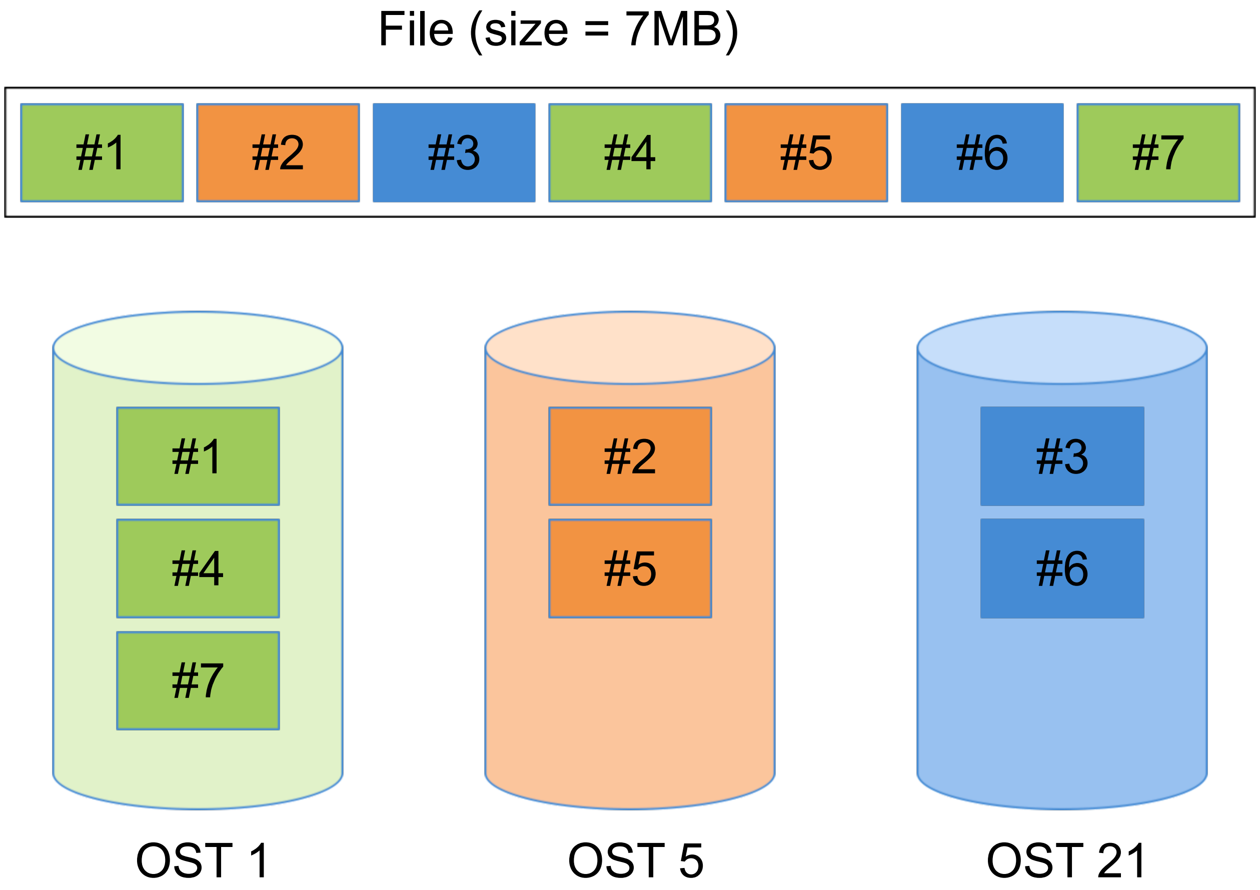


- Lustre servers connected to two different fabrics.
- LNET routers forward traffic between Infiniband networks.

File Striping Concepts

- The two most basic properties of a Lustre file are:
  - *stripe_count* (the number of OSTs to stripe across)
  - *stripe_size* (how much data is written to an OST)
- Users can control these parameters using "`lfs setstripe <file>`" or allow the file to inherit the global defaults
- When a file is created, Lustre will select *stripe_count* OSTs to use for the file.
- The first *stripe_size* bytes are written to the first OST, the second *stripe_size* bytes to the second OST, etc.

File Striping Example



stripe_count = 3 stripe_size = 1 MB

I/O Flow: A Client Perspective

- When the client opens a file, it sends a request to the MDS server
- The MDS server responds to the client with information about how the file is striped (which OSTs are used, stripe size of file, etc.)
- Based on the file offset, client can calculate which OST holds the data
- Client directly contacts appropriate OST to read/write data

I/O Optimization

- There are no hard-and-fast rules on how to optimize I/O for a Lustre file system.
- Full optimization requires in-depth knowledge of the application's I/O pattern (and may even require changes to the application).
- Optimization can also depend upon characteristics of the file system itself.
- Fortunately, significant benefits can often be achieved with relatively small changes

Lustre I/O Suggestions

- Avoid over-striping
  - More stripes does not necessarily mean faster access
  - For file sizes of O(1GB), stripe_count=1 may be best
- Avoid under-striping
  - Very large files with stripe_count=1 can fill up an OST
  - If many clients are writing to separate portions of the same large shared file, a low stripe_count could cause contention on OSTs
- Avoid small I/O requests
  - If possible, buffer many small writes into larger requests
- Know your application's I/O pattern!

Summary

- Lustre is a scalable parallel file system that can handle some very demanding I/O loads
- Lustre can support simple small-scale configurations as well as very complex large-scale configurations
- Careful tuning of file striping parameters can yield significant improvements in application performance by avoiding I/O contention

Acknowledgements

This work was supported by the United States Department of Defense (DoD) and used resources of the DoD-HPC Program at Oak Ridge National Laboratory.