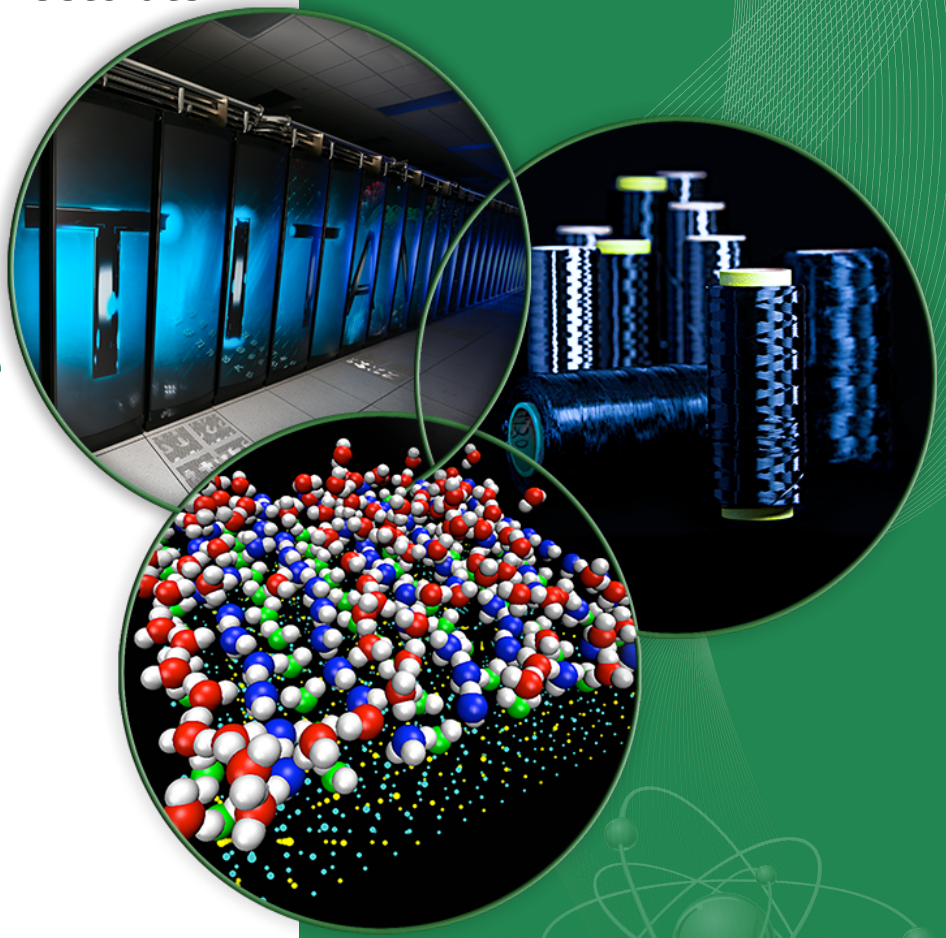
Oak Ridge National Laboratory

Computing and Computational Sciences Directorate



Hardware Selection and Benchmarking for Lustre

Rick Mohr
Jeffrey Rossiter
Sarp Oral
Michael Brim
Jason Hill
Jesse Hanley
Neena Imam

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

OAK RIDGE National Laboratory

Outline of Topics

- Part I: Hardware Selection
  - Selection criteria
  - Server guidelines for MGS/MDS/OSS
  - Networking guidelines
  - Client guidelines
- Part II: Benchmarking Methods
  - Purpose of benchmarking
  - Bottom-up approach to benchmarking
  - Benchmarking tools and techniques

Part I: Hardware Selection

Overview of Selection Process

- Appropriate hardware choices will be driven by many factors
- Consider higher level goals
  - Typical use (production, test/development, evaluation)
  - Policies and procedures
  - Integration with existing resources
- Narrow choices by considering performance requirements
  - Storage capacity and bandwidth
  - Network bandwidth and latency

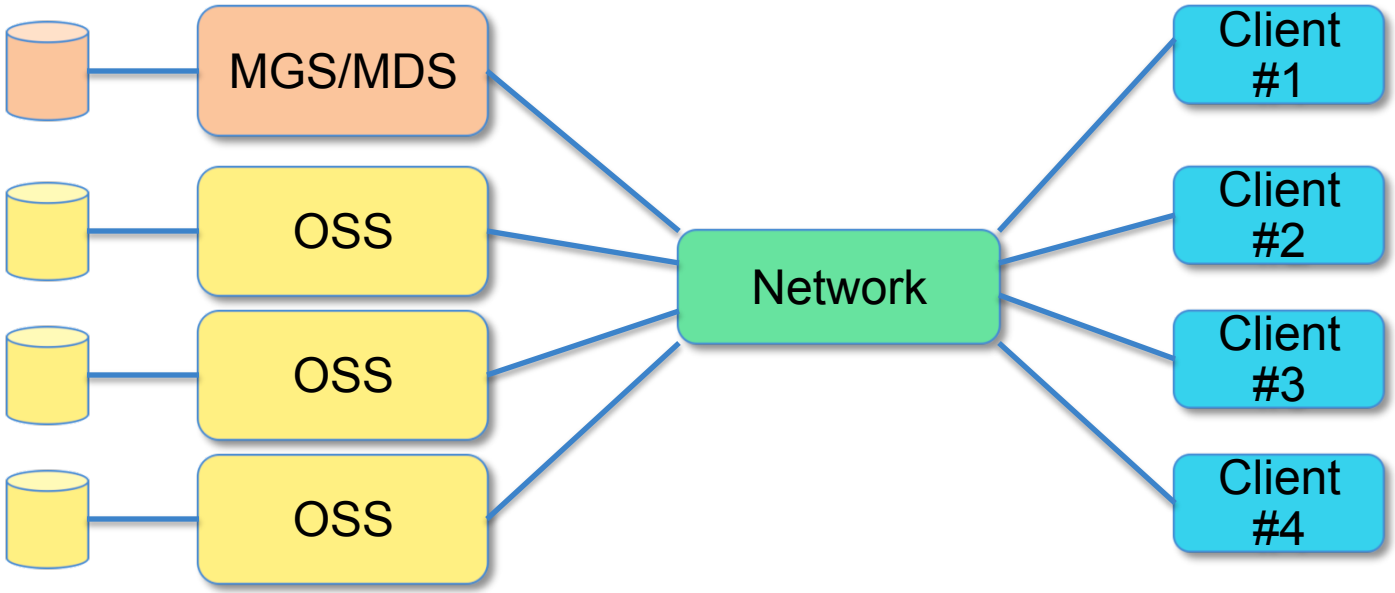
Selection Criteria

- Typical use for file system
  - Production → Consider RAID level (data protection), hardware redundancy/failover (improved uptime)
  - Testing/Development → Closely mimic existing (or expected) file system hardware
  - Evaluation → Flexibility to integrate different types of resources
- Policies and procedures
  - Security policy restrictions
  - Scratch space vs. Long-term storage

Selection Criteria (cont.)

- Integration with existing resources
  - Compatibility with currently deployed hardware
  - System management requirements
- Performance requirements
  - Capacity
  - Bandwidth (disk and network)
  - Latency
- Application I/O patterns
  - If file system is intended to support a small set of specific applications, gather info about typical workflows

Simple Lustre Setup



- Combined MDS/MGS
- All hosts directly attached to the same network fabric (no routing)
- Exact number of servers/clients in this example is not important

MGS/MDS Server Guidelines

- MGS and MDS can coexist on same server, but separate servers can be beneficial
  - Multiple file systems can use same MGS
  - MGS server can server as backup MDS server
- MDS is CPU intensive
  - Minimum 4 processor cores recommended
  - Faster cores are usually better
- More memory allows MDS to cache more metadata
  - Helps reduce lots of small I/O requests to disks
  - Allows server to maintain more client locks

MGS/MDS Storage Guidelines

- MGT Storage
  - Space requirements are small
  - Infrequent access, so performance not critical
  - Data is important so be sure to mirror disks
- MDT Storage
  - Access pattern is database-like (many seeks, small I/O)
  - Use fast disks if possible (high-RPM SAS ,SSD)
  - Data is critical! Use RAID-10.
  - External journal can improve performance
- Failover → Accessibility from multiple servers.

MGS/MDS Storage Guidelines (cont.)

- MGT requires <100 MB of disk space
- MDT space requirements are more complex
  - Size of MDT determines number of inodes available in the file system
  - Backend storage format (ldiskfs vs. ZFS) can affect calculations
  - Rough estimate: 1-2% of total file system capacity
  - Better estimate: Plan for 2 KB per inode
    - If N = (number of desired inodes for file system), then

      (MDT size) = 2 x (N x 2 KB) ← 2x fudge factor
  - If in doubt, err on the side of caution and get more space

OSS Server Guidelines

- OSS is not CPU intensive
  - Much of the time is spent waiting for I/O requests
  - May be desirable to get newer CPUs for better bus and memory access speeds
- More memory allows OSS to cache more data
  - Not necessary to enable caching, but for certain I/O patterns caching may reduce number of disk accesses
- Network interfaces, busses and motherboards
  - Pay close attention to possible hidden bottlenecks
    - PCI bus slower than your Infiniband card
    - Multiple network interface on shared PCI bus

OSS Storage Guidelines

- OSTs provide the file system storage space
- OSS servers typically serve 2-8 OSTs
- OSTs can use various technologies
  - Hardware storage controllers (e.g., DDN, EMC, etc.)
  - External JBODs w/ ZFS
  - Internal drives with LVM and/or software RAID
- Cost vs. Performance vs. Capacity vs. Complexity
- RAID is a must (even for scratch space)
  - Need to keep running in the event of drive failures
  - RAID6 (8+2) is often a good choice

Network Guidelines

- Network technology may be determined by various site factors
  - Are admins trained? Works with existing hardware?
- Make sure Lustre has LNET support for the network
  - Ethernet and Infiniband are common, but there's also support for specialized networks (e.g., Cray Aries)
- Balance network vs. disk bandwidth per OSS

network_bw > disk_bw	disk_bw > network_bw
• Network may be underutilized • Might be OK if expansion is planned	• Often the case if capacity is needed • Helps increase disk utilization if I/O pattern not optimal

Lustre Client Guidelines

- Not many hardware constraints on Lustre clients (although 64-bit clients are recommended)
- Lustre client architecture/endianess can be different from the server
  - Caveat: The PAGE_SIZE kernel macro on the client must be as large as the PAGE_SIZE on the server
    - Lustre client on ia64 with 64 KB pages can run with x86 servers with 4 KB pages
    - If servers are ia64 and clients are x86, the ia64 kernel must be compiled with 4KB pages
- In general, don't use Lustre servers as Lustre clients too

Case Study: Atlas Hardware

- Atlas is OLCF's site-wide Lustre resource
  - Two types of storage used:
    - DDN SFA12KX for OSTs with RAID-6 (8+2)
    - NetApp 5524 for MDTs with RAID-10
  - OSS Servers
    - Dual socket with 8-core 2.6 GHz Ivy Bridge processors
    - 64 GB RAM
  - MDS/MGS Servers
    - Dual socket with 6-core 2.6 GHz Sandy Bridge processors
    - 256 GB RAM
  - FDR Infiniband network

Part II: Benchmarking Methods

Goals of Benchmarking

- Benchmarking has several purposes
  - Verify hardware performance
    - Make sure hardware lives up to vendor's claims
    - Discover faulty hardware early
  - Discover hidden bottlenecks
    - Design looks good on paper, but in practice it doesn't work well
  - Record baseline behavior
    - Helps quantify what is "normal" and identify regressions later
- Need to use a bottom-up approach
  - Test individual hardware components
  - Add software/hardware layers incrementally

General Benchmark Plan

- Should create a benchmark plan as part of the file system deployment plan
  - Benchmark plan will likely have site-specific tests
  - Benchmarking may be part of formal system acceptance from vendor
- In general, benchmarks should test:
  - Storage
  - Network fabric
  - Lustre Lnet transport layer
  - Lustre file system

Storage Benchmarking

- Test performance of the block devices that will be used for MDT and OST storage. These could be:
  - Individual disks
  - LUNs exported to host by external storage controller
  - Software RAID devices
- OST benchmarks typically focus on streaming I/O performance with large (1MB+) request sizes
  - This gauges max speed of OST which ultimately determines max speed of entire file system
- MDT benchmarks focus on random I/O with small request sizes (usually 4KB)

Storage Benchmarking (cont.)

- Lustre comes with an I/O kit containing several benchmark tools, including `sgpdd-survey`
  - `sgpdd-survey` is a shell script that uses `sgp_dd` command to perform I/O
  - Measures “bare metal” performance, bypassing kernel block device layers and buffer cache
  - Runs multiple tests with varying numbers of threads and regions to create a performance profile
- This tool is useful for testing performance of a single OST or MDT
- Lustre manual has a section on `sgpdd-survey`
  - http://lustre.org/documentation/

Storage Benchmarking Tools

- In addition to Lustre I/O kit, there are many other benchmark tools available.
- XDD (eXtreme dd toolset)
  - Multi-threaded capabilities. Supports I/O to block devices or files
  - https://github.com/bws/xdd
- fair-lio
  - Developed at ORNL. Uses libaio (async I/O library).
  - Basis for OLCF benchmark suite
  - https://www.olcf.ornl.gov/wp-content/uploads/2010/03/olcf-benchmark-suite-final-rev-1.tar.gz

Network Benchmarking

- Test the speeds of network links between hosts
- Use tools appropriate to the network fabric
- Ethernet
  - iperf (https://github.com/esnet/iperf)
  - netperf (http://www.netperf.org/netperf/)
- Infiniband
  - qperf (https://www.openfabrics.org/downloads/qperf/)
    - Can test RDMA and IP performance
  - perftest (https://www.openfabrics.org/downloads/perftest/)
    - Contains ib_write_bw, ib_send_lat, etc.

LNet Benchmarking

- Lustre comes with a kernel module (lnet_selftest) that can be used to test the LNet transport layer.
- Can be used to send/receive bulk I/O data between multiple nodes simultaneously
  - Good for trying to saturate the network
  - Useful to compare LNet bandwidth test results to network bandwidth test results
- Capable of testing paths through LNet routers
- For details, see the Lustre manual.

Lustre Benchmarking

- Once hardware components have been tested and verified to be working as expected, file system benchmarks can be run.
- There are many tools available, but one of the most commonly used is IOR.
  - https://github.com/chaos/ior
  - Uses MPI to coordinate processes across multiple nodes
  - Supports file-per-process and shared file testing
  - Can be built with support for POSIX, MPIIO, and HDF5
  - Many options available to support various read/write tests
  - See doc/USER_GUIDE included in IOR source

Lustre Benchmarking (cont.)

- Different scenarios to test:
  - Max OST bandwidth
    - Use file-per-process test with all files located on a single OST
    - Test varying numbers of processes (n=1,2,4,....)
  - Max OSS bandwidth
    - Run previous test across all OSTs on one OSS server concurrently
  - Max client bandwidth
    - Multiple processes writing to different files on different OSTs
  - Max file system bandwidth (i.e. – Hero Run)
    - Best results usually achieved using file-per-process across many clients with stripe_count=1
    - May want to manually assign files to OSTs to achieve maximum throughput

Summary

- Lustre hardware choices may require sites to consider more than just capacity and bandwidth.
- The Lustre manual is a valuable resource for understanding the technical requirements and planning a file system deployment.
- A methodical, bottom-up approach to benchmarking can help prevent hidden surprises or hours of debugging when things don't work.
- Many benchmarking tools exist to help ensure sites can get the most out of their Lustre file system.

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.