Future of storage: Lustre

Dimitri Bourilkov, Yu Fu, Bockjoo Kim, Craig Prescott,
Jorge L. Rodiguez, Yujun Wu

Outline

- Introduction;
- Lustre features;
- Lustre architecture and setup;
- Preliminary experience in using Lustre
- FIU to UF-HPC test
- Questions and outlook
- Summary

Introduction

- Lustre filesystem, is a multiple-network, scalable, open-source cluster filesystem;
- Lustre components:

- MDS(Meta Data Server):

Manages the names and directories in the filesystem, not “real data”;

- OSS(Object Storage Servers)

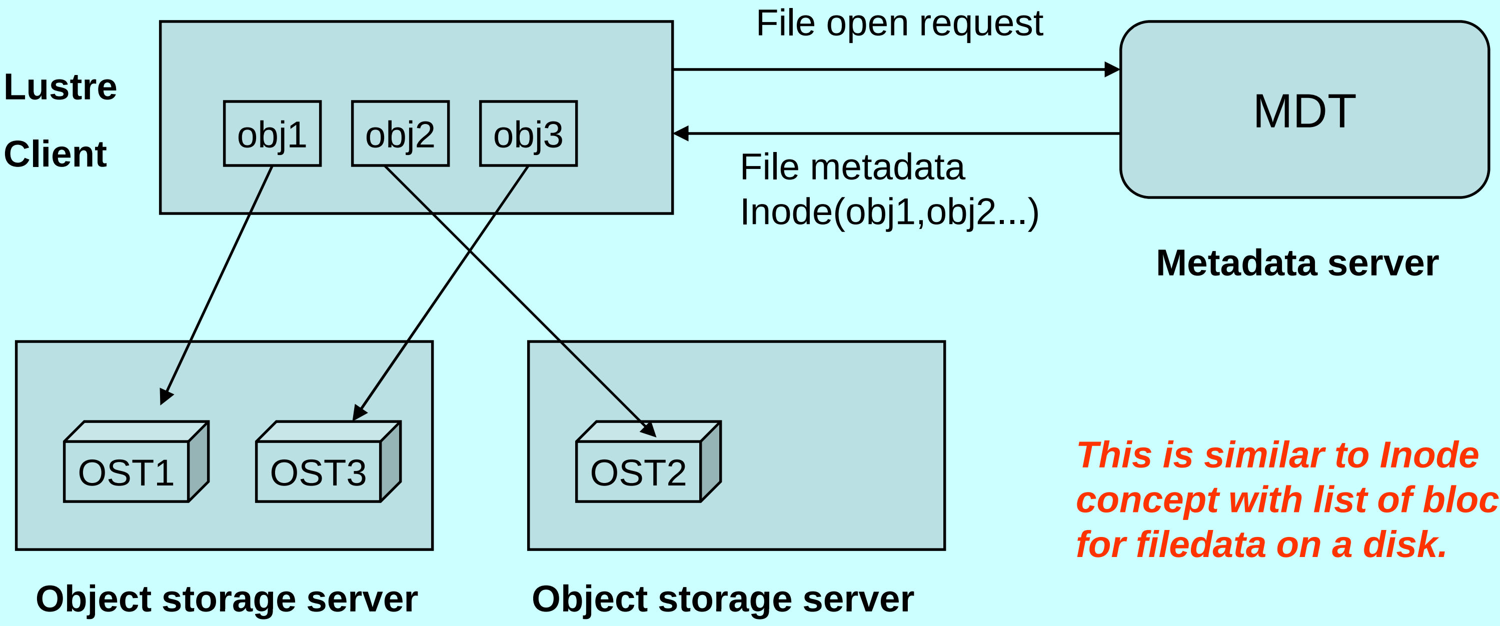
*- Contains **OST**(Object Storage Target)*

- Does the real work to store, receive, and send data

- Lustre Clients

Lustre features

- Lustre achieves high I/O performance through distributing the data objects across OSTs and allowing clients to directly interact with OSSs



This is similar to Inode concept with list of blocks for filedata on a disk.

Lustre features (2)

- Lustre is POSIX(portable operating system interface) compliant, general purpose filesystem;
- IO aggregate bandwidth scales with number of OSSs;
- Storage capacity is the total of OSTs, can grow/shrink online;
- Automatic failover of MDS, automatic OST balancing;
- Single, coherent, synchronized namespace;
- Support user quota;
- Security: supports Access Control Lists (ACLs). Kerberos is being developed;

  - *Not so good*: need clients and servers to have an identical understanding of UIDs and GIDs;
- Good WAN access performance;
- Simultaneously support multiple network types (TCP, InfiniB, Myricom, Elan….);

MSOffice1

Lustre features (3)

- Some technical numbers (from Sun whitepaper)

✓ MDS: 3,000 – 15,000 op/s

✓ OSS: ~1000 OSSs and multiple OSTs on each OSS; Maximum OST is 8TB/each;

✓ Scalability with size on a single system:

 - Production used: 1.9PB;

 - Deployed: 5PB;

 - Tested: 32PB (with 4000 OSTs);

✓ Client nodes: 25,000 nodes for a single production filesystem;

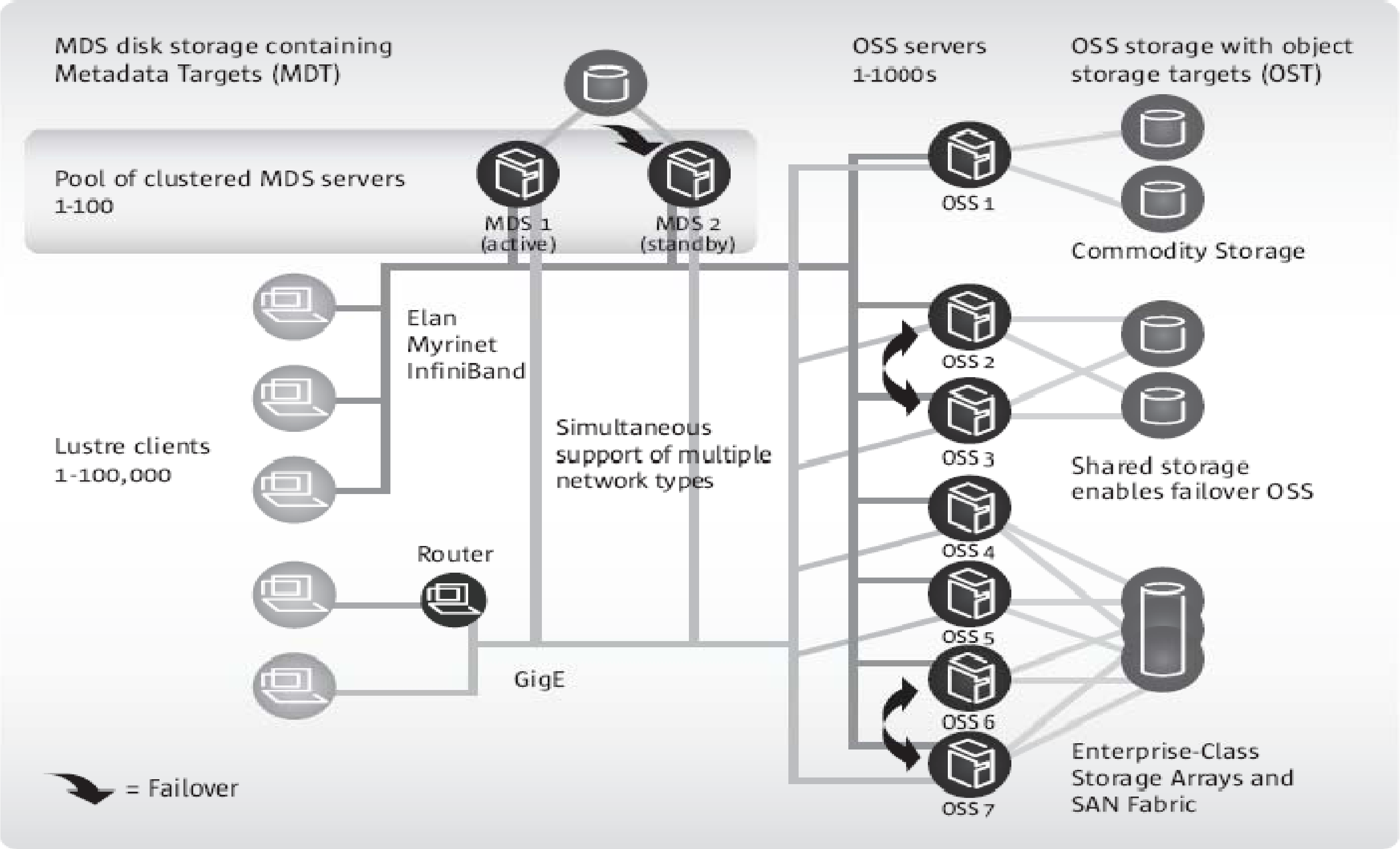
✓ IO aggregate rate can increase linearly with number of OSSs, best IO rate seen is >130GB/s (maximum seen at UF is 2GB/s);

Slide 6

MSOffice1 what kind of hardware needed for Lustre?

, 2/25/2009

Lustre architecture and setup



Lustre architecture and setup(2)

- Typical setup
  - ✓ MDS: 1-2 servers with good CPU and RAM, high seek rate;
  - ✓ OSS: 1-1000 servers. Need good bus bandwidth, storage;
- Installation itself is simple
  - ✓ Install the Lustre kernel and RPMs (download or build yourself);
  - ✓ Setup Lustre modules and modify the /etc/modprobe.conf file;
  - ✓ Format and mount the OST and MDT filesystems;
  - ✓ Start the client with mount, similar to NFS mount (client can use patchless client without modifying of the kernel);
- Notes
  - ✓ Can play with all the services(MDS,OSS) on a single node;
  - ✓ Give some time to learn and get familiar with it: 3 months(?);
  - ✓ Once it is up, manpower need is small;

Preliminary experience in using Lustre

- UF, FIU and FIT have been testing Lustre with CMS storage and analysis jobs since last year with a lot of help from UF HPC. We have basically tried with a couple of things:

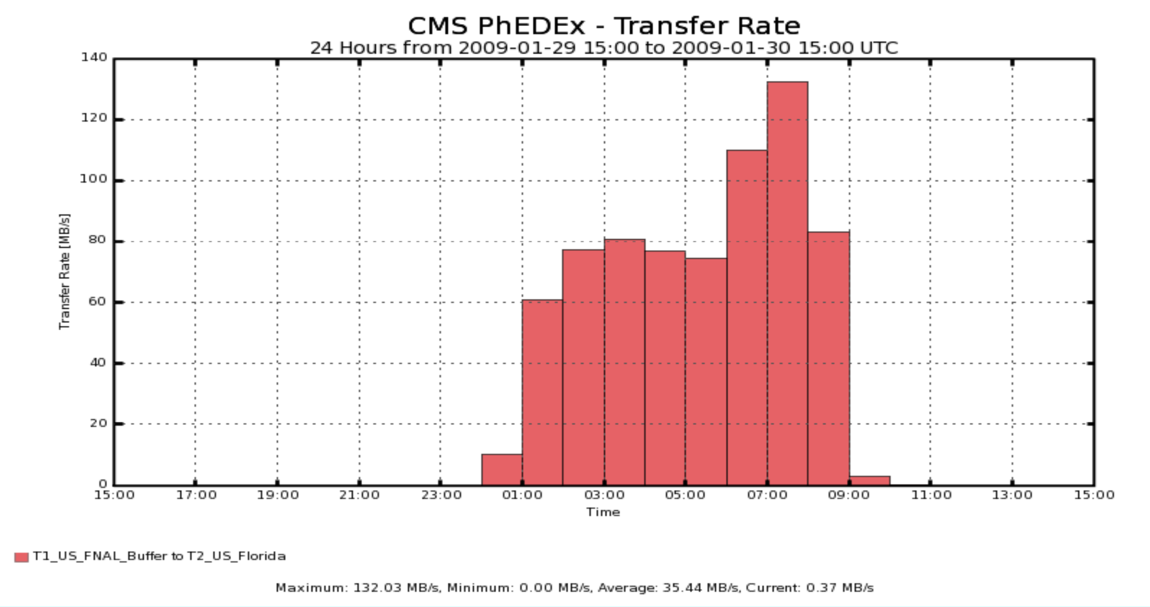
✓ Using Lustre as dCache storage;

✓ Data access performance: test data access performance of CMS analysis jobs with data stored on Lustre filesystem and comparing with the performance using dcap;

✓ Test remote access performance from FIU and FIT;

Preliminary experience in using Lustre(2)

- For dCache storage use, we have tried with using Lustre filesystem as tertiary storage (like tape) and directly as dCache pool;
- The transfer rate was able to reach over 130MB/s from a single Lustre backend pool node;



Preliminary experience in using Lustre(3)

- For CMS data access, files in Lustre can be integrated with CMS applications seamlessly without any modification:

  - Once Lustre filesystem is mounted, it acts just like you run your jobs accessing data at local disk;

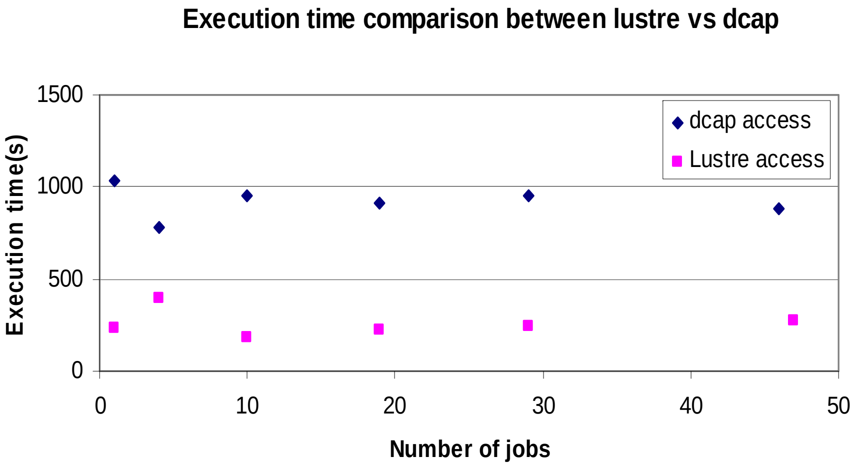
  - The IO extensive job execution time can reach 2.6 time faster when accessing files directly through Lustre mounted filesystem comparing with accessing files of the same dataset using dcap protocol that are located at a dCache raid pool with xfs filesystem (the hardware are similar);

  - The execution time can be improved even with dcap protocol when the files are put on Lustre backend pools;

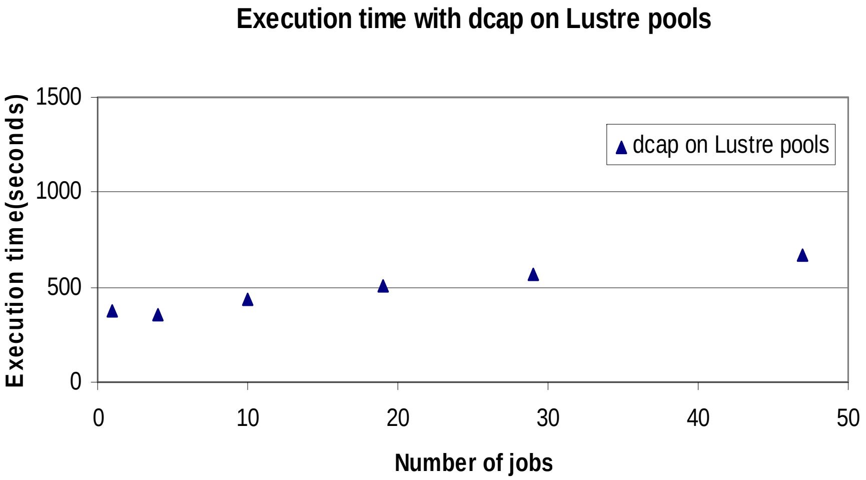
  - "Recent Storage Group report Hepix quotes number *2 better performance for jobs running on Lustre.“ --- Alex Kulyavtsev from FNAL;

Preliminary experience in using Lustre(4)

- **Execution time comparison between directly Lustre access and dcap with xfs raid pool**



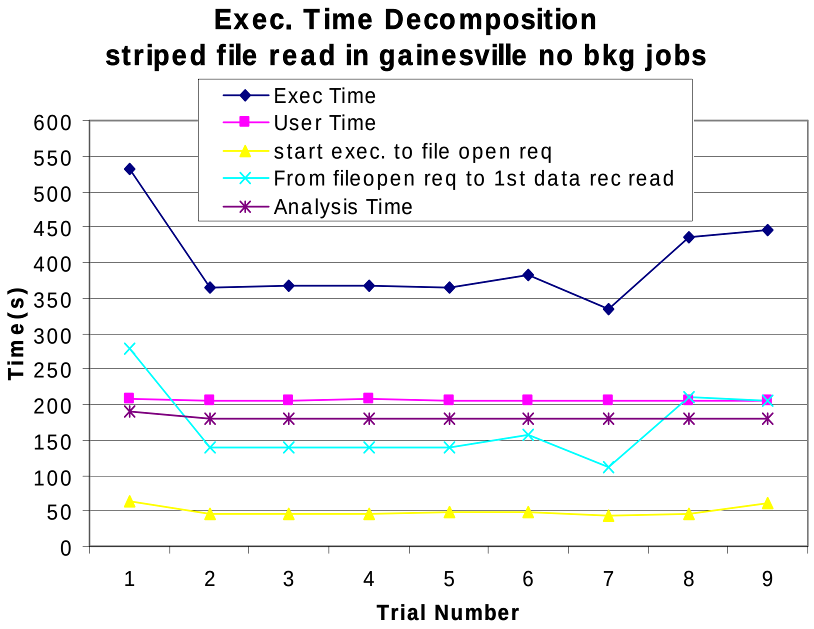
- **Execution time with dcap access using dCache pool on Lustre filesystem**

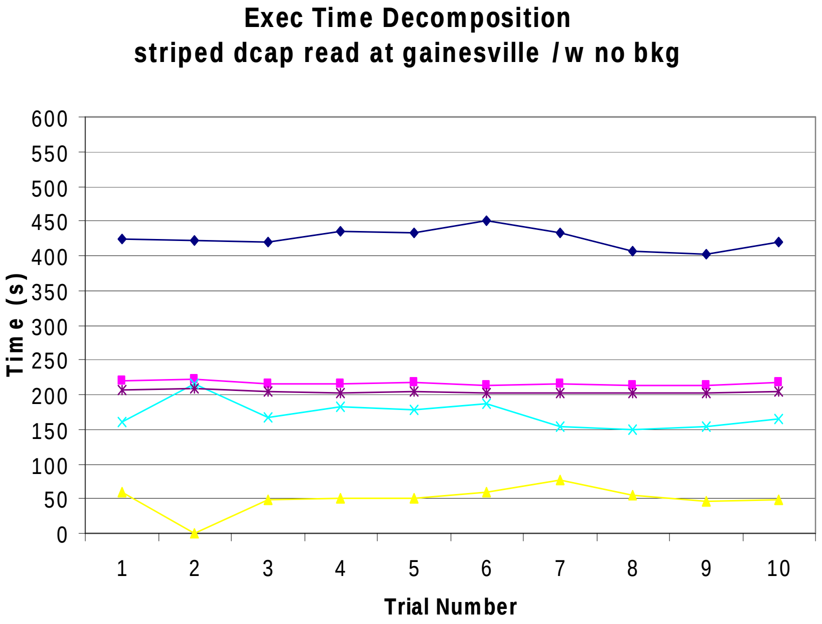


Preliminary experience in using Lustre(5)

- Bockjoo did some further detailed comparison tests on CMSSW jobs using Lustre and dcap on striped files in dcache Lustre pool:

  - One can see the major delay comparing with Lustre and dcap read comes from the analysis time and from file open request to first data record read





Preliminary experience in using Lustre(6)

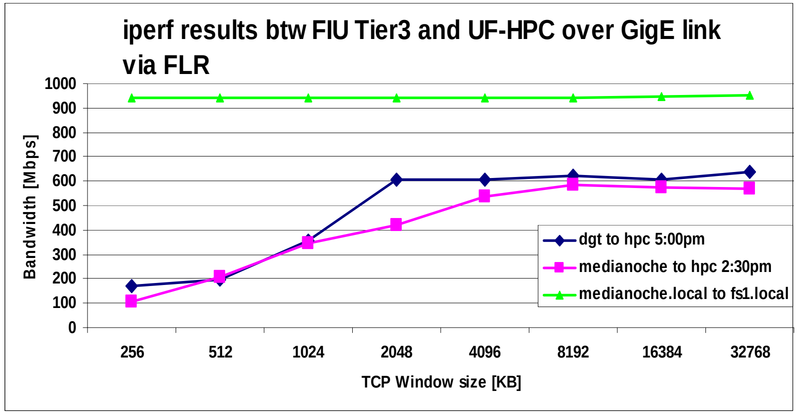
- Remotely, FIU (Jorge) has been able to run CMS application with directly mounted Lustre filesystem for data stored at UF HPC Lustre without any noticeable performance downgrade;

- UF and FIT have been testing the Lustre performance between our two sites and the performance has been only limited by our network connection. They are now able to access the CMS data stored at UF;

  - Good collaboration examples for T2 and T3 to share data and resources;

FIU to UF-HPC tests

Configuration at FIU



- Networking:
  - Connection to FLR via “dedicated link thru our CRN” border router
  - Hardware issues limit BW to ~ 600 Mbps from FIU’s CRN to FLR, RTT is 16ms
  - Server’s TCP buffer sizes set to max of 16 MB
- Systems used
  - analysis server- “medianoche.hep.fiu.edu
    - Dual 4 core with 16GB RAM with dual NICs 1GigE- priv/pub
    - Filesystems: (local), (NFS 16 TB 3ware RAID), (Lustre mounted 81 TB UFL-HPC CRN storage)
  - “gatekeeper” – dgt.hep.fiu.edu
    - Dual 2 core with 2GB RAM single NIC 1GigE private (an ageing ’03 Xeon system)
    - Also used in experiments mounting lustre over NAT on nodes on private subnet
- System configuration RHEL 4. (medianoche) and SL 5.0 (dgt)
  - Booted patched kernel 2.6.9-55.0.9.EL_lustre 1.6.4.2 on both system
  - Modified modeprob.conf
  - Mounted UF-HPC’s CRN on /crn/scratch

FIU to UF-HPC tests(2)

Configuration at UF-HPC

- Networking:
  - Connection to FLR via “dedicated link ur CRN via 2x 10Gbps links
- RAID Inc. Falcon III Storage (104 TB Raw, 81 TB volume)
  - Six shelves each with 24 ea. SATAII drives with dual 4GBps FC RAID
  - Fronted by two dual 4 core servers with 3 FC cards dual ports, 16GB RAM, infiniband and 10 Gbps Chelsio NIC
  - Mounts to FIU via Lustre over TCP/IP
  - System can read/write natively at well over 1 GBps via TCP/IP

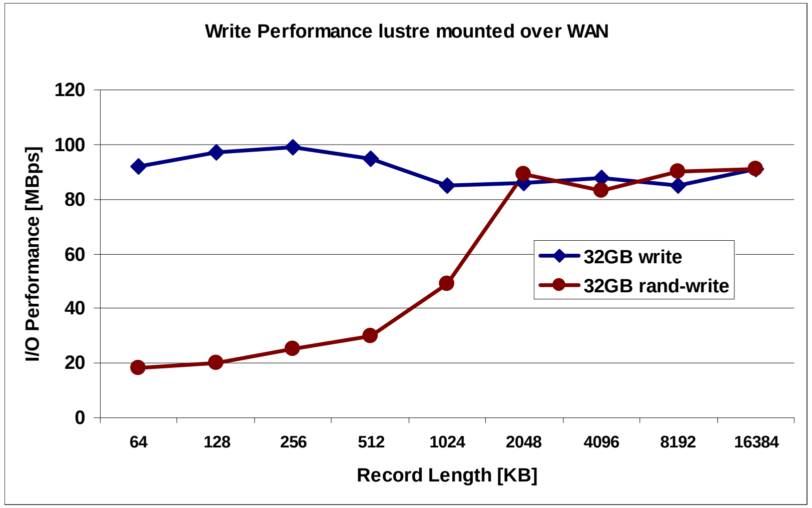
Site synchronization and security

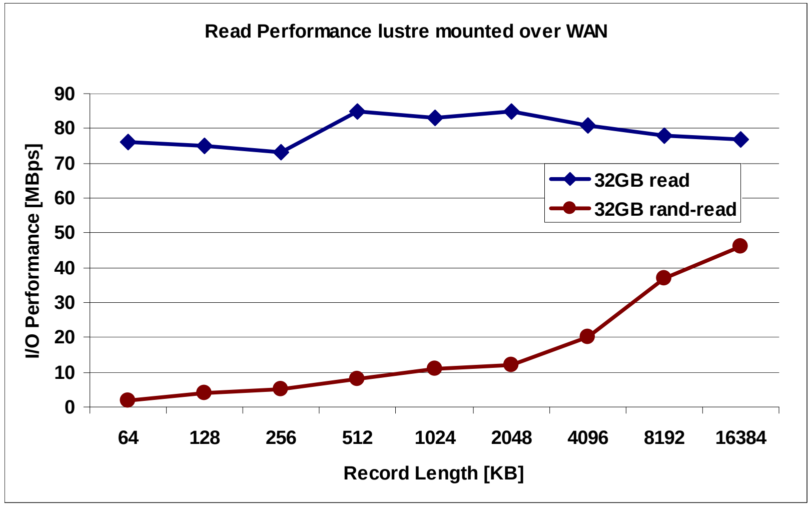
- HPC Lustre systems configured to allow mounts to particular remote servers at FIU and FIT
  - Access is granted to specific IP’s
  - Other nodes on the site can do remote Lustre mount via NAT: known to work but not tested yet
- Remote servers and HPC lustre share a common UID/GID domain
  - Not a problem for Tier3 with dedicated servers and small user community
- ACL’s and root_squash etc., available in this version of Lustre but not yet explored

FIU to UF-HPC test (3)

IOzone tests results

- Ran tests on both medianoche & dgt
  - Filesize set to 2xRAM,
  - Checked with various record lengths
  - Checked with IOzone multi-processor mode up to 8 processes running simultaneously
    - Results are robust
    - Haven't prepared plots yet…
  - Results consistent with dd read/write
  - CMSSW tested but I/O rates for a single job are a fraction of IOzone rates
- Conclusions:
  - **Either hardware configuration with single or multi-stream can completely saturate campus network link to FLR via lustre!**





FIU to UF-HPC Use Cases

- Use case scenario 1: Provide distributed scratch space read/write
  - A convenient way of collaborating
  - Both Tier2 and Tier3 analysis servers have access to same scratch space
  - **This is how we are currently using the FIU/UF-HPC space now**

- Use case scenario 2: Provide access to data storage via Lustre
  - Read-only access to dedicated remote (Tier3) server
  - Eliminate the need to deploy CMS data management services at Tier3s
    - Utilize resources including data managers at Tier2
    - No need to deploy hardware and services Phedex, SRM enabled storage… etc.
  - Tier3 compute resources could be used to access CMS data directly
    - Via NAT enabled gateway or switch or
    - Export remote Lustre mount via NFS to the rest of the Tier3 nodes.
  - This also allows interactive access to CMS data from a Tier3 remotely
    - Sometime the only way to find the answer to a problem!
  - **This has not yet been fully explored**

Questions and outlook

- Q: Can we use Lustre to simplify our dCache storage architecture, e.g., using Lustre as dCache pool storage?
- Q: What about putting an srm interface directly on Lustre filesystem?
- Q: Can we use Lustre as the primary source of CMS data sharing and distribution between T2 and T3?

  - Advantage: lower manpower/resource cost --- only fraction of data to access in CMS jobs
- Q: Use Lustre as home/data directory?
- Q: How about Lustre comparing with pNFS 4.1?

Summary

- Lustre has shown to have good performance, scalability and relatively easy to deploy and admin;
- CMS user analysis jobs have been able to run with the data stored on Lustre filesystem without any problems. And the performance can be significantly improved when the data are accessed through Lustre than being accessed through dCache directly;
- CMS T3 physicists have been able to share CMS data remotely located at UF T2 site;
- UF has been able to integrate Lustre with existing dCache infrastructure. There is a potential to simplify our storage architecture using Lustre.
- There are really a lot of potential with Lustre!