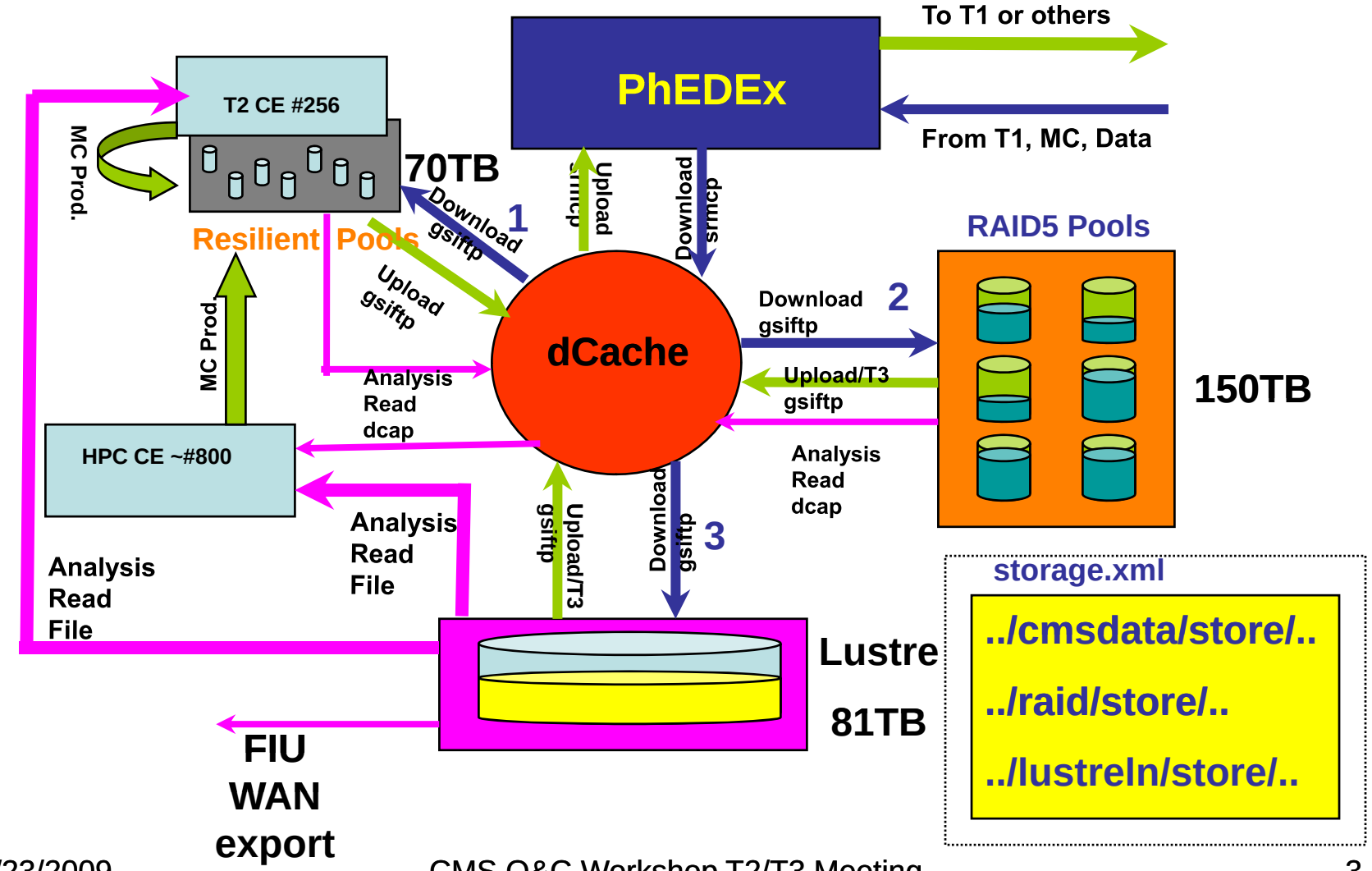
Use Lustre as the SE for a CMS Tier2 site

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

Background

- Florida T2 works closely with its institutional HPC
- The HPC tried various FS, Ibrix, gpfs, and Lustre to provide its university users with a P/IO
- They eventually chose the Lustre FS as their P/IO
- In 2008, Florida T2 had to host 60TB per local users' request. The Lustre at HPC provided the 60TB space. (Manual Transfer+Publication)
- We decided to use the Lustre FS as part of our SE since (Yujun picked up work to (PhEDEx+dcache)fy the Lustre)

Florida T2 SE



Lustre? What?

- 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)

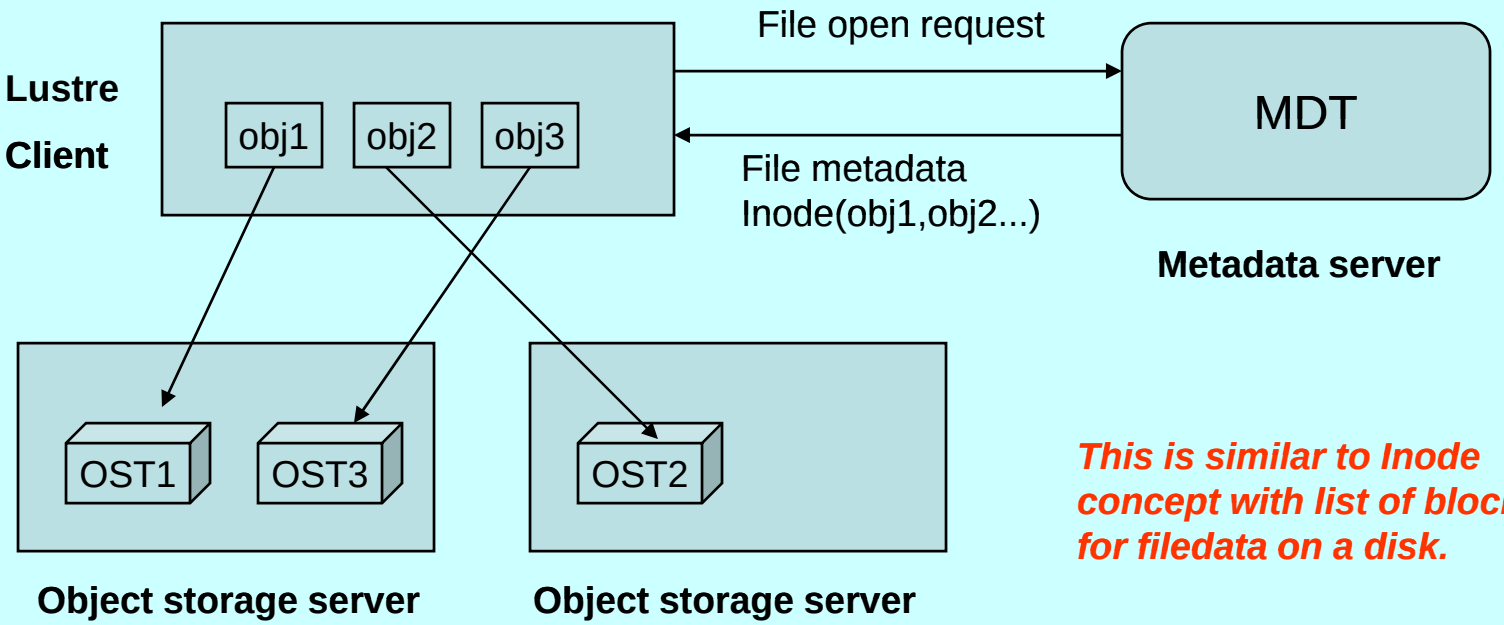
    *- Contains **OST**(Object Storage Target)*

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

  - Lustre Clients

Lustre Features (1)

- 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.

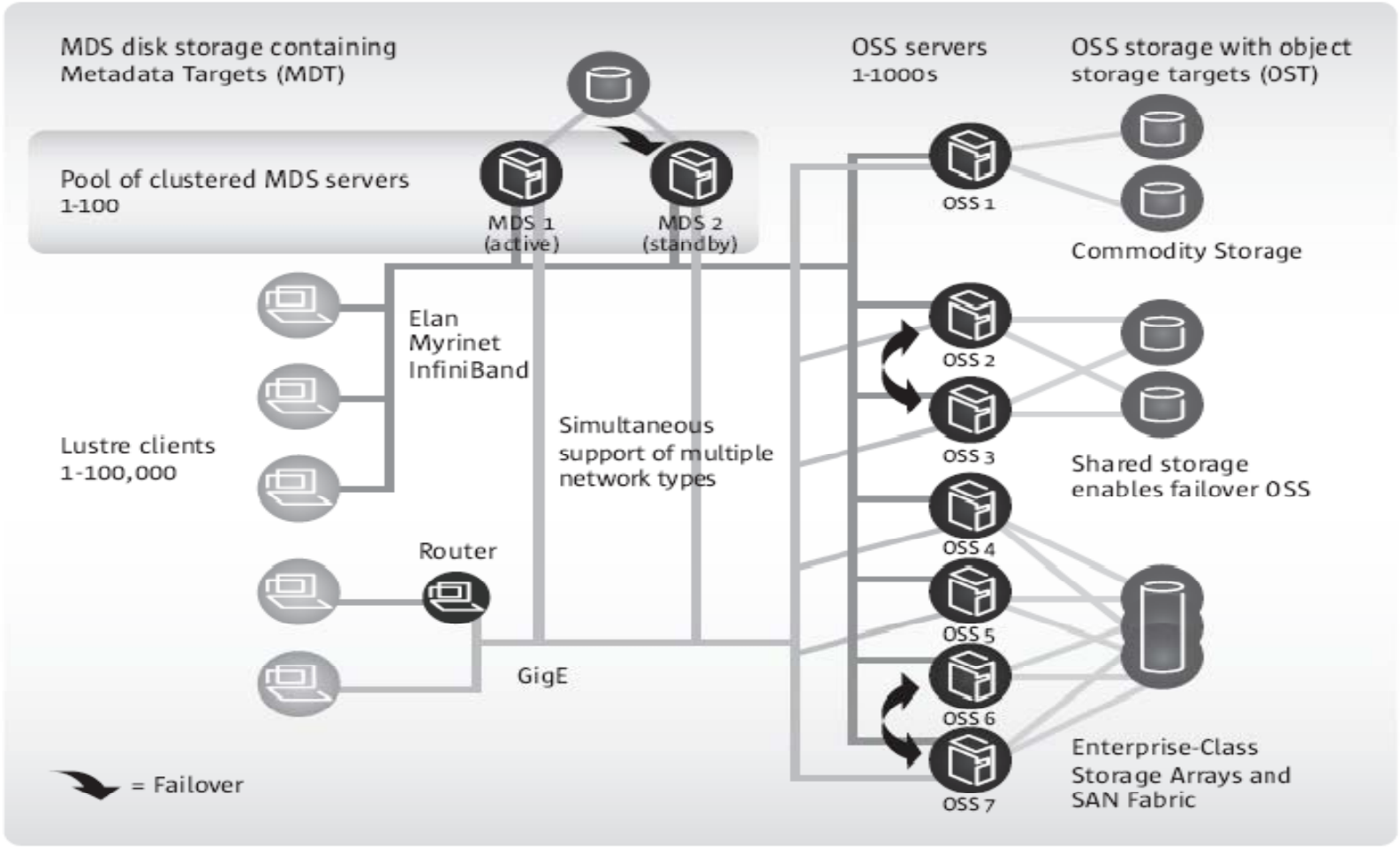
Lustre Features (2)

- 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, grow/shrink online
- Data Safety: Redundancy or RaidX
- Automatic failover of MDS, automatic OST balancing
- Single, coherent, and synchronized namespace
- Support user quota
- Security: supports Access Control Lists (ACLs). Kerberos is being developed
- Good WAN access performance
- Simultaneously support multiple network types (TCP, InfiniB, Myricom, Elan….)

Lustre Envelopes (Sun WP)

- ✓ 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)

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
  - ✓ Server: Format and mount the OST and MDT filesystems
  - ✓ Client : Install the Lustre kernel and RPMs (download or build yourself) or load Kernel modules (patcless) and mount
- 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

SRM Interface to Lustre

- Since Lustre is POSIX compliant, it is easy to add an SRM interface on top of it
- We use dCache with Lustre which worked. However, it had some function redundancy with Lustre since both Lustre and dCache manage storages on multiple servers
- Recently, we started testing Berkeley Storage Manager (BeStMan), a lightweight full implementation of SRM v2.2. BeStMan has the advantages:

✓ Works on top of existing disk-based unix file systems

✓ Full implementation of SRM v2.2, works well with dCache clients and FTS

✓ Very easy to configure and need minimal administrative efforts to maintain

✓ Works well with existing grid services, e.g., gridftp, gums, etc

Lustre Experience

- 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 Storage Element (SE)

✓ 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

Lustre Experience (2)

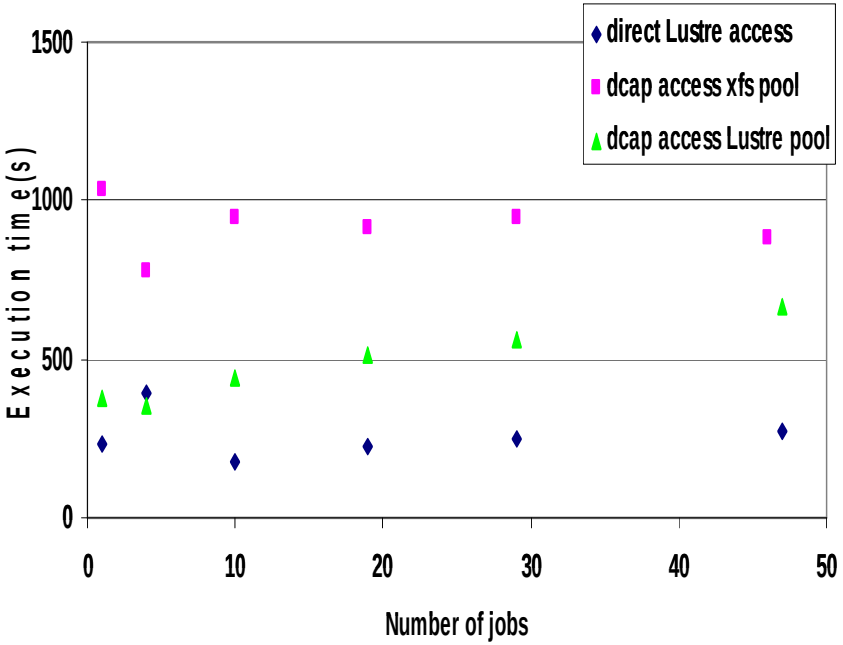
- For dCache storage use, we have tried with using Lustre filesystem as tertiary storage (like tape) and directly as dCache pools. The transfer rate was able to reach over 130MB/s from a single Lustre backend pool node
- We started to test BeStMan with Lustre for data transfer. In our PhEDEx loadtest, we were able to reach near 100MB/s from FNAL for some period of time when the injection rate was 50MB/s.

Test With Lustre

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 the jobs accessing data at local disk. We also found the performance improvement when running jobs through Lustre:

✓ 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)

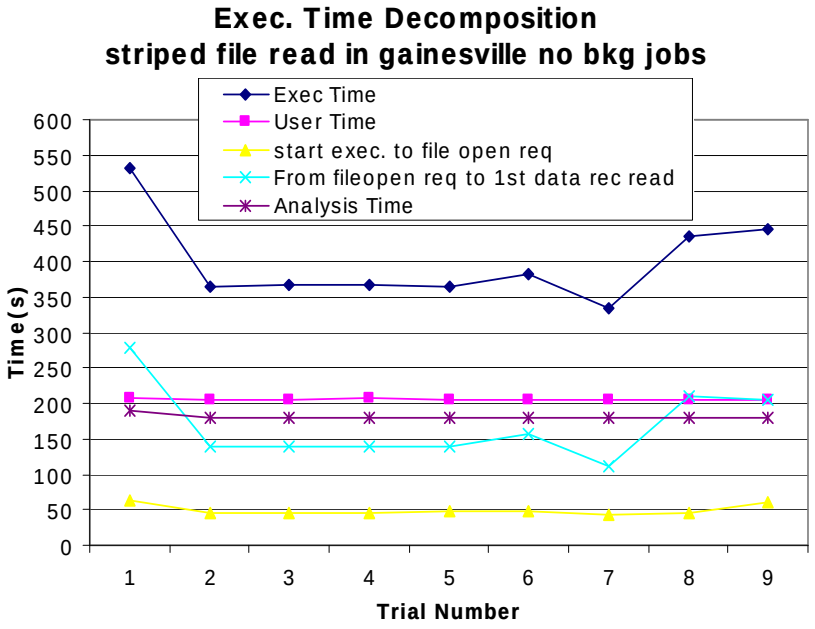
✓The execution time can be improved even with dcap protocol when the files are put on Lustre backend pools

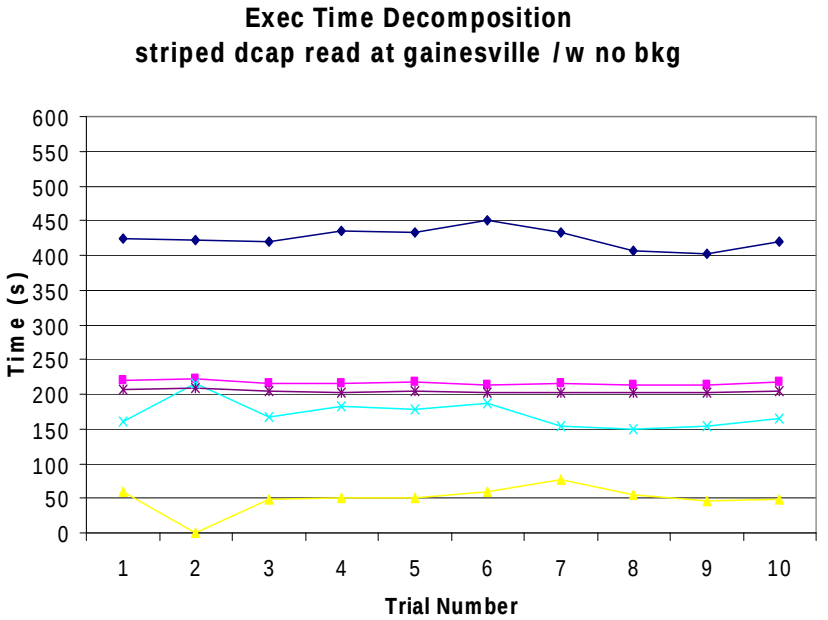


Test With Lustre (2)

- We 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





Lustre Wan Test

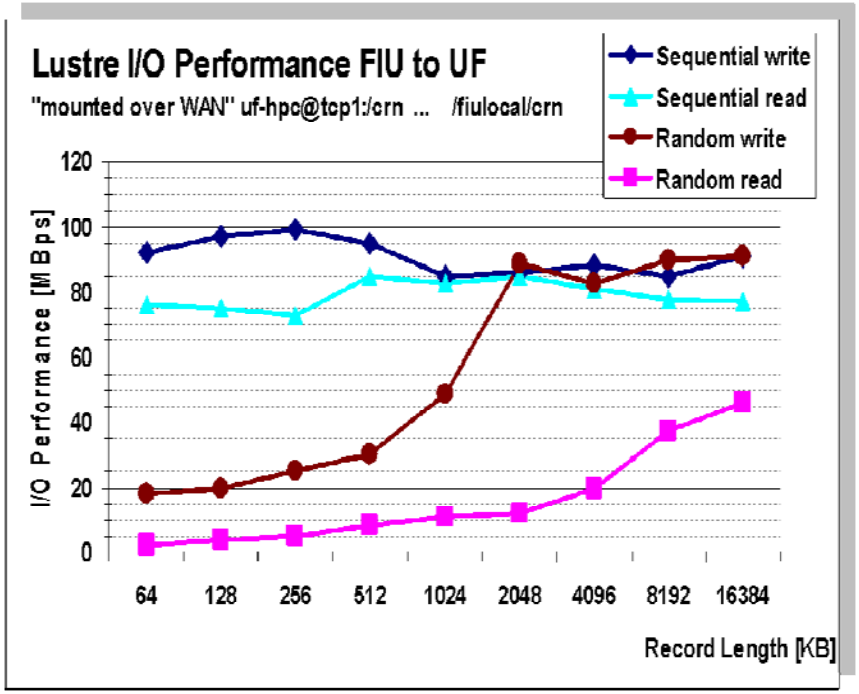
- Remotely, FIU (Jorge) has been able to run CMS application with directly mounted Lustre filesystem for data stored at UF HPC Lustre

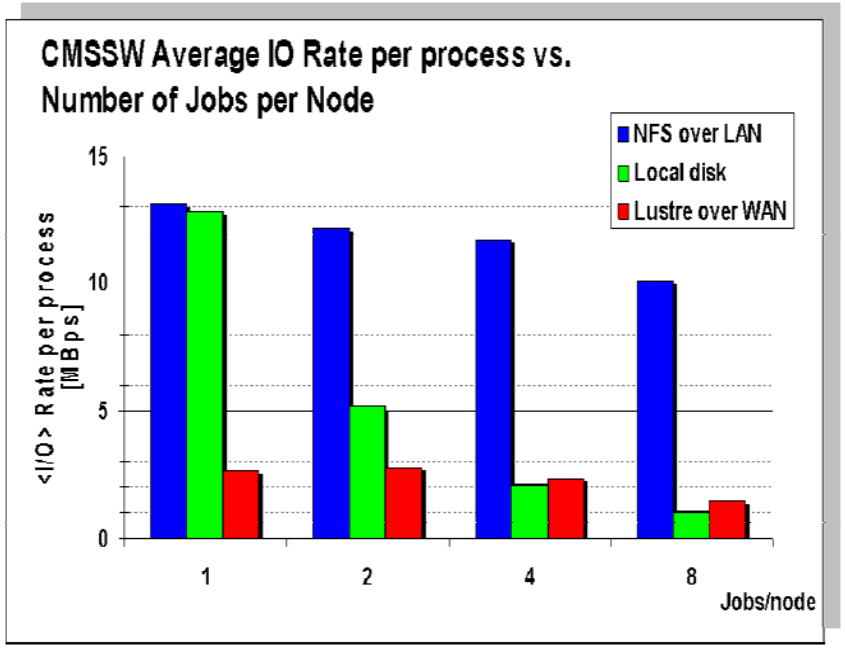
- UF and FIT have been testing the Lustre performance between our two sites and the performance has been only limited by our network connection. They are now able to access the CMS data stored at UF

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

Lustre Wan Test (2)

✓ IO performance test using IO benchmark tool IOZone can easily saturate the network link between UF and FIU

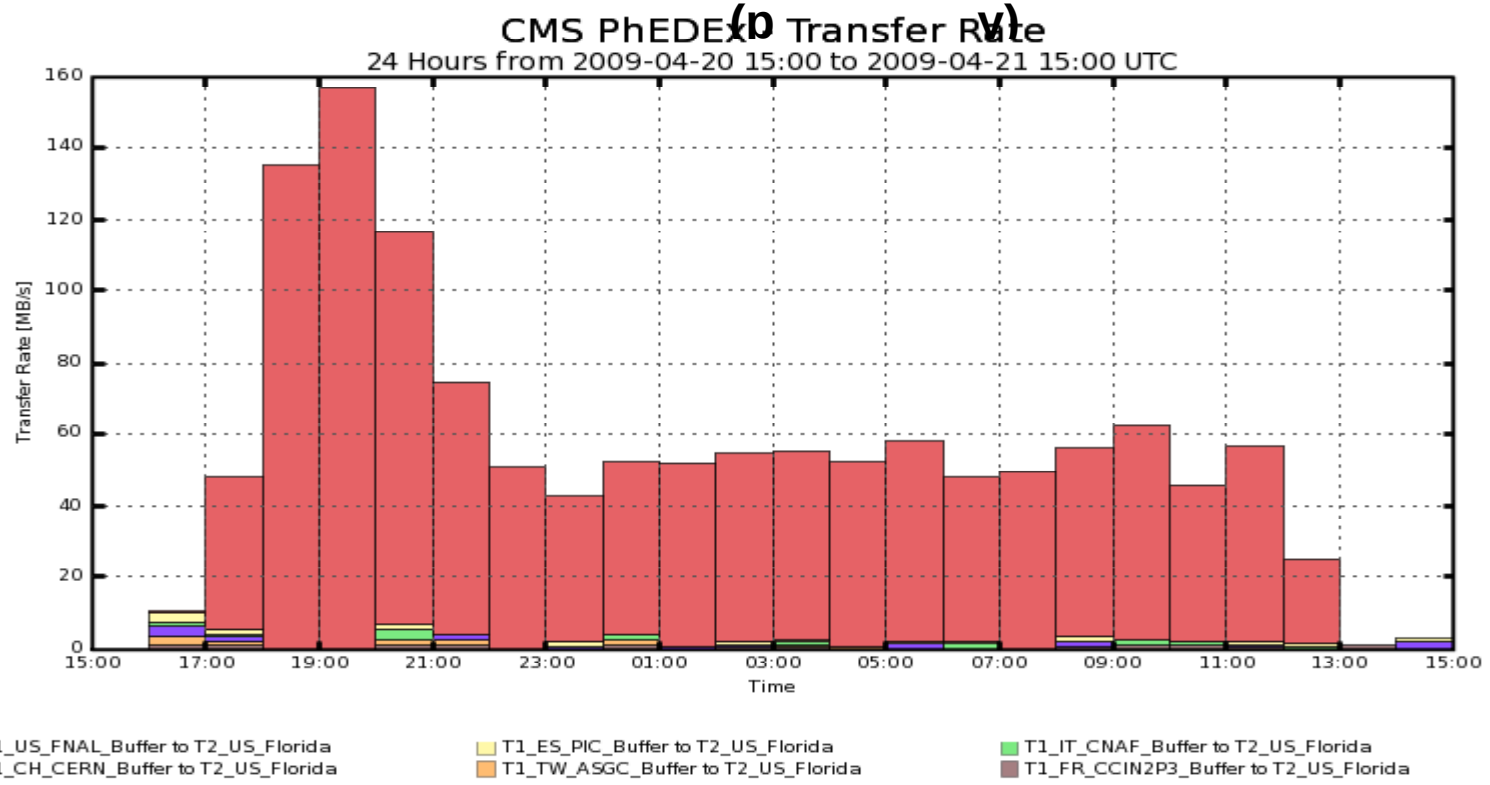




✓ CMSSW application can access data at UF through mounted Lustre filesystem from FIU (300miles away)

✓Work on understanding the low CMSSW IO rate

PhEDEx+BeStMan+Lustre



PhEDEx LoadTest using BeStMan+Lustre with 8 worker nodes gridftp servers (was running production jobs). We were able to reach 156MB/s before one gridftp server was down. Then running short of files due to the injection rate was only 50MB/s. Working on improving the rate.

Outlook and 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. This has the potential to avoid the need to deploy CMS data management services at small Tier3s and allow physicists to focus on physics
- Using lightweight SRM implementation on top of Lustre can potentially reduce our efforts in deploying and admin SE at a Tier2 site