Investigation of storage options for scientific computing on Grid and Cloud facilities

Overview

- Context
- Test Bed
- Lustre Evaluation
  - Standard benchmarks
  - Application-based benchmark
  - HEPiX Storage Group report
- Current work (Hadoop Evaluation)

Mar 24, 2011

Keith Chadwick for Gabriele Garzoglio
Computing Division, Fermilab

 Work supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359

Acknowledgements

- *Ted Hesselroth, Doug Strain – IOZone Perf. measurements*
- *Andrei Maslennikov* – HEPiX storage group
- *Andrew Norman, Denis Perevalov* – Nova framework for the storage benchmarks and HEPiX work
- *Robert Hatcher, Art Kreymer* – Minos framework

- *Steve Timm, Neha Sharma*
- *Alex Kulyavtsev, Amitoj Singh*

- This work is supported by the U.S. Department of Energy under contract No. DE-AC02-07CH11359

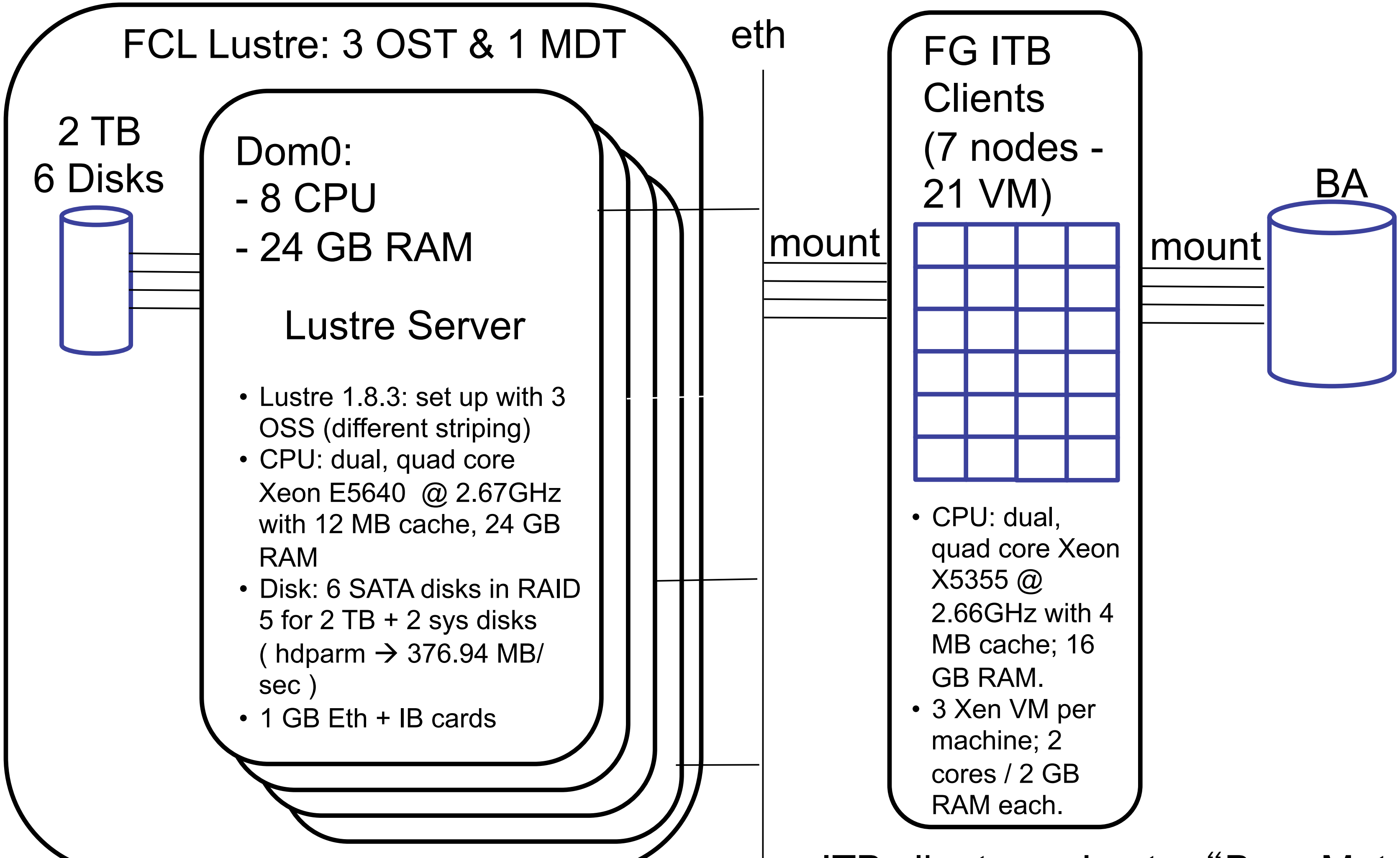
Context

- Goal
  - Evaluation of storage technologies for the use case of data intensive jobs on Grid and Cloud facilities at Fermilab.
- Technologies considered
  - Lustre (**DONE**)
  - Hadoop Distributed File System (HDFS) (**Ongoing**)
  - Blue Arc (BA) (**TODO**)
  - Orange FS (new request) (**TODO**)
- Targeted infrastructures:
  - FermiGrid, FermiCloud, and the General Physics Computing Farm.
- Collaboration at Fermilab:
  - FermiGrid / FermiCloud, Open Science Grid Storage area, Data Movement and Storage, Running EXperiments

Evaluation Method

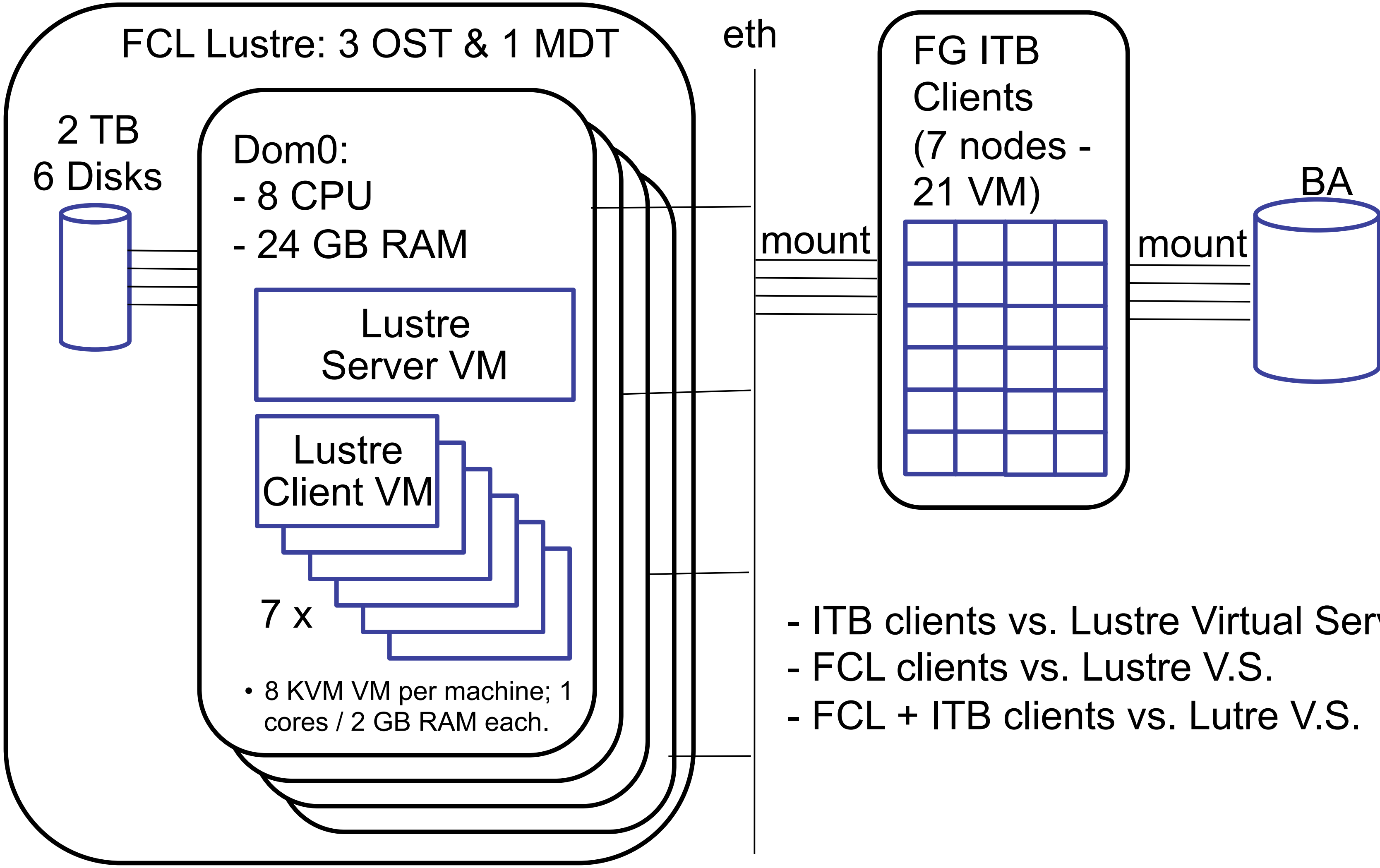
- **Set the scale**: measure storage metrics from running experiments to set the scale on expected bandwidth, typical file size, number of clients, etc.
  - http://home.fnal.gov/~garzogli/storage/dzero-sam-file-access.html
  - http://home.fnal.gov/~garzogli/storage/cdf-sam-file-access-per-app-family.html
- **Measure performance**
  - run standard benchmarks on storage installations
  - study response of the technology to real-life applications access patterns (root-based)
  - use HEPiX storage group infrastructure to characterize response to IF applications
- **Fault tolerance**: simulate faults and study reactions
- **Operations:** comment on potential operational issues

Lustre Test Bed: FCL “Bare Metal”



- ITB clients vs. Lustre “Bare Metal”

Lustre Test Bed: FCL “Virtual Server”



- ITB clients vs. Lustre Virtual Server
- FCL clients vs. Lustre V.S.
- FCL + ITB clients vs. Lutre V.S.

Data Access Tests

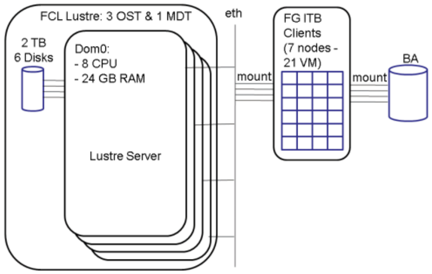
- **IOZone** – Writes (2GB) file from each client and performs read/write tests.
- **Setup:** 3-48 clients on 3 VM/nodes.

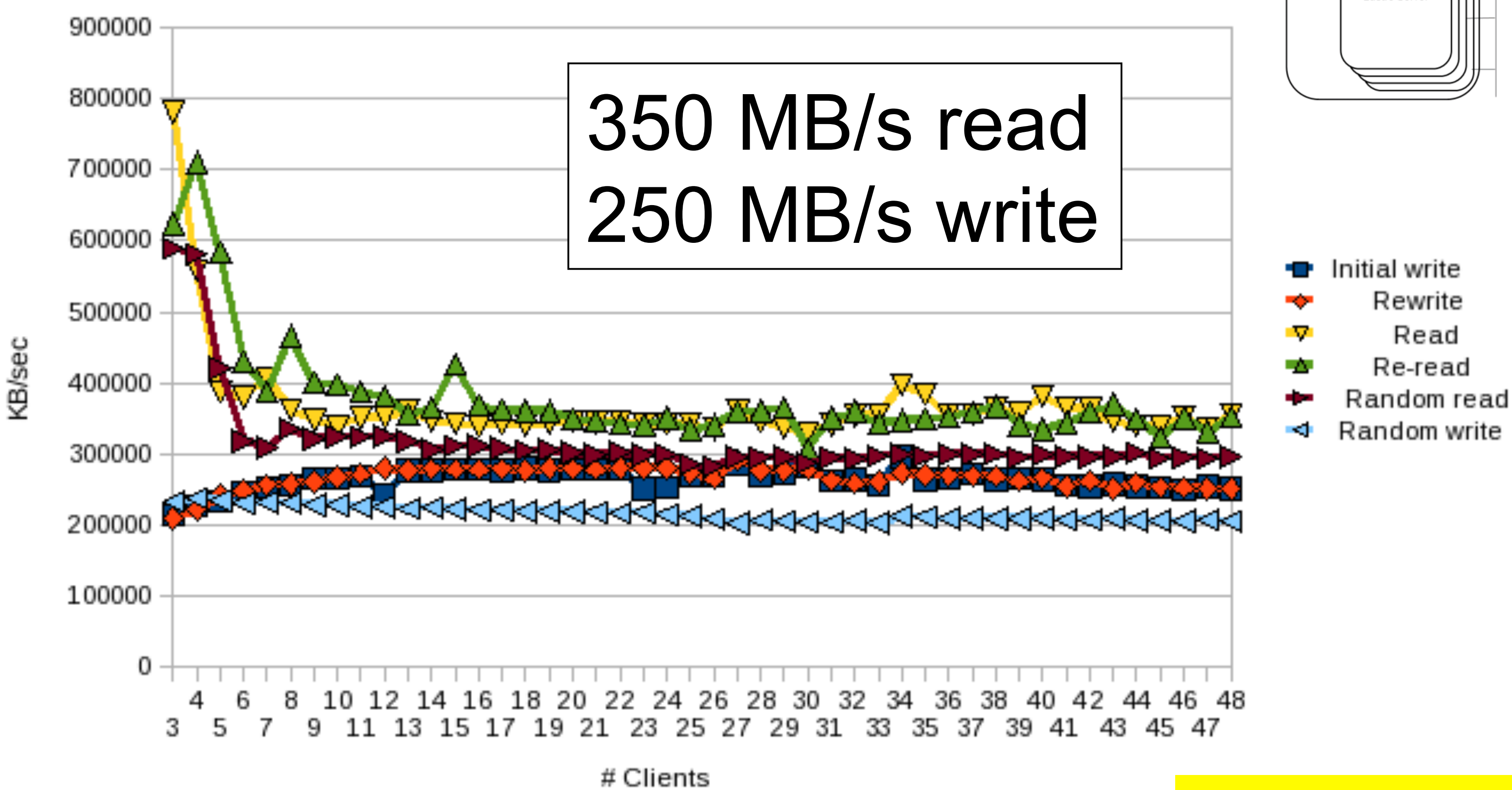
Tests Performed

difference)

  – read vs. different types of disk and net drivers for the virtual server.
  – read and write vs. number of virtual server CPU (*no difference*)
- FCL clts vs. virt. Lustre - **“on-board” vs. “remote” IO**
  – read and write vs. number of idle VMs on the server
  – read and write w/ and w/o data striping (*no significant difference*)

ITB clts vs. FCL Bare Metal Lustre





350 MB/s read
250 MB/s write

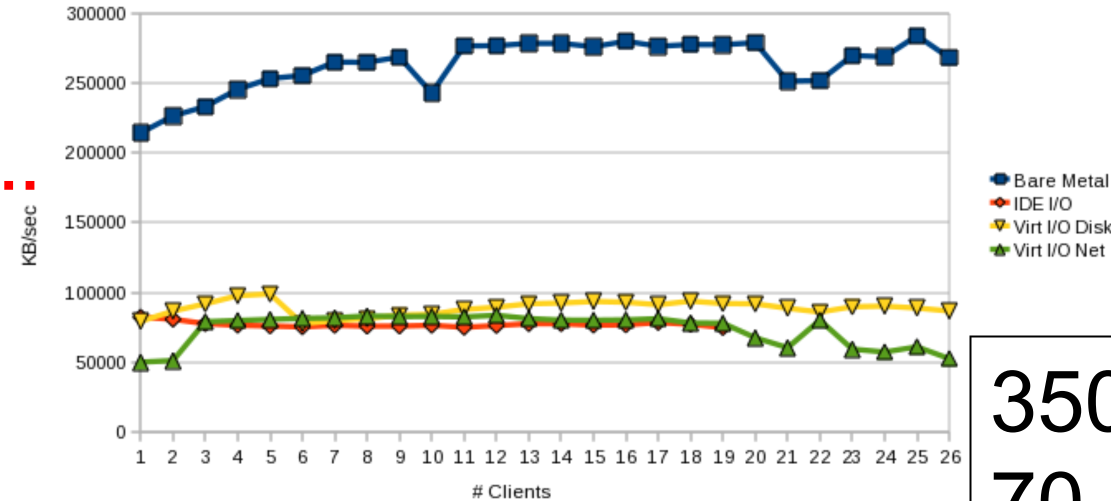
Our baseline…

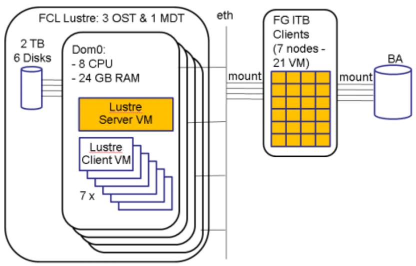
ITB clts vs. FCL Virt. Srv. Lustre

Changing Disk and Net drivers on the Lustre Srv VM…

Use Virt I/O drivers for Net

Write I/O Rates





Read I/O Rates

350 MB/s read
70 MB/s write
(250 MB/s write on Bare M.)

Bare Metal

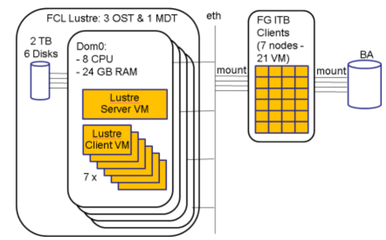
Virt I/O for Disk and Net

Virt I/O for Disk and default for Net

Default driver for Disk and Net

ITB & FCL clts vs. FCL Virt. Srv. Lustre

FCL client vs. FCL virt. srv. compared to ITB clients vs. FCL virt. srv. w/ and w/o idle client VMs...



FCL clts 15% slower than ITB clts: not significant

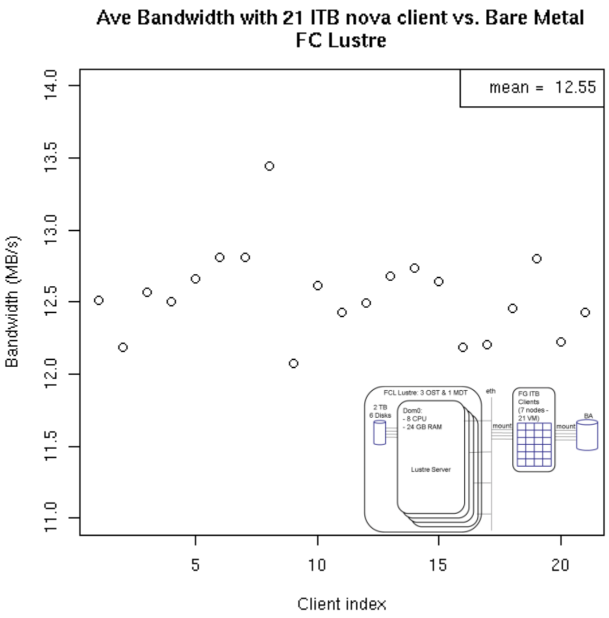
Application-based Tests

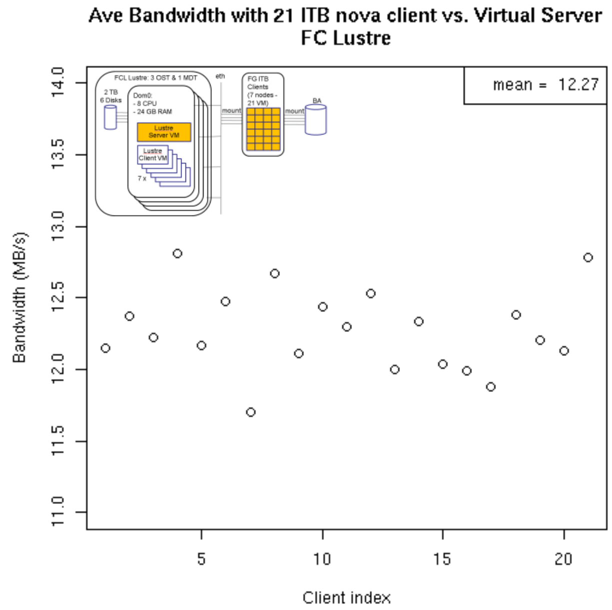
- Focusing on root-based applications:
  - Nova: ana framework, simulating skim app – read large fraction of all events → disregard all (read-only) or write all.
  - Minos: loon framework, simulating skim app – data is compressed → access CPU bound (does NOT stress storage)

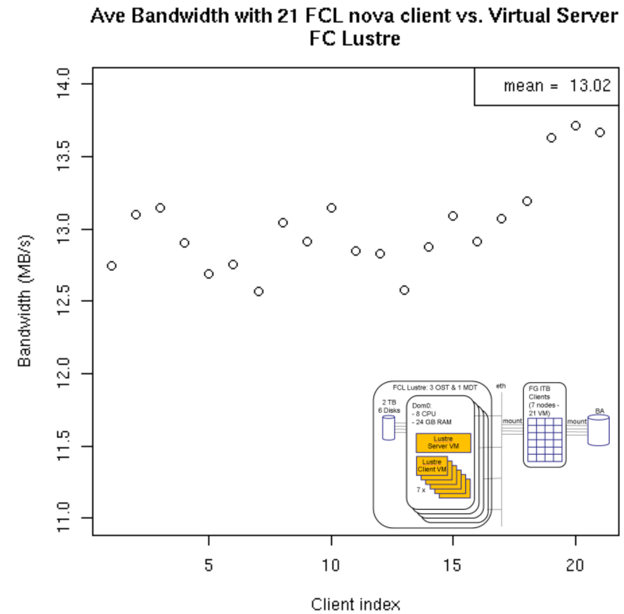
Tests Performed

- Nova ITB clts vs. bare metal Lustre – **Write and Read-only**
- Minos ITB clts vs. bare m Lustre – Diversification of app.
- Nova ITB clts vs. virt. Lustre – **virt. vs. bare m. server.**
- Nova FCL clts vs. virt. Lustre – **"on-board" vs. "remote" IO**
- Nova FCL / ITB clts vs. striped virt Lustre – **effect of striping**
- Nova FCL + ITB clts vs. virt Lustre – **bandwidth saturation**

21 Nova clt vs. bare m. & virt. srv.







Read – ITB vs. bare metal
BW = 12.55 ± 0.06 MB/s
(1 cl. vs. b.m.: 15.6 ± 0.2 MB/s)

Read – ITB vs. virt. srv.
BW = 12.27 ± 0.08 MB/s
(1 ITB cl.: 15.3 ± 0.1 MB/s)

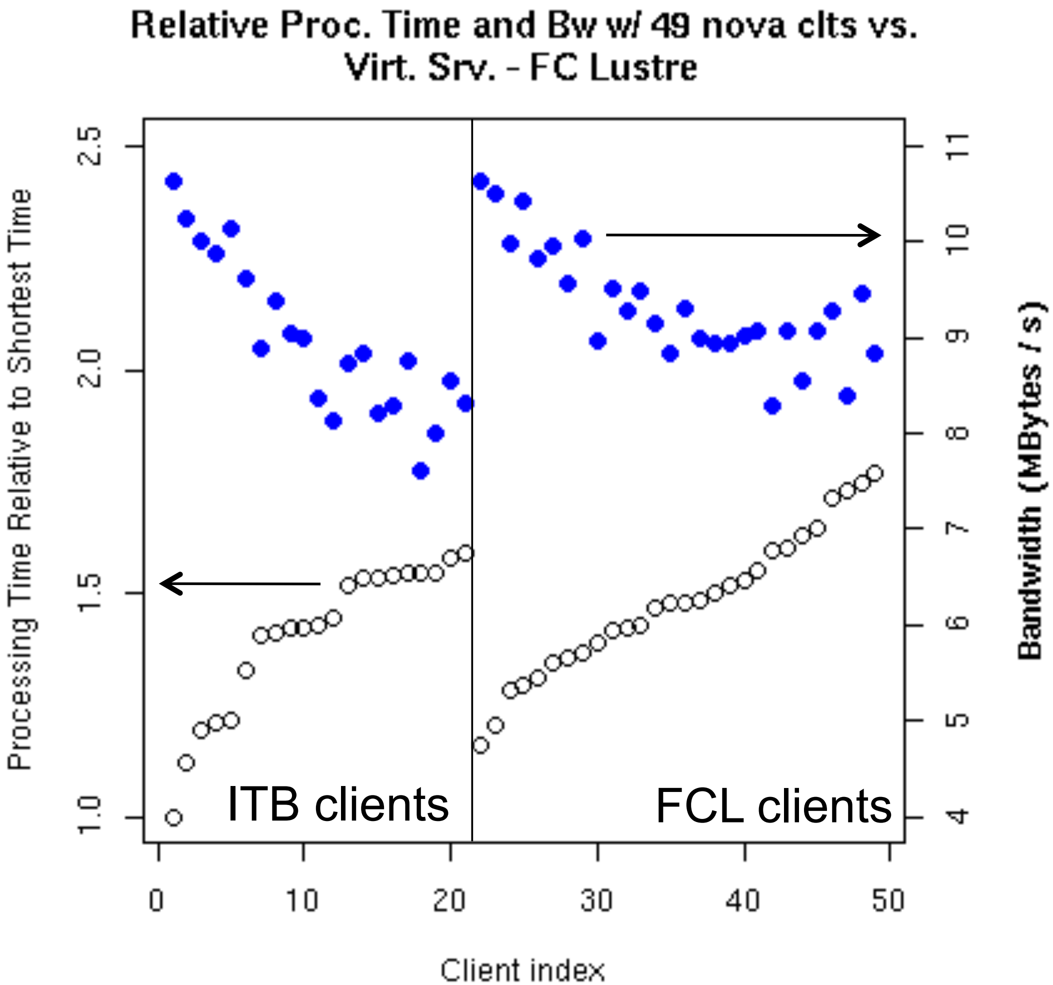
Read – FCL vs. virt. srv.
BW = 13.02 ± 0.05 MB/s
(1 FCL cl.: 14.4 ± 0.1 MB/s)

Virtual Server is almost as fast as bare metal for read

Virtual Clients on-board (on the same machine as the Virtual Server) are as fast as bare metal for read

49 Nova ITB / FCL clts vs. virt. srv.

49 clts (1 job / VM / core) saturate the bandwidth to the srv. Is the distribution of the bandwidth fair?



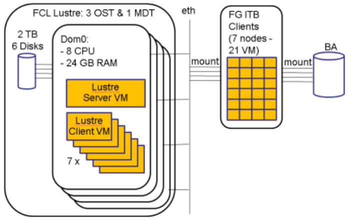
- Minimum processing time for 10 files (1.5 GB each) = 1268 s
- Client processing time ranges up to **177**% of min. time

Clients do NOT all get the same share of the bandwidth (within 20%).

- ITB clts:
  - •Ave time = 141 ± 4 %
  - •Ave bw = 9.0 ± 0.2 MB/s
- FCL clts:
  - •Ave time = 148 ± 3 %
  - •Ave bw = 9.3 ± 0.1 MB/s

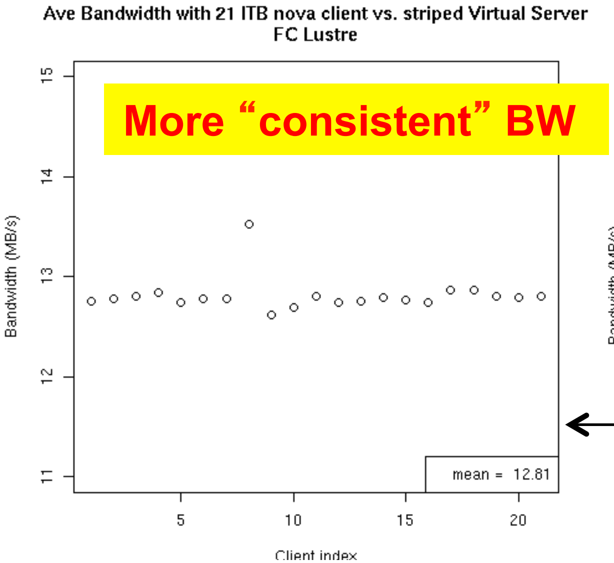
No difference in bandwidth between ITB and FCL clts.

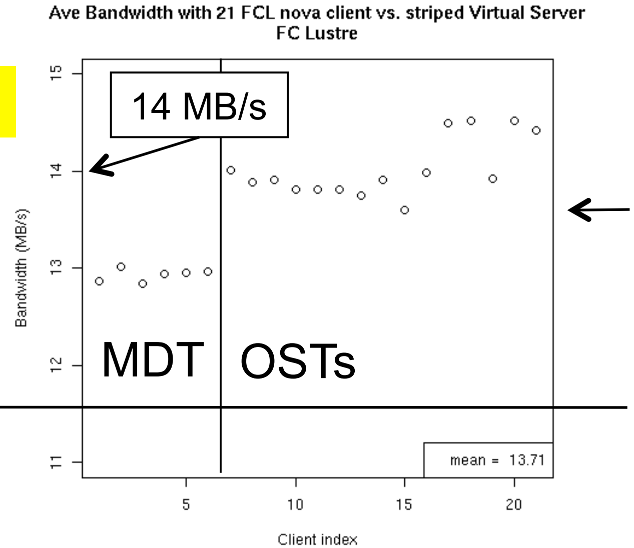
21 Nova ITB / FCL clt vs. striped virt. srv.



What effect does striping have on bandwidth?

More "consistent" BW



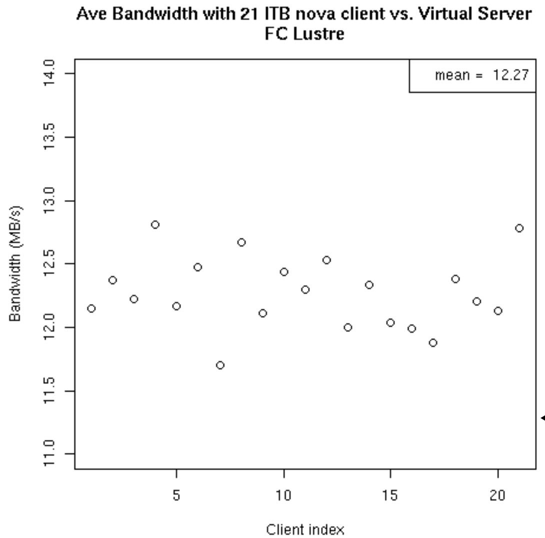


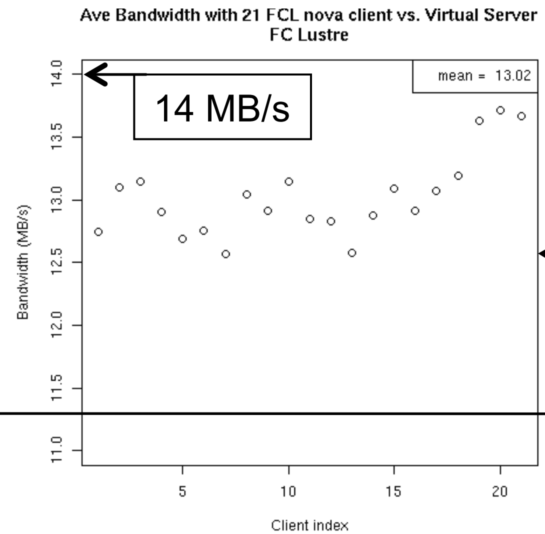
STRIPED

4MB stripes on 3 OST

Read – FCL vs. virt. srv.
BW = 13.71 ± 0.03 MB/s

Read – ITB vs. virt. srv.
BW = 12.81 ± 0.01 MB/s





Slightly better BW on OSTs

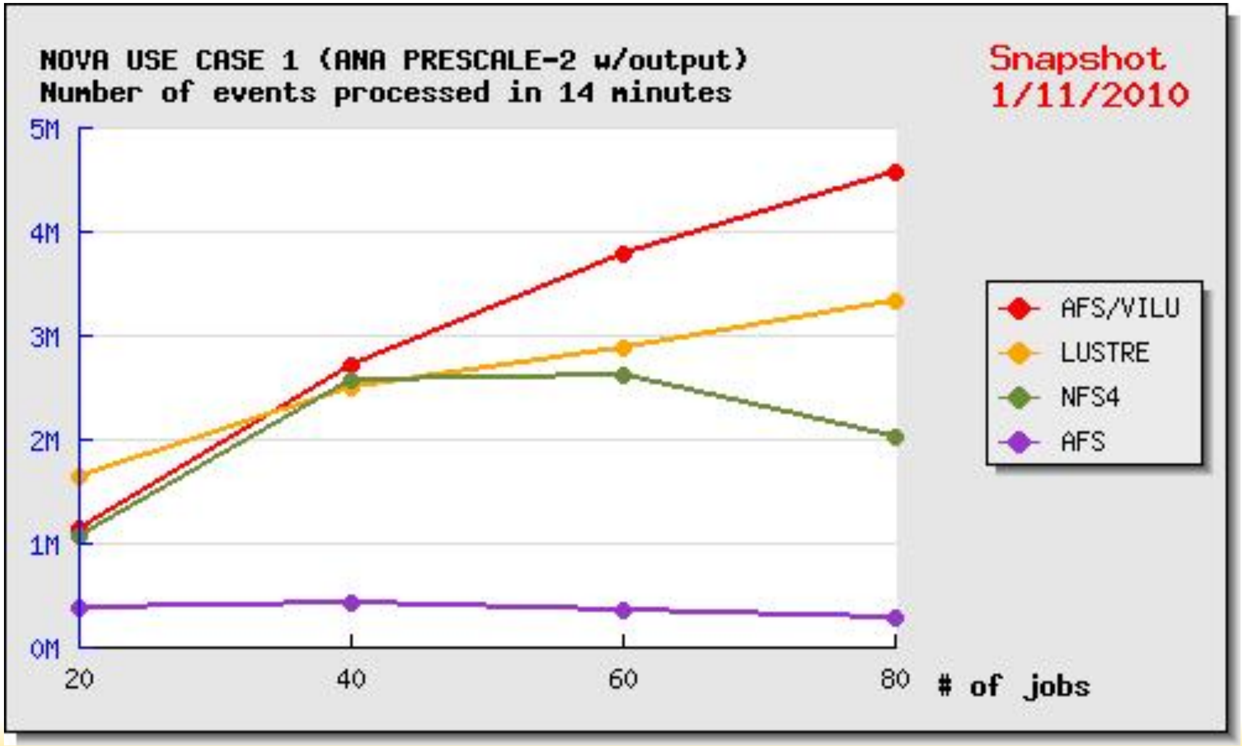
NON STRIPED

Read – FCL vs. virt. srv.
BW = 13.02 ± 0.05 MB/s

Read – ITB vs. virt. srv.
BW = 12.27 ± 0.08 MB/s

HEPiX Storage Group

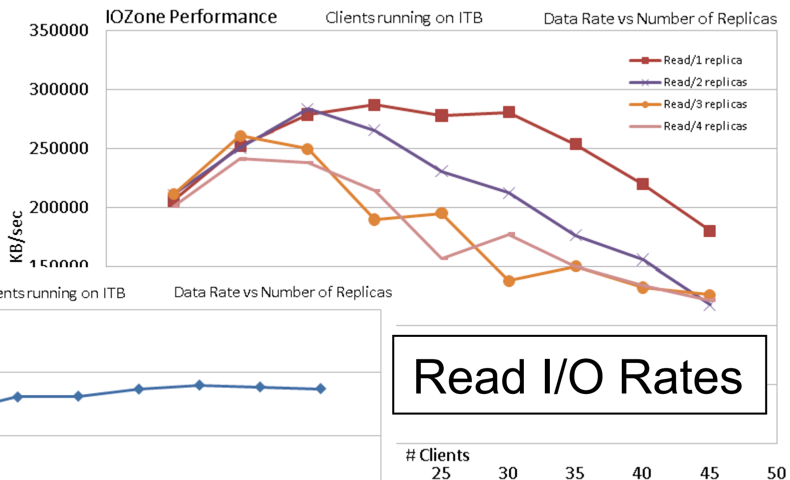
- Collaboration with Andrei Maslennikov
- Nova offline skim app. used to characterize storage solutions
- Lustre with AFS front-end for caching has best performance (AFS/VILU).

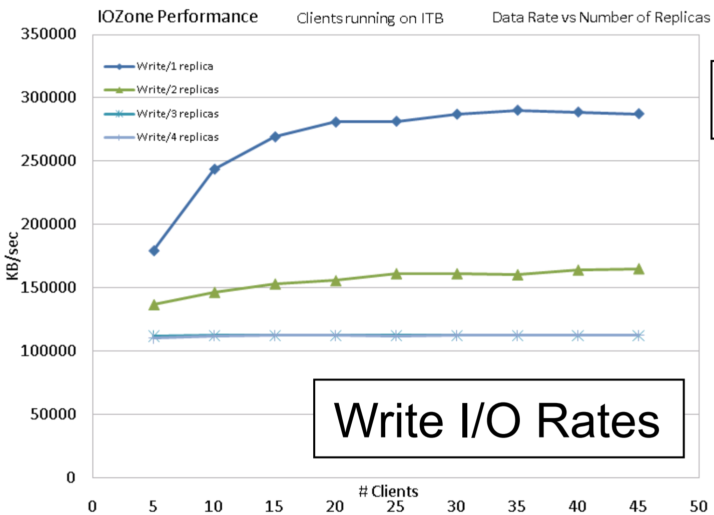


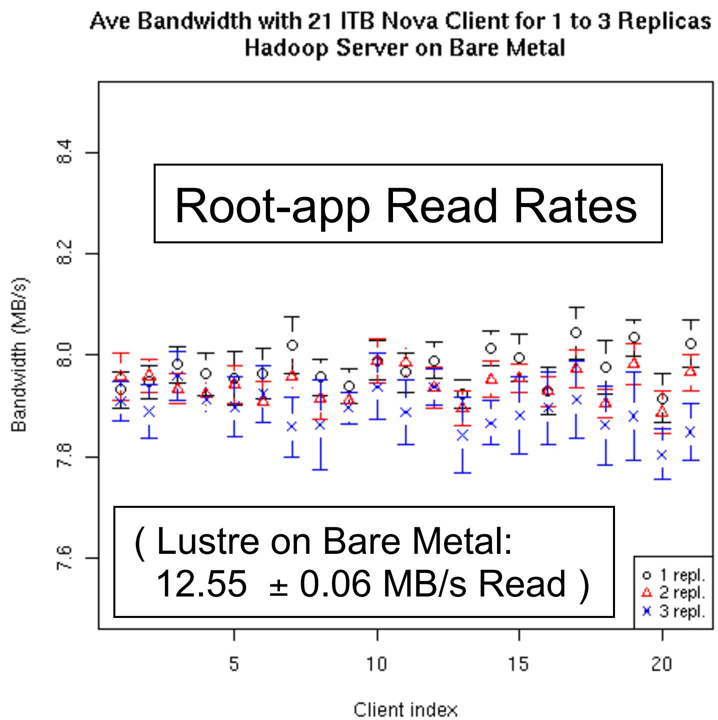
Current Work: Hadoop Eval.

- Hadoop: 1 meta-data + 3 storage servers.
  Testing access rates with different replica numbers.
- Clients access data via Fuse. Only semi-POSIX: root app.: cannot write; untar: returned before data is available; chown: not all features supported; …

Lustre on Bare Metal:
350 MB/s read
250 MB/s write







Conclusions

- **Performance**
  - Lustre Virtual Server writes 3 times slower than bare metal. Use of virtio drivers is necessary but not sufficient.
  - The HEP applications tested do NOT have high demands for write bandwidth. Virtual server may be valuable for them.
  - Using VM clts on the Lustre VM server has the same performance as "external" clients (within 15%)
  - Data striping has minimal (5%) impact on read bandwidth. None on write.
  - Fairness of bandwidth distribution is within 20%.
  - More data will be coming through HEPiX Storage tests.
- **Fault tolerance (results not presented)**
  - Fail-out mode did NOT work
  - Fail-over tests show graceful degradation
- **General Operations**
  - Managed to destroy data with a change of fault tolerance configuration. Could NOT recover from MDT vs. OST de-synch.
  - Some errors are easy to understand, some very hard.
  - The configuration is coded on the Lustre partition. Need special commands to access it. Difficult to diagnose and debug.

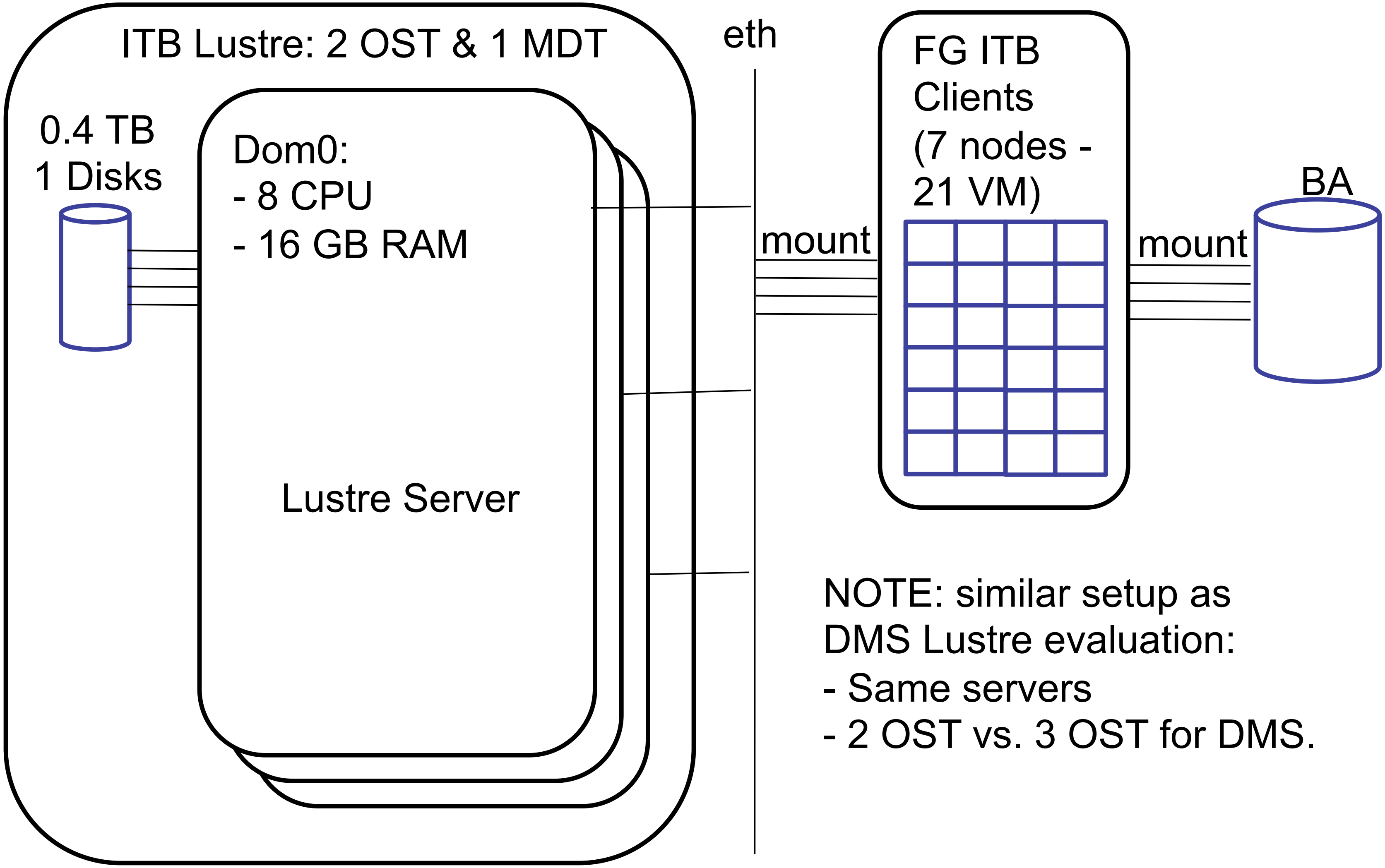
EXTRA SLIDES

Storage evaluation metrics

Metrics from Stu, Gabriele, and DMS (Lustre evaluation)

- Cost
- Data volume
- Data volatility (permanent, semi-permanent, temporary)
- Access modes (local, remote)
- Access patterns (random, sequential, batch, interactive, short, long, CPU intensive, I/O intensive)
- **Number of simultaneous client processes**
- **Acceptable latencies requirements (e.g for batch vs. interactive)**
- **Required per-process I/O rates**
- **Required aggregate I/O rates**
- **File size requirements**
- Reliability / redundancy / data integrity
- Need for tape storage, either hierarchical or backup
- Authentication (e.g. Kerberos, X509, UID/GID, AFS_token) / Authorization (e.g. Unix perm., ACLs)
- User & group quotas / allocation / auditing
- Namespace performance ("file system as catalog")
- Supported platforms and systems
- Usability: maintenance, troubleshooting, problem isolation
- Data storage functionality and scalability

Lustre Test Bed: ITB “Bare Metal”



NOTE: similar setup as DMS Lustre evaluation:
- Same servers
- 2 OST vs. 3 OST for DMS.

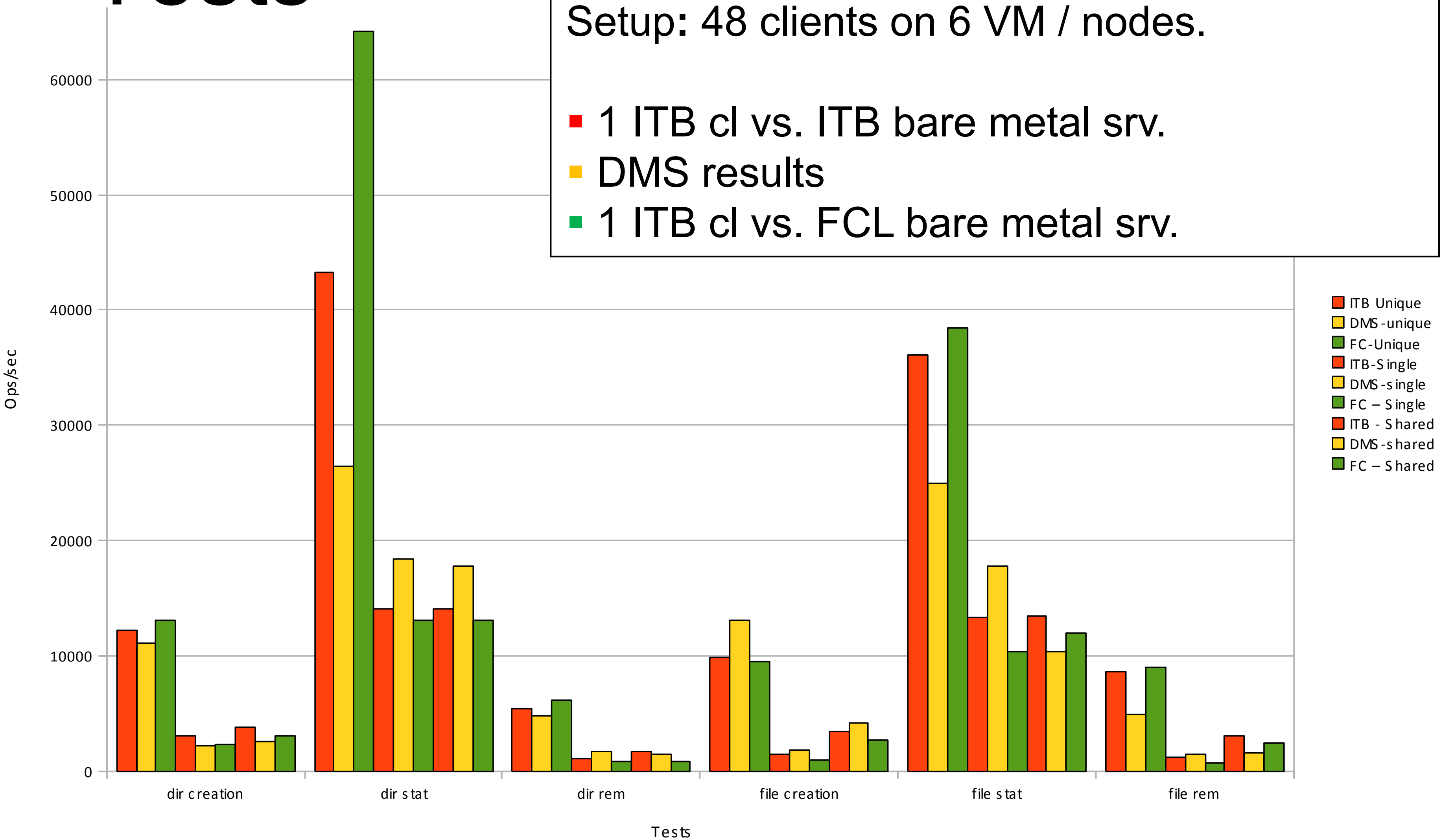
Machine Specifications

- FCL Client / Server Machines:
  - Lustre 1.8.3: set up with 3 OSS (different striping)
  - CPU: dual, quad core Xeon E5640 @ 2.67GHz with 12 MB cache, 24 GB RAM
  - Disk: 6 SATA disks in RAID 5 for 2 TB + 2 sys disks (hdparm → 376.94 MB/sec)
  - 1 GB Eth + IB cards
- ITB Client / Server Machines:
  - Lustre 1.8.3 : Striped across 2 OSS, 1 MB block
  - CPU: dual, quad core Xeon X5355 @ 2.66GHz with 4 MB cache: 16 GB RAM
  - Disk: single 500 GB disk (hdparm → 76.42 MB/sec)

Metadata Tests

Mdtest: Tests metadata rates from multiple clients. File/Directory Creation, Stat, Deletion.
Setup: 48 clients on 6 VM / nodes.

- 1 ITB cl vs. ITB bare metal srv.
- DMS results
- 1 ITB cl vs. FCL bare metal srv.



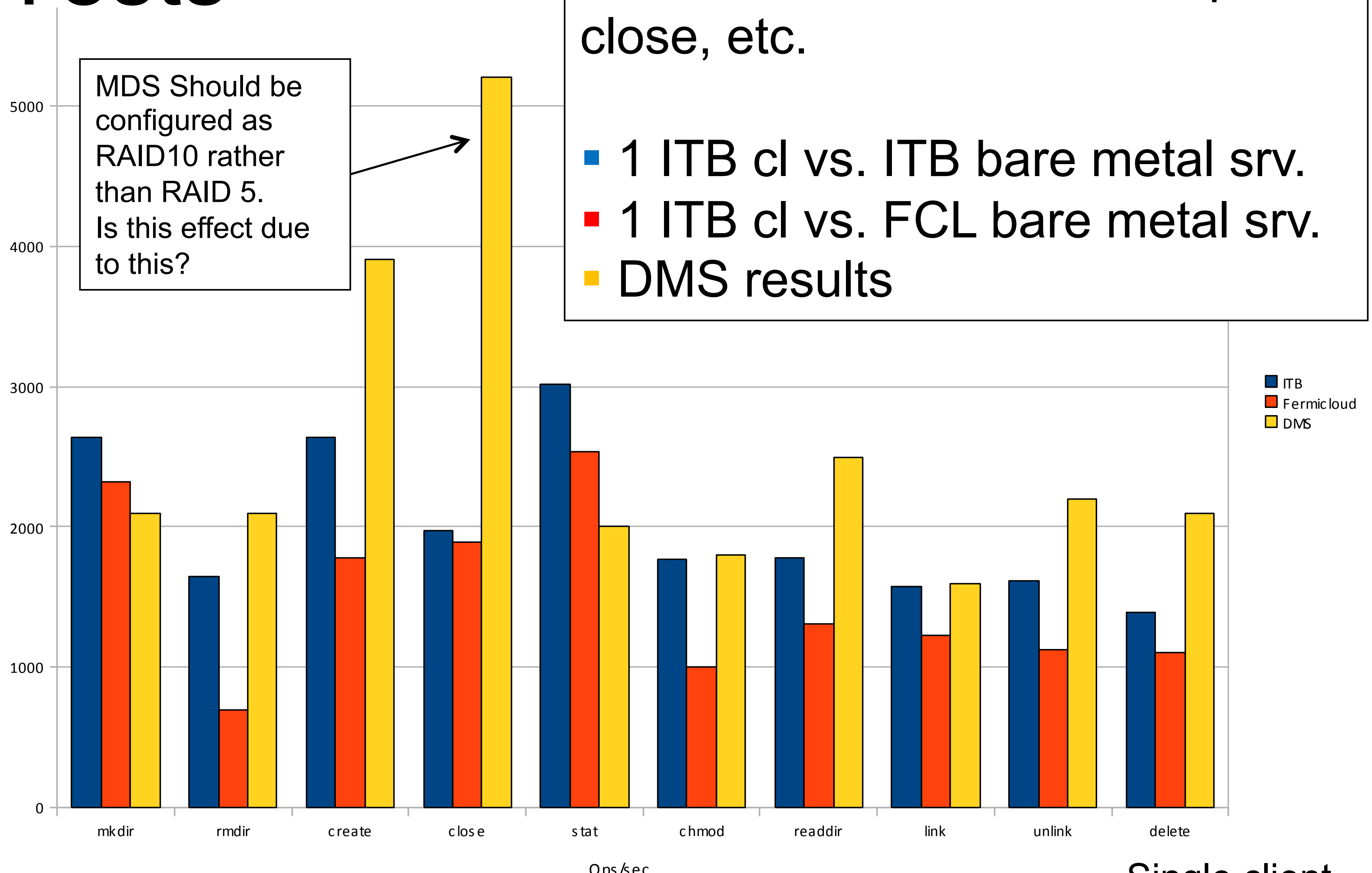
48 clients on 6 VM on 6 different nodes

Metadata Tests

Fileop: Iozone's metadata tests. Tests rates of mkdir, chdir, open, close, etc.

- 1 ITB cl vs. ITB bare metal srv.
- 1 ITB cl vs. FCL bare metal srv.
- DMS results

MDS Should be configured as RAID10 rather than RAID 5.
Is this effect due to this?



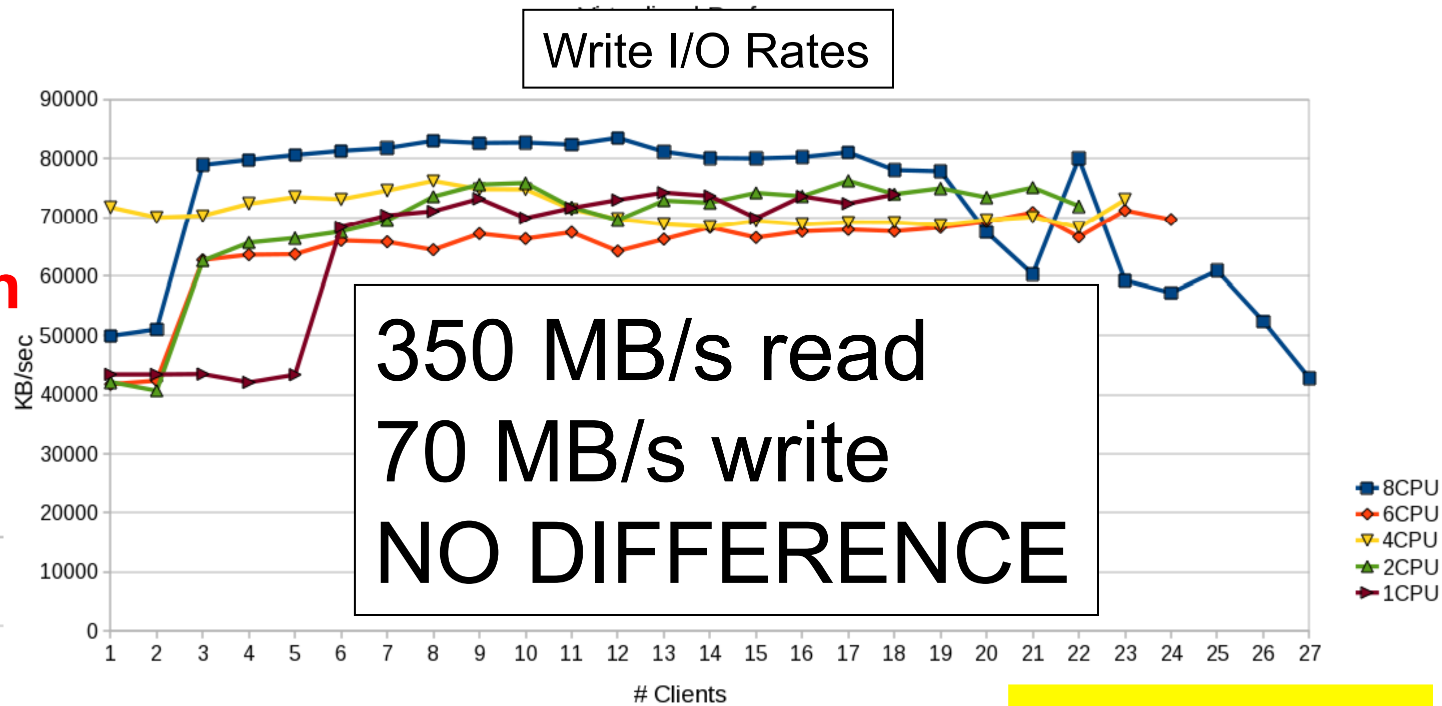
Single client

Status and future work

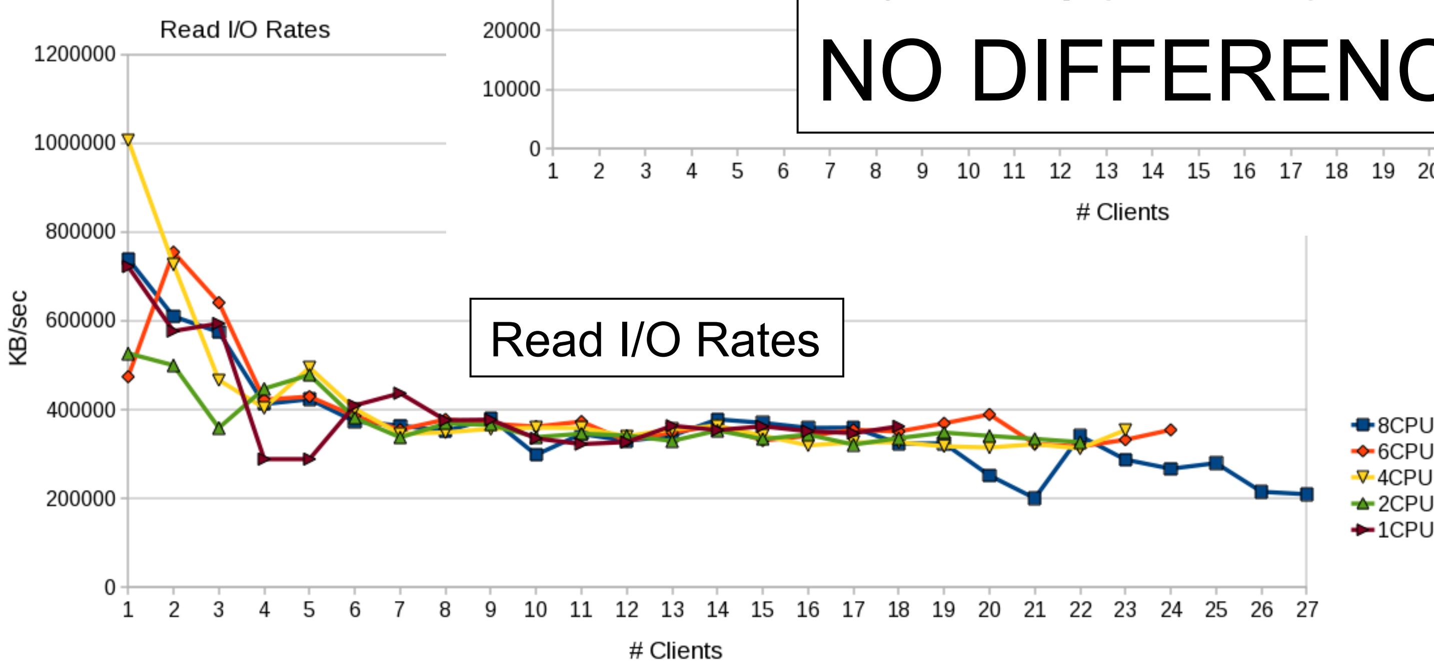
- Storage evaluation project status
  - Initial study of data access model: DONE
  - Deploy test bed infrastructure: DONE
  - Benchmarks commissioning: DONE
  - Lustre evaluation: DONE
  - Hadoop evaluation: STARTED
  - Orange FS and Blue Arc evaluations TODO
  - Prepare final report: STARTED
- Current completion estimate is May 2011

ITB clts vs. FCL Virt. Srv. Lustre

Trying to improve write IO changing num of CPU on the Lustre Srv VM…

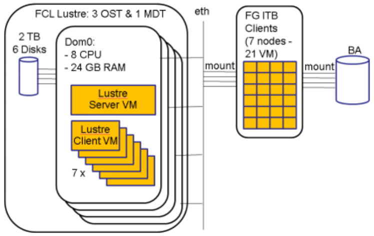


350 MB/s read
70 MB/s write
NO DIFFERENCE

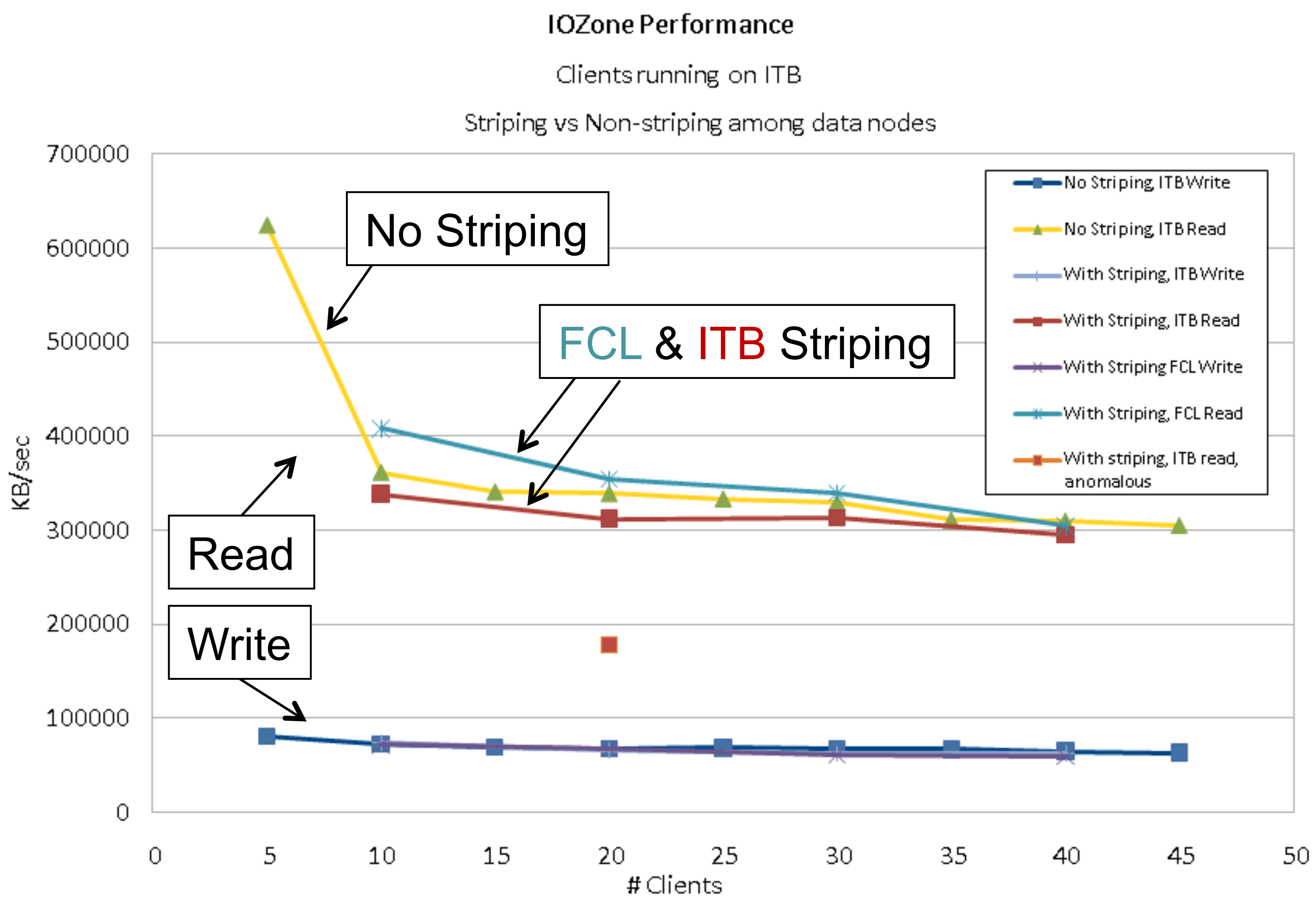


Write IO does NOT depend on num. CPU. 1 or 8 CPU (3 GB RAM) are equivalent for this scale

ITB & FCL clts vs. Striped Virt. Srv.



What effect does striping have on bandwidth?



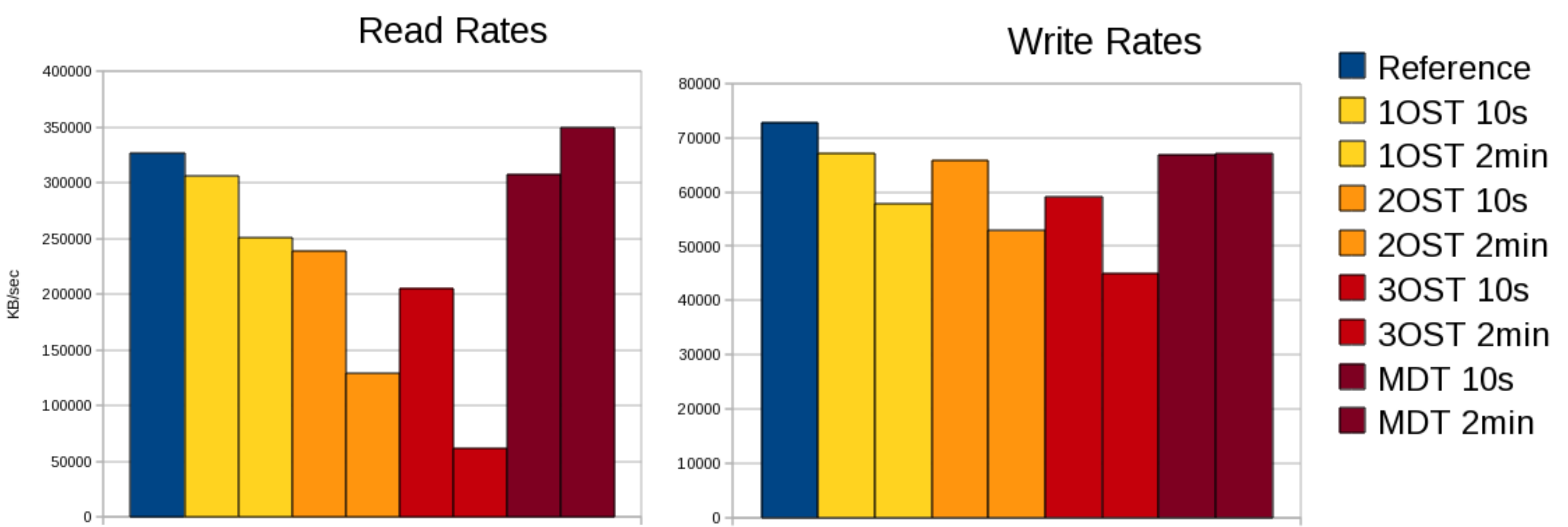
Writes are the same

Reads w/ striping:
- FCL clts 5% faster
-ITB clts 5% slower

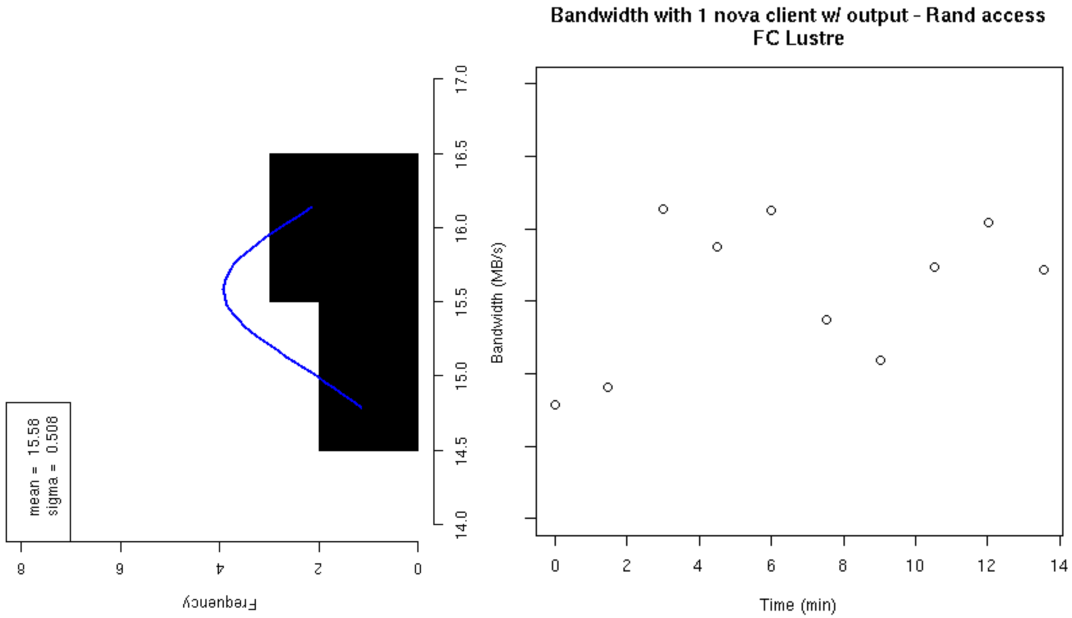
Not significant

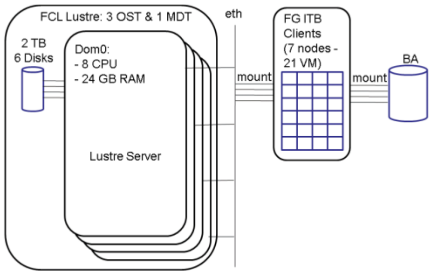
Fault Tolerance

- Basic fault tolerance tests of ITB clients vs. FCL lustre virtual server
- Read / Write rates during iozone tests when turning off 1,2,3 OST or MDT for 10 sec or 2 min.
- 2 modes: Fail-over vs. Fail-out. **Fail-out did not work**.
- **Graceful degradation**:
  - If OST down → access is suspended
  - If MDT down → ongoing access is NOT affected

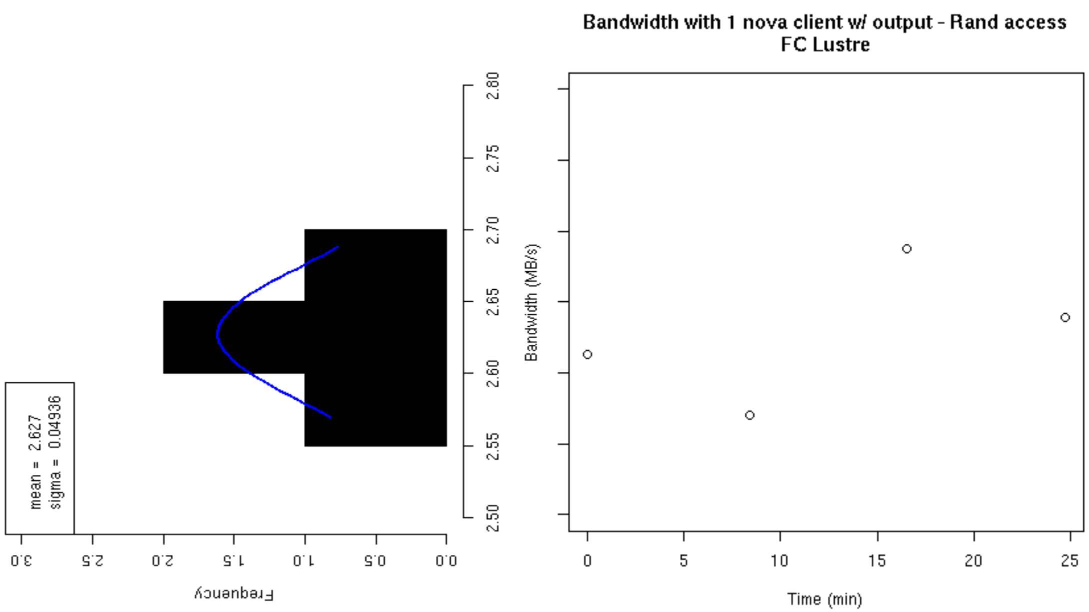


1 Nova ITB clt vs. bare metal





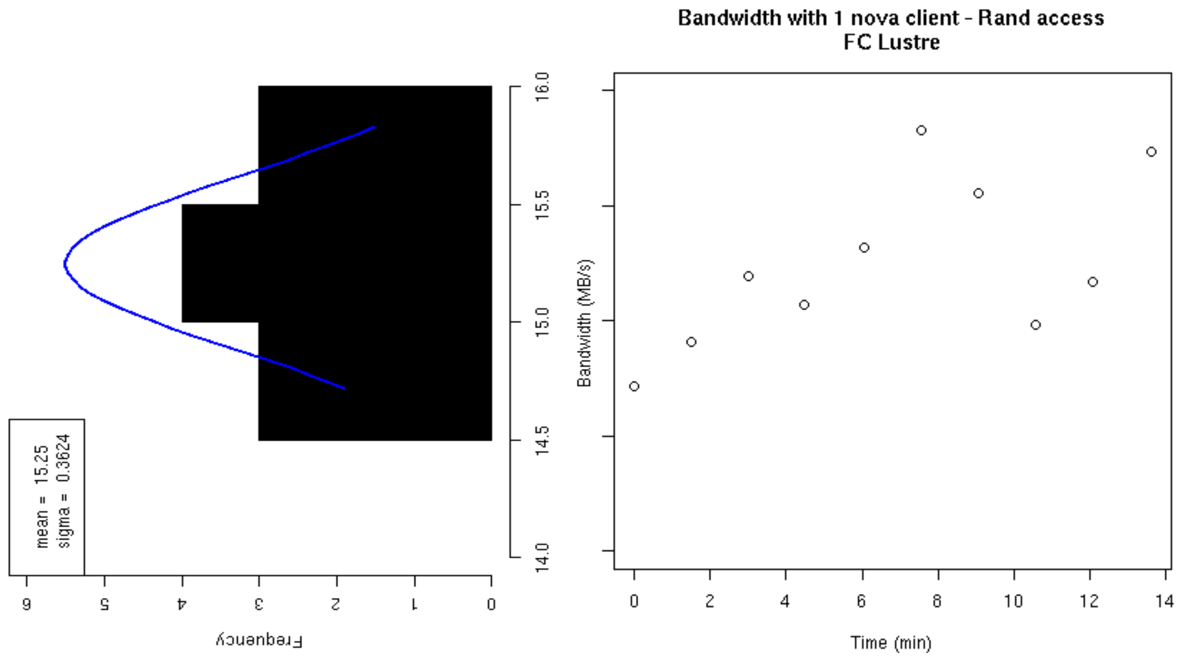
Read
BW = 15.6 ± 0.2 MB/s

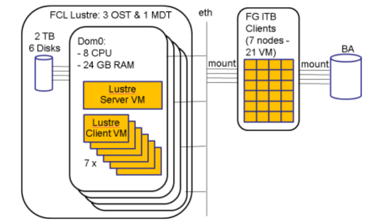


Read & Write
BW read = 2.63 ± 0.02 MB/s
BW write = 3.25 ± 0.02 MB/s

Write is always CPU bound – It does NOT stress storage

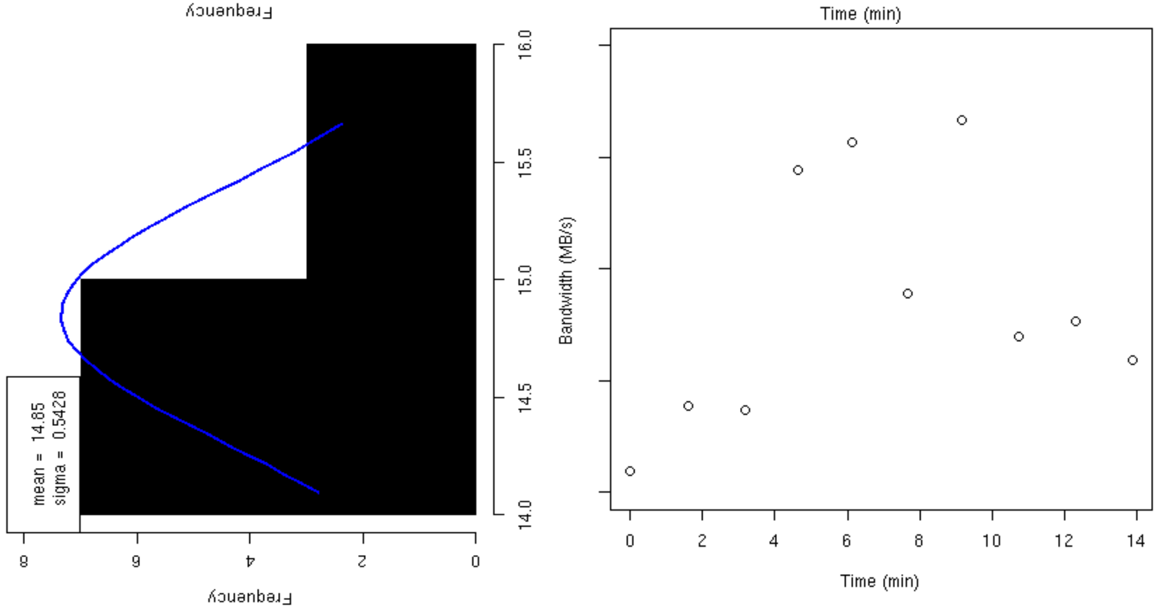
1 Nova ITB / FCL clt vs. virt. srv.





1 ITB clt – Read
BW = 15.3 ± 0.1 MB/s
(Bare m: 15.6 ± 0.2 MB/s)

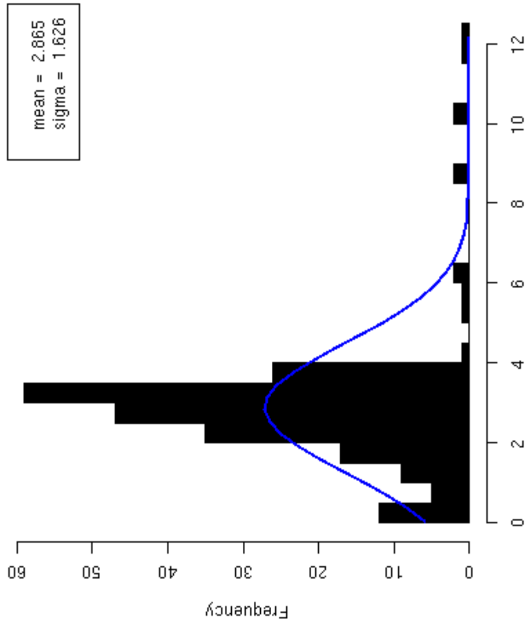
Virtual Server is as fast as bare metal for read

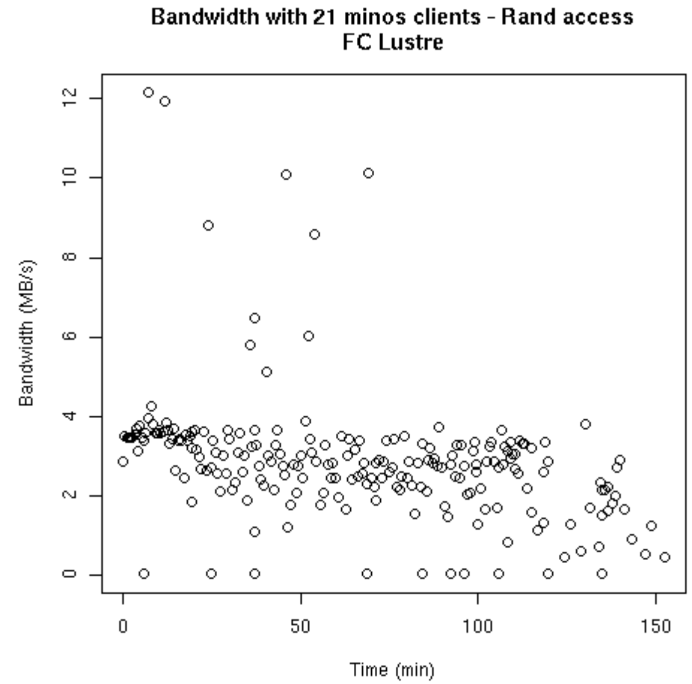


1 FCL clt – Read
BW = 14.9 ± 0.2 MB/s
(Bare m: 15.6 ± 0.2 MB/s)
w/ default disk and net drivers:
BW = 14.4 ± 0.1 MB/s

On-board client is almost as fast as remote client

Minos





- 21 Clients
- Minos application (loon) skimming
- Random access to 1400 files

Loon is CPU bound – It does NOT stress storage