InfiniBand Performance Metrics/Testing

Susan Coulter
Los Alamos National Laboratory

High Performance Computing Division
HPC-3 Production Systems
skc@lanl.gov

April 19, 2013

Testing Fundamental Fabric Performance

- **ib_read_bw, ib_write_bw**
  - Speed * encoding = theoretical maximum data bandwidth
    - QDR ➔ 40Gb/s * 80% = 32Gb/s
    - FDR ➔ 56Gb/s * 97% = 54.5Gb/s
- **ib_read_lat, ib_write_lat**
- **mpi synthetics**
- **Previously: manual process run as time permits on cluster standup**
- **Now: automated process run regularly**

- **3 problems uncovered at LANL with these processes**

First automation: Lustre LNet to OSS Aggregate Testing

… from the wiki

Client script for cielo LNet aggregate

The iterations are increased to 100,000 to insure significant overlap of clients.

```
for x in `seq 1 52`; do ib_read_bw -p 18515 -n 100000 10.149.13.${x} & done | grep 65536 >> /tmp/agg
```

Check MDSes

```
To verify IB connection on MDSes:

pdsh -R ssh -w ci-mds1,ci-mds2,ci-mds3,ci-mds4 ibstat
```

Find the LNet nodes

```
To get the list of current LNet systems, on boot node:

xtopview -e "cat /etc/sysconfig/ethcfg" | grep lnet > /tmp/lnet_nids

cat /tmp/lnet_nids | xargs | sed 's/ /,/g' > /tmp/lnet_arg  (may need to whack extra stuff from /tmp/lnet_nids)
```

Lustre – LNet to OSS Aggregate Testing (continued)

Server

```
for x in `seq 18515 18562`; do pexec -t0 -P52 -pm `cat /tmp/lnet_arg` --ping --ssh /usr/bin/perftest-1.2.3/ib_read_bw -p ${x}; done
```

Client

```
# copy to all OSSs
for x in `seq 1 48`; do scp client ci-oss${x}:/tmp/foo; done

# modify port used on each client
for x in `seq 1 48`; do port=18514; port=`expr $port + ${x}`; ssh ci-oss${x} "cat /tmp/foo | sed 's/18515/$port/' > /tmp/client";
done

# make it executable
for x in `seq 1 48`; do ssh ci-oss${x} chmod +x /tmp/client; done

# launch
for x in `seq 1 48`; do ssh ci-oss${x} /tmp/client; done
```

Grab the data

```
for x in `seq 1 48`; do echo ci-oss${x} to LNets >> /root/markus/agg; scp ci-oss${x}:/tmp/agg /root/markus/skc; cat /root/markus/skc
>> /root/markus/agg.fta; done
```

Clear data

```
for x in `seq 1 48`; do ssh ci-oss${x} rm /tmp/agg; done
```

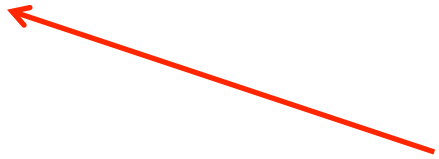
Typhoon vs Luna Case study

Typhoon (older)

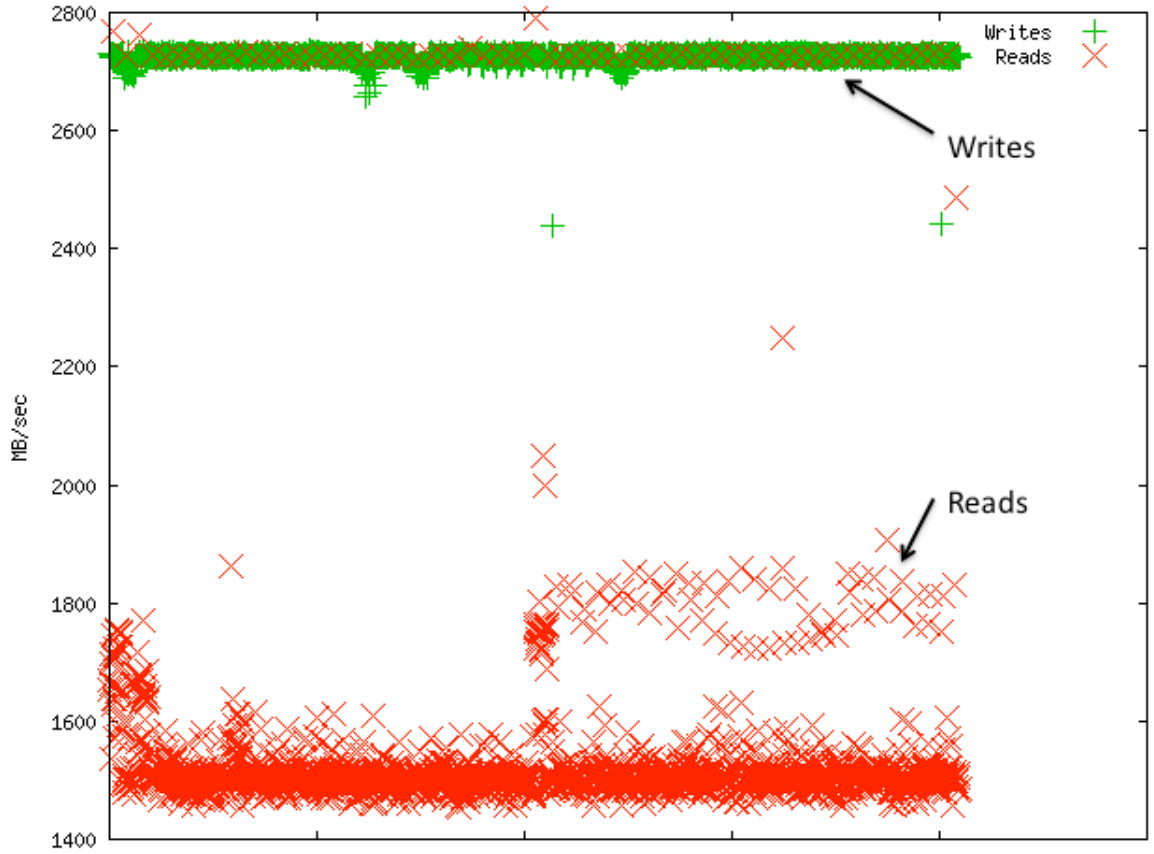
- Appro cluster in classified partition
- 416 compute nodes / 32 AMD 2GHz processors each
- 2 Voltaire QDR 4700 Grid Director chassis
- FatTree routing

Luna (newer)

- Appro cluster in classified partition
- 1636 compute nodes / 16 Intel XEON 2.6GHz processors each
- 3 QLogic/Intel QDR chassis and 90 36port edge switches
- FatTree routing
- User codes reporting 5x speed up !!!



Typhoon verbs performance – what ?!?



Comparison of ib_read_bw and ib_write _bw performance

Not the fabric … check the nodes (dmidecode)

tya001: System Information

tya001: Manufacturer: APPRO

tya001: Product Name: 1343H-LANL-CN

tya001: Family: 1234567890

tya414: System Information

tya414: Manufacturer: Supermicro

tya414: Product Name: H8QG6

tya414: Family: 1234567890

tya413: System Information

tya413: Manufacturer: APPRO

tya413: Product Name: APPRO-1343H

tya413: Family: 1234567890

Base Board Information

Manufacturer: Supermicro

Product Name: H8QG6

tya182: System Information

tya182: Manufacturer: APPRO

tya182: Product Name: 1343H-LANL-CN

tya182: Family: Server

tya002: System Information

tya002: Manufacturer: Supermicro

tya002: Product Name: H8QG6

tya002: Family: Server

Motherboard or BIOS ?

BIOS Date: 07/01/2008 Ver: 6.24.00.00

BIOS Date: 04/11/2012 Ver: 2.0b

BIOS Date: 09/02/10 Rev: 1.0b

BIOS Date: 09/08/10 11:37:43 Ver: 1.0b

BIOS Date: 10/11/10 16:32:43 Ver 1.0c

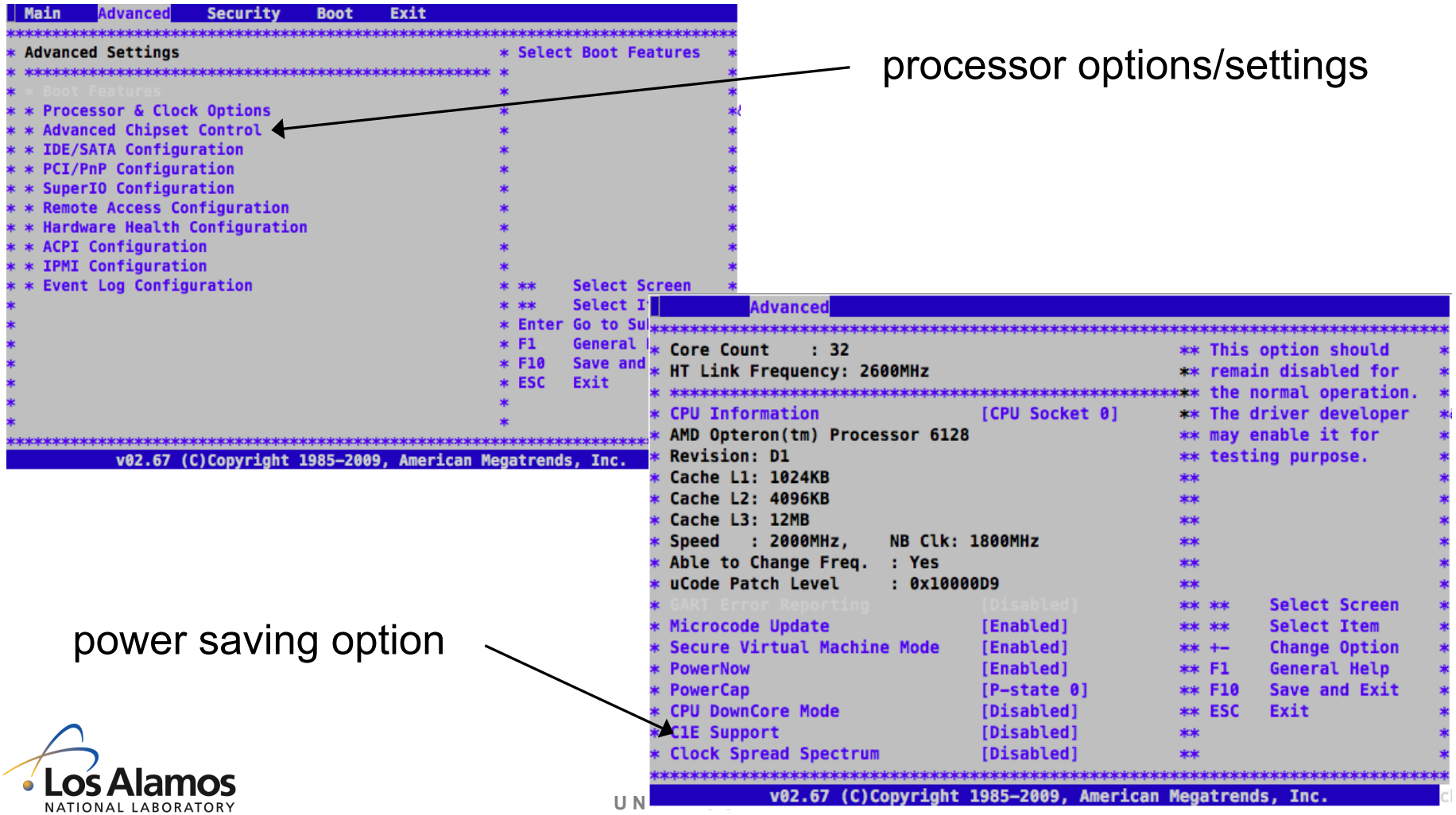
BIOS Date: 10/28/2011 Ver: 2.0a

BIOS Date: 11/04/10 10:59:38 Ver 1.10.t06

7 different bios versions !!

Only 1, resident on 6 nodes, resulted in full IB performance

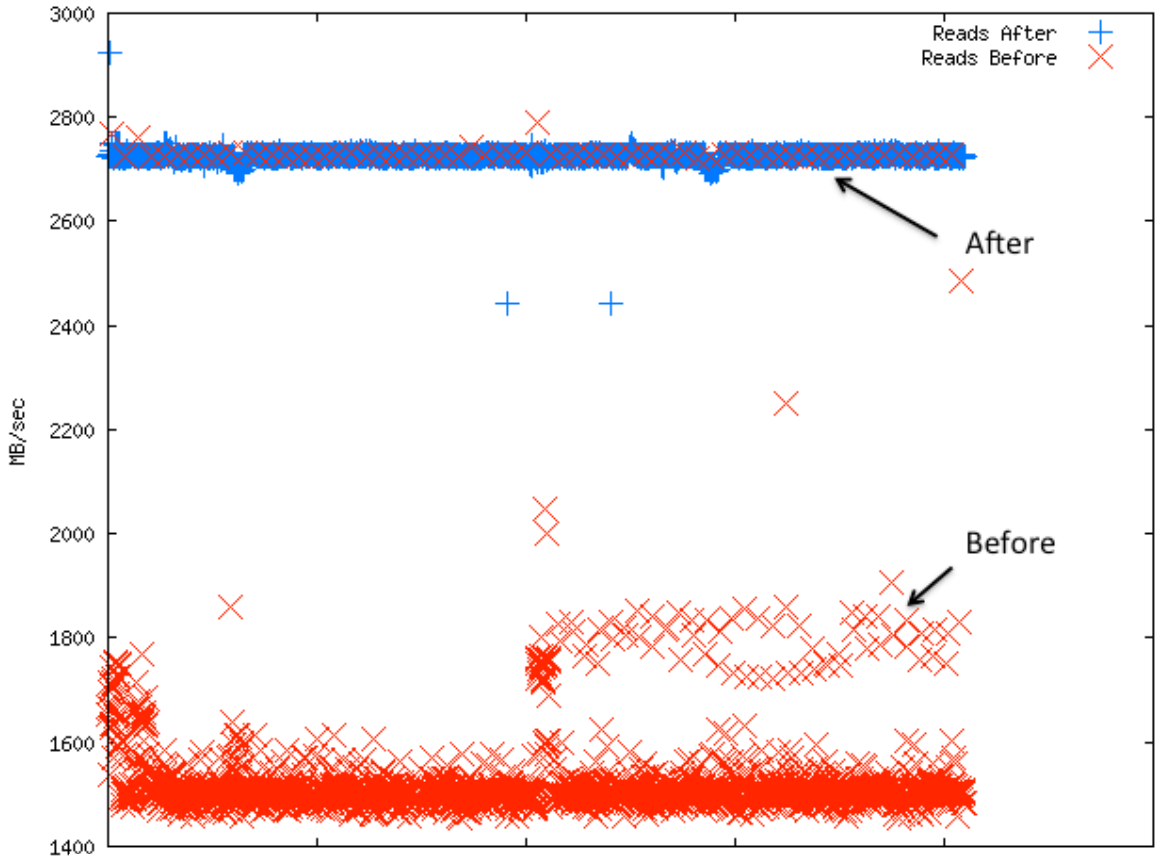
BIOS settings



processor options/settings

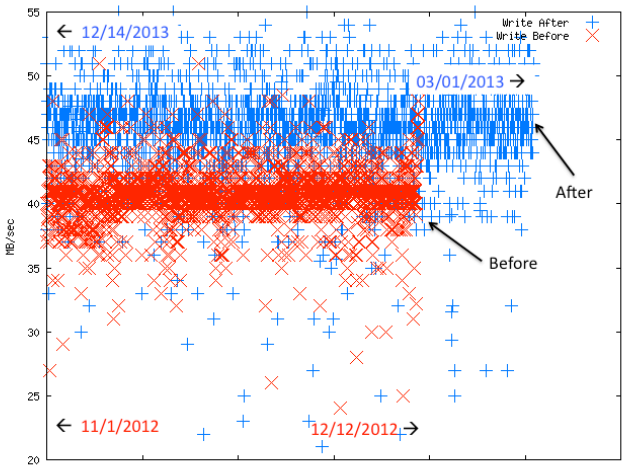
power saving option

After BIOS change

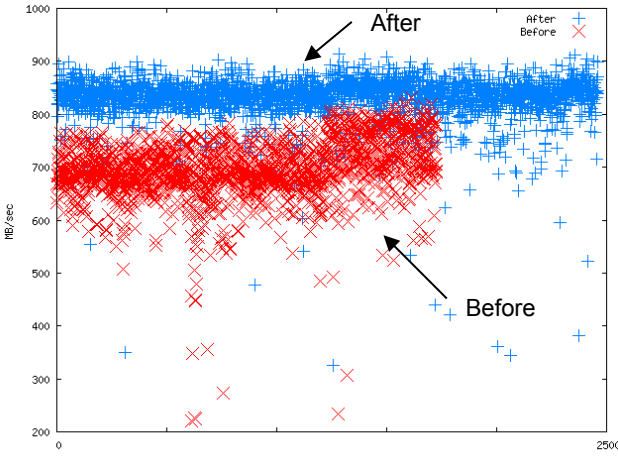


Comparison of ib_read_bw performance before and after

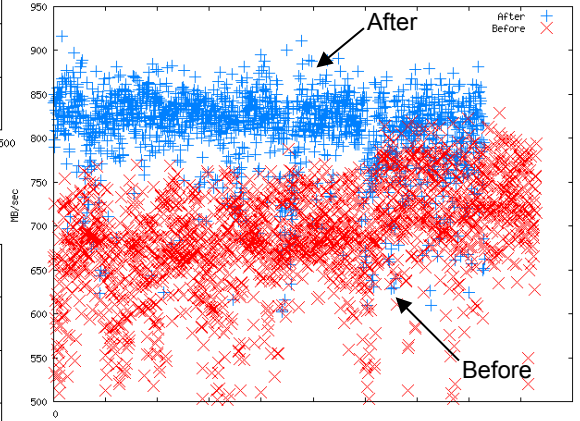
File System Write Performance



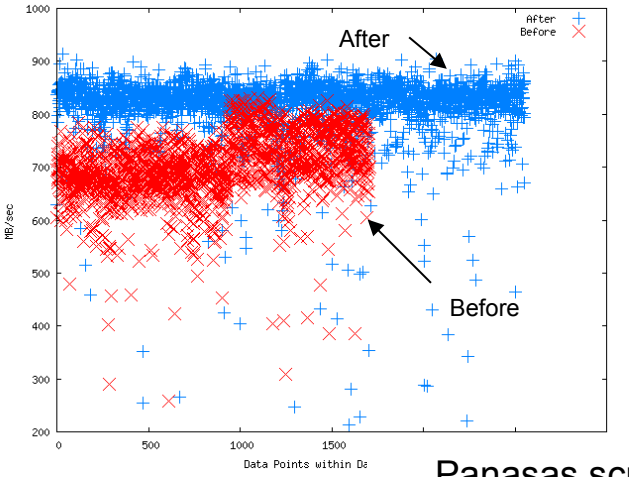
NFS



Panasas scratch9



Panasas scratch6

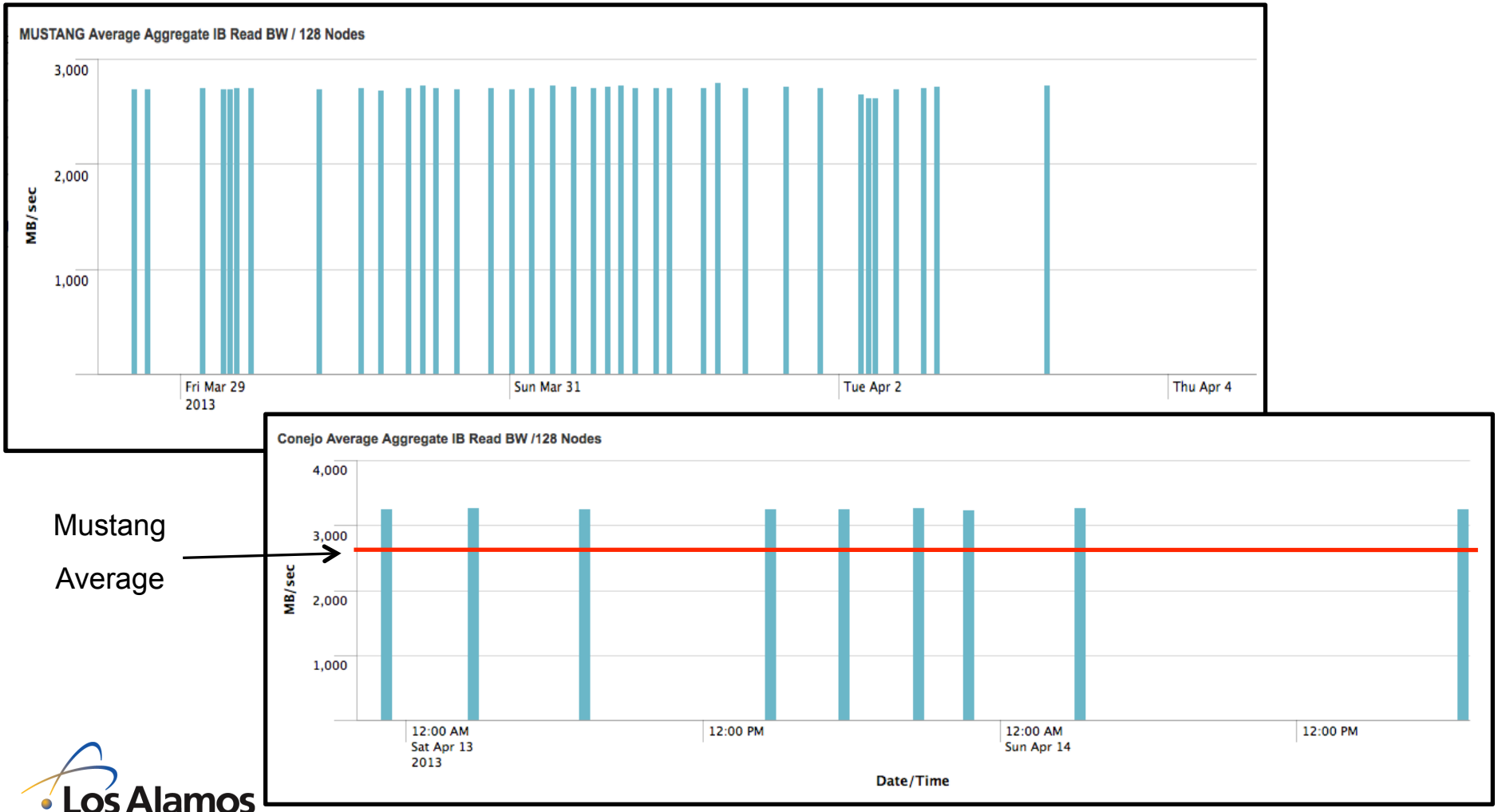


Panasas scratch8

Automate for baseline trending - Gazebo / Splunk

- gazebo: LANL-written test framework
  - allows setup of ongoing process to continually submit jobs
  - can control how much of the machine your tests cover
  - sends results directly to splunk
- splunk: Tool for handling/indexing/querying large amounts of data
  - allows for trending and graphing data
  - can create baselines and thresholds
  - can send notices given certain events or combination of events

Splunk easily shows … Mustang is slower than Conejo?!



Mustang Average

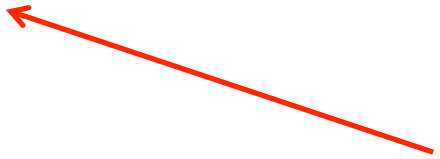
Mustang vs Mapejo Case study

Conejo/Mapache (older)

- SGI cluster in open partition
- 620 compute nodes / 8 Intel XEON 2.6GHz processors each
- 1 QDR Grid Director chassis
- FatTree routing

Mustang (newer)

- Appro cluster in open partition
- 1600 compute nodes / 24 AMD 2.3GHz processors each
- 3 QDR Grid Director chassis and 91 36 port edge switches
- FatTree routing
- Why is it so much slower ?



Mustang and Mapejo – IB configuration

QDR Mellanox – FatTree routing

Product name: EFM_PPC_M460EX

Product release: EFM_1.1.2500

Build ID: #1-dev

Build date: 2011-02-22 15:51:54

Target arch: ppc

Target hw: m460ex

Built by: alia@fit01

Conejo/Mapache:

CA 'mlx4_0'

CA type: MT26428

Number of ports: 1

Firmware version: 2.8.0

Hardware version: b0

board_id: MT_0D90110009

standalone HCAs

vs

on board HCAs

Mustang:

CA 'mlx4_0'

CA type: MT26428

Number of ports: 1

Firmware version: 2.9.1000

Hardware version: b0

board_id: SM_2121000001000

Mustang and Mapejo – Processors

Mustang:

vendor_id: AuthenticAMD
cpu family: 16
model name: AMD Opteron(tm) Processor 6176
stepping: 1
cpu MHz: 2300.082
cache size: 512 KB
siblings: 12
cpu cores: 12
bogomips: 4600.04
TLB size: 1024 4K pages
clflush size: 64
cache_alignment: 64
address sizes: 48 bits physical, 48 bits virtual
power management: ts ttp tm stc 100mhzsteps hwpstate

Conejo/Mapache:

vendor_id: GenuineIntel
cpu family: 6
model name: Intel®Xeon® CPUX5550@2.67GHz
stepping: 5
cpu MHz: 2668.000
cache size: 8192 KB
siblings: 4
cpu cores: 4
bogomips: 5333.51

clflush size: 64
cache_alignment: 64
address sizes: 40 bits physical, 48 bits virtual
power management: [8]

Mustang and Mapejo – PCI settings

Conejo/Mapache:

01:00.0 InfiniBand: Mellanox Technologies MT26428 [ConnectX VPI PCIe 2.0 5GT/s - IB QDR / 10GigE] (rev b0)

Subsystem: Mellanox Technologies Device 0022

Control: I/O- Mem+ BusMaster+ SpecCycle-MemWINV- VGASnoop- ParErr+ Stepping- SERR+ FastB2B- DisINTx+

Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0, Cache Line Size: 256

Mustang:

02:00.0 InfiniBand: Mellanox Technologies MT26428 [ConnectX VPI PCIe 2.0 5GT/s - IB QDR / 10GigE] (rev b0)

Subsystem: Super Micro Computer Inc Device 673c

Control: I/O- Mem+ BusMaster+ SpecCycle- MemWINV- VGASnoop- ParErr- Stepping- SERR+ FastB2B- DisINTx+

Status: Cap+ 66MHz- UDF- FastB2B- ParErr- DEVSEL=fast >TAbort- <TAbort- <MAbort- >SERR- <PERR- INTx-

Latency: 0, Cache Line Size: 64 bytes

Conclusion

- Performance metrics do more than validate the fabric !!
- Great for baseline and trending
- Useful for validating (or not) user reports of changes

End

Questions?