Flash Based Lustre Filesystem for the ARM-based Supercomputer Astra

Lee Ward (lee@sandia.gov)

Sandia National Laboratories

Michael Timmerman (michael.timmerman@hpe.com)

Hewlett-Packard Enterprise

Ruth Klundt (rklundt@sandia.gov)

Sandia National Laboratories

Vanguard Program

Sandia's advanced machines/architecture test program

Purpose

- Acquire, test, and evaluate machines and, usually, small platforms
- To evaluate how well alternative CPU/Architecture/Memory/Storage/Accelerators support our applications
- To pioneer small and/or emerging, potential platform suppliers

After experience with multiple, small, ARM-based machines and platforms

- Cavium processor and roadmap could be a viable alternative for NNSA production work at scale
- Pitched a "small" but reasonable proxy platform acquisition
  - Of sufficient size, composition, and capability to provide evidence to justify (or not) a full-scale purchase

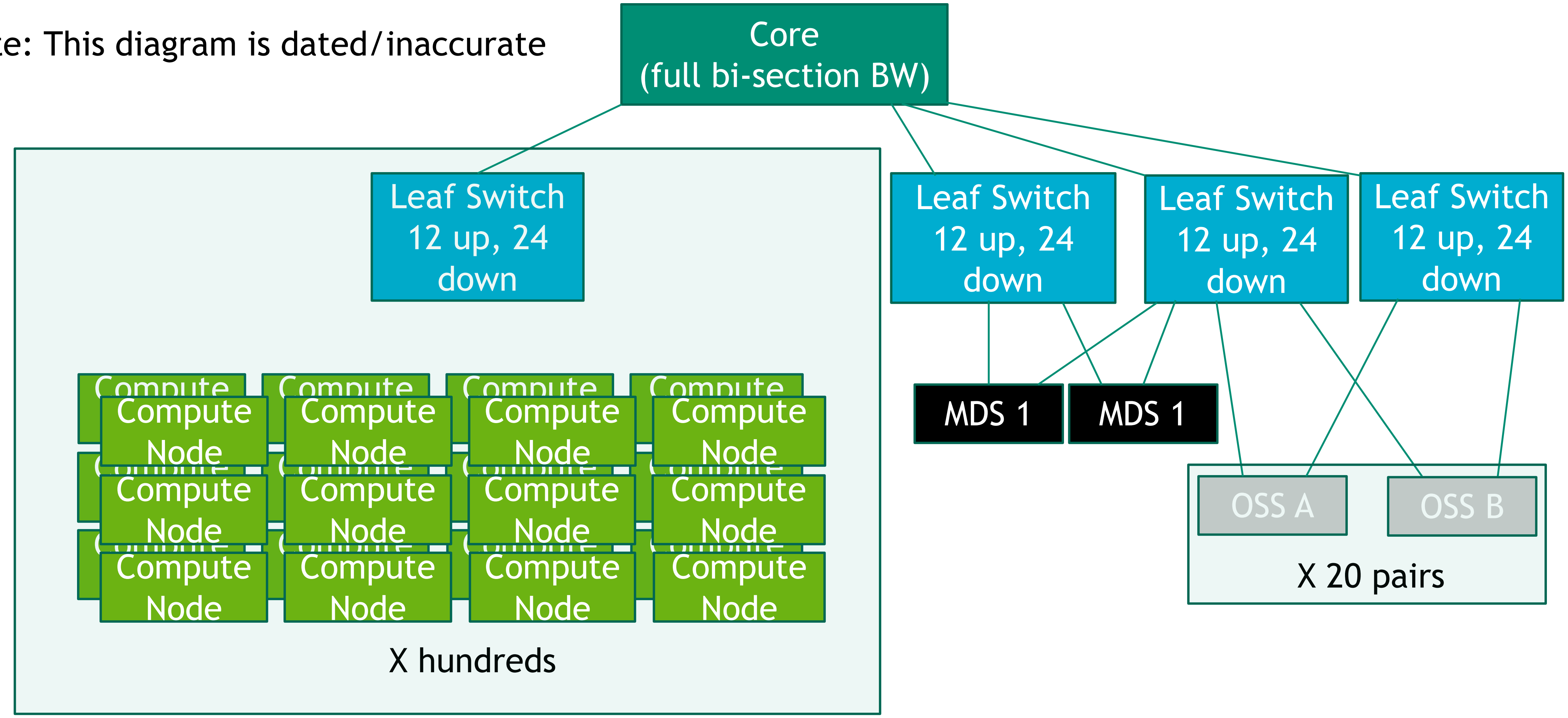
NNSA agrees!

Result, Astra

- It's alive!
- Now what?
- Guess we make it work; That'll teach us. ☺

Network

Note: This diagram is dated/inaccurate



Compute Section Composition

Three Ply, Fat-Tree Network

- Compute section is 2:1 oversubscribed

72 racks, 36 nodes per rack

- Or, 2592 nodes

Node

- Two NUMA socket, Cavium ThunderX2, 28 cores/socket
- 128 GB memory
- Infiniband adapter is a Mellanox ConnectX-5 EDR IB
  - Note; Two virtual rails.

Core IO Requirements

Minimal IO section in the RFP, focus on fundamental requirements

- It's a *test* (ish) platform, unusually sparse requirements
- IO hardware, bandwidths, interface, and suggest acceptance tests

Implicitly, two storage tiers

- Sandia provided, external capacity store running Lustre v2.12.0

> 1x aggregate node memory capacity

$\frac{JMTTI}{\delta_{chkpt}} > 200$

- JMTTI: Job Mean Time To Interrupt
  - Average time until some MPI rank experiences a fatal error
- δ_{chkpt}: Time to check-point 80% of aggregate compute memory
- J. T. Daly, "A higher order estimate of the optimum checkpoint interval for restart dumps"
  - Future Generation Computer Systems, Vol 22, Issue 3, 2006, Pages 303-312
  - ISSN 0167-739X

Hard IO Requirements

JMTTI is ~72 hours

$\delta_{chkpt} \leq \frac{(72)(60)(60)(seconds)}{200}$, or 1296s

Aggregate memory: (2592)(128GB) = 331776GB

Required ***write*** bandwidth, then, is at least

$$\frac{(.8)(331776GB)}{1296s} = 204.8GB/s$$

IO Section Composition

36 links to the network core, aggregate of 432GB/s, unidirectional

Node

- Apollo 4520 chassis (two nodes)
- Dual socket with Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz (14 cores) cpus
- Memory 256GB (2400 RDIMM)
- IB adapter HPE InfiniBand EDR/Ethernet 100Gb 1-port 840QSFP28 Adapter
  - Note; Single rail
- Primary for 2 ZFS 7+1 storage targets

Organized as Lustre active-active pairs

- No HA; Sandia experience is too many false positives for comfort

Lustre MDS also active-active, and have DNE 1

IO Node performance

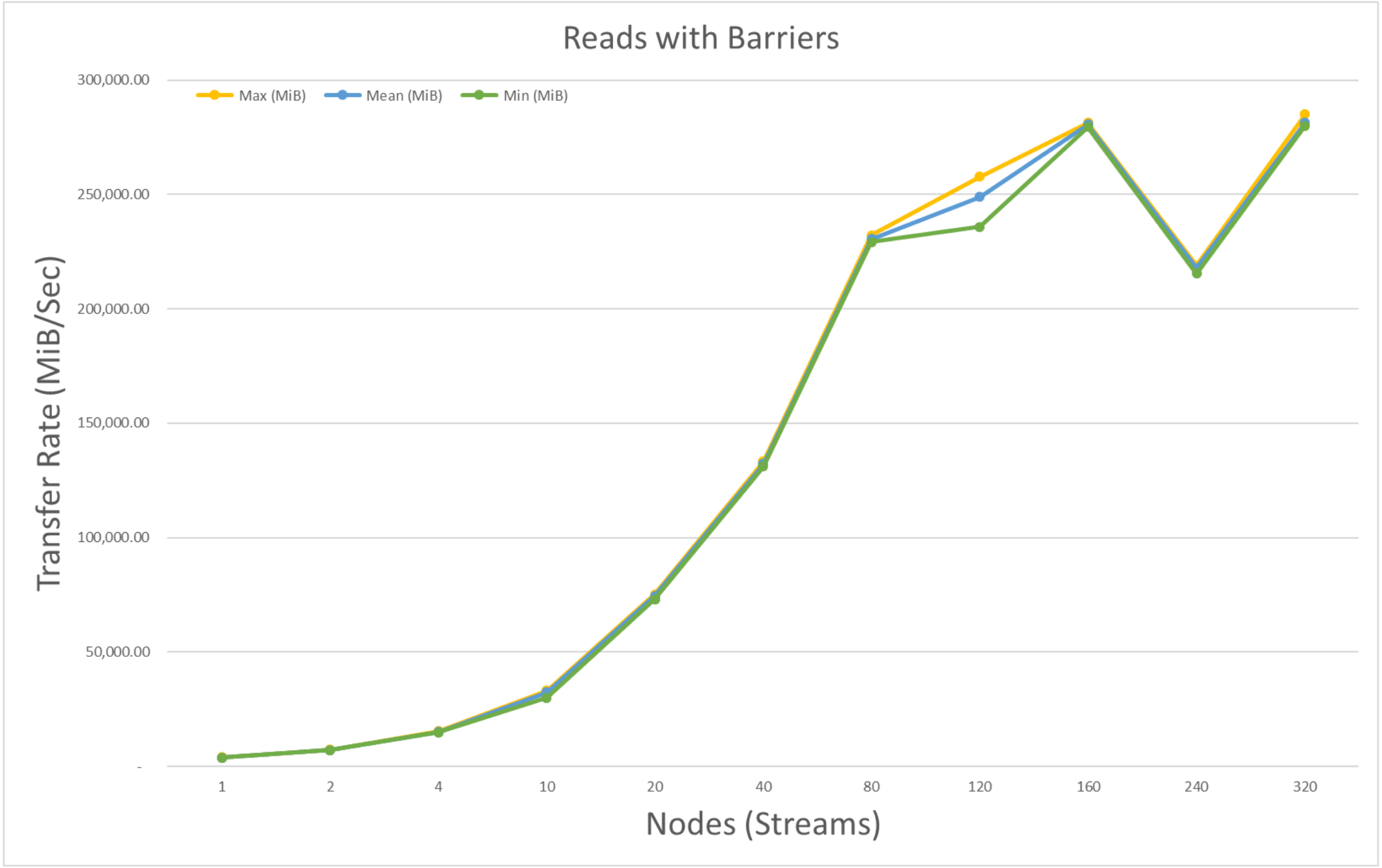
OST is a ZFS 7+1 target utilizing SSDs; 7.5/7.2 GiB/s read/write

OSS, serving two targets, then, has 15/14.4 GiB/s read/write

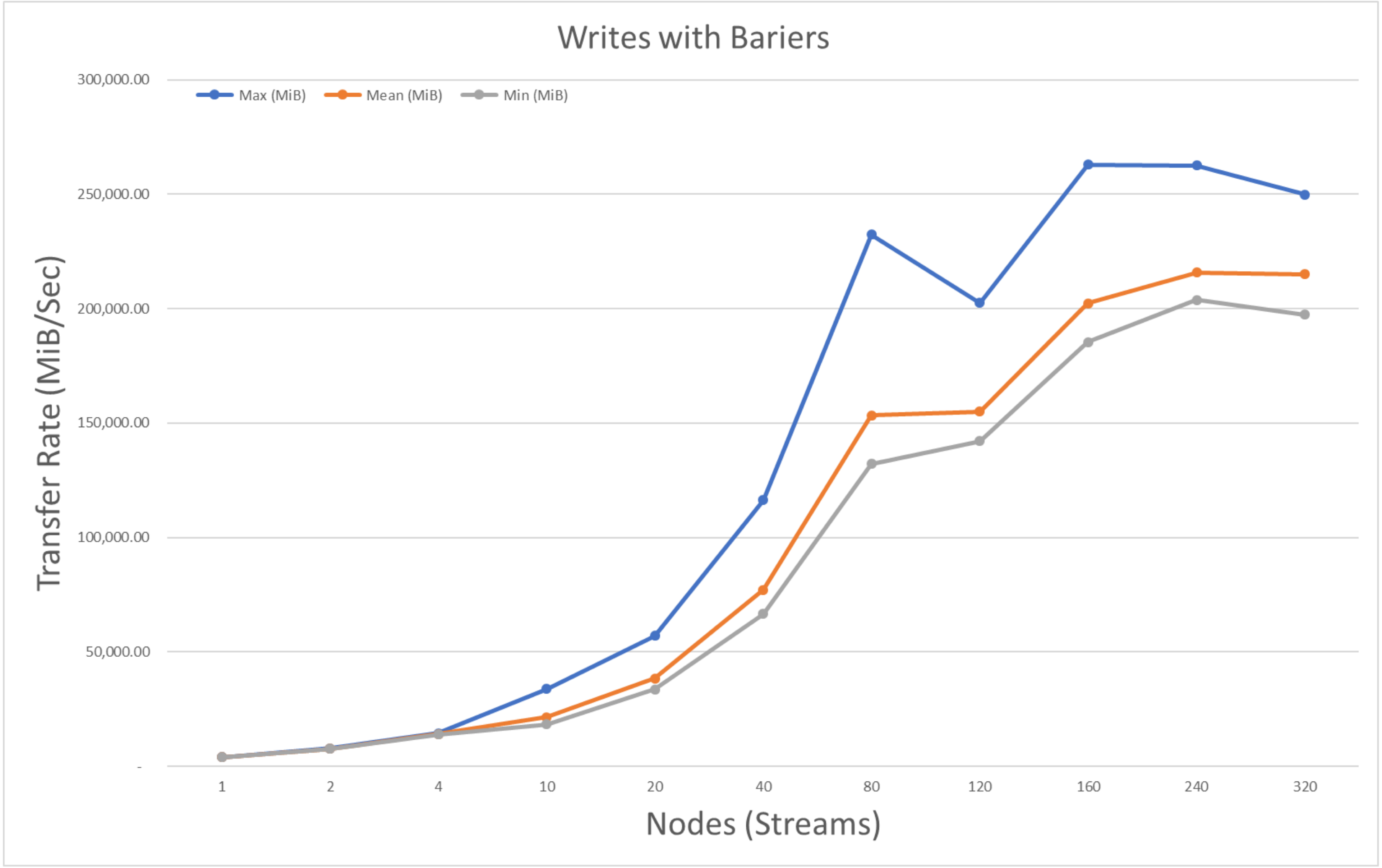
But the IB adapter theoretical, unidirectional, peak is only 12.2 GiB/s

We're leaving at least 15% of our bandwidth untapped ☹

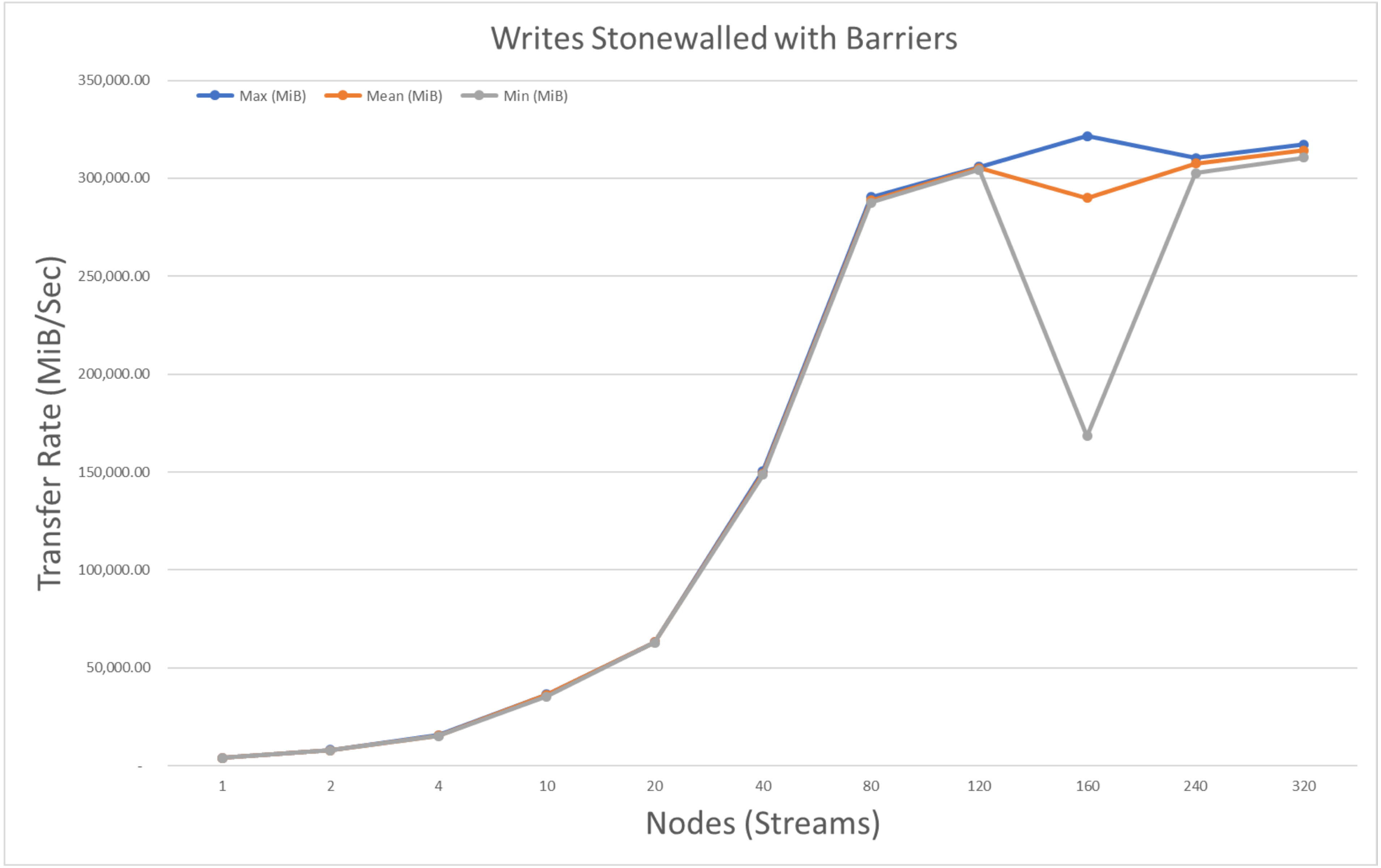
IO Performance Scaling: Read



IO Performance Scaling: Write



IO Performance Scaling: Long Tails ☹



Lessons Learned

Prepare FLASH for benchmarking

Unsynchronized job launches

Care with cabling

Unexpected purposes for the machine

FLASH infant mortality

HPE/Sandia partnership

Early adoption

Prepare the Flash for testing

Conservatively, we want our bandwidth requirements met on *aged* file systems.

Native page size for FLASH is often 256KB

- Flash Translation Layer supports TRIM but…
- Does the on-disk file system? Really? How well? Not fun to contemplate.

Early benchmark results had wild variability.

Solution was to just write the drives, end-to-end

- Yes, I know, that doesn't quite get it – FTL manages a larger pool than is exposed
- But, hey, it worked!
- Deterministic, now, to a large degree

Beware of job launch

HPE MPI 1.3 did not effectively synchronize task starts

- IOR file creates were mixed with large IO transfers
- Result, mean aggregate 188GB/s while writing
  - Very, very, very long tails

IOR –g option is your friend

- Barrier between file create and IO phases
- Could also pre-create the files, I suppose, no?

HPE MPI 1.4 launcher synchronizes start

- Works just as well as the IOR –g option

Cabling Matters ☹

Still…

Turns out it's important that data cable connectors need to be well-seated

- And reseated, and reseated again

It's best if cables to the node are long enough to open the drawer, fully, without having to remove the cables from that and the adjacent node.

- Operational considerations; Which adjacent node to shut down as well?
- See previous, first level, bullet

Will we ever learn?

Test Machines Sometimes Aren't ☹

When management tells you that you are building a test machine, don't believe it!

- Specify the requirements for a production machine anyway.
- As in, maybe not production level software but *all* the needed components should be present.

The capability to pre-stage and post-migrate data between the two stores on Astra is missing.

- Original intent was to develop this capability in the year after deployment.
  - … and the road to hell is paved with good intentions
- But the machine will enter "production" prior to the start of that effort.
- Somebody tell me what the users will do, beyond wishing me an ignominious end?

Infant Mortality is Relative

SSD failures show up quickly under use.

But really do seem to demonstrate the early part of the "bathtub" curve.

- Unlike most failures with new magnetic media
  - Most fail at initial power-on
  - The rest, at first conveyance pass
  - Then a long period of stability

SSD powers on, works fine, for awhile

- Then, write errors

Just an interesting, new to us, mode of infant mortality

- Not really enough evidence to be confident
- Only one large batch of SSD, all from a single vendor, and in the same family
- Still, something to watch for

Coequal partners

The two organizations have different experience managing machines of this size

- You mean there's another way to do this? ☺
- Have to evolve cluster mgmt/operations
  - HP Cluster Manager evolves to accommodate
    - Network boot in the IO section
    - Rapid, somewhat frequent updates to the IO section
  - DOE/NNSA has to learn a new toolset and paradigm
    - We usually use home-grown GMI
      - Easily adapted to other clusters, but…
    - Quite an ask to expect HPe to productize or even support our toolset
      - We need to be part of a community, not those-weird-folks-in-the-corner
- You really want good communication, negotiating skills, and attitudes

Early Adoption Comes With Excitement

The conservative approach isn't, always

- Let's take all the risk on the client side! Right?
  - Need only a good Lustre port to ARM for the client
  - But ARM page size is different!
    - Can tune this away
    - Better, for us on initial 2.11 deployment, was to back-patch and rebuild from source

Lustre 2.12.0 bug

- Server sometimes sees two clients as a single end-point.
  - Then round-robins between them
  - Affected clients don't receive responses, abort operations with errors
  - Service side enters recovery
    - A mere two minutes later, life is good again
- Haven't seen this since we turned off dynamic discovery.

ZFS 0.79 sometimes panics on clean shutdown

- No corruption, to date, so only an annoyance

Conclusions/Thoughts

The dense, low footprint, hence FLASH-based storage, IO service section has a large amount of untapped IO bandwidth

- One IB interface just can't move it.
- But
  - Can we show that the OSS hosts can actually tap that bandwidth?
  - Can we justify a second IB adapter per host?
    - And two additional switches?
    - And the port usage in the core?
      - Since the users would certainly rather use them to support additional compute
- Probably not, but one can always dream

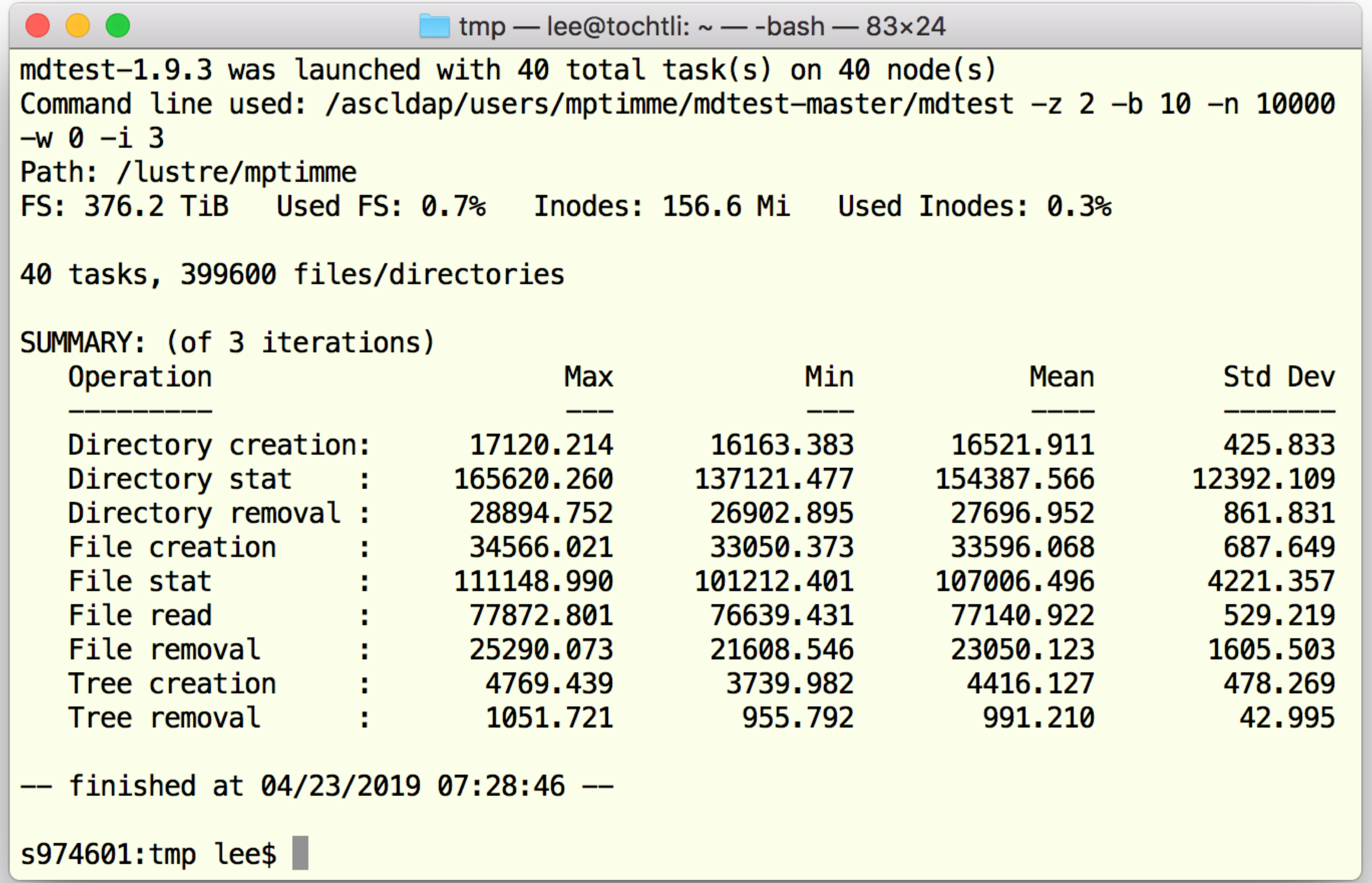
The interesting take-away seems to be that we bought for capacity, not performance

- Which means we've come full-circle
  - Twenty years ago, we bought spinning rust for capacity and captured performance en pasant
  - Then networks improved markedly, and we started buying for performance
  - Now, with FLASH-based devices…

The ball is back in the networking court

- Right where it belongs ☺

Extra: mdtest



```
mdtest-1.9.3 was launched with 40 total task(s) on 40 node(s)
Command line used: /ascldap/users/mptimme/mdtest-master/mdtest -z 2 -b 10 -n 10000
-w 0 -i 3
Path: /lustre/mptimme
FS: 376.2 TiB   Used FS: 0.7%   Inodes: 156.6 Mi   Used Inodes: 0.3%

40 tasks, 399600 files/directories

SUMMARY: (of 3 iterations)
   Operation                      Max            Min           Mean        Std Dev
   ---------                      ---            ---           ----        -------
   Directory creation:      17120.214      16163.383      16521.911        425.833
   Directory stat    :     165620.260     137121.477     154387.566      12392.109
   Directory removal :      28894.752      26902.895      27696.952        861.831
   File creation     :      34566.021      33050.373      33596.068        687.649
   File stat         :     111148.990     101212.401     107006.496       4221.357
   File read         :      77872.801      76639.431      77140.922        529.219
   File removal      :      25290.073      21608.546      23050.123       1605.503
   Tree creation     :       4769.439       3739.982       4416.127        478.269
   Tree removal      :       1051.721        955.792        991.210         42.995

-- finished at 04/23/2019 07:28:46 --

s974601:tmp lee$
```