Scalability Testing of DNE2 in Lustre 2.7
and
Metadata Performance using Virtual Machines

Tom Crowe, Nathan Lavender, Stephen Simms

Research Technologies
High Performance File Systems
hpfs-admin@iu.edu
Indiana University

RESEARCH TECHNOLOGIES
INDIANA UNIVERSITY
University Information Technology Services

PERVASIVE TECHNOLOGY INSTITUTE
INDIANA UNIVERSITY

Indiana University Metadata – Current Status

Multiple Compute Clusters

- Over 150 Disciplines served
- Mixed workloads, various I/O patterns

Current Metadata Challenge

- Single MDS/MDT comprised from 24 SAS drives (RAID-10)
- over 1B inodes
- Lustre 2.1.6 with plans to move to 2.5.X soon.

More metadata performance please

- SSD + DNE2 = goodness?

Very heavy metadata workflows can harm other users

- Can we use multiple Virtual MDS to isolate “unique” users?

Distributed Namespace Environment (DNE)

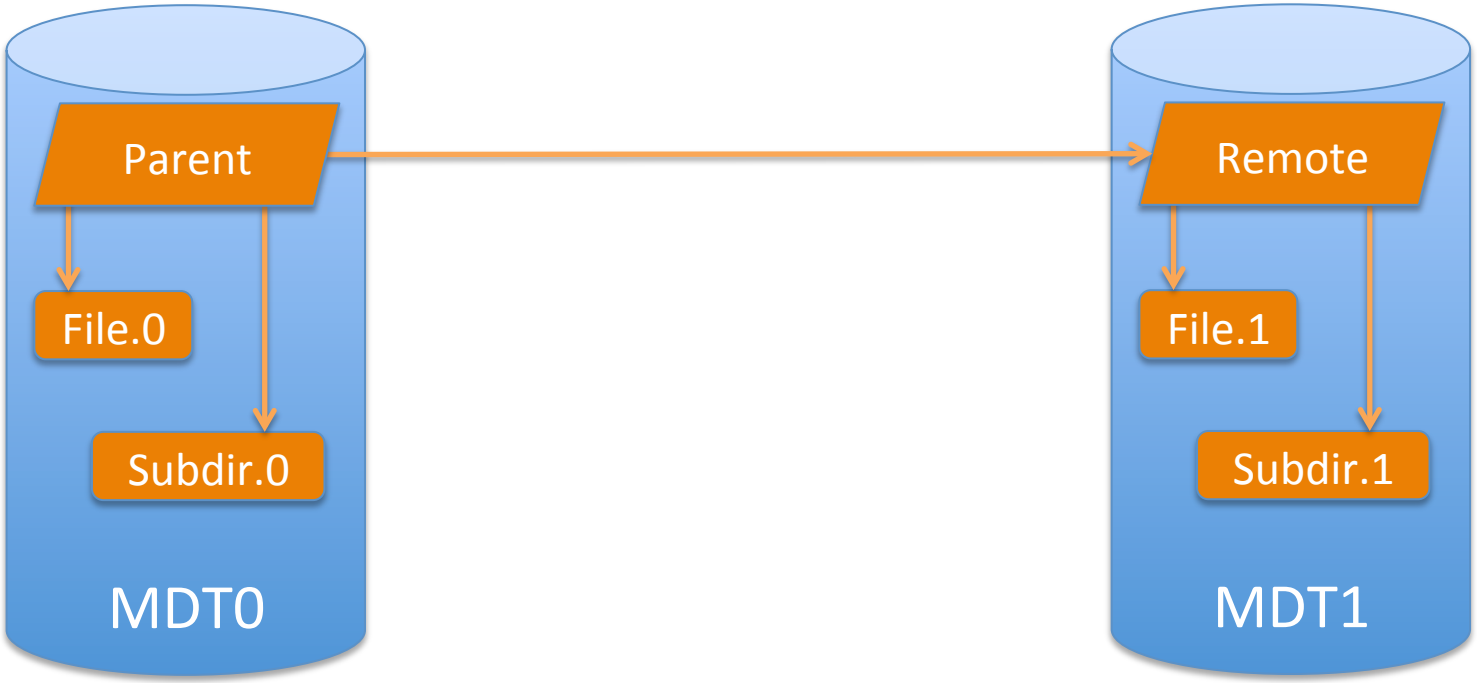
DNE Phase 1 – Lustre 2.4

- Enables deployment of multiple MDTs on one or more MDS nodes
- create directories on a specific remote MDT

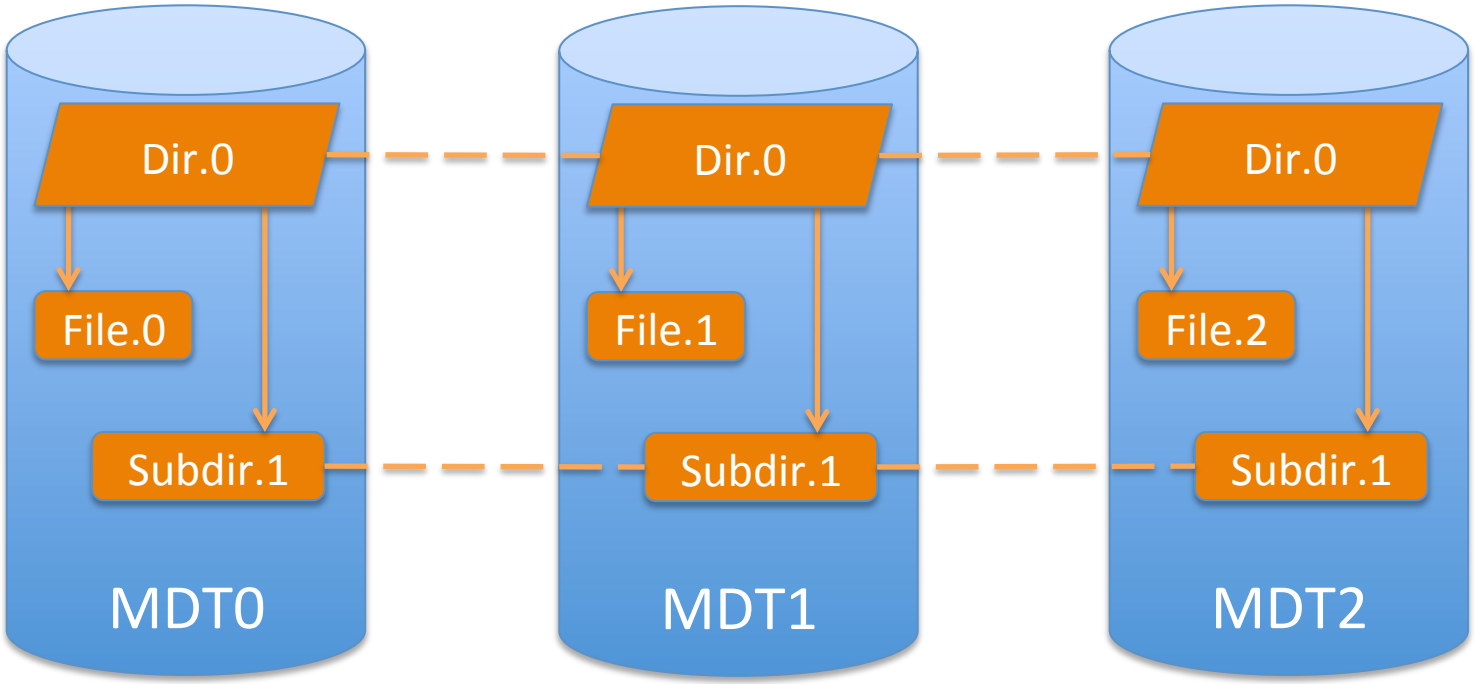
DNE Phase 2 – preview in Lustre 2.6/2.7 to be released in 2.8

- Enables deployment of striped directories on multiple MDS nodes
- Improved versatility

Distributed NamespacE (DNE) – Remote Directory



Distributed NamespacE (DNE) – Striped Directory



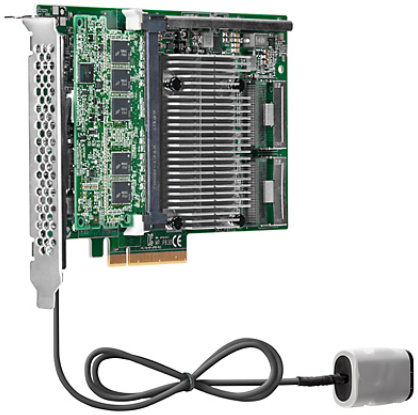
Building Blocks

(6) Servers, identical specs

- HP ProLiant DL380p Generation8 (Gen8)
- Dual socket Intel(R) Xeon(R) 2x E5-2667v2 "Ivy Bridge-EP" @ 3.30GHz 8 core
- 128GB - (16) 8GB @ 1866MHz memory
- HP Smart Array P830 controller with 4GB battery backed cache
- (6) Intel SSD DC S3500 drives (800GB drives)
- (1) SAS drive (146GB, 15,000 RPM)
- Mellanox ConnectX-3

Building Blocks (cont)

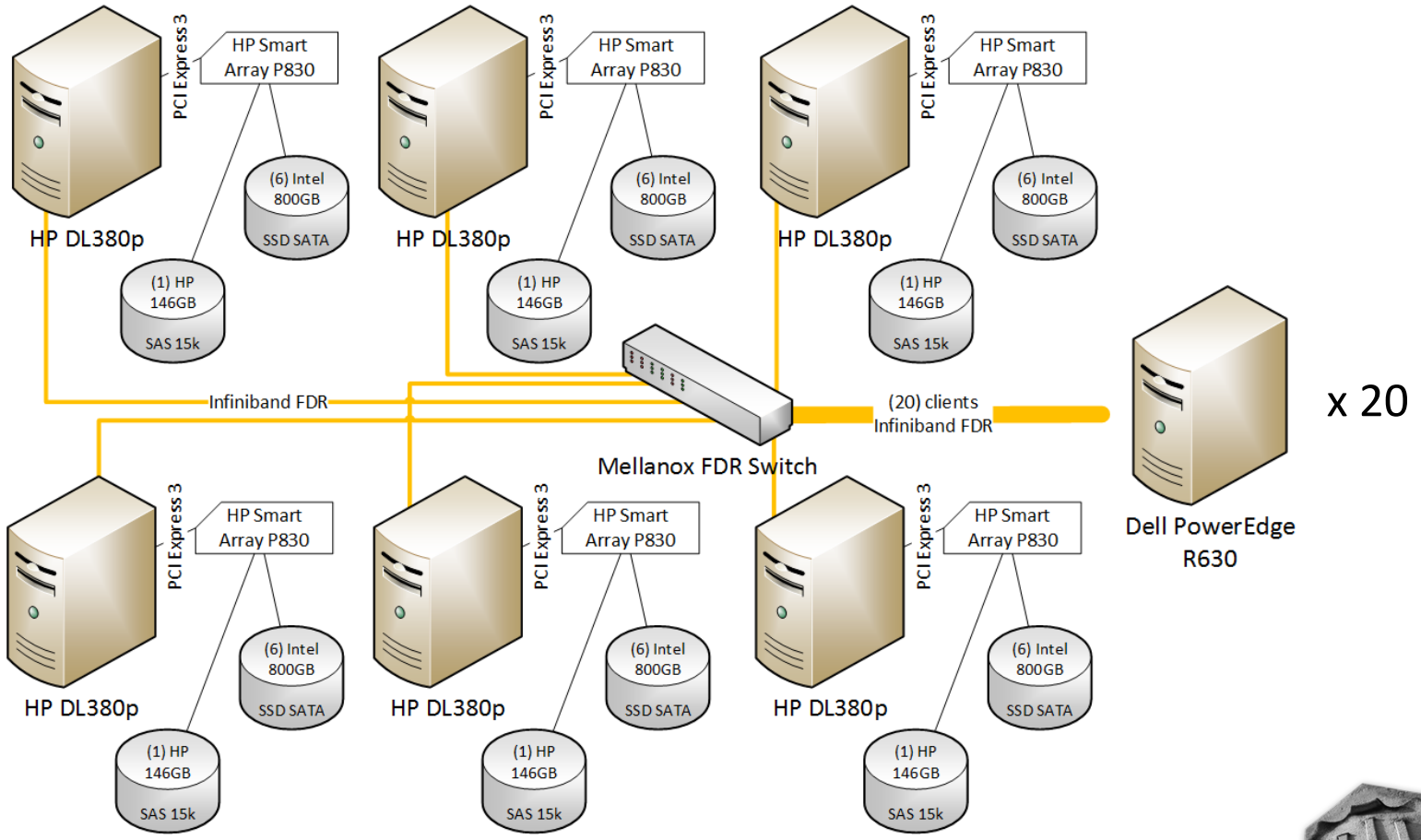
(6) HP DL380p G8 Servers



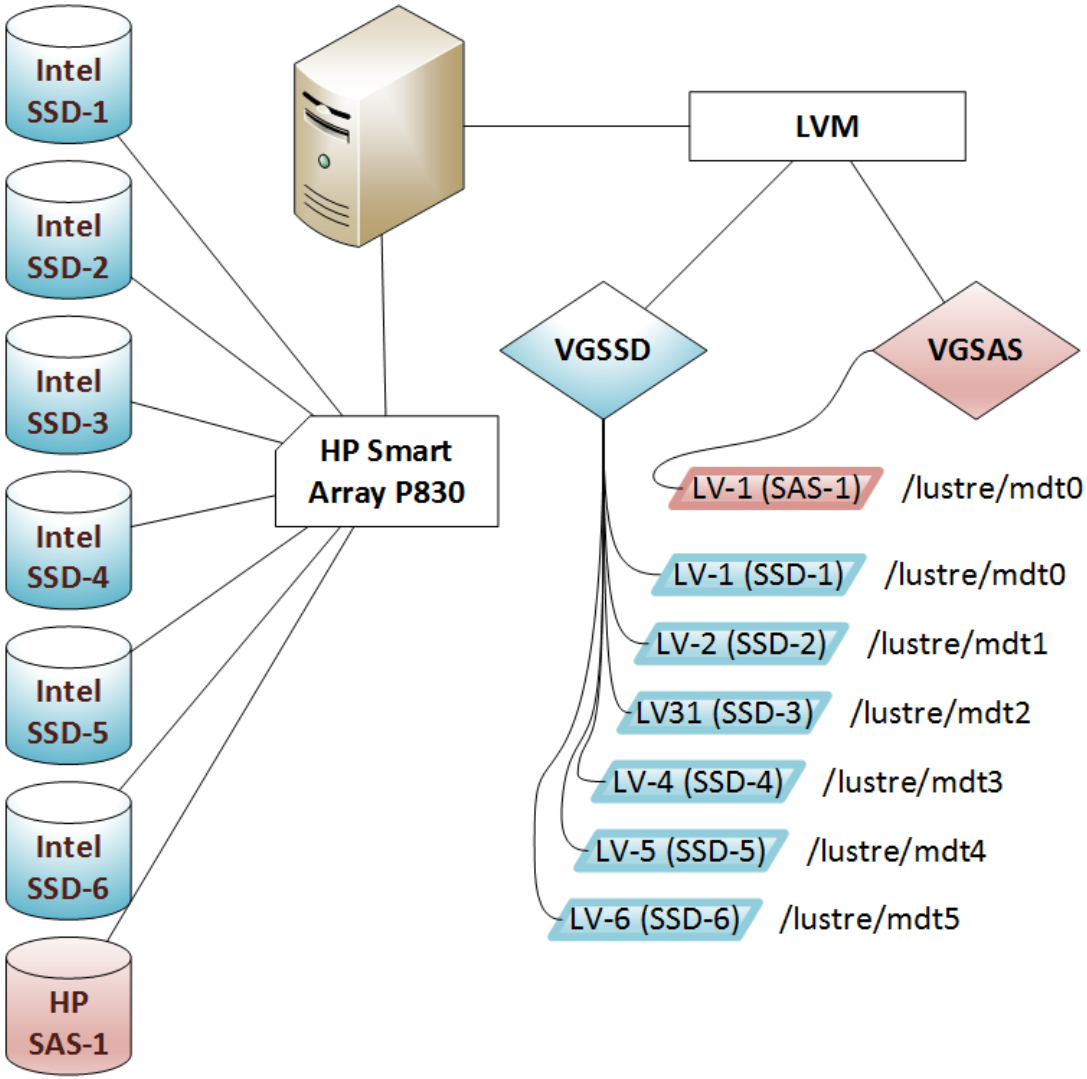
HP Smart Array
830p controller

Intel SSD DC S3500

Building Blocks (cont)



Logical Setup



Logical Setup

Block Devices

- 50GB LUNs were provisioned from each drive, preserving 1:1 layout
  - » 50GB LUNs allowed mkfs to complete in a reasonable time

File System Options

- 8GB journal
- lazy_itable_init=0
  - » Enabled by default resulting in file system activity directly following mkfs/mount

Methodology

Software

- mdsrate – lustre aware metadata benchmark in Lustre test suite
- operation - mknod (create with no OST object allocation)

Wide parameter sweep

– 20 clients, 32 mounts each, for 640 mounts simulating 640 clients

– Varied number of directories from 1 to 128 by powers of 2

– 4 threads per directory, each on a separate mount point

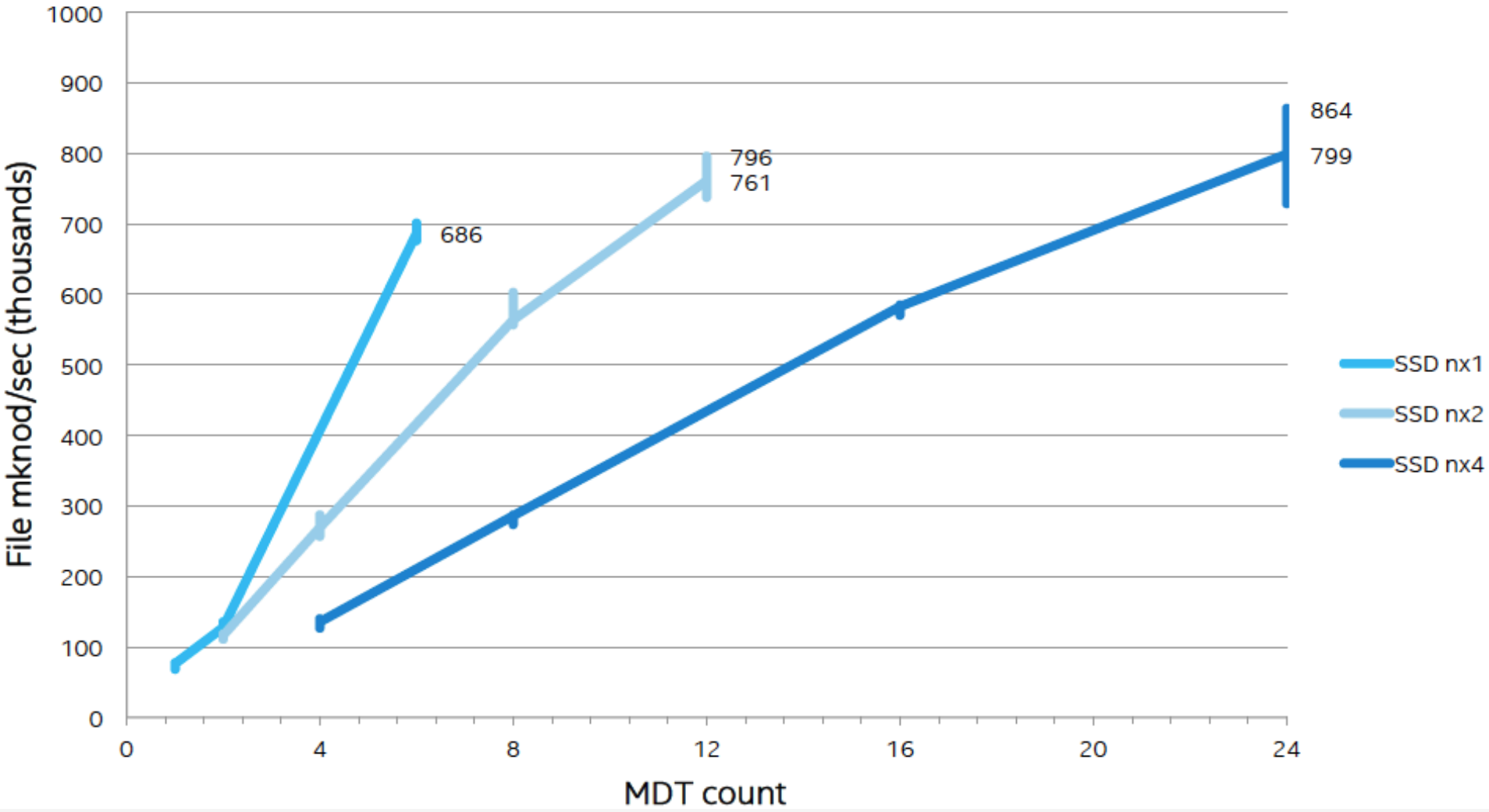
– Directory stripe count increased matching MDT count

Hardware Configurations Tested

- Single MDS, multiple MDTs
- Multiple MDSs, single MDT per MDS
- Multiple MDSs, multiple MDTs per MDS

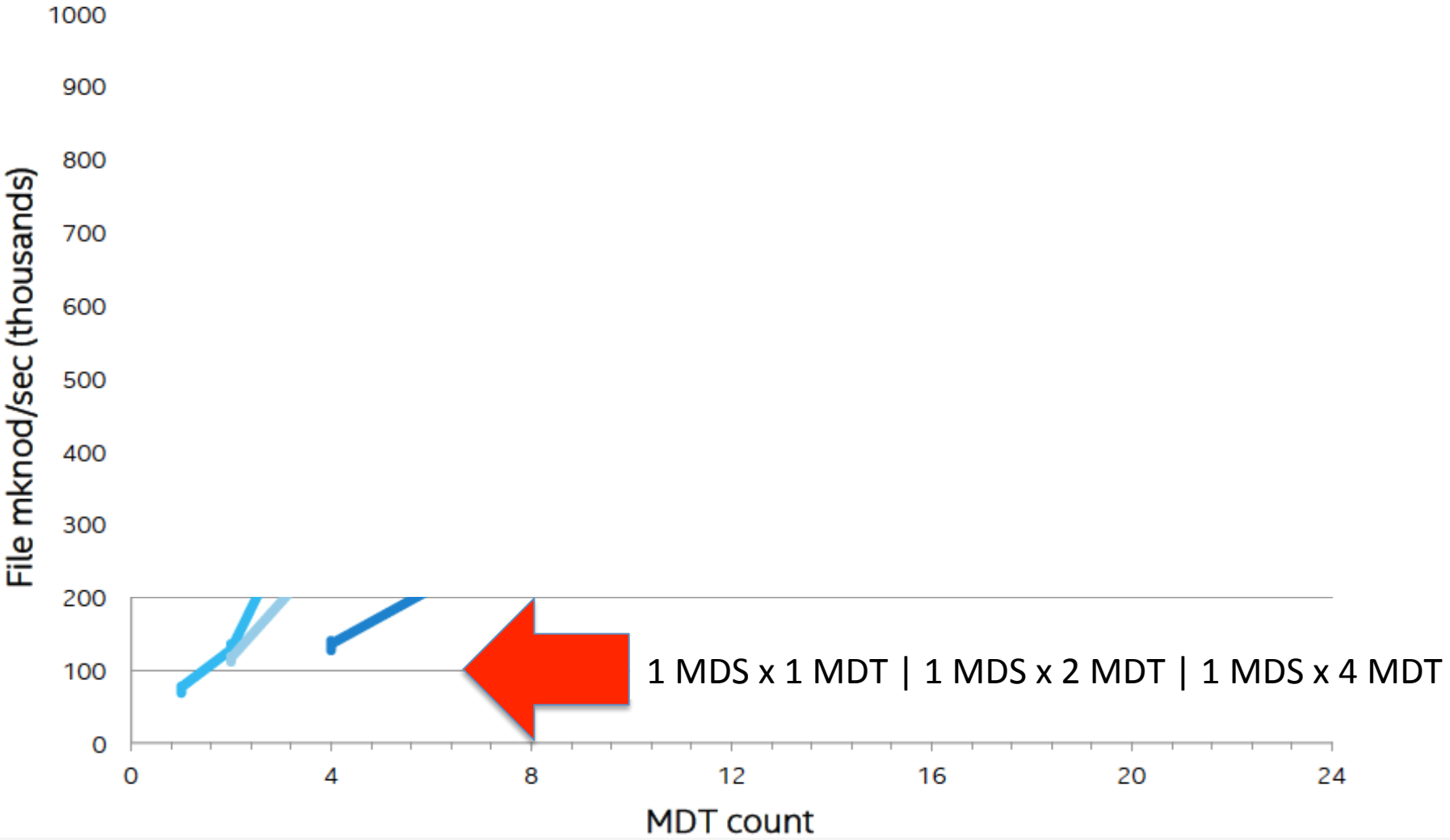
Results - mknod scaling with increasing MDS count

256 threads



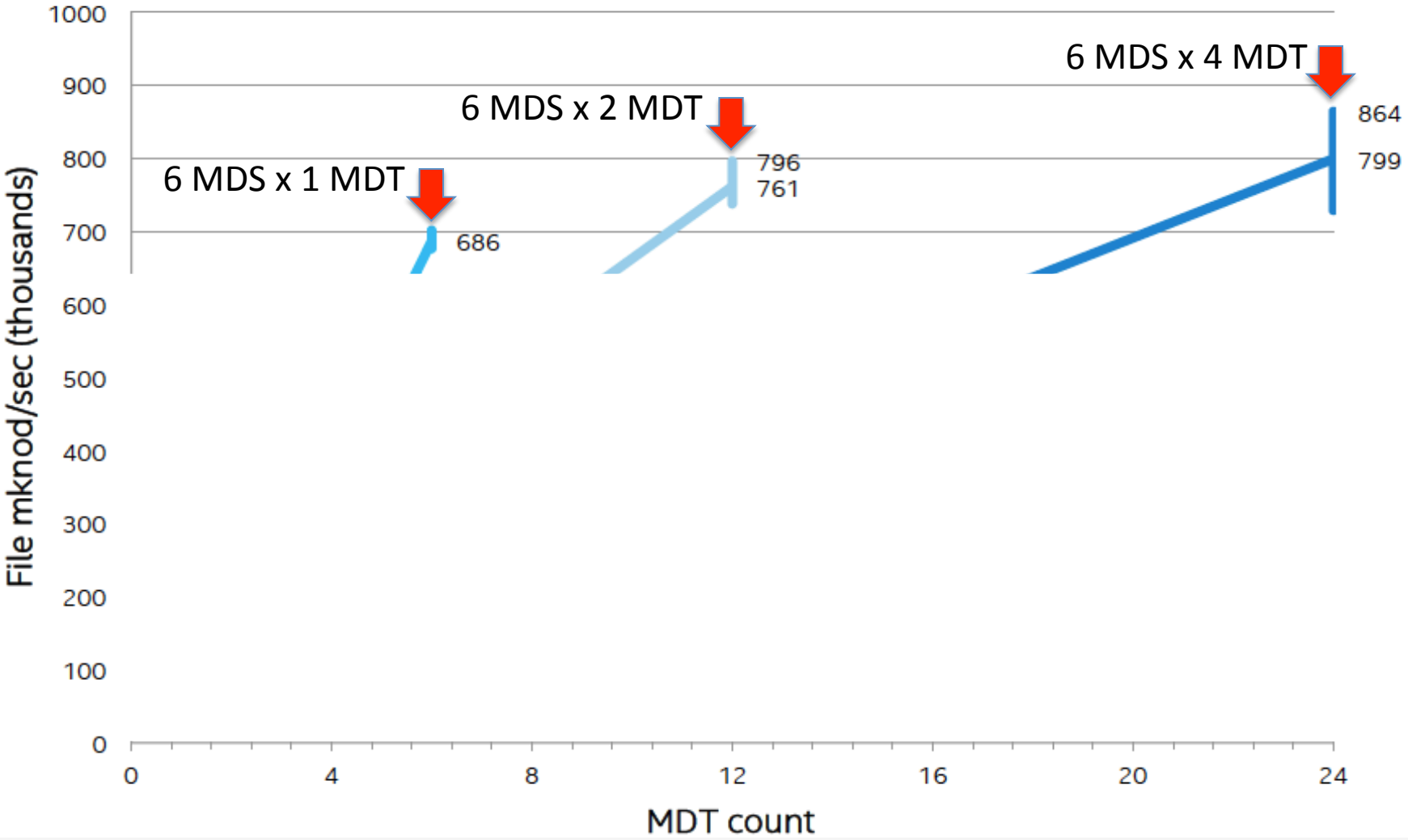
Results - mknod scaling with increasing MDS count

256 threads



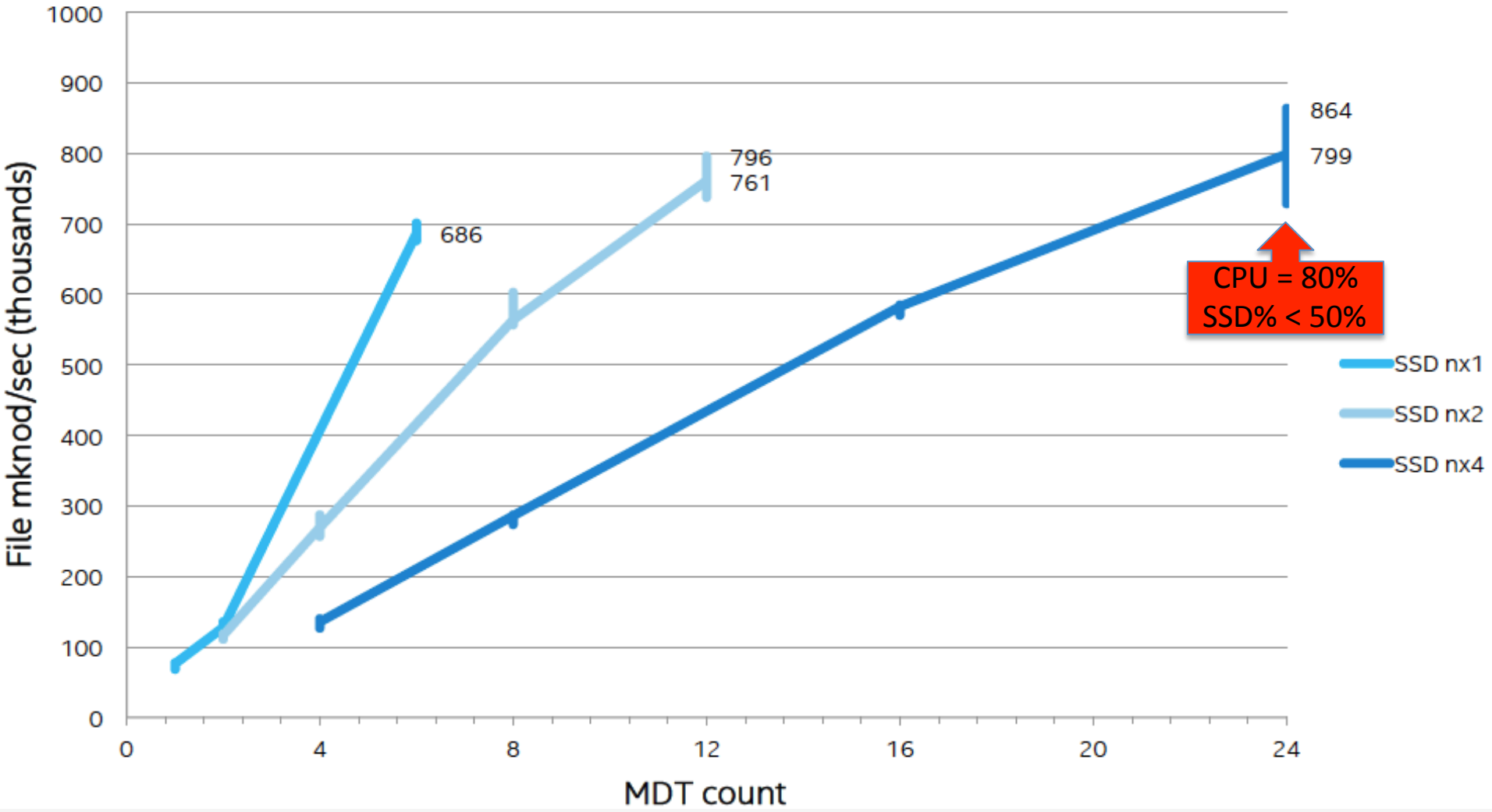
Results - mknod scaling with increasing MDS count

256 threads



Results - mknod scaling with increasing MDS count

256 threads



Summary

- DNE2 works
  - Metadata performance improves by increasing MDTs
  - Metadata performance improves by increasing MDSs
  - Adding MDSs (physical cores) overshadows adding MDTs
  - Performance in a single directory increases
  - Diminishing returns beyond 4 MDTs per MDS
- Peak performance costs 80% CPU
- Peak performance is only driving 50% of disk subsystem

Could we increase performance using VMs to drive hardware harder?

Virtual? Really? For HPC?
Well, talented people have looked at it…

Suichi Ihara

- Virtualizing Lustre LUG 2011

Robert Read

- Lustre on Amazon Web Services LUG 2013

Marc Stearman

- Per User Lustre File Systems LUG 2015

Virtual is taking up less and less overhead and is flexible:

Resize the guest: bigger/smaller.

Duplicate the guest.

Snapshot the guest.

Migrate the guest.

All before your morning coffee break…

A Possible Use Case

Stearman articulated IU's situation pretty well at LUG 2015

Some people are bad actors where metadata are concerned and don't know it
Some people have to run code that is metadata intense
Some people want to have their own separate file system

Why not use ZFS as Stearman described and create (if not user) project based file systems that put caps on size and performance. If needs are extreme or would burden other members of the research community, give them their own space.

Performance? Try SR-IOV

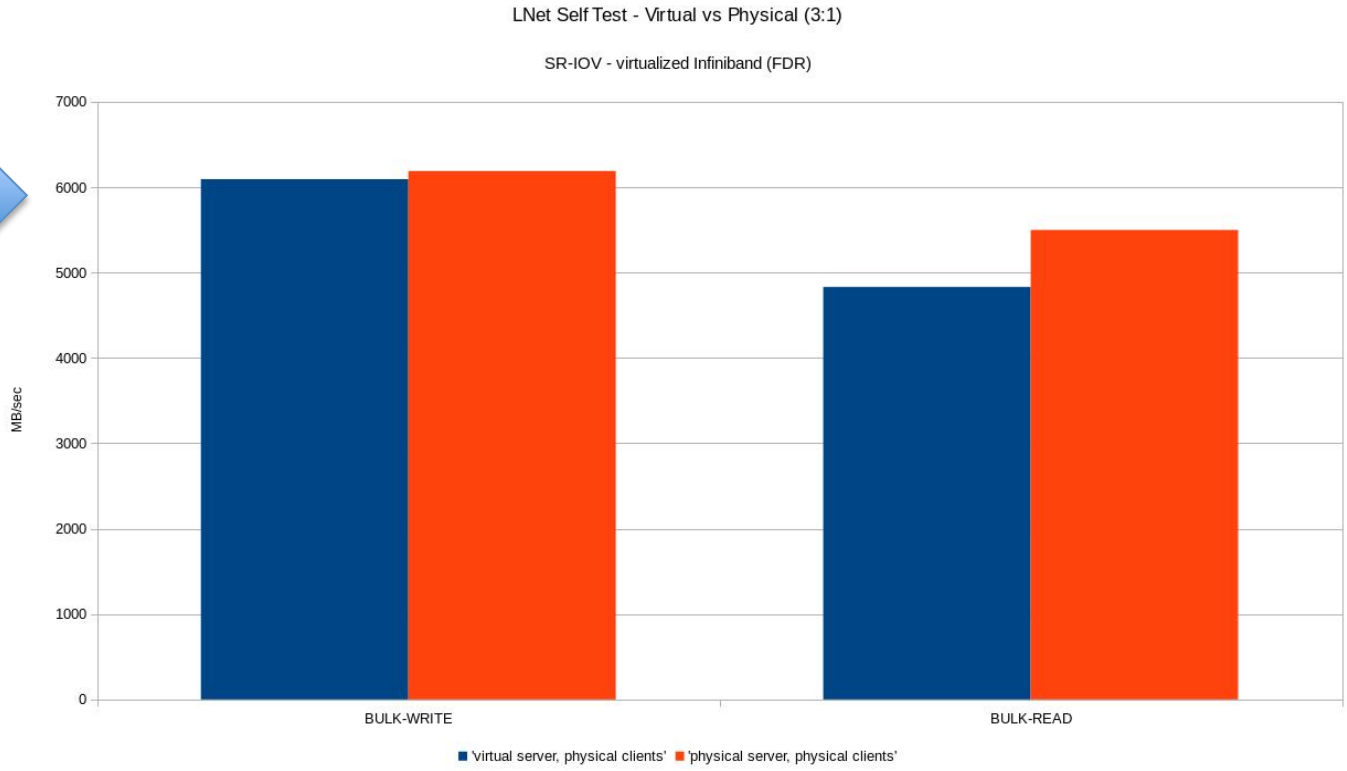
Warned that performance would be terrible

SR-IOV – Single Root I/O Virtualization

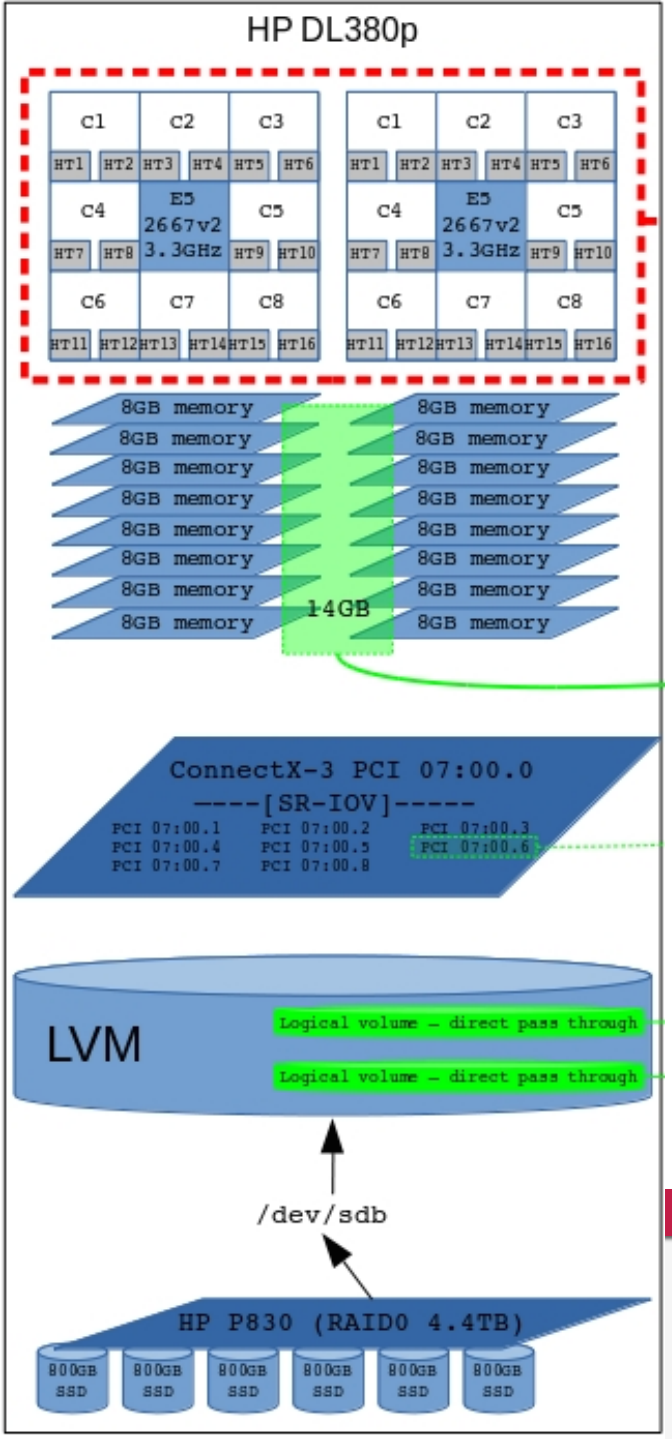
It allows a device to appear in the virtual world as multiple separate physical PCIe devices. A virtual guest thinks it has its own IB card

Lnet Self Test over IB

3 physical -> 1 virtual
VS
3 physical – 1 physical



Virtual Configuration

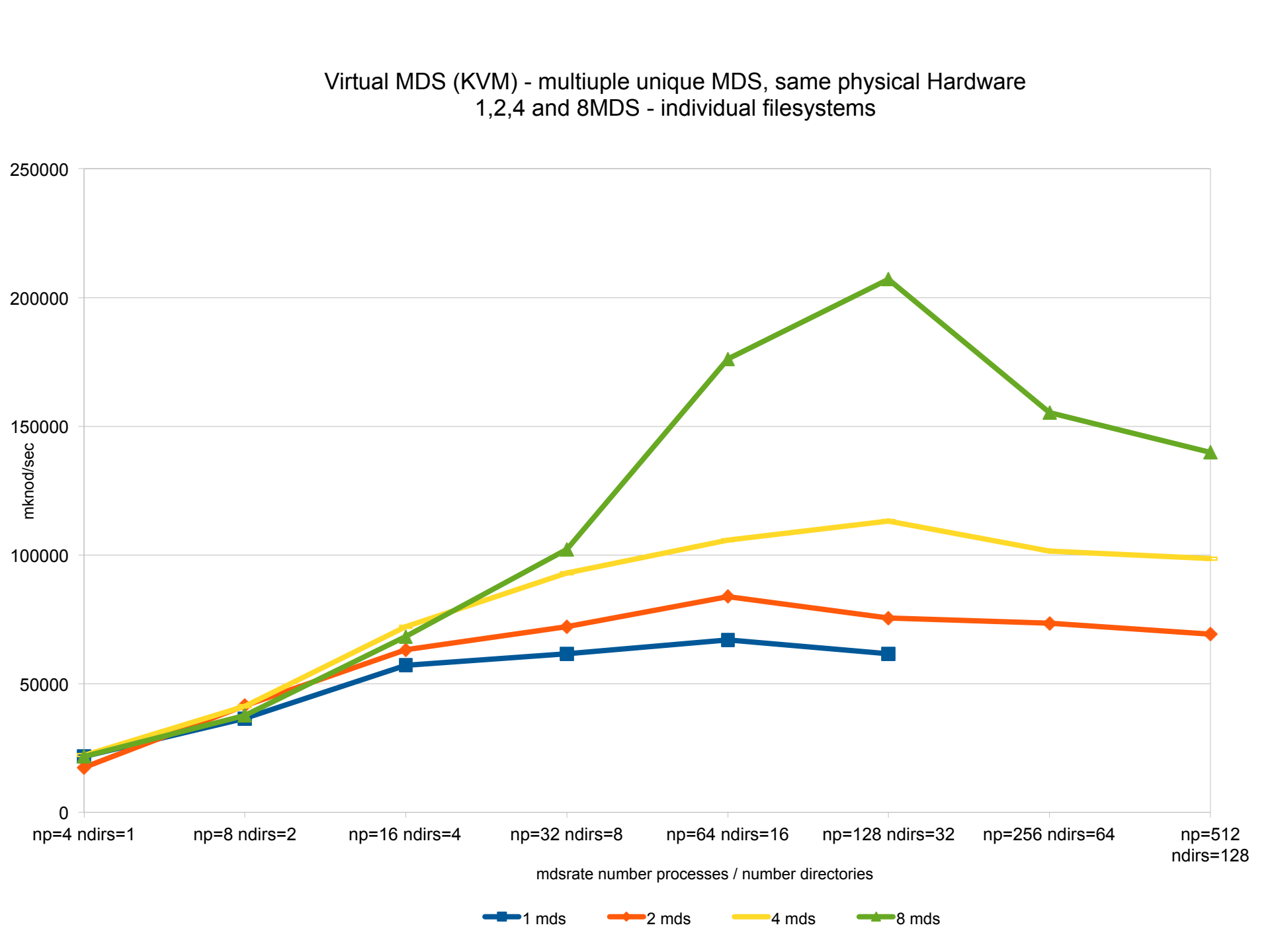


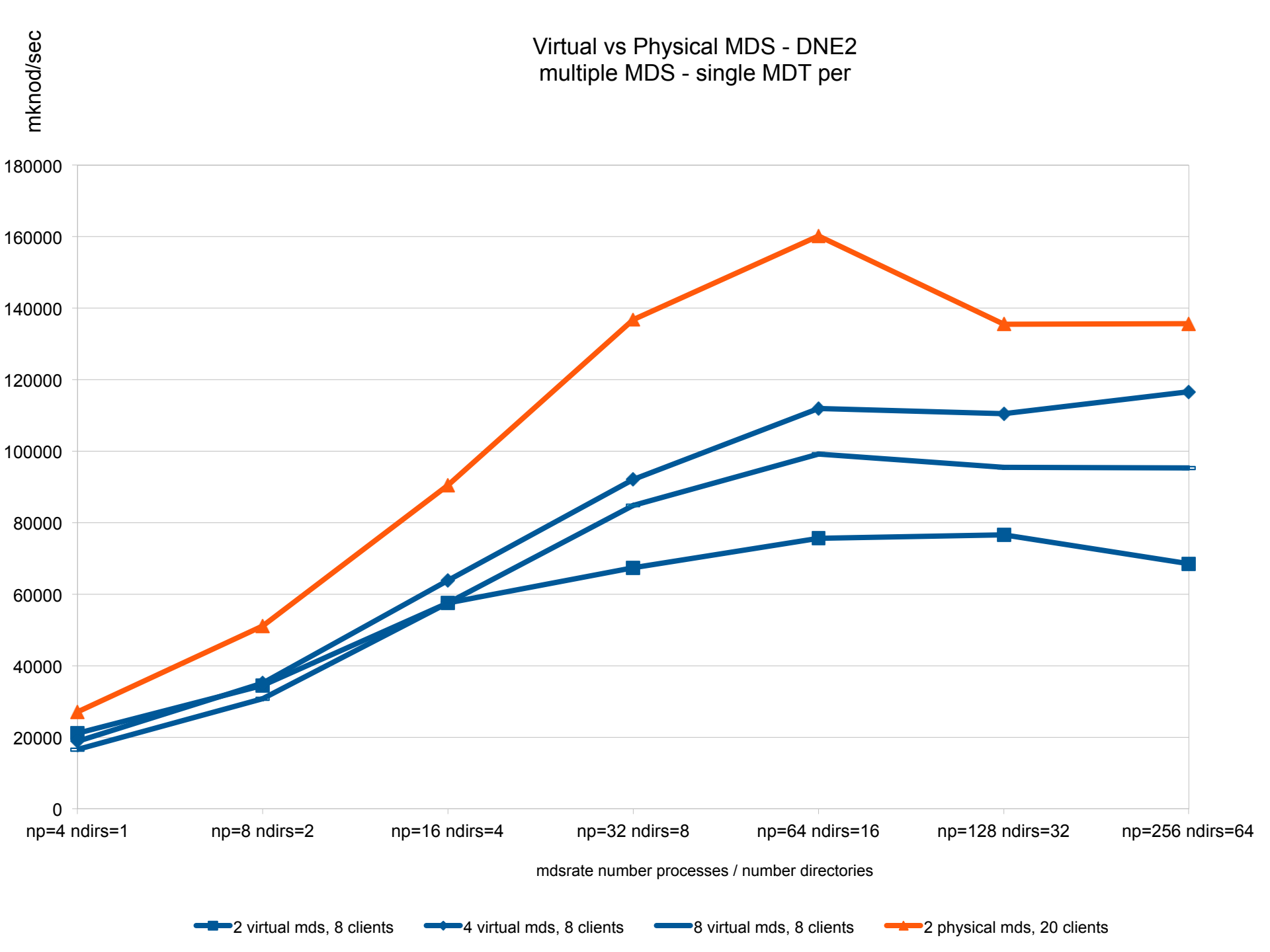
8 MDS – 8 vcpu per guest
oversubscribing physical cores 2:1

Same clients / servers / OS / Lustre as before
Same tests as before with fewer clients

All guests run through single ConnectX-3 via SR-IOV

LVM pass through allows flexibility


Virtual MDS (KVM) - multiuple unique MDS, same physical Hardware
1,2,4 and 8MDS - individual filesystems
250000
200000
150000
100000
50000
0
mknod/sec
np=4 ndirs=1
np=8 ndirs=2
np=16 ndirs=4
np=32 ndirs=8
np=64 ndirs=16
np=128 ndirs=32
np=256 ndirs=64
np=512 ndirs=128
mdsrate number processes / number directories
1 mds
2 mds
4 mds
8 mds


Virtual vs Physical MDS - DNE2
multiple MDS - single MDT per
mknod/sec
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
np=4 ndirs=1
np=8 ndirs=2
np=16 ndirs=4
np=32 ndirs=8
np=64 ndirs=16
np=128 ndirs=32
np=256 ndirs=64
mdsrate number processes / number directories
2 virtual mds, 8 clients
4 virtual mds, 8 clients
8 virtual mds, 8 clients
2 physical mds, 20 clients

Conclusions

Greater aggregate performance can be achieved using VMs
- 20% greater than best 1 MDS numbers
- Increase in service threads?
- Increase on bare metal showed no significant improvement
- Possibly a good fit for creating separate file systems for users

DNE2 performance is worse on VMs
- No magic bullet here

Future Work

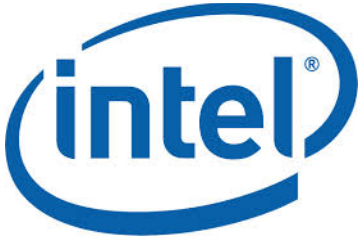
MDS work

- Always more data to be taken and sifted through
  - Adding file creation to the mix
  - Mdsrate create in lieu of mknod
- Application testing
  - Trinity Bio code for example

VM Work

- Testing and Creation of a pilot at IU

Acknowledgments



- IU's High Performance File System Team
- IU Scientific Application and Performance Tuning Team
- Matrix Integration
- Intel
- HP
- IU's Wrangler grant (NSF 13-528) partners TACC and ANL

This material is based in part upon work supported by the National Science Foundation under Grant No. NSF 13-528. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

Thank You!

Questions?