I/O Performance Traps

Erich Focht
NEC HPC Europe

WSSP, Stuttgart, October 28-29, 2013

NEC

Overview

- Motivation
- Parallel Filesystems: Lustre
  - Architecture
  - LXFS
  - LXFS in a Data Center
- High performance I/O on node local filesystem

Motivation

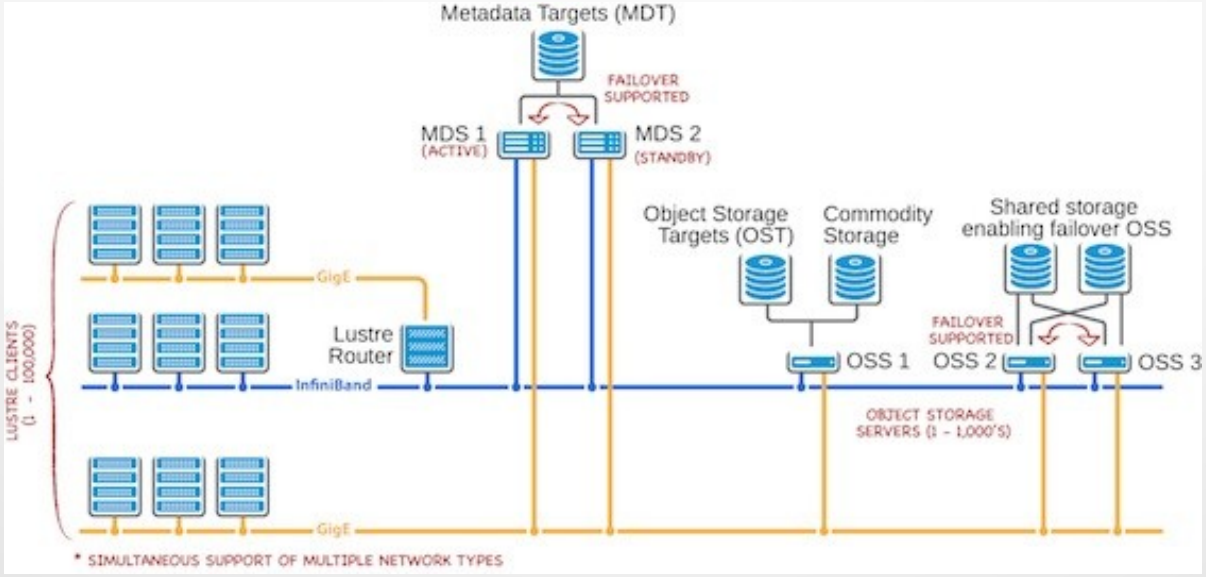
- I/O in High Performance Computing
  - Checkpoints
    - mostly write, maybe read, bandwidth
  - Input / Output
    - read, write, bandwidth
  - Databases, eg. genome/protein/...
    - Random I/O, IOPS
  - Intermediate results, eg. out of core solvers, ...
    - Random I/O, IOPS, bandwidth
  - Mostly to/from parallel filesystems

Motivation

- Big Data
  - Structured data: SQL databases
    - In memory, or not
    - IOPS, memory bandwidth, alignment, compression
  - Unstructured data
    - NoSQL
    - HDFS
    - Node local filesystems
    - MapReduce, analytics...
    - MPI

Parallel Filesystem: Lustre

- Architecture: built for scalability
  - Scale with number of
    - OSS/OST: capacity, bandwidth
    - MDS/MDT: number of files, metadata performance (DNE)
  - Lnet Router: decouple I/O network from compute network



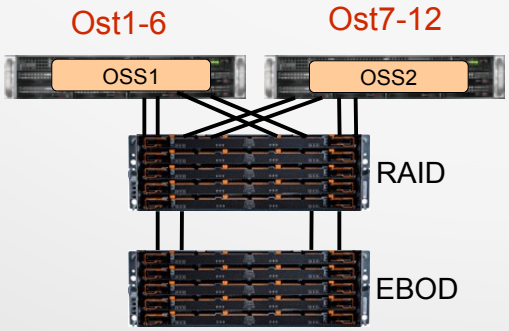
Lustre Based LXFS Appliance

- Since 2007: parallel filesystem solution integrating validated HW (servers, networks, high performance storage), deployment, management, monitoring SW.
- Each LXFS block is a pre-installed appliance
  - Extremely simple provisioning
- Entire storage cluster configuration is in one place: LxDaemon
  - Describes the setup, hierarchy
  - Contains all configurable parameters
  - ***Single data source*** used for deployment, management, monitoring
- Deploying the storage cluster is very easy:
  - Auto-generate configuration of components from data in LxDaemon
  - Storage devices setup, configuration
  - Servers setup, configuration
  - HA setup, services, dependencies, STONITH
  - Formatting, Lustre mounting
- Managing & monitoring LXFS
  - Set of simple commands, hiding complexity of underlying setup
  - Health monitoring: pro-active, sends messages to admins, reacts to issues
  - Performance monitoring: time-history, GUI, discover issues and bottlenecks

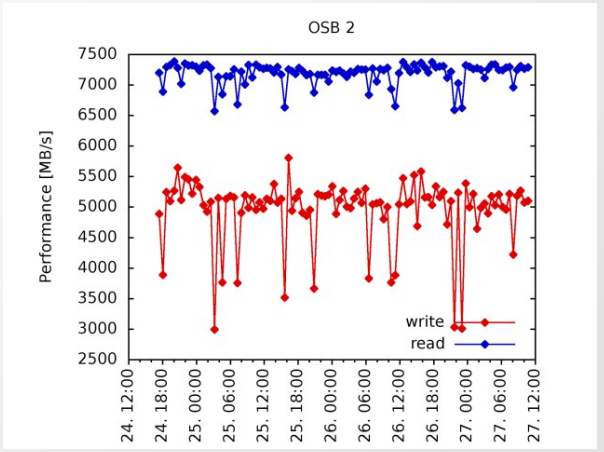
NEC

Eg.: Performance Optimized OSS Block

- Specs
  - > 6GB/s write, > 7.5GB/s read, up to 384TB in 8U (storage)
  - 2 Servers: highly available, active-active
    - 2 x 6core E5-2620(-v2)
    - 2 x LSI9207 HBA, 1 x Mellanox ConnectX 3 FDR HCA
  - Storage
    - NEC SNA460 + SNA060 (built on NetApp E5500)
- ...

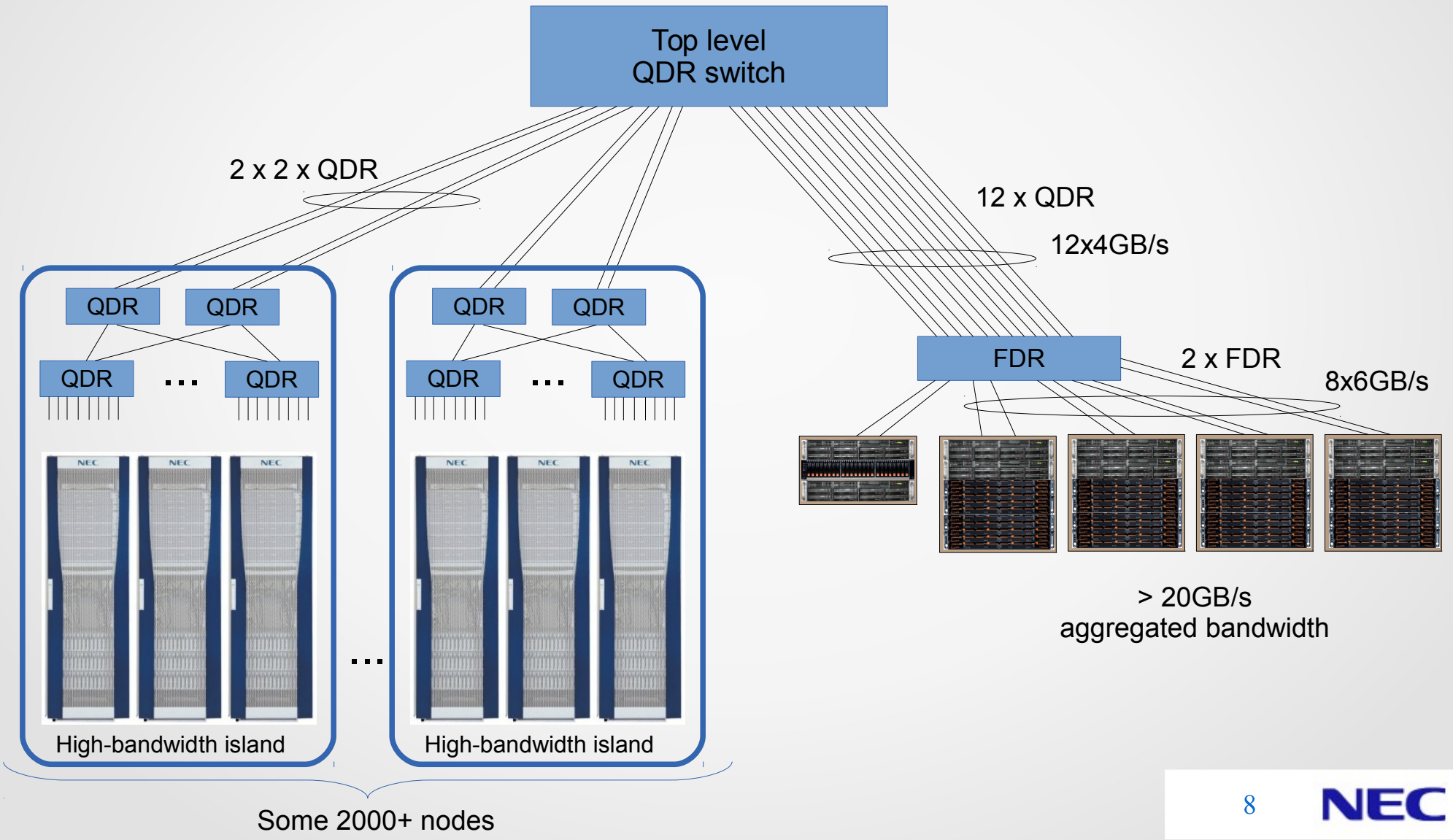


Performance example
80 x 1TB NL-SAS 7.2k
IOR
In noisy environment



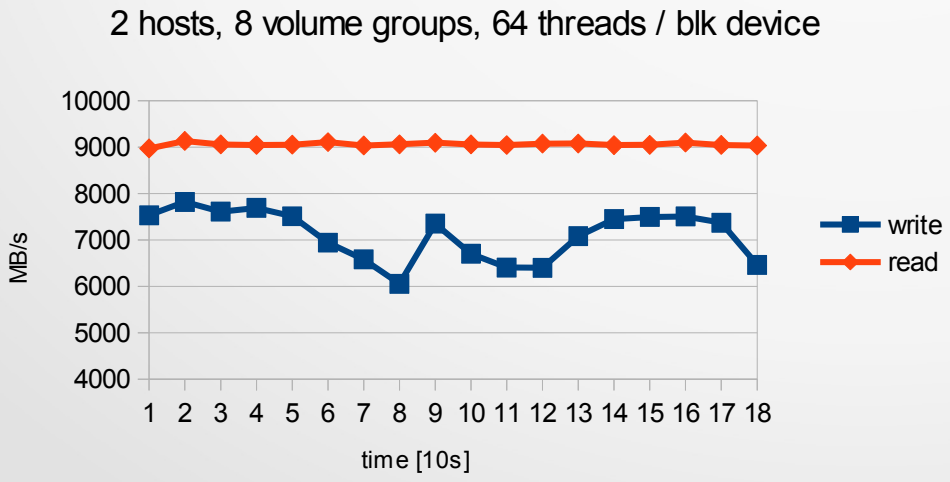
NEC

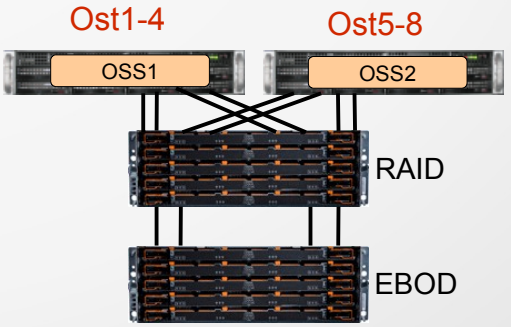
Integrated In The Datacenter



Demonstrate 20GB/s on 8 or 16 Nodes!

- Check all layers!
- OSS to Block Devices
  - vdbench, sgpdd-survey, with Lustre: obdfilter-survey
  - not fio! (multiple threads hit same address: controller cache)
    - Seen with blktrace
  - Read/write to controller cache! Check PCIe, HBAs (SAS)





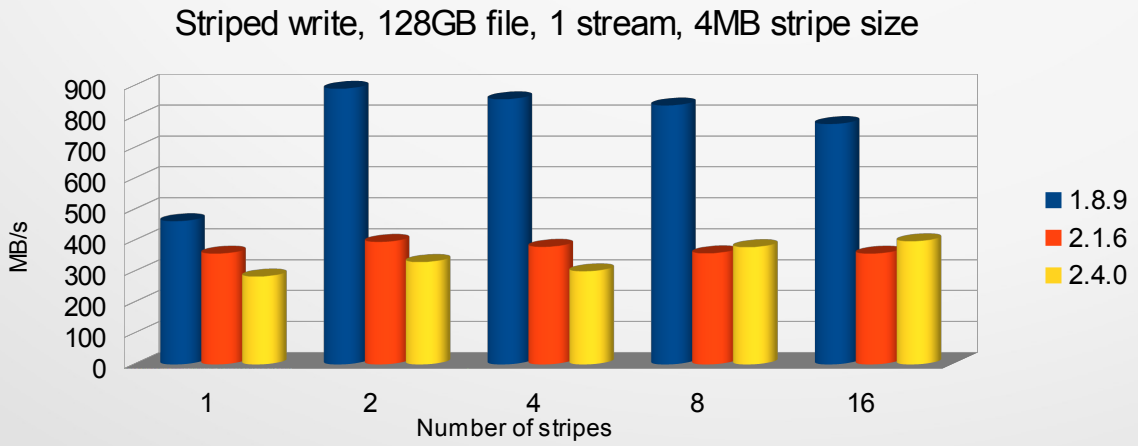
NEC

Demonstrate 20GB/s on 8 or 16 Nodes!

- Lnet (Lustre Network Layer)
  - Derived from Portals
  - TCP/IP (socklnd), InfiniBand Verbs (o2iblnd), ...
  - added routing between LNETs
  - Test with lnet-selftest (lst)
  - peer_credits? Enough in-flight transactions?
    - Default: 8, small. Number of streams, clients, OSSes...

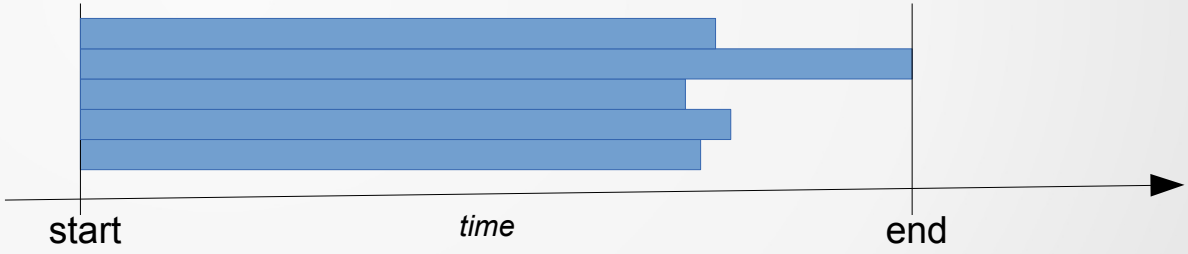
Demonstrate 20GB/s on 8 or 16 Nodes!

- Client nodes
  - InfiniBand BW: QDR, PCIe gen2 x8: max 3.2GB/s... often 2.8GB/s
  - 8 nodes: peak QDR bandwidth ~24GB/s, close to 20GB/s
  - Client Lustre Version!
    - 1.8.9 still fastest client!
    - 2.1.6 : striped I/O performance doesn't scale
    - 2.4.0 : striped I/O performance doesn't scale



Multi-Streams I/O Benchmark Problems

- IOR, IOZONE, ...
- When „just“ measuring IOR on 8/16/24 nodes:
  - Bandwidth varies, 10-18GB/s
  - Storage is ... idle
- Stragglers!
  - Hide performance of I/O subsystem

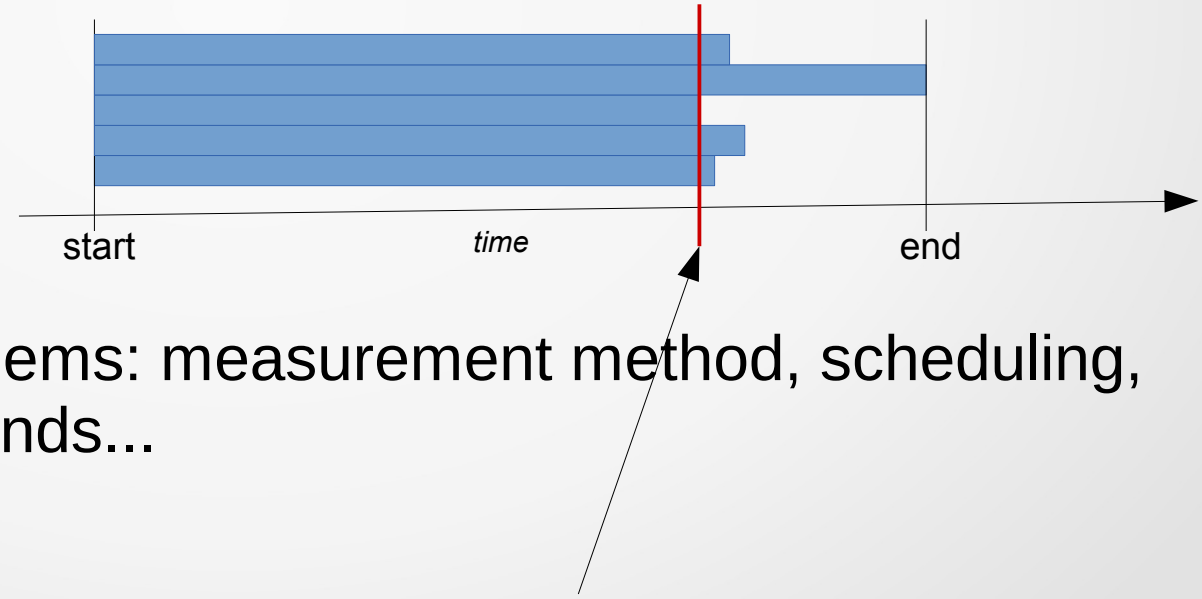


  - Might point to problems: measurement method, scheduling, imbalances of all kinds...

NEC

Multi-Streams I/O Benchmark Problems

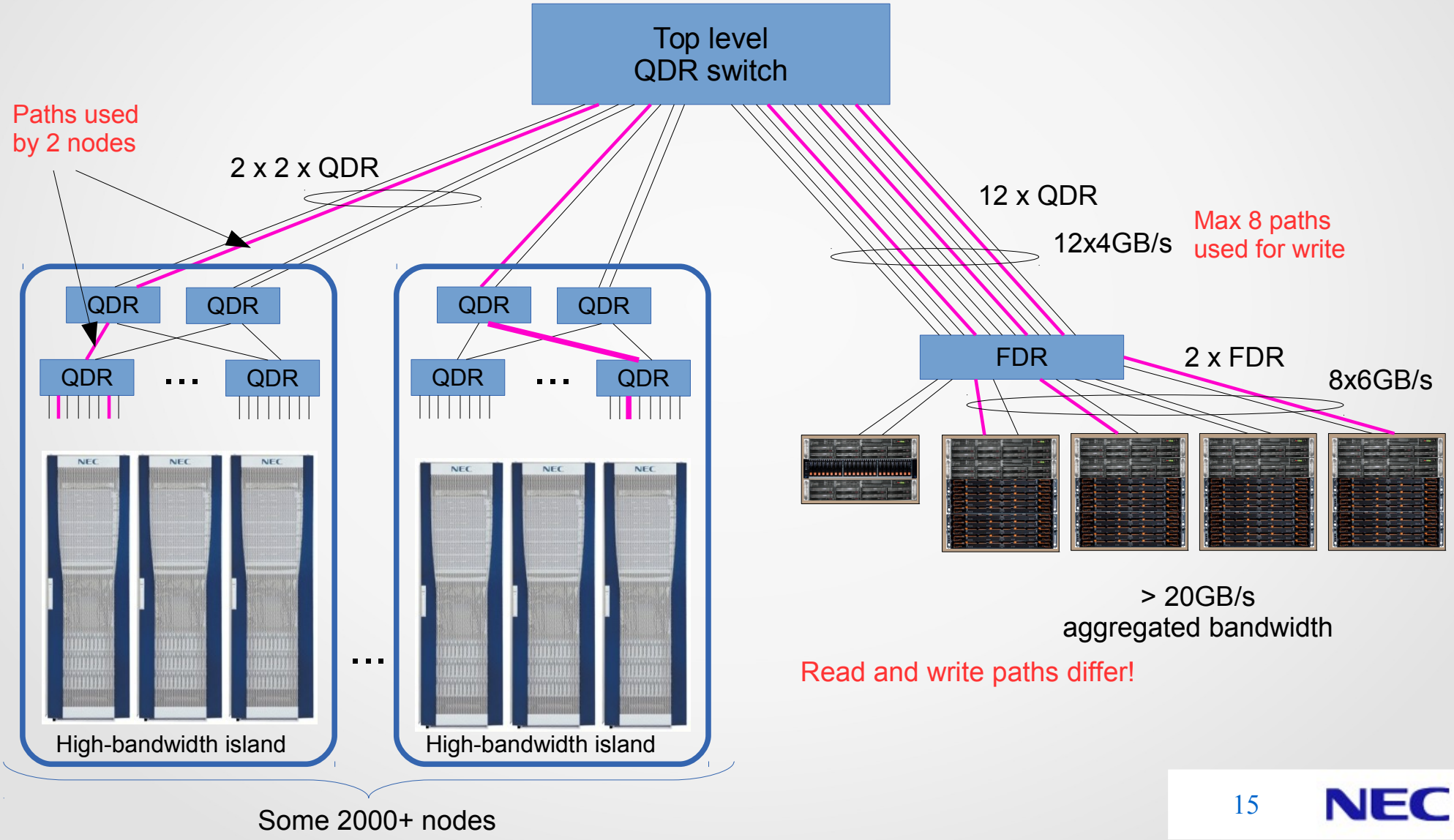
- IOR, IOZONE, ...
- Naively measuring IOR on 8/16/24 nodes:
  - Bandwidth varies, 10-18GB/s
  - Storage is ... idle
- Stragglers!
  - Hide performance of I/O subsystem
  - Might point to problems: measurement method, scheduling, imbalances of all kinds...
- Stonewalling
  - Stop measuring once first thread finished



Multi-Streams I/O Benchmark Problems

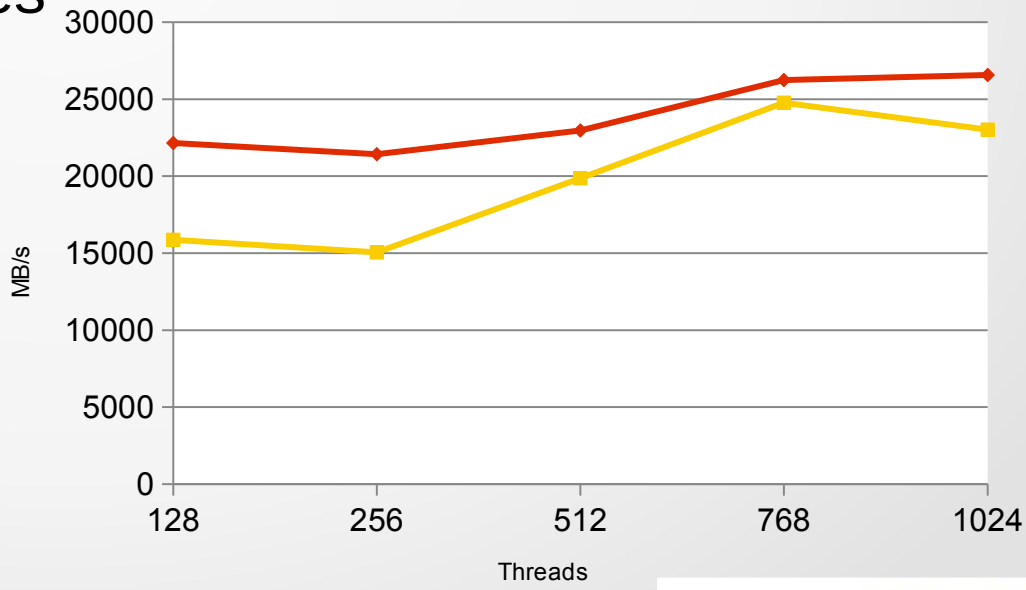
- InfiniBand
  - Is great!
  - Brings in new problems due to static routing
  - Each IB switch chip has a static routing table
    - Go to LID x through port y
  - Not dynamically changeable:
    - Existing connections (QPs) contain the route
    - Mellanox UFM helps, eg. for jobs, changes routing, but before job starts and connenctions are built up
  - Why do I mention this here?

Demonstrate 20GB/s on 8 or 16 Nodes!



Demonstrate 20GB/s on 8 or 16 Nodes!

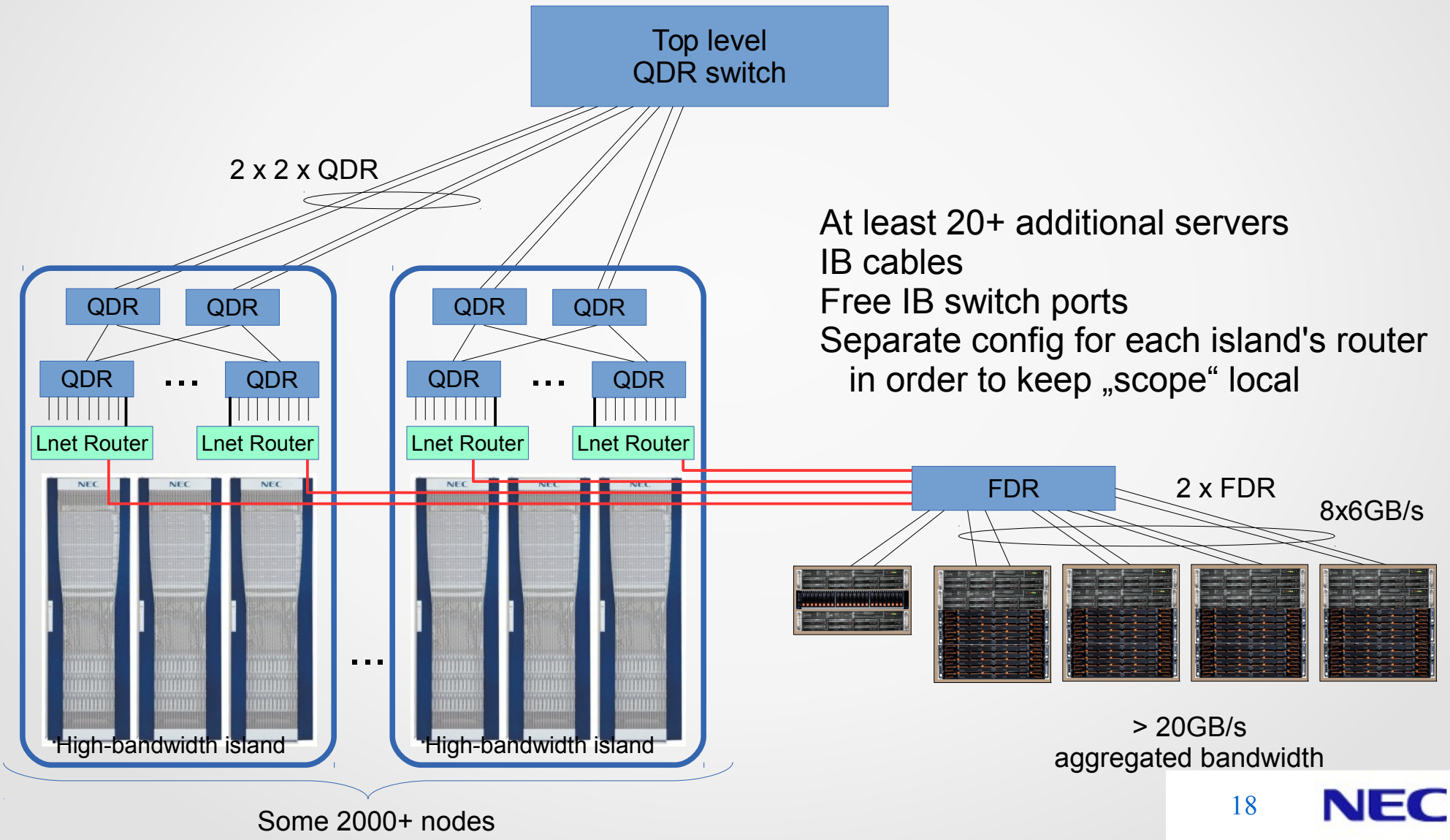
- Given the routing is static: select nodes carefully
  - Out of a pool of reserved nodes
  - Minimize oversubscription of InfiniBand links on both paths
  - ... to OSTs/OSSes that are involved
  - ... and back to client nodes
- Use stonewalling
- Result from 16 clients
  - Write: max 24.7GB/s
  - Read: max 26.6GB/s



Remarks

- Performance of storage devices is huge
  - Performance of one OSS is O(IB link bandwidth)
- Sane way of integrating is more expensive and more complex:
  - Lustre/Lnet routers
    - Additional hardware
    - Each responsible for its „island“
    - Can also limit I/O bandwidth in an „island“

Integrated In Datacenter: The Right Way



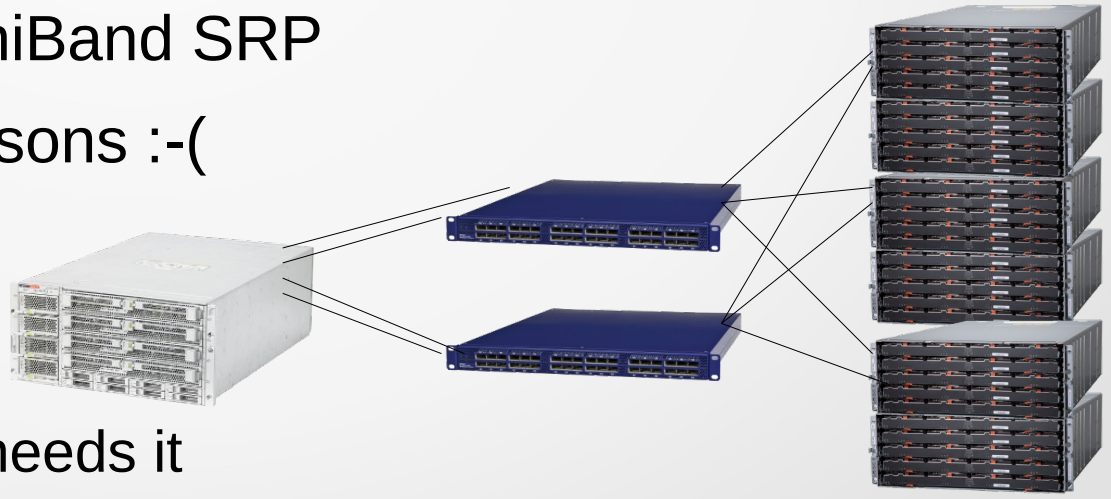
At least 20+ additional servers
IB cables
Free IB switch ports
Separate config for each island's router
in order to keep „scope“ local

And now to something much simpler

Server doing I/O to own block storage

- Quite fat server
  - 8 sockets Intel E7-8800, 80 cores, 1TB RAM
- Quite fast storage (SNA400 AKA E5500)
  - O(300) disk drives, NL-SAS
- Accessed through InfiniBand SRP
- QDR, for technical reasons :-(

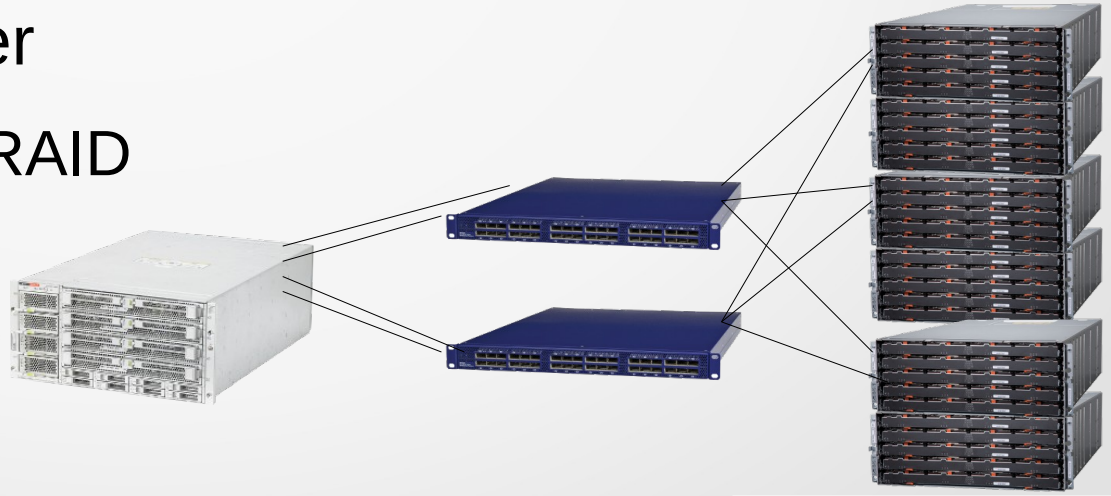
- Big Data, somehow...
  - Storage on demand
  - Assigned to server that needs it



NEC

Local Block Storage: Software

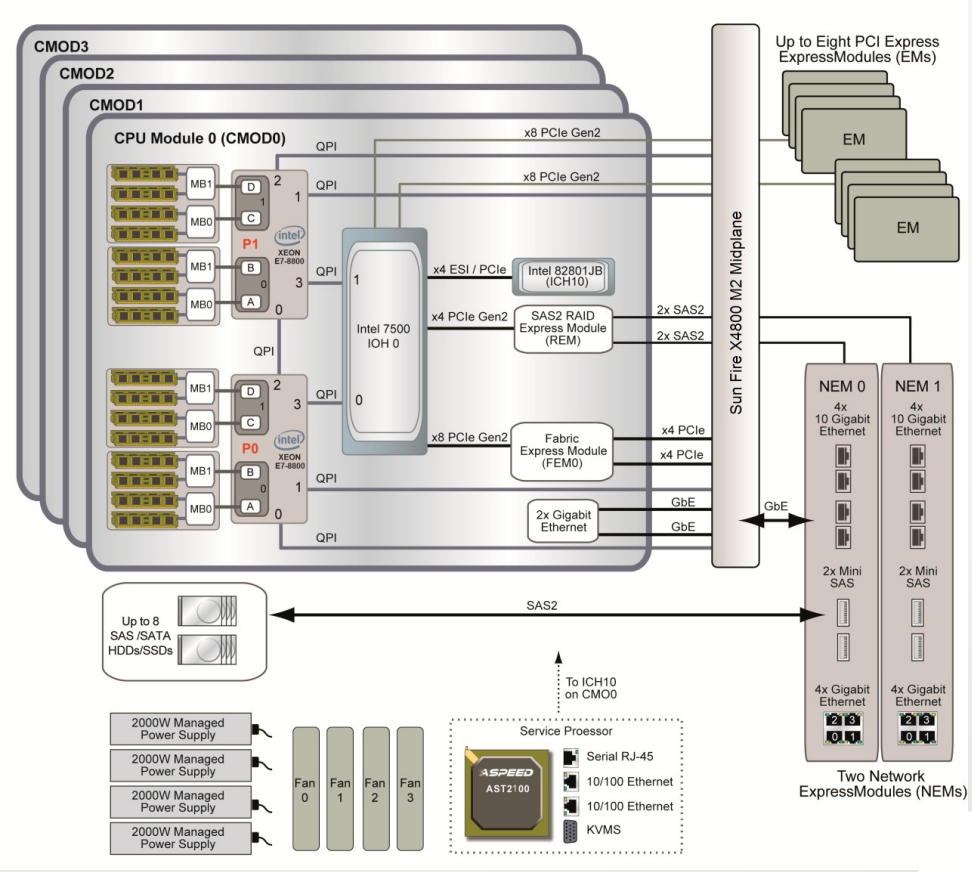
- RedHat Enterprise Linux 6.4 (basically)
- OFED InfiniBand Stack
  - SRP daemon
- Device Mapper Multipath
  - Each block device seen through 8 paths (at least)
- Logical Volume Manager
  - O(10) block devices / RAID
- XFS Filesystem
  - Or ext4



NEC

Expect Trouble When Going To Limits!

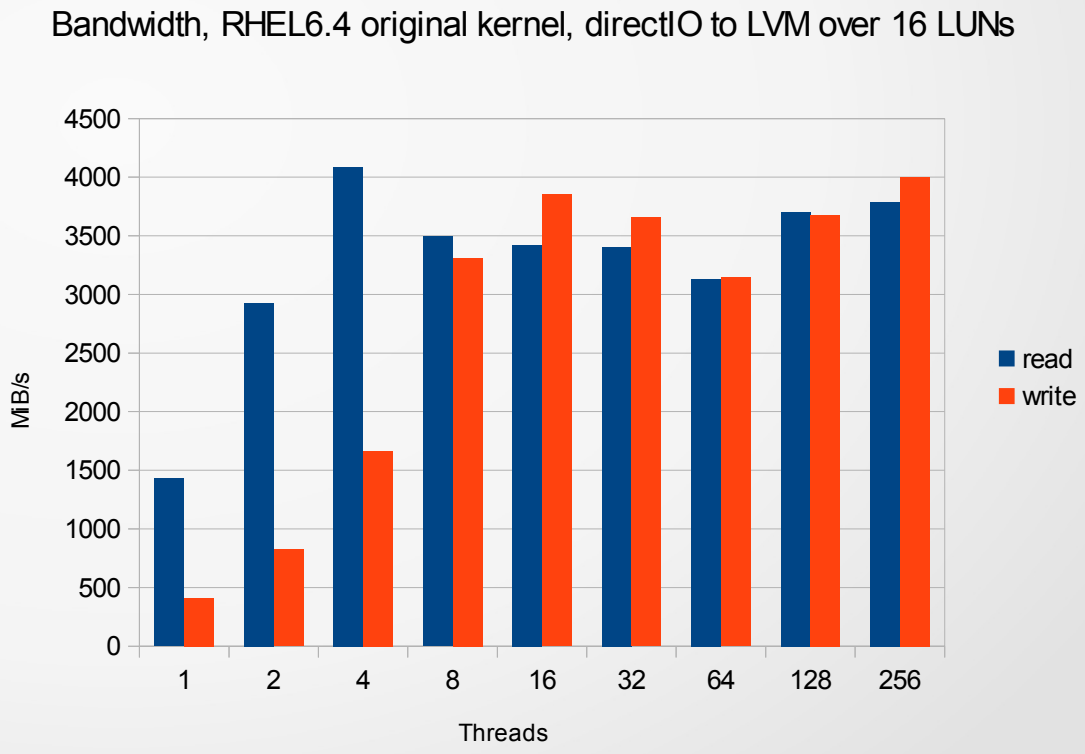
- Storage: 3 x (6-10GB/s)
- Server: 4-8 InfiniBand HCAs, QDR (max 3.2GB/s each)
- PCIe is limiting...



NEC

LVM Scales ... Not

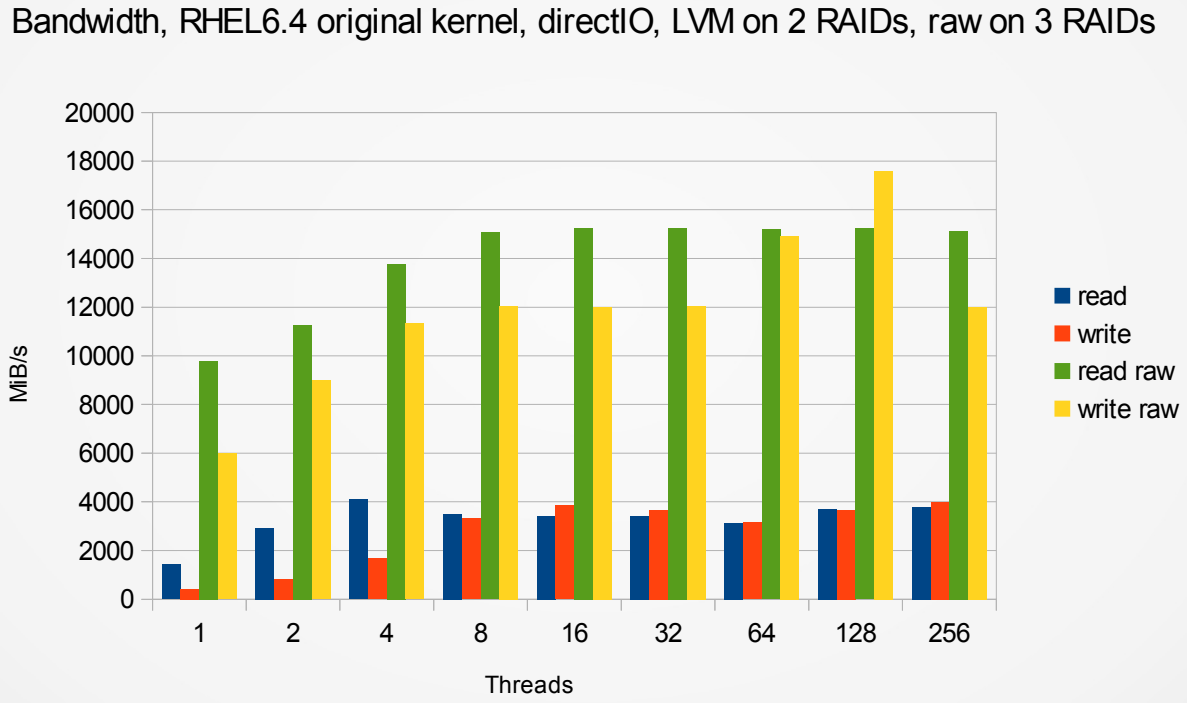
- Surprise: LVM doesn't scale!
  - Huge system time
  - Kernel wastes time
    > 85% spent in:
    dm_table_put
    dm_table_get
    dm_table_find_target
    _read_lock_irqsave
    _spin_lock_irqsave



NEC

I/O to raw devices

- Read / write directly to block devices, no LVM



  - Ok, we speak of huge bandwidths. To Disk!
  - Compare with STREAM a few years ago

2011.06.28	Intel_Core_i7-2600	4	12235.0	12294.0	13780.0	13779.0

NEC

LVM Patch Get's Us Closer

- Replace dm table locking by SRCU
  - Was „out“ since mid 2012
  - Went into mainline with Linux kernel 3.11 (September 2013)
  - Working with RedHat to get into RHEL6.6
- Results: LVM, 3 RAIDs, 24 LUNs
  - Read: reaches 15GB/s
  - Write: tops at 10-11GB/s
- New limitation in write path unveiled:

```
10,54%  [dm_mod]          [k] dm_table_unplug_all
 5,74%  [kernel]          [k] put_page
 5,66%  [kernel]          [k] blk_unplug
 5,31%  [kernel]          [k] get_page
 4,07%  [dm_mod]          [k] stripe_map_sector
```


NEC

I/O To Local Filesystem

- XFS and ext4 behave quite similarly
  - For the multi-threaded iozone tests we did
- Setup: XFS on LVM on 24 LUNs from 3 RAIDs
- Results:
  - Read: up to 12.5GB/s
  - Write: up to 10GB/s
- Tuning:
  - The usual: align for RAIDs stripe line size (1MB)
  - Surprise: hugepages are not useful!
    - At these performance levels compound huge pages are expensive

Conclusion

- New, ultra-fast storage devices reveal performance bottlenecks where we didn't see them before
  - And help reach new performance levels!
- Lustre parallel filesystem performance requires careful consideration of all layers, including InfiniBand in odd ways.
- O(10GB/s) for local filesystem I/O is doable, but challenging
  - Works for DAS and SAN
- Lessons learned for rotating storage apply for SSDs!
  - Only nobody could afford buying me enough of them ;-)