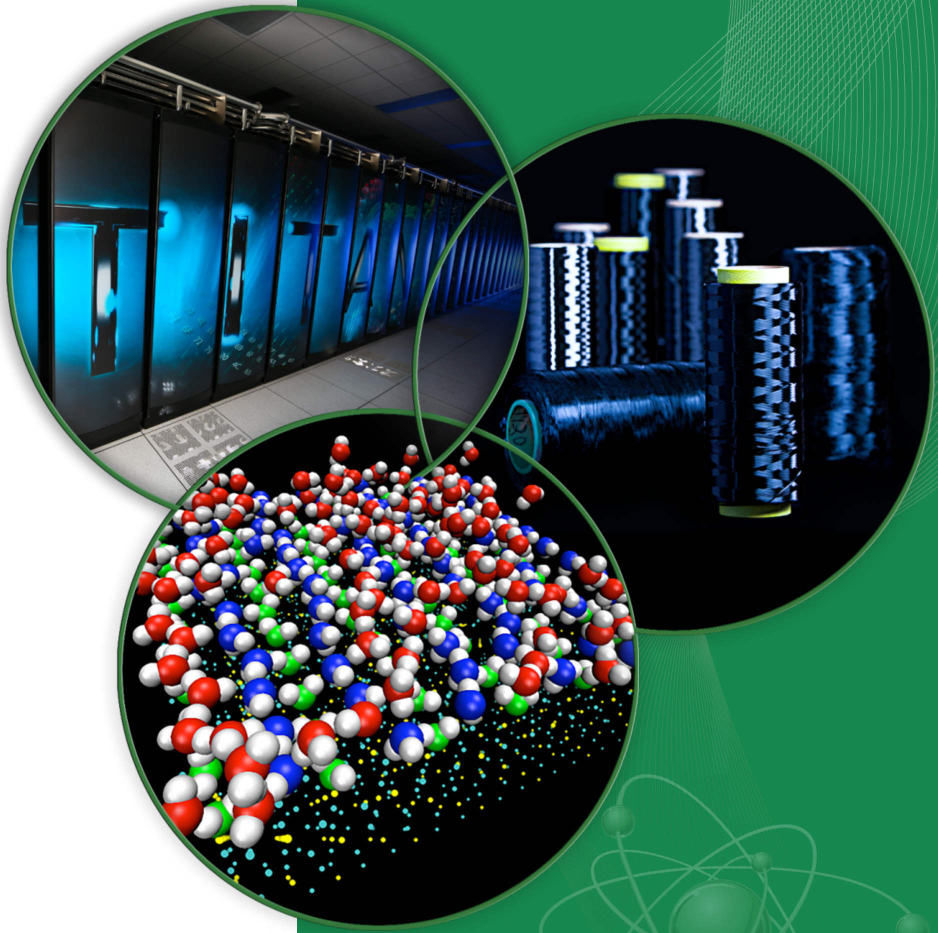
Improving Block Level Efficiency

with **scsi-mq**



Blake Caldwell
NCCS/ORNL

March 4th, 2015

ORNL is managed by UT-Battelle
for the US Department of Energy

OAK RIDGE National Laboratory | OAK RIDGE LEADERSHIP COMPUTING FACILITY

Block Layer Problems

- Global lock of request queue per block device
- Cache coherency traffic
  - If servicing part of a request on multiple cores, the lock must be obtained on the new core and invalidated on the old core
- Interrupt locality
  - Hardware interrupt may occur on wrong core, requiring sending soft-interrupt to proper core

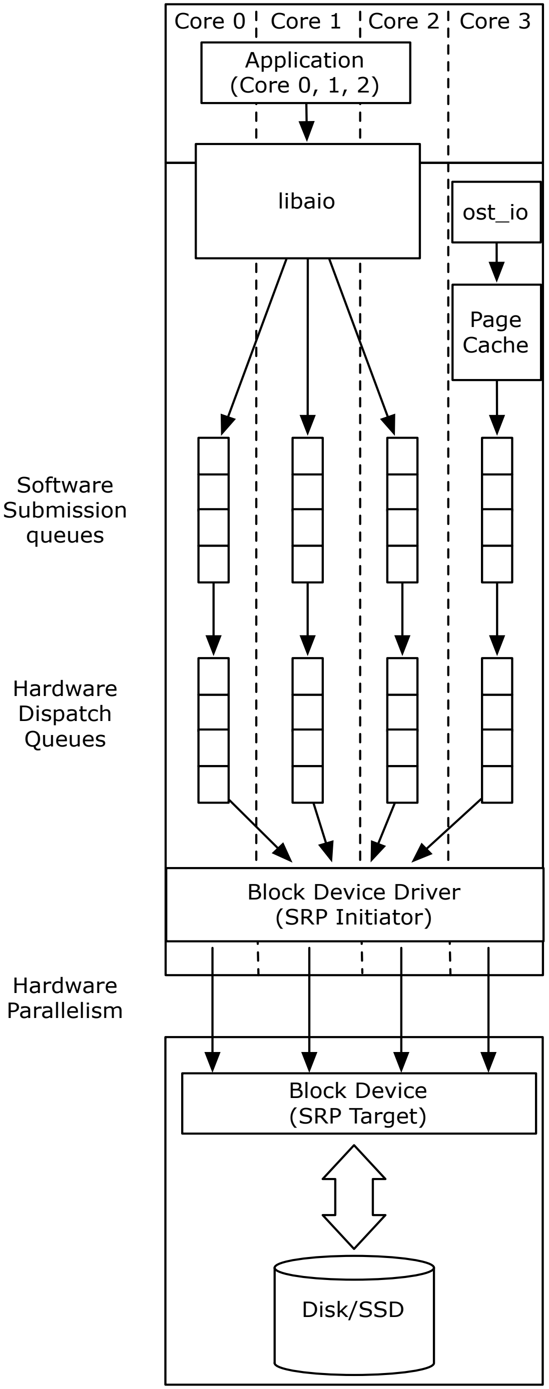
Linux Block Layer

- Designed with rotational media in mind
  - Time spent in the queue allows sequential request reordering – a very ***good*** thing
  - Completion latencies 10ms to 100ms
- Single request queue
  - Staging area for merging, reordering, scheduling
- Drivers are presented with the same interface for each block device

blk-mq (multi-queue)

- Rewrite of the Linux block layer (since kernel 3.13)
- Two levels of queues
  1) Per-core submission queues
  2) 1 more more hardware dispatch queues with affinity to NUMA nodes/CPU's (device-driver specific)
- IO scheduling within software queues
  – Inserted in FIFO order, then interleaved to hardware queues
- Tags IOs that are reused for lookup on completion

blk-mq



Core 0 Core 1 Core 2 Core 3

Application
(Core 0, 1, 2)

libaio

ost_io

Page
Cache

Software
Submission
queues

Hardware
Dispatch
Queues

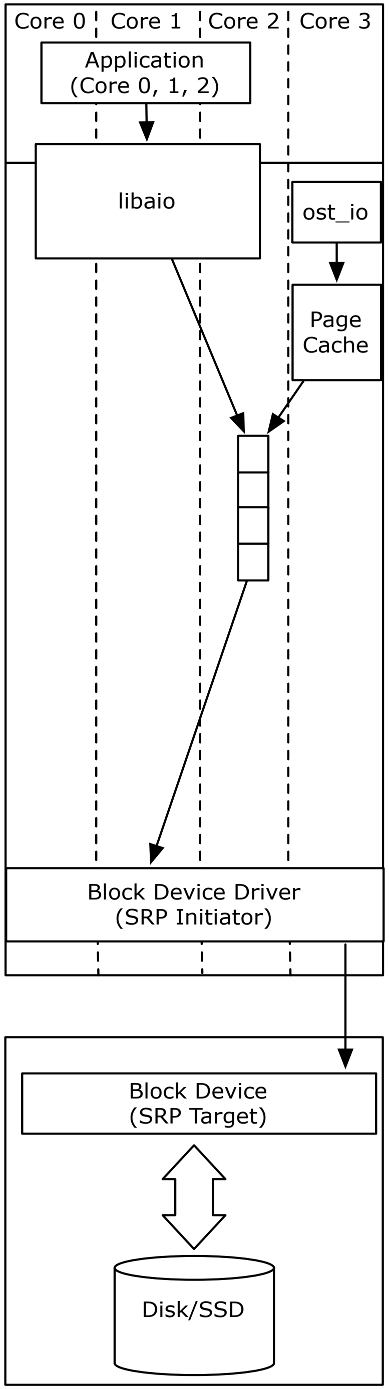
Block Device Driver
(SRP Initiator)

Hardware
Parallelism

Block Device
(SRP Target)

Disk/SSD

single queue



Core 0 Core 1 Core 2 Core 3

Application
(Core 0, 1, 2)

libaio

ost_io

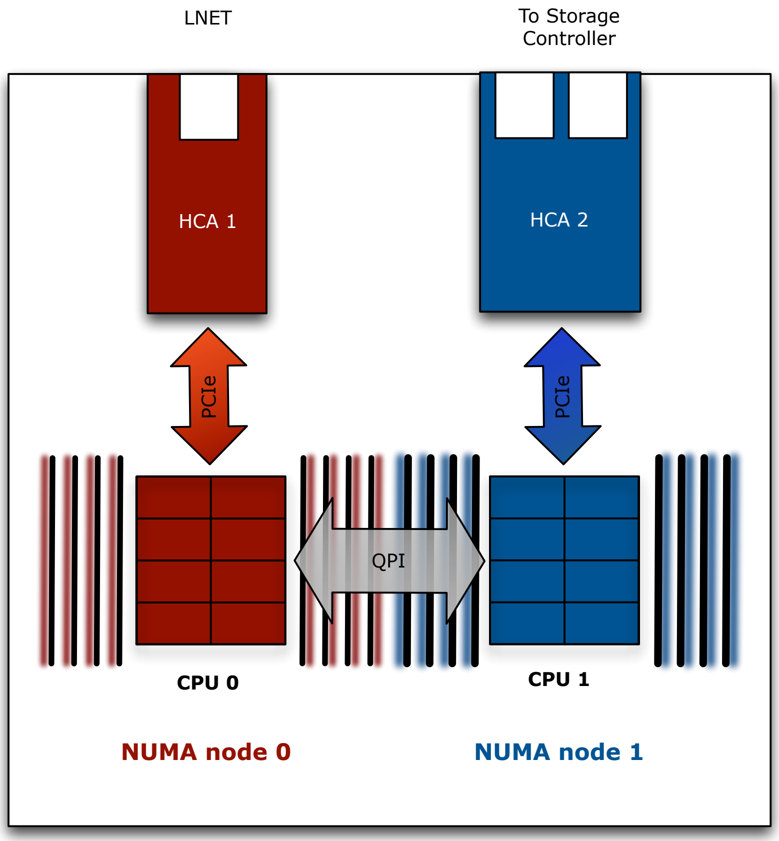
Page
Cache

Block Device Driver
(SRP Initiator)

Block Device
(SRP Target)

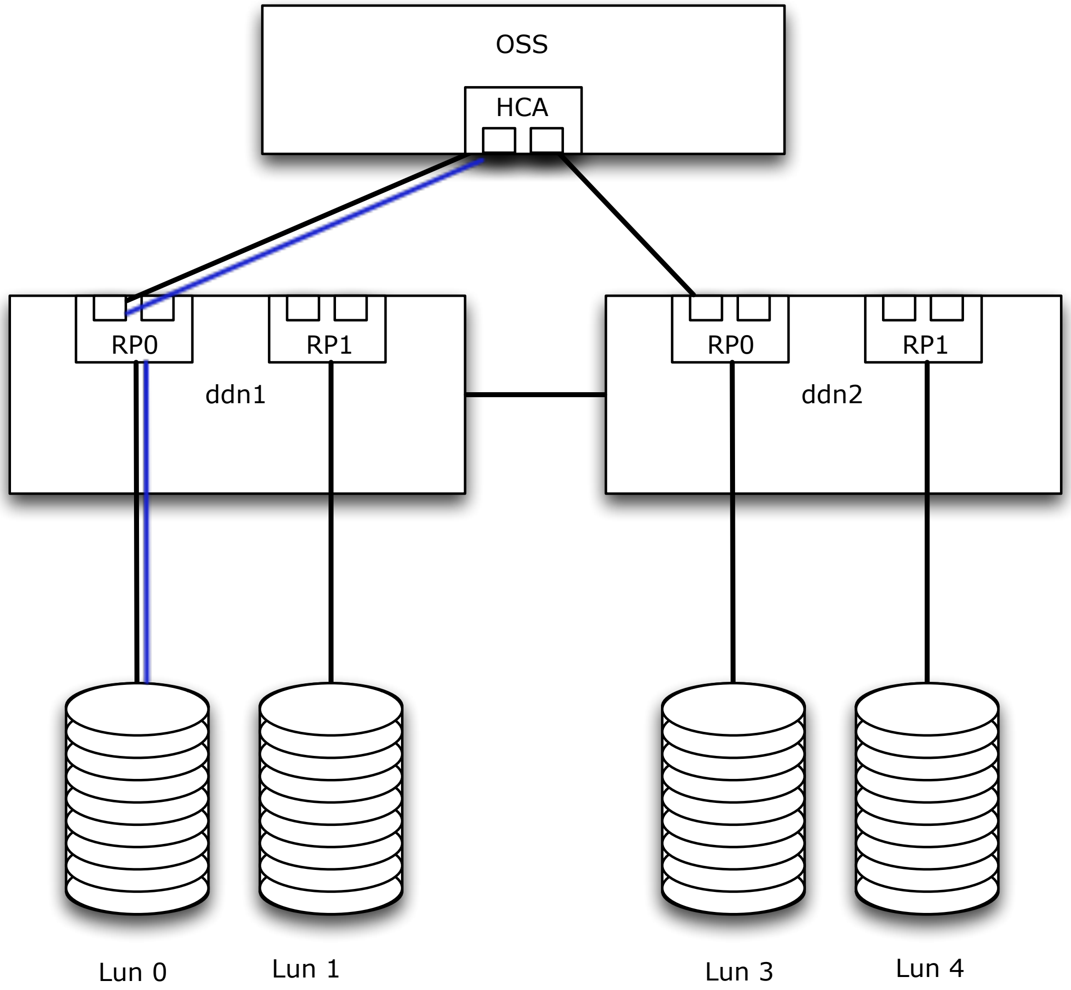
Disk/SSD

IO Device Affinity

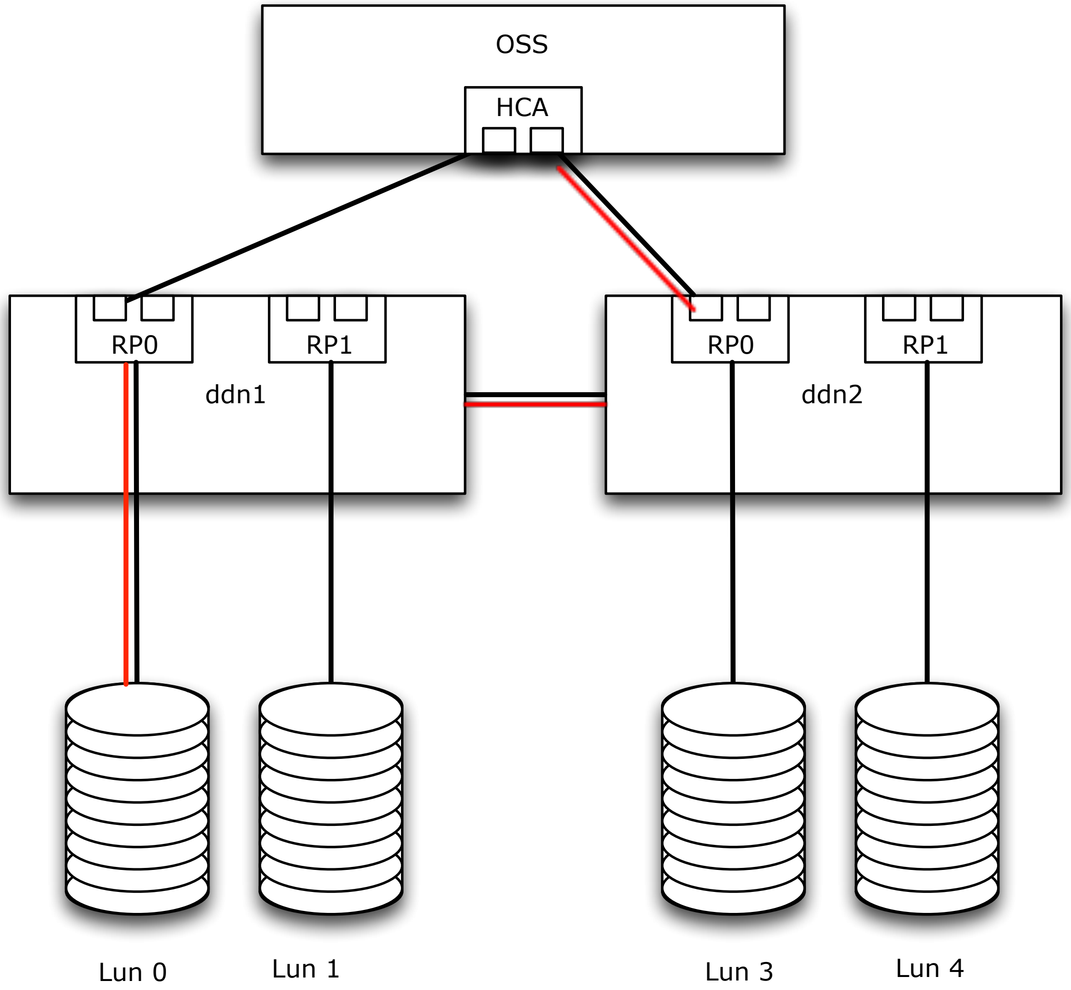


OSS Node

Controller Caching (direct)



Controller Caching (indirect)



Evaluation Setup

- Linux 3.18
  - blk-mq (3.13)
  - scsi-mq (3.17)
  - ib-srp multichannel (3.18)
  - dm-multipath support (4.0)
- Lustre 2.7.0 rc1
  - ldiskfs patches rebased to 3.18

- OSS
  - Dual Ivy Bridge E5-2650 (2 NUMA nodes)
  - 64GB
  - Dual-port QDR IB to array
- Storage Array
  - Dual DDN 10k controllers
  - 8GB write-back cache
  - RAID 6 (8+2) LUNs
  - SATA disk

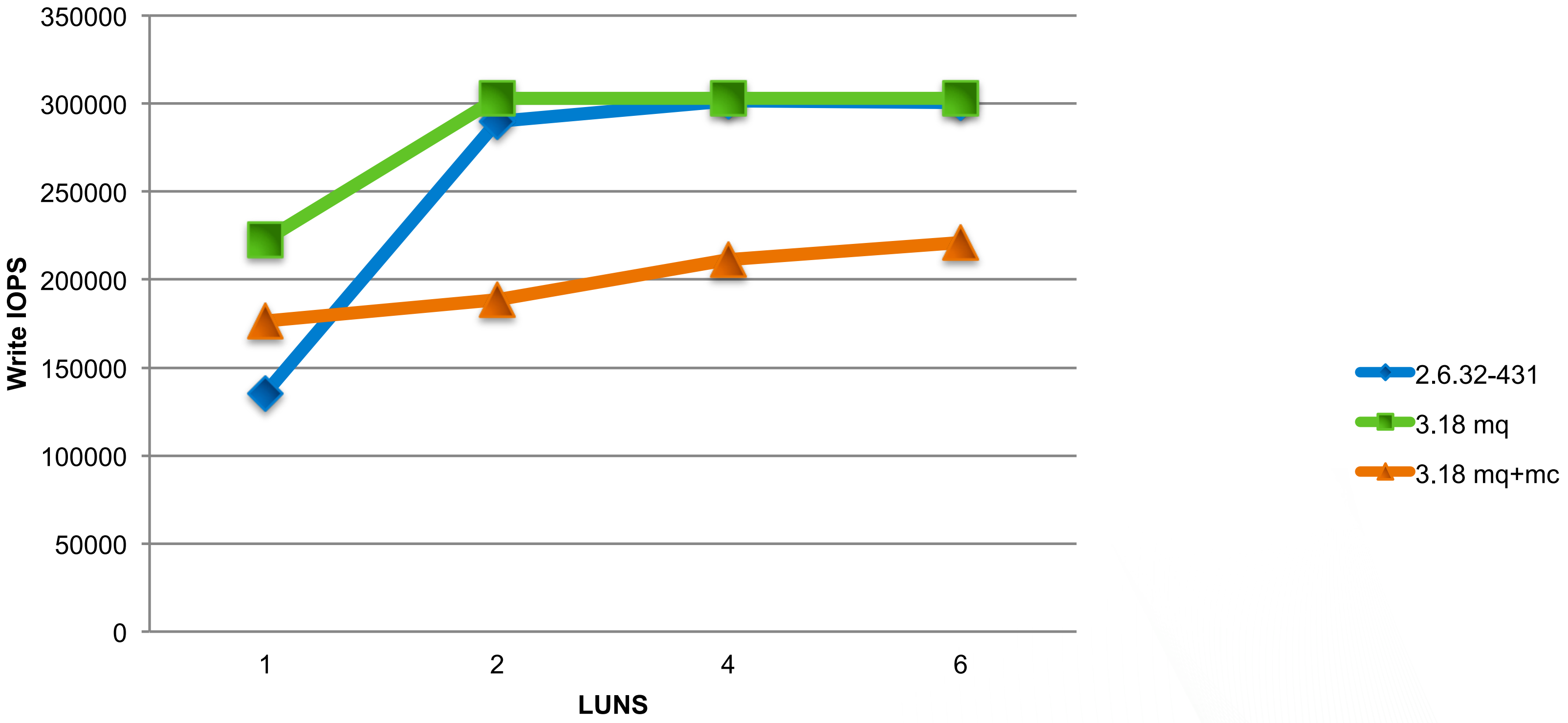
Evaluation Goals

- Block level testing: adapt tests done with null-blk device to a real storage device with scsi-mq
  - Does NUMA affinity awareness lead to efficiency
    - Increased bandwidth
    - Decreased request latency
- Multipath performance
- Explore benefits for filesystems
  - Ready for use with fabric attached storage?
  - Are there performance benefits?

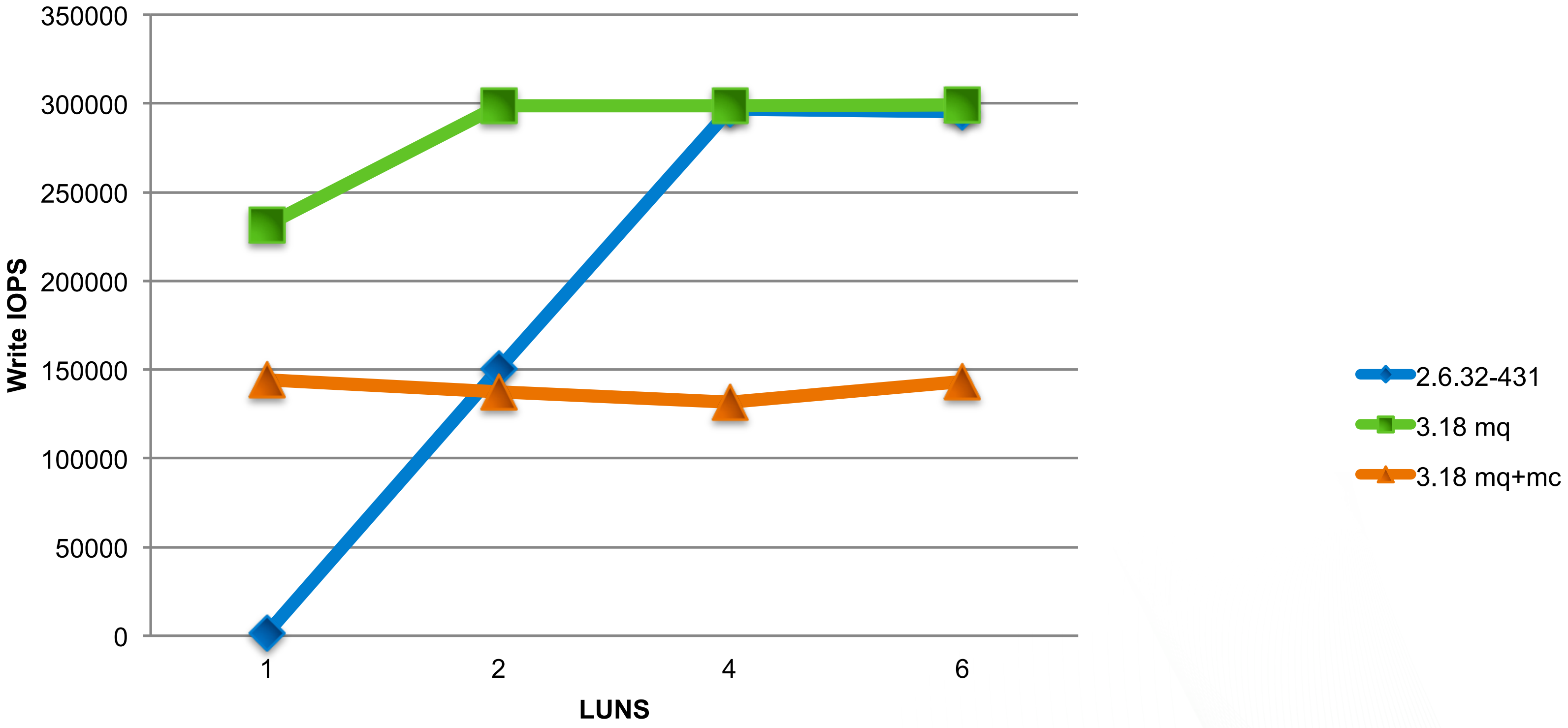
Evaluation Parameters

- Affinity
  - IB MSI-x interrupts spread among cores on NUMA node 1
  - Fio threads bound to NUMA node 1 (closest to HCA)
  - Block device rq_affinity=2 – completion happens on submitting core
- Block tuning
  - Noop scheduler
  - max_sectors_kb = max_hw_sectors_kb
- IB-srp module
  - 16 channels
  - Max_sect 8192
  - Max_cmd_per_lun 62
  - Queue size 127 (fixed by hardware)

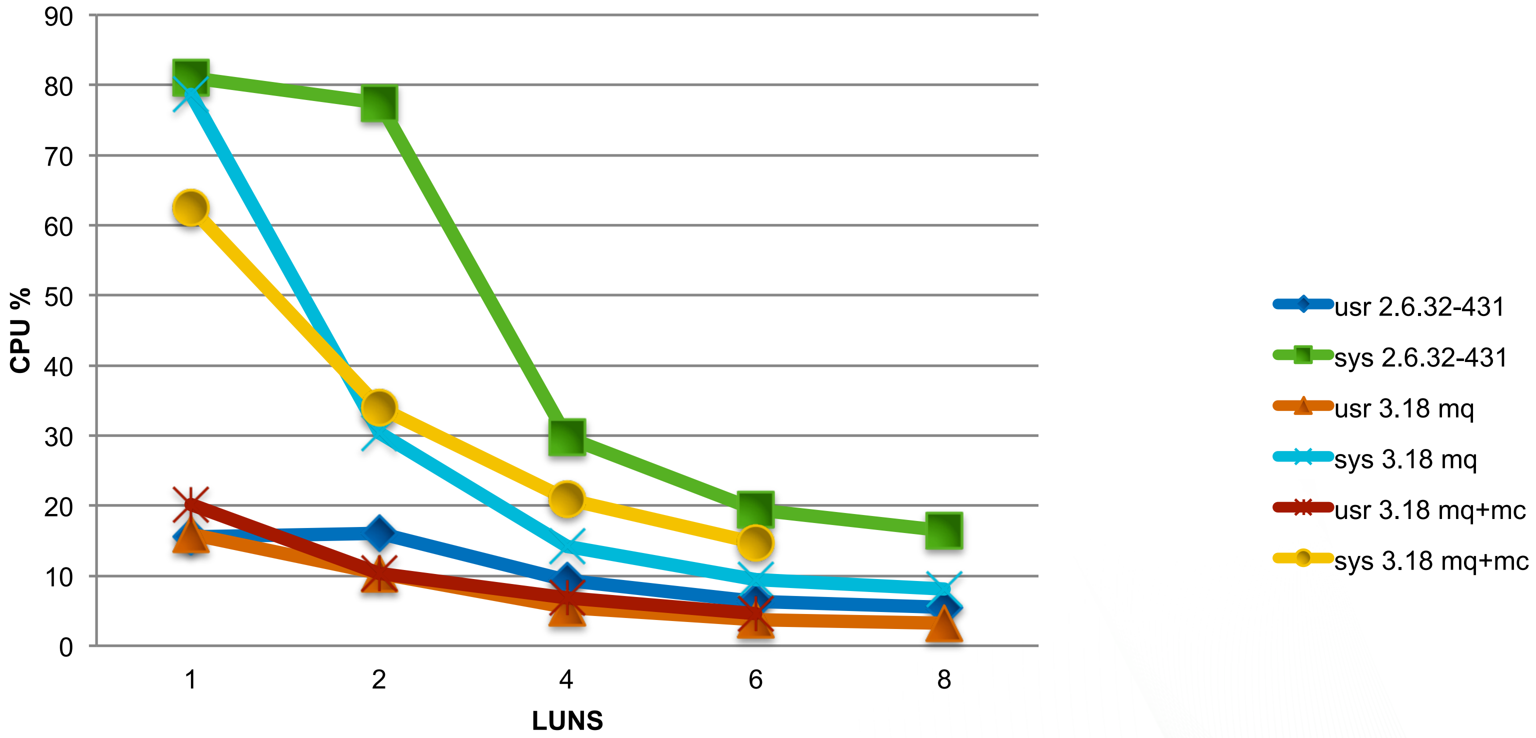
Throughput (direct path) thread/LUN



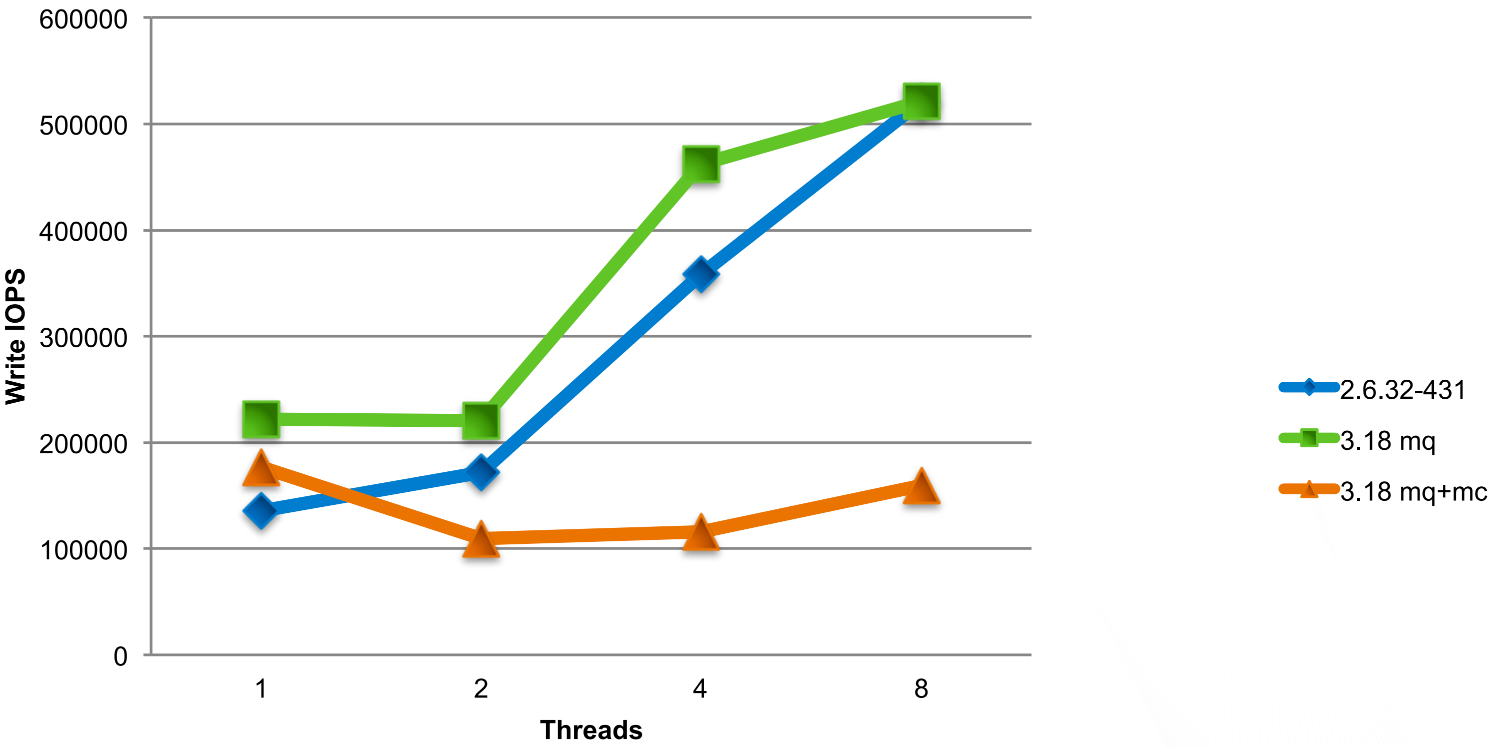
Throughput (indirect path) thread/LUN



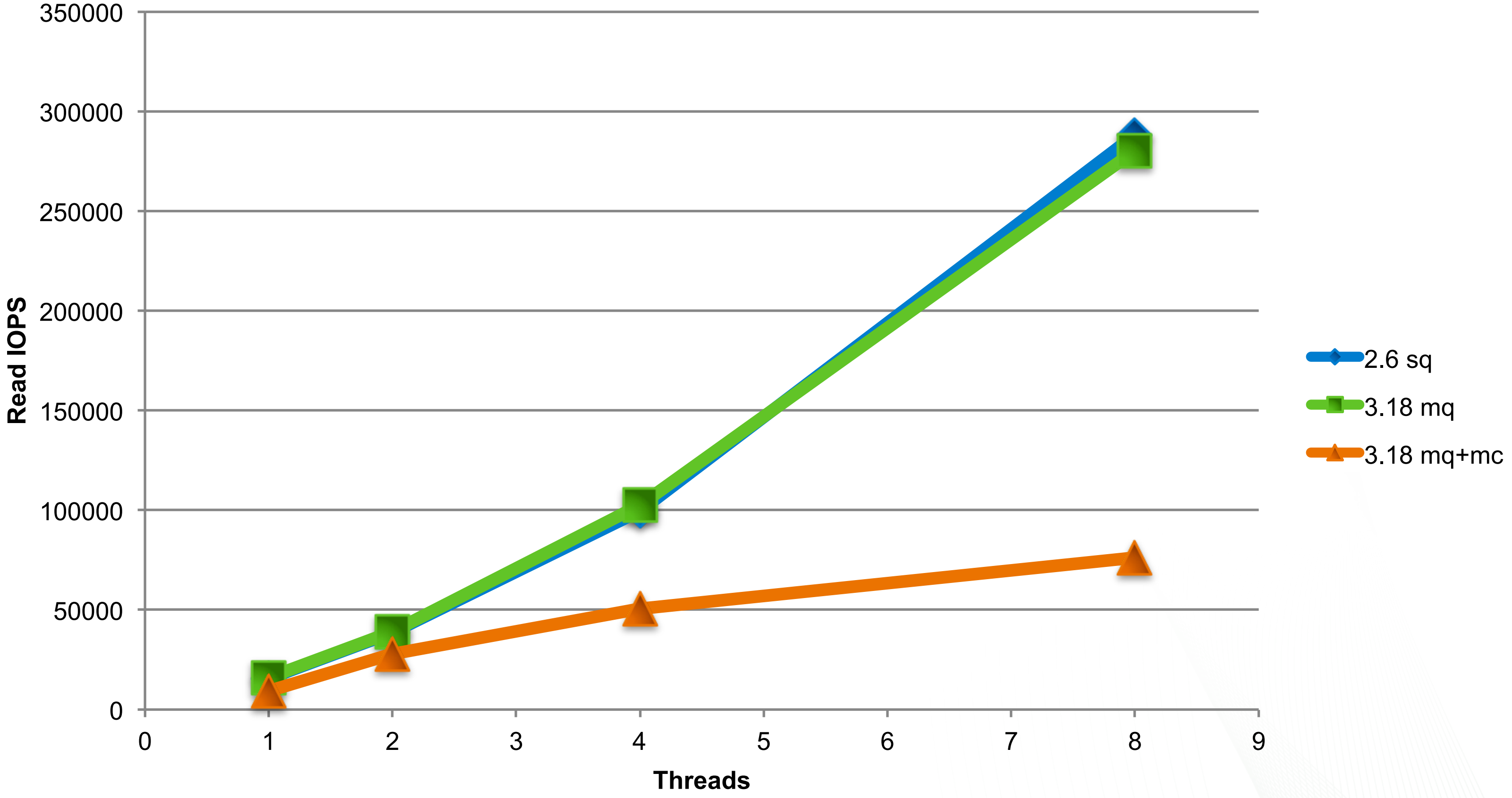
CPU Usage (direct path) thread/LUN



Throughput (direct path) 1 LUN



Read Throughput (direct path) 1LUN



dm-multipath support

- Evaluated on Linux 4.0rc1 with patches for dm blk-mq support
  - 4k IO, libaio, iodepth=127, SRP multi-channel support enabled

Multipath	Direct (no IO FWD)	indirect (IO FWD)
393.2 MB/s	849.5 MB/s	854.8 MB/s
100658 IOPs	217468 IOPs	218829 IOPs

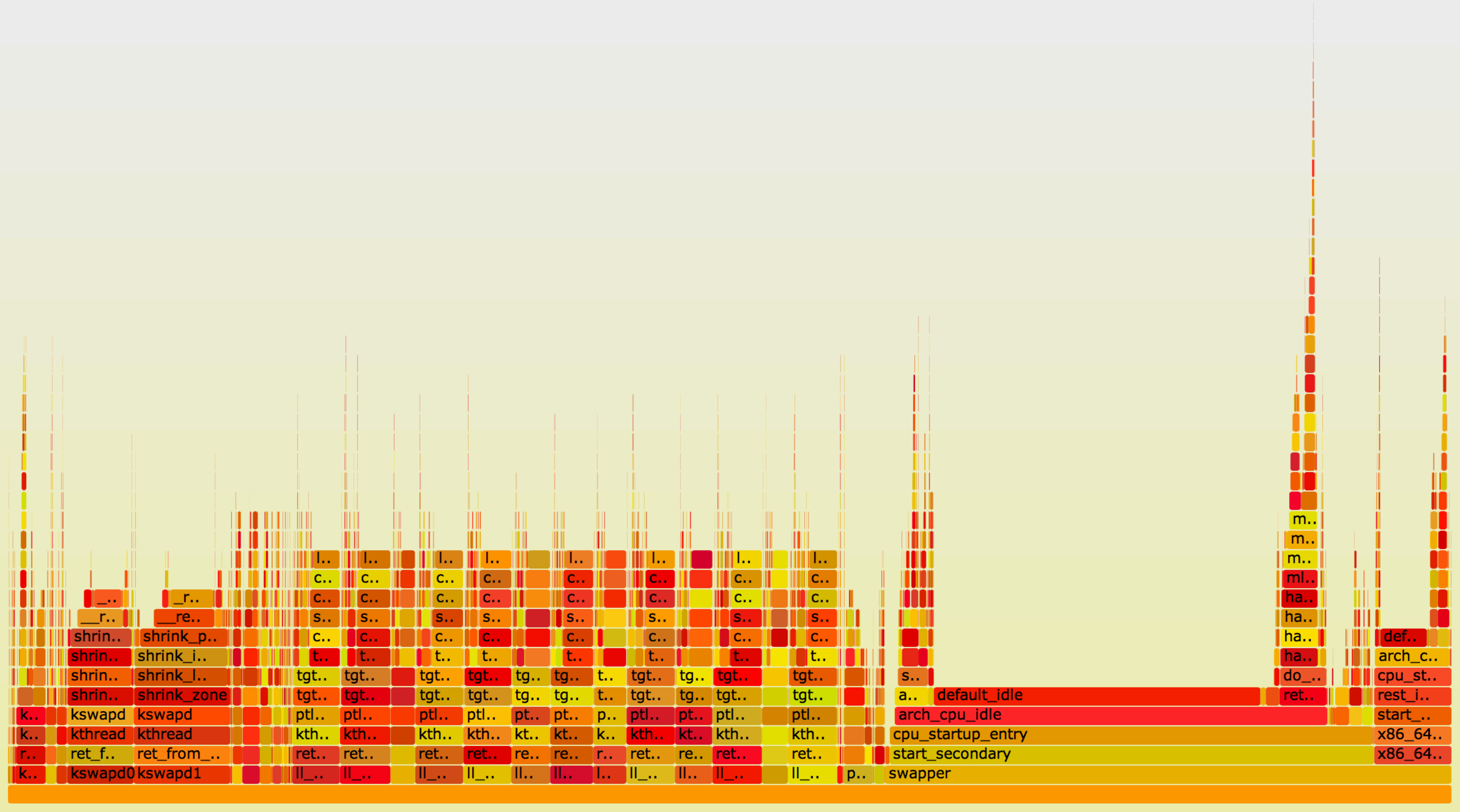
Request latency

- Average for 100 4k IOs, fio with synchronous IO engine

kernel	Multipath	Direct (no IO FWD)	Indirect (IO FWD)
2.6.32-431.17.1	130.50	119.92	169.22
4.0rc1 (**blk-mq**)	130.56	103.58	142.84
Improvement	-0.04%	13.6%	15.6%

Lustre Profiling

- FlameGraph of kernel code using Perf, 100Hz, Linux 3.18, Lustre 2.7.0rc1, 1MB writes to a single OST



Lustre Applications

- Metadata IO
  - Improve single request latency
  - Is bandwidth necessary during flushing metadata to MDT?
- Object IO
  - Scheduling
    - Request size
    - Request tagging

Future Directions

- Lustre with Linux 4.0
- Testing with hardware capable of 600k+ 4k IOPs
  - Random write performance for multiple thread/LUN
- Evaluate multiple threads/LUN sequential writes
- Read and random tests needs further investigation

Conclusion

- scsi-mq has potential to lower CPU usage even with rotational media
- scsi-mq has lower IO completion latency
- Further evaluation needed of device drivers that support multiple hardware dispatch queues

Thank You

Blake Caldwell <blakec@ornl.gov>