SanDisk®

Increasing SCSI LLD Driver Performance by Using the SCSI Multiqueue Approach

Bart Van Assche, Ph.D.

March 17, 2015

Overview

- Introduction
- SCSI Architecture Concepts
- Linux SCSI Initiator Stack
- Linux SCSI Initiator Scalability Issues
- SCSI Multiqueue Approach a.k.a. scsi-mq
- SCSI RDMA Protocol (SRP)
- Importance of MSI-X
- Multiqueue SRP initiator Performance Results

Introduction

- Today's SSDs and all-flash arrays support more than one million IOPS and sub-millisecond latency.
- Until recently the Linux block layer and SCSI core were a bottleneck for these fast storage devices.
- Hence the introduction of multiqueue support in the block layer core (blk-mq) and SCSI mid-layer (scsi-mq).
- Leveraging full multiqueue potential requires requires SCSI LLD driver modifications.
- Results will be shown for the InfiniBand SRP initiator driver.

About myself

- Linux kernel InfiniBand SRP initiator maintainer.
- SCST co-maintainer.
- Member of the SanDisk ION team.
- ION = all-flash array.
- In our performance tests we noticed that there was a bottleneck at the initiator side.

SCSI Architecture Concepts

- SCSI command: READ, WRITE, REPORT LUNS, INQUIRY, ...
- Transport protocol: e.g. FC, iSCSI, iSER, SRP.
- LUN = Logical Unit Number.
- Initiator system: submits SCSI commands.
- Target system: processes SCSI commands.

Linux SCSI Initiator Stack

□Upper level drivers: sd (disk), sr (CD-ROM), st (tape), …

□Mid level: SCSI command processing; error handling; interface between UL and LL drivers.

□Lower level drivers: SCSI transport protocol implementation + HBA driver. Examples: FC, iSCSI, iSER and SRP initiator drivers.

Linux SCSI Initiator Command Processing

□Mid-level submits SCSI command to LLD via queuecommand().

□LLD submits command to HCA.

□LLD receives command completion from HCA via interrupt or via polling.

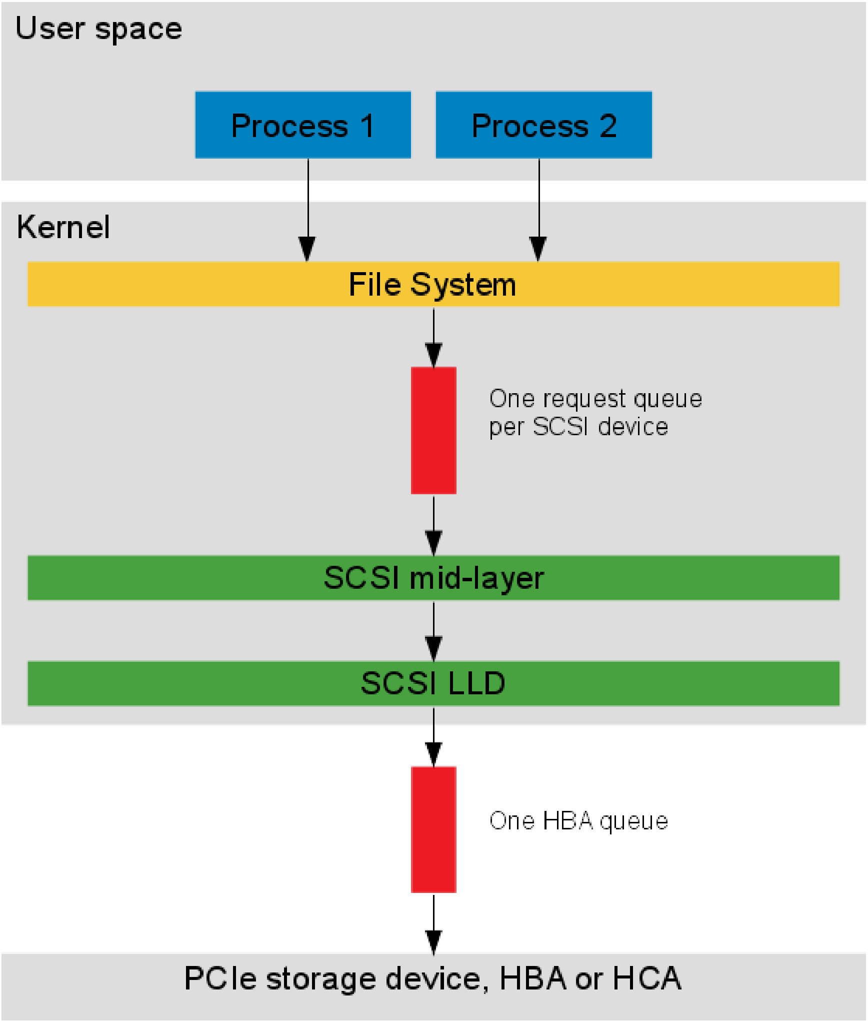
□LLD reports command completion via cmd->scsi_done().

Linux SCSI Initiator Scalability Issues

- At most 400.000 IOPS per LUN.
- Lock contention in mid-layer.
- Previous attempts to use polling resulted in limited performance improvements (about 5%).
- Interrupt coalescing increases latency too much.
- Hence the limitation of the SCSI command processing rate to about the speed at which a single CPU can process interrupts.

SCSI Single Queue Approach

One SCSI command queue per SCSI host shared by all CPU cores.

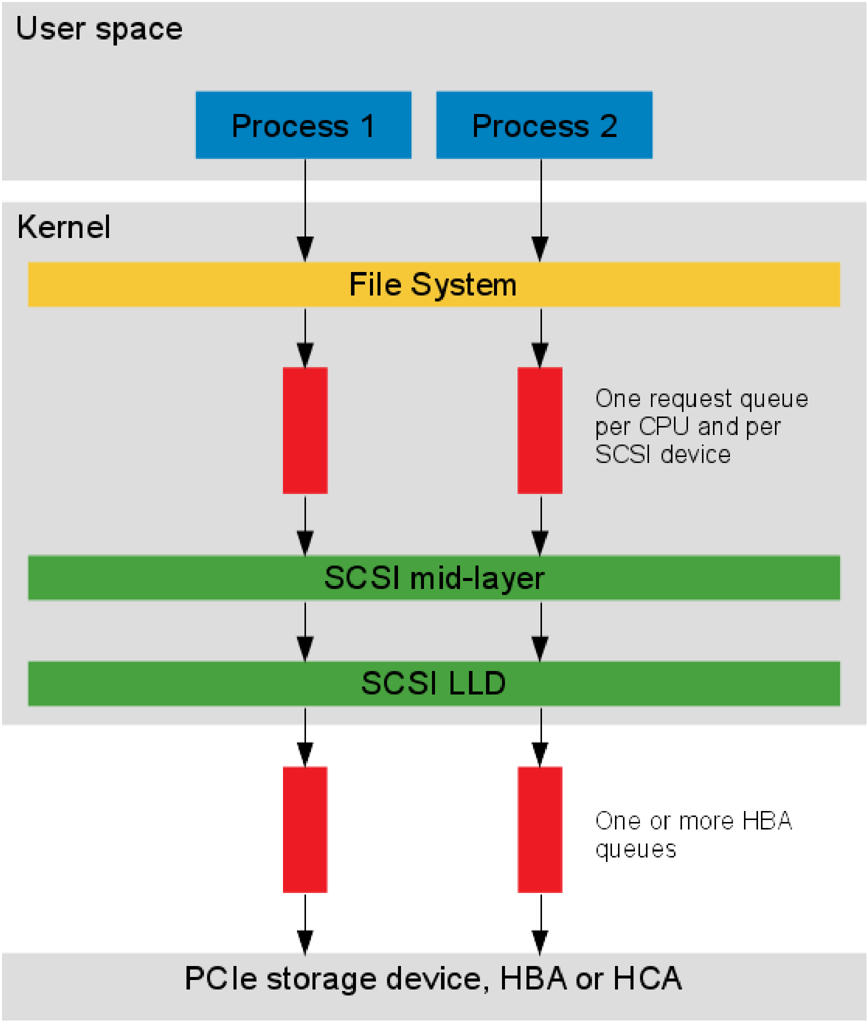


SCSI Multiqueue Approach a.k.a. scsi-mq

□One SCSI command queue per SCSI host and per CPU core.

□Number of queues between LLD and HBA depends on LLD implementation.

□Note: Linux SCSI initiator stack does not guarantee that SCSI commands submission order is preserved.



SCSI RDMA Protocol (SRP)

□Allows one computer to access SCSI devices attached to another computer via remote direct memory access (RDMA).

□Advantages of RDMA are low latency, low CPU utilization and high bandwidth.

□ANSI T10 SRP specification defines how to use multiple RDMA channels for a single SRP session.

□ib_srp kernel driver implements SRP over InfiniBand.

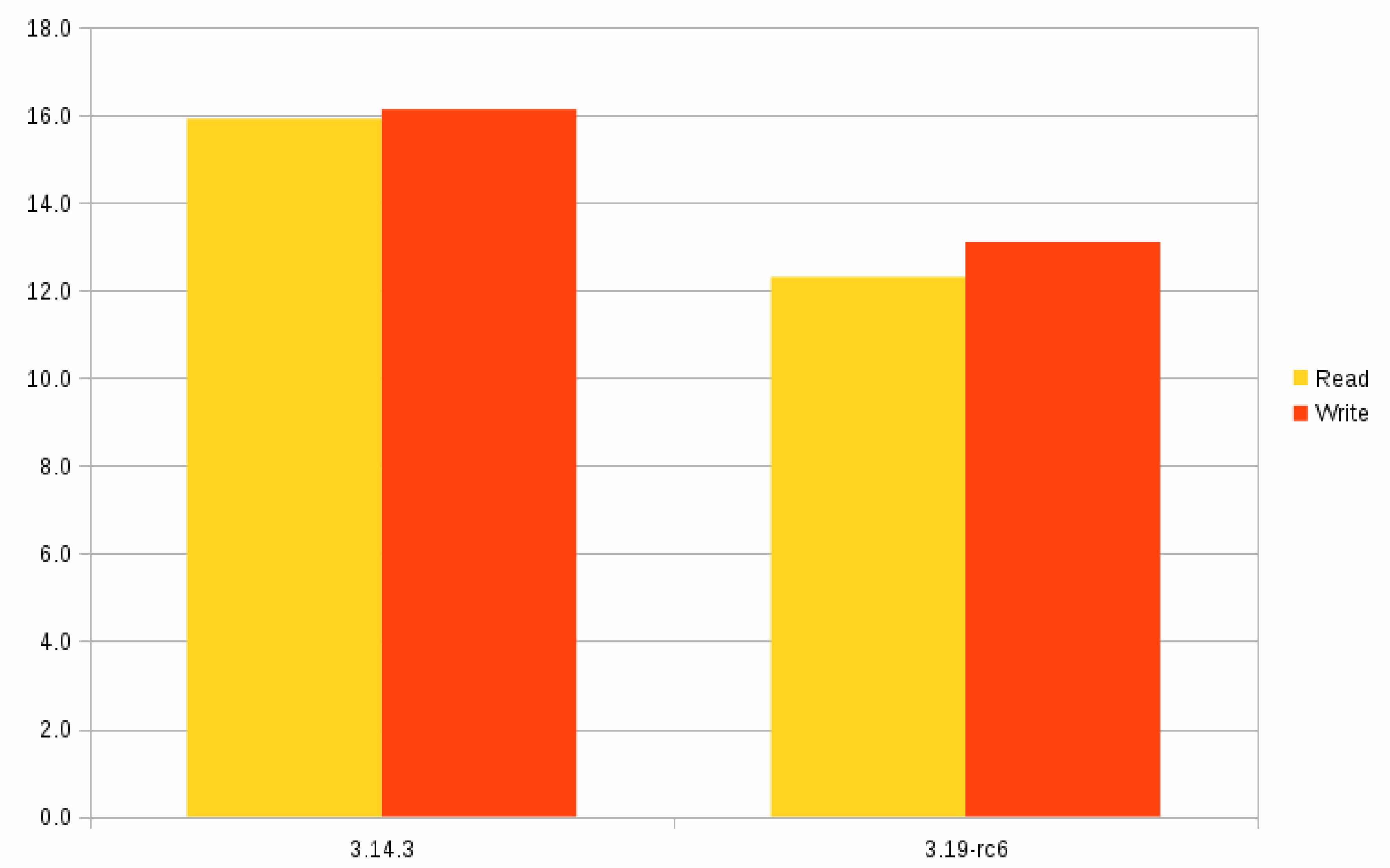
Multiqueue SRP initiator

- Available in Linux kernel 3.19 (February 2015).
- Supports scsi-mq:
  - set SCSI_MQ_DEFAULT=y in kernel config
  - - or -
  - echo Y > /sys/module/scsi_mod/parameters/use_blk_mq
- Configurable number of RDMA channels:
  - echo options ib_srp ch_count=$n > /etc/modprobe.d/ib_srp.conf
- Performance depends on number of MSI-X vectors supported by RDMA HCA.
- Test setup: RDMA HCAs with eight MSI-X vectors.

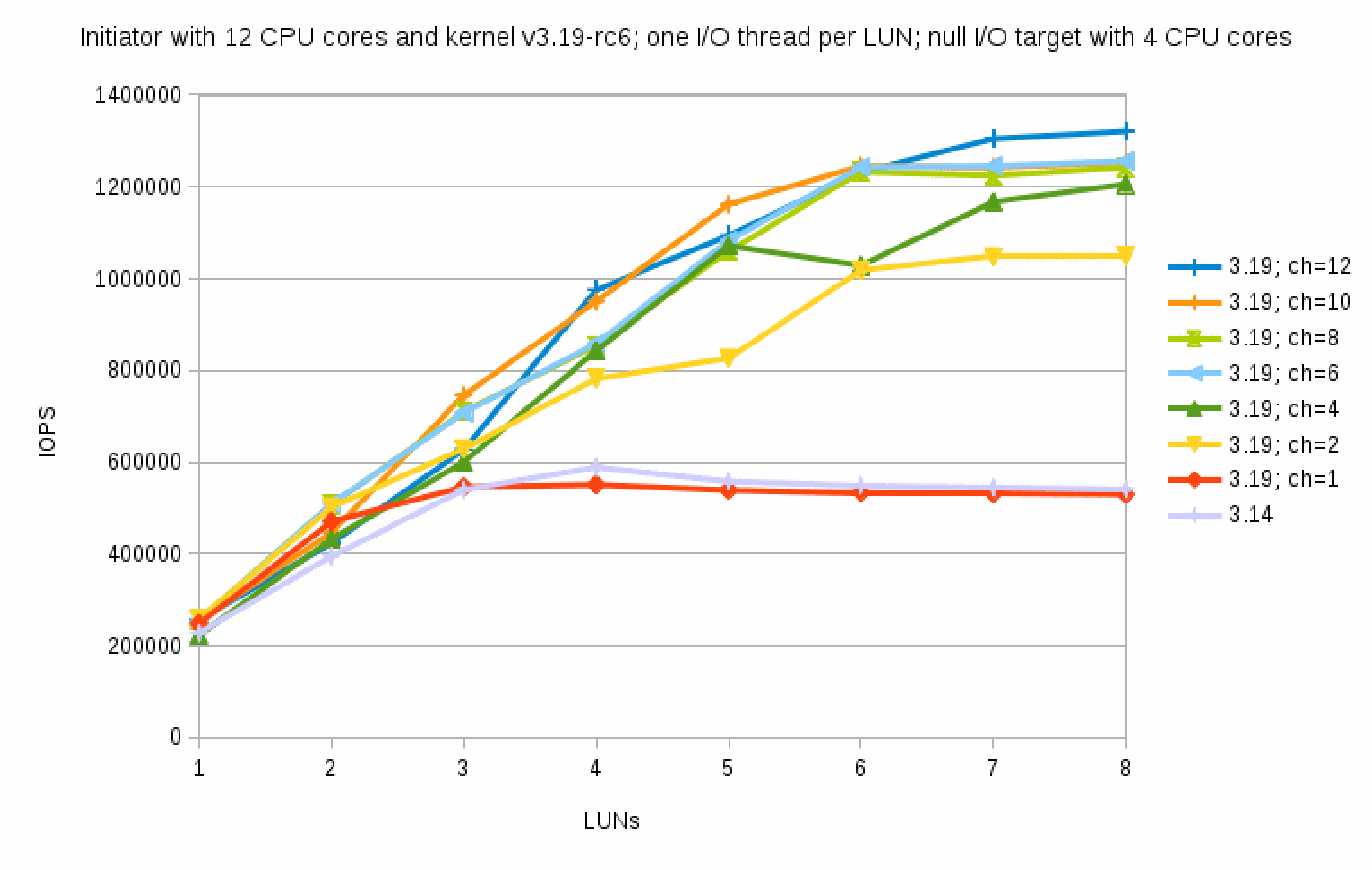
Multiqueue and NUMA Systems

- Achieving optimal performance on NUMA systems means constraining communication between CPU sockets.
- Hence, process each I/O completion on the CPU socket that submitted the I/O.
- Setting rq_affinity=2 helps but is not sufficient. MSI-X interrupt must be processed by CPU that submitted I/O request.
- Requires knowledge of which MSI-X interrupt is associated with which CPU core: /proc/irq/$n/smp_affinity.
- SRP initiator driver assumes that MSI-X vectors are spread uniformly over CPU sockets.
- E.g. MSI-X vectors 0-3 are associated with first CPU socket and vectors 4-7 are associated with second CPU socket.
- SRP initiator driver selects MSI-X interrupt via RDMA the RDMA API – last argument of ib_create_cq() is MSI-X completion vector index.

Latency Comparison (µs)



IOPS Performance for 50/50 R/W Workload



Performance Conclusions

- Scsi-mq approach results in a significant latency reduction.
- Kernel 3.14+sq / 3.19+mq+ch=1 results illustrate lock contention: IOPS decrease for increasing number of LUNs.
- Single channel (ch=1) scsi-mq performance better than that of kernel 3.14.3 for #LUNs <= 2.
- Initiator CPU usage was 100% for <= 4 LUNs and below 100% for > 4 LUNs due to target system saturation.
- With multiple channels almost linear scalability of IOPS in terms of LUNs (for #LUNs >= 4).
- Multiple channels more than doubles maximum IOPS.
- Note: CPU cores that ran I/O also processed IB interrupts.

Linux kernel 3.15

- Several SCSI mid-layer optimizations were merged in kernel 3.15.
- Optimizations apply to both traditional and multiqueue LLDs.
- New field in **struct scsi_host_template**, namely **cmd_size**.
- Allows drivers to specify size of per-command private data.
- Makes SCSI core perform a single allocation for core + LLD per-command data instead of a separate allocation by the SCSI core and another allocation by the LLD.
- See also James Bottomley, *First round of SCSI updates for the 3.15 merge window*, April 2014 (https://lkml.org/lkml/2014/4/1/441).

Linux kernel 3.17

- A second series of optimizations and scsi-mq support were merged in kernel 3.17.
- The only way to enable scsi-mq with kernel 3.17 is as follows:
- echo Y > /sys/module/scsi_mod/parameters/use_blk_mq
- See also James Bottomley, *First round of SCSI updates for the 3.17 merge window*, August 2014
- (https://lkml.org/lkml/2014/8/6/378).

Linux kernel 3.18

- The CONFIG_SCSI_MQ_DEFAULT kernel configuration option was merged in kernel 3.18.
- See also James Bottomley, *First round of SCSI updates for the 3.18 merge window*, October 2014
- (https://lkml.org/lkml/2014/10/7/839).

Linux kernel 3.19

- New field in struct scsi_host_template: use_blk_tags.
  - Allows to use scsi-mq style tags even with scsi-mq disabled.
  - Allows to use the same LLD code with and without scsi-mq.
- Support for multiple hardware queues was added to scsi-mq.
  - New functions for querying hardware queue index and tag from inside SCSI LLD:
  - u32 hwq_and_tag = blk_mq_unique_tag(scmnd->request);
  - u16 hwq = blk_mq_unique_tag_to_hwq(hwq_and_tag);
  - u16 tag = blk_mq_unique_tag_to_tag(hwq_and_tag);
  - These functions also work with scsi-mq disabled.
- scsi-mq support was added in a SCSI LLD, namely the SRP initiator driver.
- See also James Bottomley, *First round of SCSI updates for the 3.19 merge window*, December 2014
- (https://lkml.org/lkml/2014/12/8/585 / http://www.spinics.net/lists/linux-scsi/msg81290.html).

Future Work

- Integrating blk-mq support in the dm-multipath driver (Mike Snitzer and Keith Busch are working on this).
- Adding I/O scheduler support in the blk-mq layer.
- Adding scsi-mq support in the iSCSI initiator.
- Adding scsi-mq support in the FC initiator drivers.
- Automatic and scsi-mq aware IRQ affinity configuration, e.g. in irqbalanced or in the kernel.

Thanks to

- Christoph Hellwig for the hard work of implementing scsi-mq.
- Jens Axboe for the blk-mq changes and improvements needed for scsi-mq.
- Robert Elliott and Steve Cameron for helping with scsi-mq testing.
- Sagi Grimberg and Christoph Hellwig for reviewing the IB/SRP scsi-mq patches.
- Fusion-io/SanDisk for sponsoring Christoph's scsi-mq and blk-mq work and for allowing me to work on scsi-mq.

References

- Jonathan Corbet, *Interrupt mitigation in the block layer, LWN.net*, August 2009 (html).
- Matias Bjørling e.a., *Linux Block IO: Introducing Multi-queue SSD Access on Multi-core Systems*, 6th Systems and Storage Conference, ACM, June 2013 (pdf).
- Bart Van Assche, *Scsi-mq Performance Measurements*, Google Drive, June 2014 (pdf).
- Christoph Hellwig, *High Performance Storage with blk-mq and scsi-mq*, Linuxcon Europe, Oct 2014 (pdf).

Any questions or comments ?

SanDisk®