# Linux Multi-Queue Block IO Queueing Mechanism (blk-mq)

blk-mq (Multi-Queue Block IO Queueing Mechanism) is a new framework for the Linux block layer that was introduced with Linux Kernel 3.13 and has become feature-complete with Kernel 3.16.
Blk-mq allows for over 15 million IOPS with high-performance flash devices (e.g. PCIe SSDs) on multi-socket servers, though even single and dual socket servers also benefit considerably from blk-mq.
To use a device with blk-mq, the device must support the respective driver.

# Overview of blk-mq

Blk-mq integrates into the storage stack and provides basic functions to device drivers for mapping I/O enquiries to multiple queues.
The tasks are distributed across multiple threads and therefore to multiple CPU cores (per-core software queues).
Blk-mq compatible drivers request count for parallel hardware queues a device supports.
All device drivers that use the previous block I/O layer continue to work independently of blk-mq.

**blk-mq-based** device drivers bypass the previous Linux I/O scheduler.

# Device Drivers

| Driver | Device Name | Supported Devices |
|---|---|---|
| null_blk | /dev/nullb* | none (test drivers) |
| virtio-blk | /dev/vd* | Virtual guest drivers |
| mtip32xx | /dev/rssd* | Micron RealSSD PCIe |
| scsi (scsi_mq) | /dev/sd* | e.g. SAS and SATA SSDs/HDDs |
| NVMe | /dev/nvme* | e.g. Intel SSD DC P3600 DC P3700 Series |
| rbd | /dev/rdb* | RADOS Block Device (Ceph) |
| ubi/block | /dev/ubiblock* | |
| loop | /dev/loop* | Loopback-Device |
| dm / dm-mpath | | |

# Enable blk-mq

Enabling blk-mq must happen at boot time. You need to add **scsi_mod.use_blk_mq=1** to your lilo.conf

/etc/lilo.conf

```
# LILO configuration file
```

```
# generated by 'liloconfig'
#
# Start LILO global section
# Append any additional kernel parameters:
append=" scsi_mod.use_blk_mq=1"
```

Enabling blk_mq makes it impossible to use non blk_mq schedulers. You lose noop cfq and the non mq deadline

# Automatic IO scheduler selection

udev rules can be used to select IO schedulers for different device types.

/etc/udev/rules.d/60-ioscheduler.rules

```
# set deadline scheduler for non-rotating disks
ACTION=="add|change", KERNEL=="sd[a-z]", TEST!="queue/rotational",
ATTR{queue/scheduler}="deadline"
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="0",
ATTR{queue/scheduler}="bfq"

# set cfq scheduler for rotating disks
ACTION=="add|change", KERNEL=="sd[a-z]", ATTR{queue/rotational}=="1",
ATTR{queue/scheduler}="cfq"
```

# Sources

* Originally written by lamerix
* Source https://www.thomas-krenn.com/en/wiki/Linux_Multi-Queue_Block_IO_Queueing_Mechanism_(blk-mq)
* Source http://kernel.dk/blk-mq.pdf
* Source https://lwn.net/Articles/552904/

howtos, blk-mq, multi-queue, io, author lamerix