Linux & NVM

File and Storage System Challenges

Ric Wheeler
Senior Engineering Manager
Kernel File Systems
Red Hat, Inc.

Overview

- The Linux Kernel Process
- Linux Support for SSD Devices
- Current Challenges with NVM Devices
- Future Challenges

Linux Kernel Process

What is Linux?

- A set of projects and companies
  - Various free and fee-based distributions
  - Hardware vendors from handsets up to mainframes
  - Many different development communities
- Can be a long road to get a new bit of hardware enabled
  - Open source code allows any party to write their own file system or driver
  - Different vendors have different paths to full support
  - No single party can promise your feature will land in all distributions

Not Just the Linux Kernel

- Most features rely on user space components
- Red Hat Enterprise Linux (RHEL) has hundreds of projects each with
  - Its own development community (upstream)
  - Its own rules and processes
  - Choice of licenses
- Enterprise Linux vendors
  - Work in the upstream projects
  - Tune, test and configure
  - Support the shipping versions

The Life Span of a Linux Enhancement

- Origin of a feature
  - Driven through standards like T10 or IETF
  - Pushed by a single vendor
  - Created by a developer or at a research group
- Proposed in the upstream community
  - Prototype patches posted
  - Feedback and testing
  - Advocacy for inclusion
- Move into a “free” distribution
- Shipped and supported by an enterprise distribution

The Linux Community is Huge

- Most active contributors in 3.7 kernel – lines changed:
  - Red Hat – 18.2%
  - No affiliation – 9.3%
  - Unknown – 8.3%
  - Cavium – 5.4%
  - IBM - 4.5%
  - Intel - 3.9%
  - Linaro – 3.4%
  - Texas Instruments – 3.3%
  - ARM - 2.9%
- No pure storage company in the top 20
- Statistics from: http://lwn.net/Articles/527191

Linux Storage & File & MM Summit 2012

Linux and Current SSD Devices

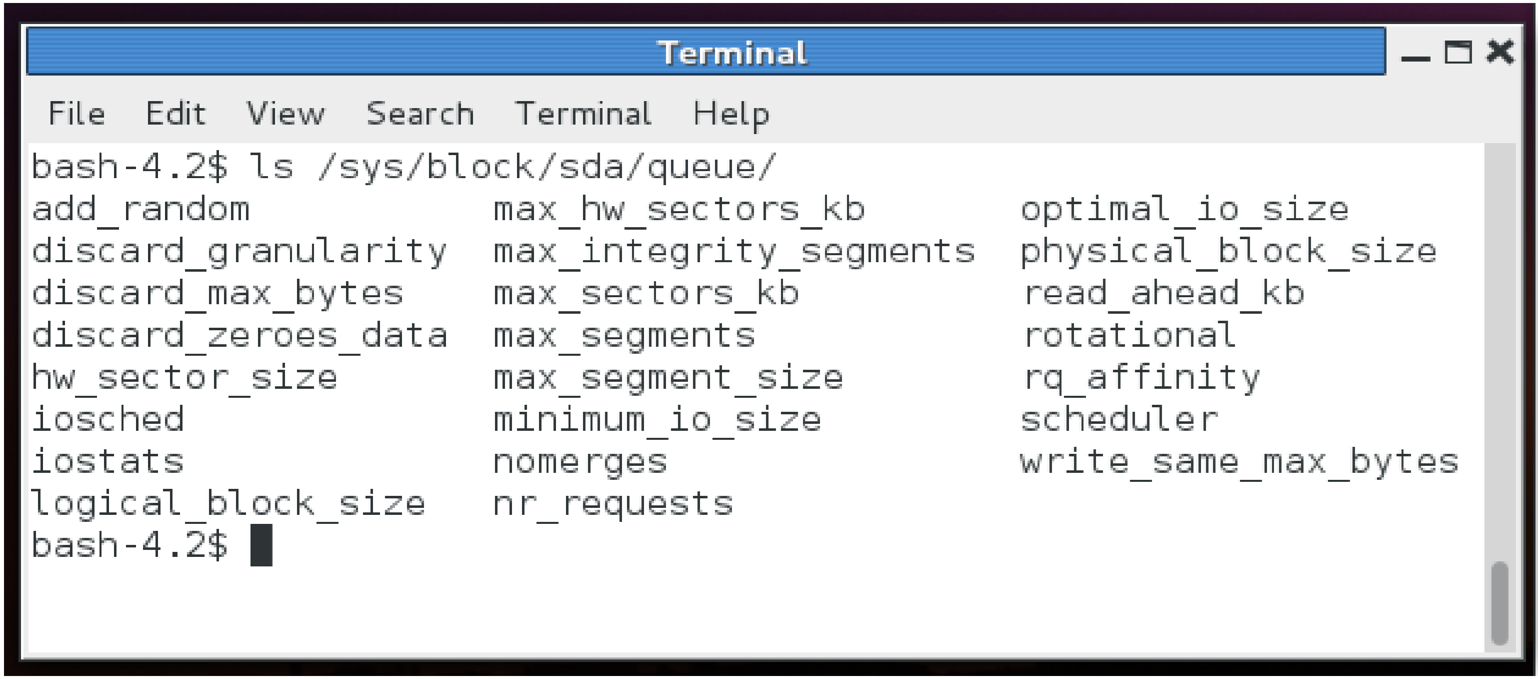
Early SSD's and Linux

- The earliest SSD's look like disks to the kernel
  - Fibre channel attached high end DRAM arrays (TMS, etc)
  - S-ATA and SAS attached FLASH drives
- Plugged in seamlessly to the existing stack
  - Block based IO
  - IOP rate could be sustained by a well tuned stack
  - Used the full block layer

PCI-e SSD Devices

- Push the boundaries of the Linux IO stack
    - Some devices emulated AHCI devices
    - Many vendors created custom drivers to avoid the overhead of using the whole stack
- Performance challenges
    - Linux block based IO has not been tuned as well as the network stack to support millions of IOPS
    - IO scheduling was developed for high latency devices

Tuning Linux for an SSD

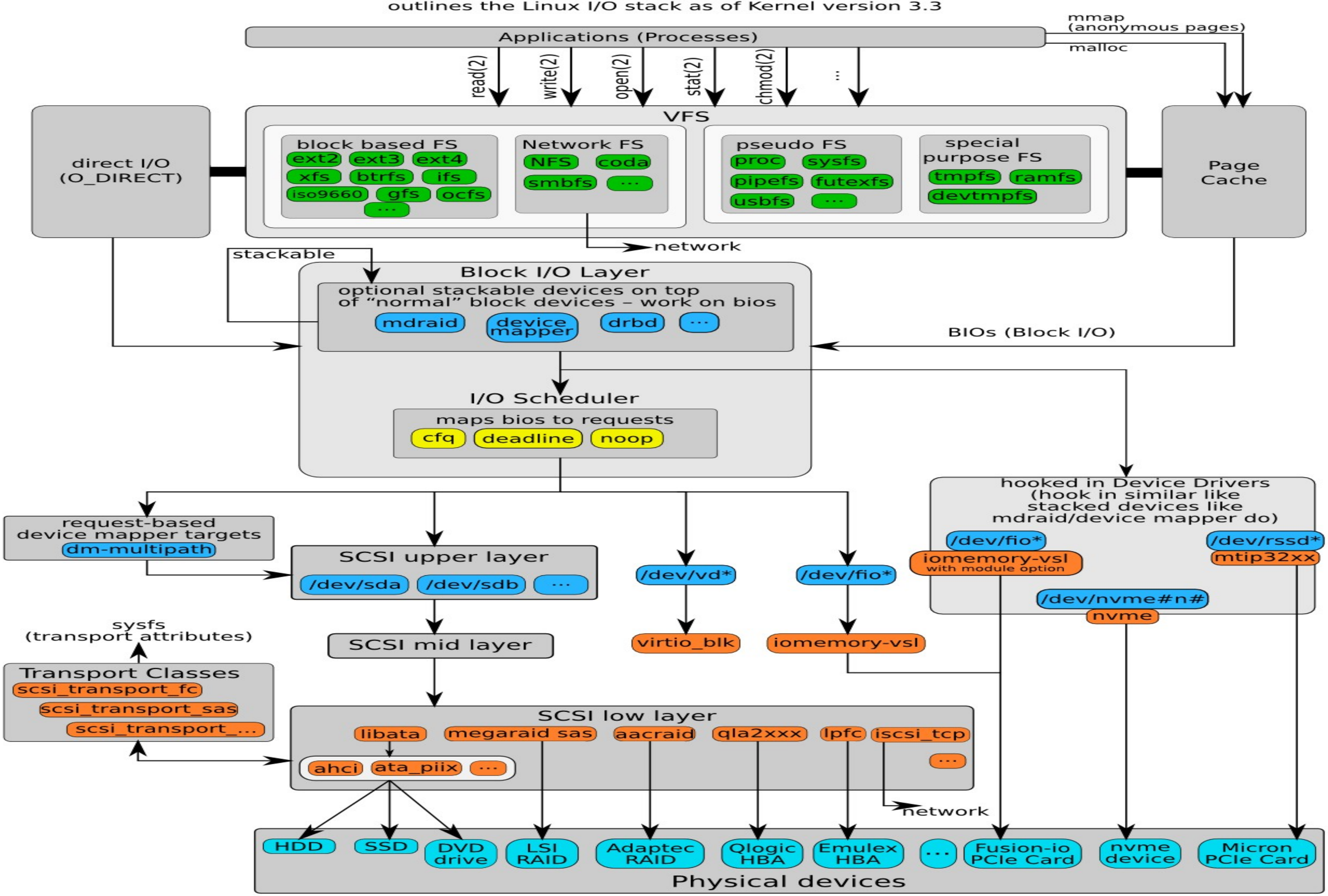


```
bash-4.2$ ls /sys/block/sda/queue/
add_random             max_hw_sectors_kb        optimal_io_size
discard_granularity    max_integrity_segments   physical_block_size
discard_max_bytes      max_sectors_kb           read_ahead_kb
discard_zeroes_data    max_segments             rotational
hw_sector_size         max_segment_size         rq_affinity
iosched                minimum_io_size          scheduler
iostats                nomerges                 write_same_max_bytes
logical_block_size     nr_requests
bash-4.2$ 
```

- Take advantage of the Linux /sys/block parameters
  - rotational is key
  - Aligment fields can be extremely useful
  - http://mkp.net/pubs/storage-topology.pdf
- Almost always a good idea not to use CFQ

The Linux I/O Stack Diagram

version 1.0, 2012-06-20
outlines the Linux I/O stack as of Kernel version 3.3



The Linux I/O Stack Diagram (version 1.0, 2012-06-20)
http://www.thomas-krenn.com/en/oss/linux-io-stack-diagram.html
Created by Werner Fischer and Georg Schönberger
License: CC-BY-SA 3.0, see http://creativecommons.org/licenses/by-sa/3.0/

Current Challenges with NVM Devices

Performance Limitations of the Stack

- PCI-e devices are pushing us beyond our current IOP rate
  - Looking at a target of 1 million IOPS/device
- Working through a lot of lessons learned in the networking stack
  - Multiqueue support for devices
  - IO scheduling (remove plugging)
  - SMP/NUMA affinity for device specific requests
  - Lock contention
- Some fixes gain performance and lose features

Device Driver Choice

- Will one driver emerge for PCI-e cards?
  - NVMe: http://www.nvmexpress.org
  - SCSI over PCI-e: http://www.t10.org/members/w_sop-.htm
  - Vendor specific drivers
  - Most Linux vendors will end up supporting a range of open drivers
- Open vs closed Source drivers
  - Linux vendors have a strong preference for open source drivers
  - They ship with the distribution, no separate installation
  - Our support & development teams can fix things

Performance & Driver Issues Cross Groups

- Developers focus in relatively narrow areas of the kernel
- SCSI, S-ATA and vendor drivers are all different teams
- Block layer expertise is a small community
- File system teams per file system
- Each community of developers spans multiple companies

Caching Implementation Choice

- Bcache from Kent Overstreet at Google is moving into the upstream kernel
  - http://bcache.evilpiepirate.org
- A new device mapper's dm-cache target
  - Simple cache target can be a layer in device mapper stacks.
  - Modular policy allows anyone to write their own policy
  - Reuses the persistent-data library from thin provisioning
  - https://www.redhat.com/archives/dm-devel/2012-December/msg00029.html
- Vendor specific caching schemes (STEC)

Future Challenges

Non-Block NVM Technology

- DRAM is used to cache all types of objects – file system metadata and user data
  - Moving away from this model is a challenge
  - IO sent in multiples of file system block size
  - Rely on journal based or btree based updates for consistency
  - Must be resilient over crashes & reboots
  - On disk state is consistent and perfect and not in sync with DRAM view
- MRAM class devices do not need block IO

Thought Experiments

- Tmpfs is a DRAM only file system
  - Just refuses to do write back when asked
  - No crash consistency or backing store
  - Endian/size issues forbid cross platform sharing
  - Linux VFS does not tolerate corruption well
  - Must map NVM device to the same address each boot
- Separate metadata and user data
  - Use traditional virtual block device for metadata
  - Bypass page cache for updating user data

Resources & Questions

- Resources
  - Linux Weekly News: http://lwn.net/
  - Mailing lists like linux-scsi, linux-ide, linux-fsdevel, etc
- Storage & file system focused events
  - LSF workshop
  - Linux Foundation events
  - Linux Plumbers
- IRC
  - irc.freenode.net
  - irc.oftc.net