TwitterCache

Replacing Hardware RAID with Persistent Memory and software RAID stack

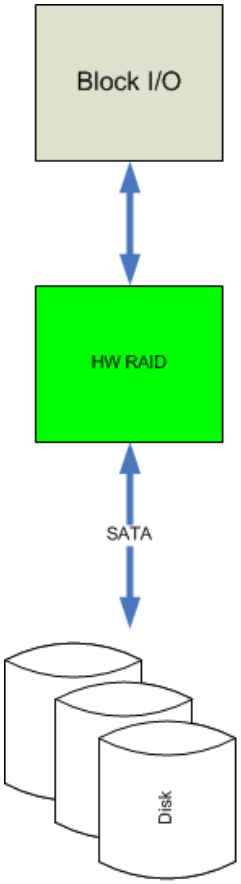
#Linuxcon2014

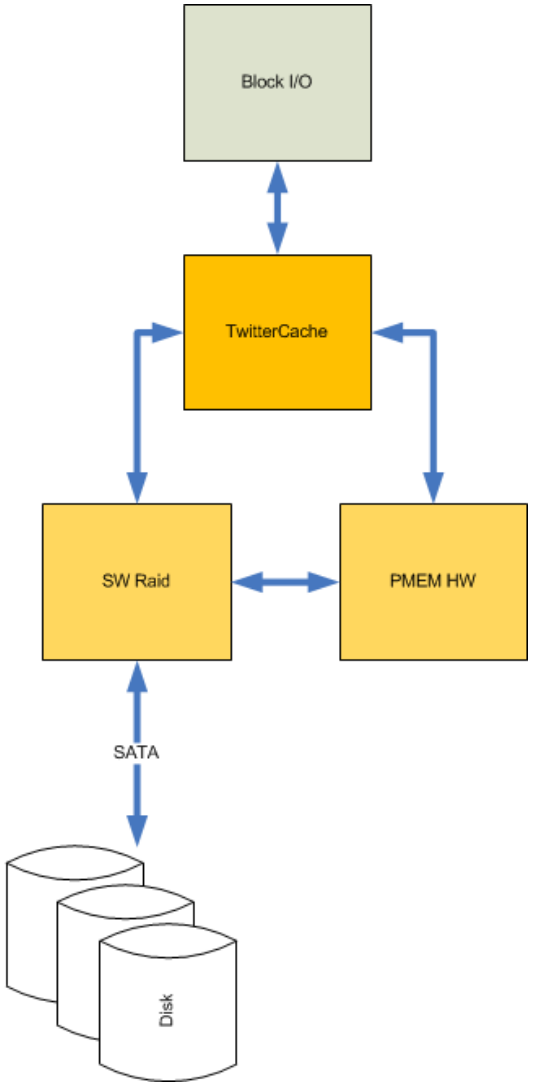
Fio Cattaneo
@fiorenzo1963

Hardware RAID

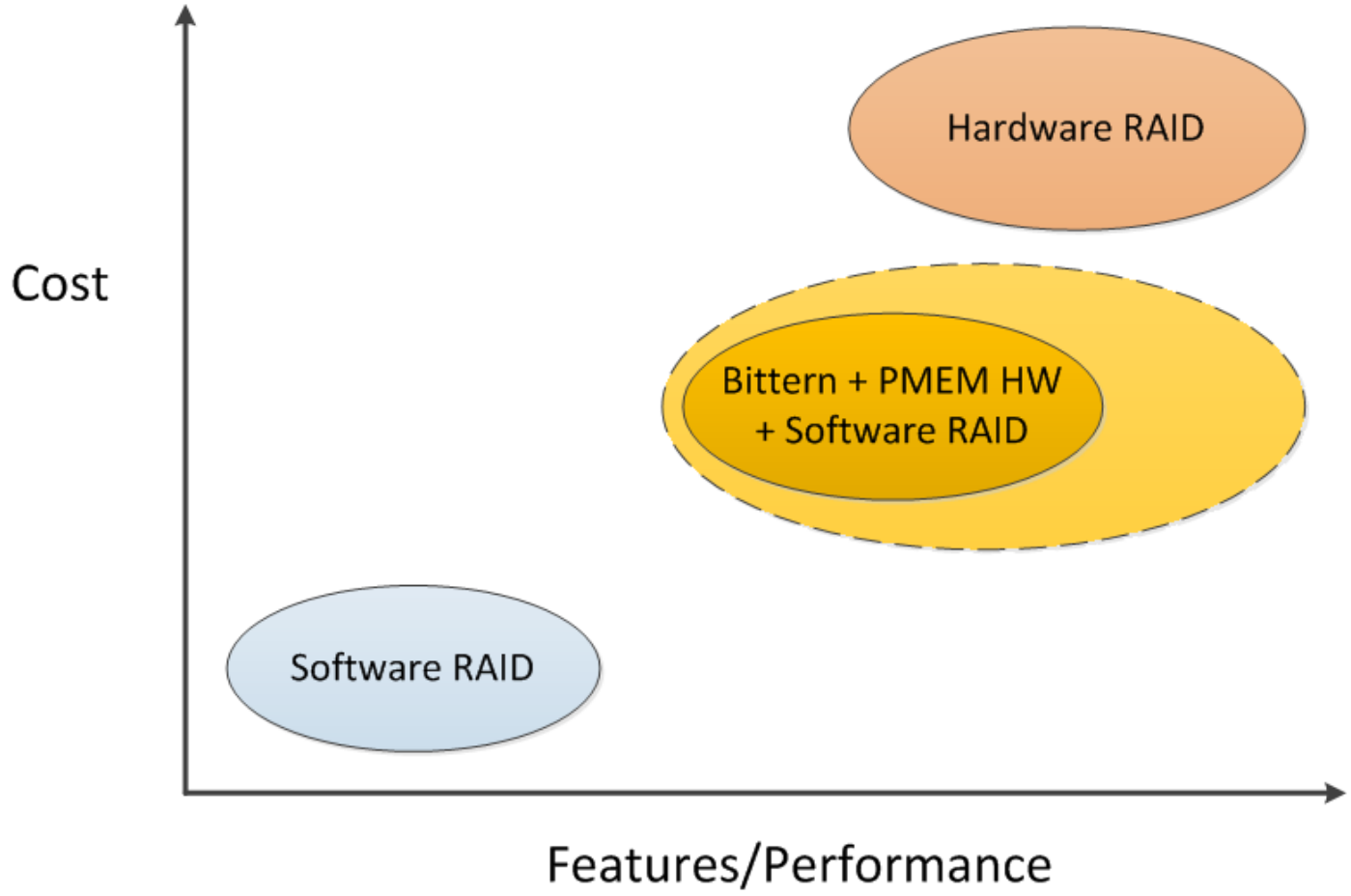
- Provides storage acceleration for I/O intensive workloads
  - Onboard NVRAM allows for writeback caching
  - Writeback caching key differentiator between HW RAID and software RAID solutions
- Commoditization of Persistent Memory ("PMEM") now allows for a cheaper full open source alternative

HW RAID vs. TwitterCache

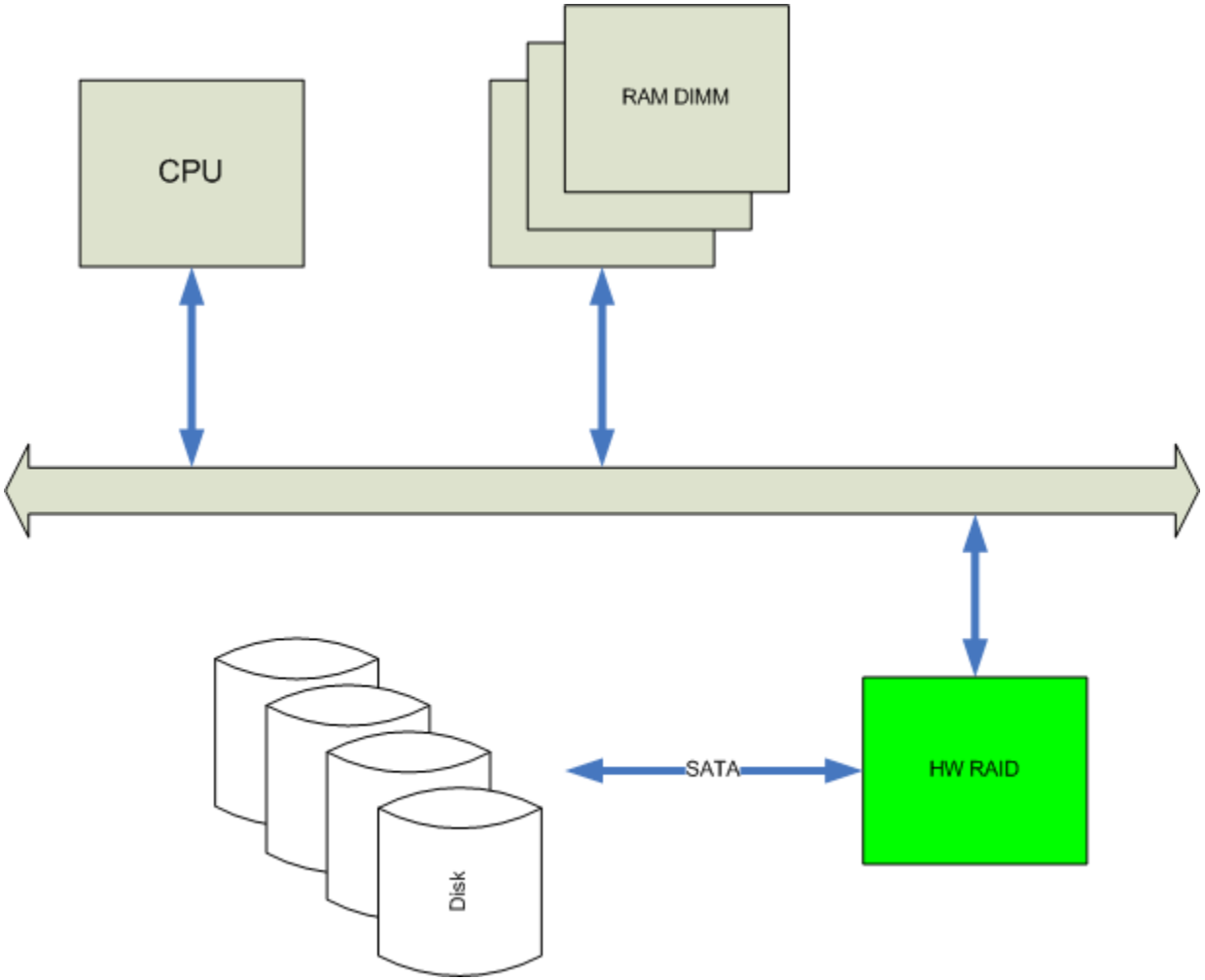




TwitterCache goals



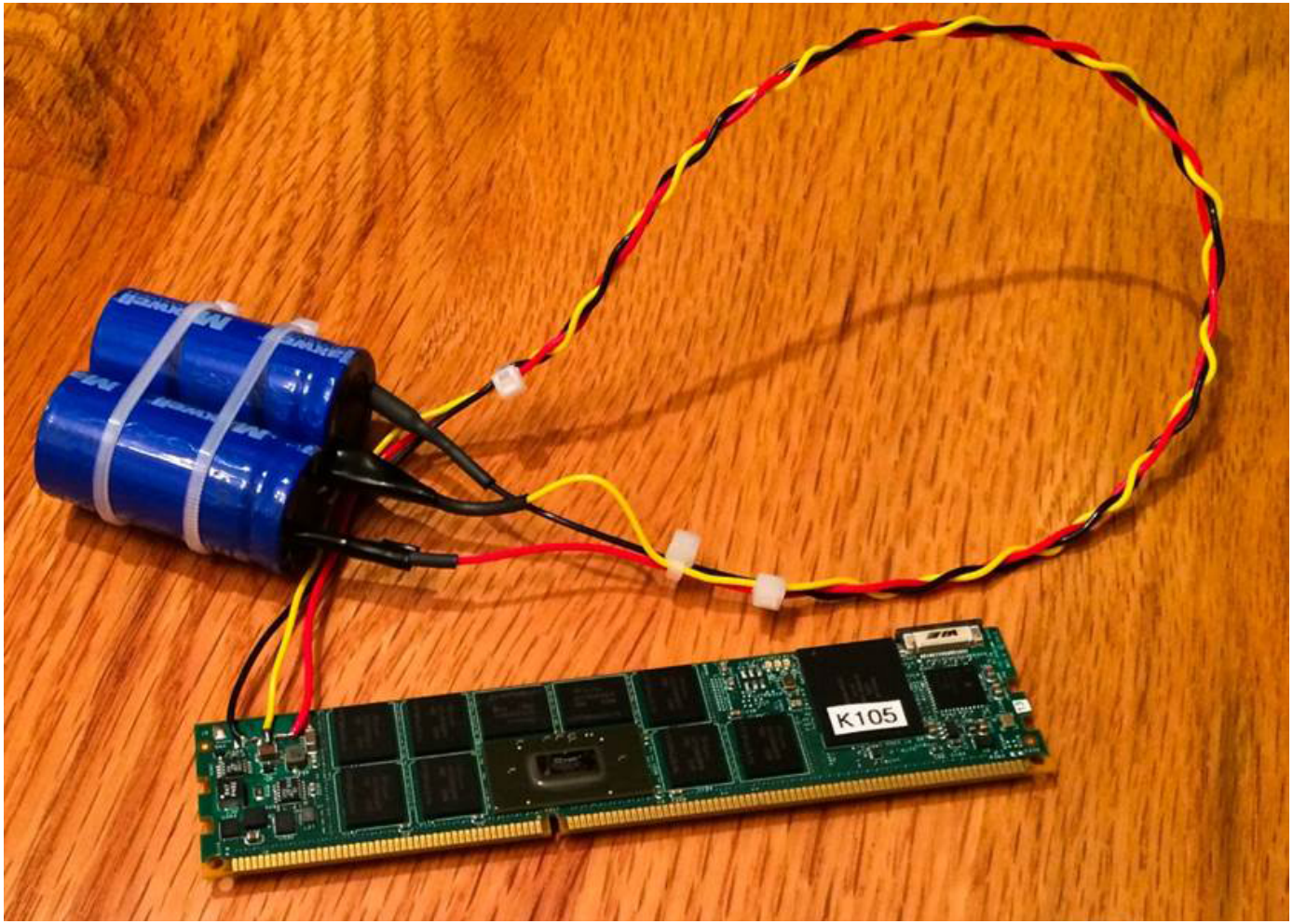
Hardware Raid



Persistent Memory

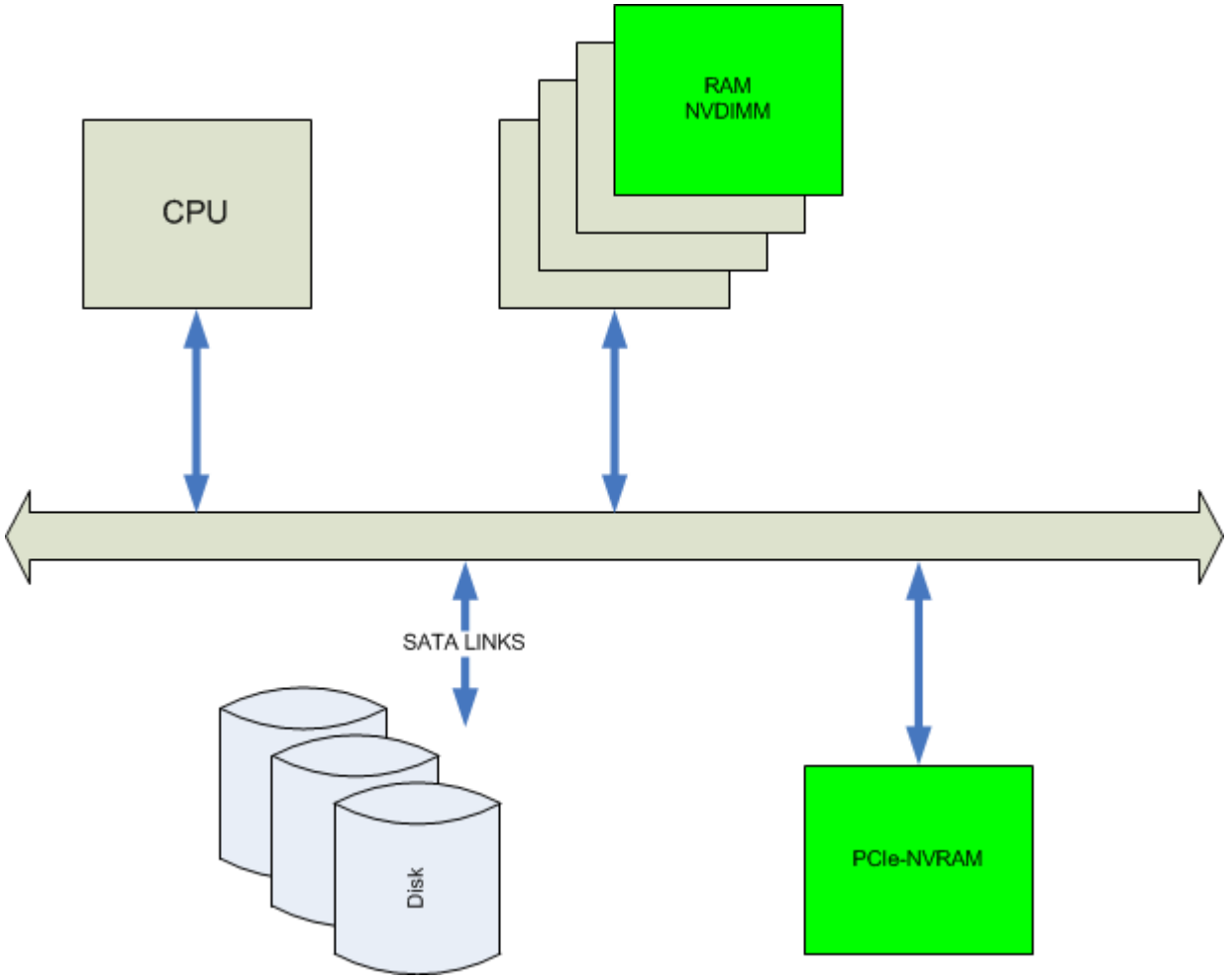
- Family of technologies which allow memory content to persist across power failure
- PCIe-NVRAM
  - PCIe card with DRAM backed by flash memory
- NVDIMM
  - DDR3/DDR4 DRAM with embedded flash memory
- NVMe
  - PCIe-attached flash memory

NVDIMM module



PCIe-NVRAM module

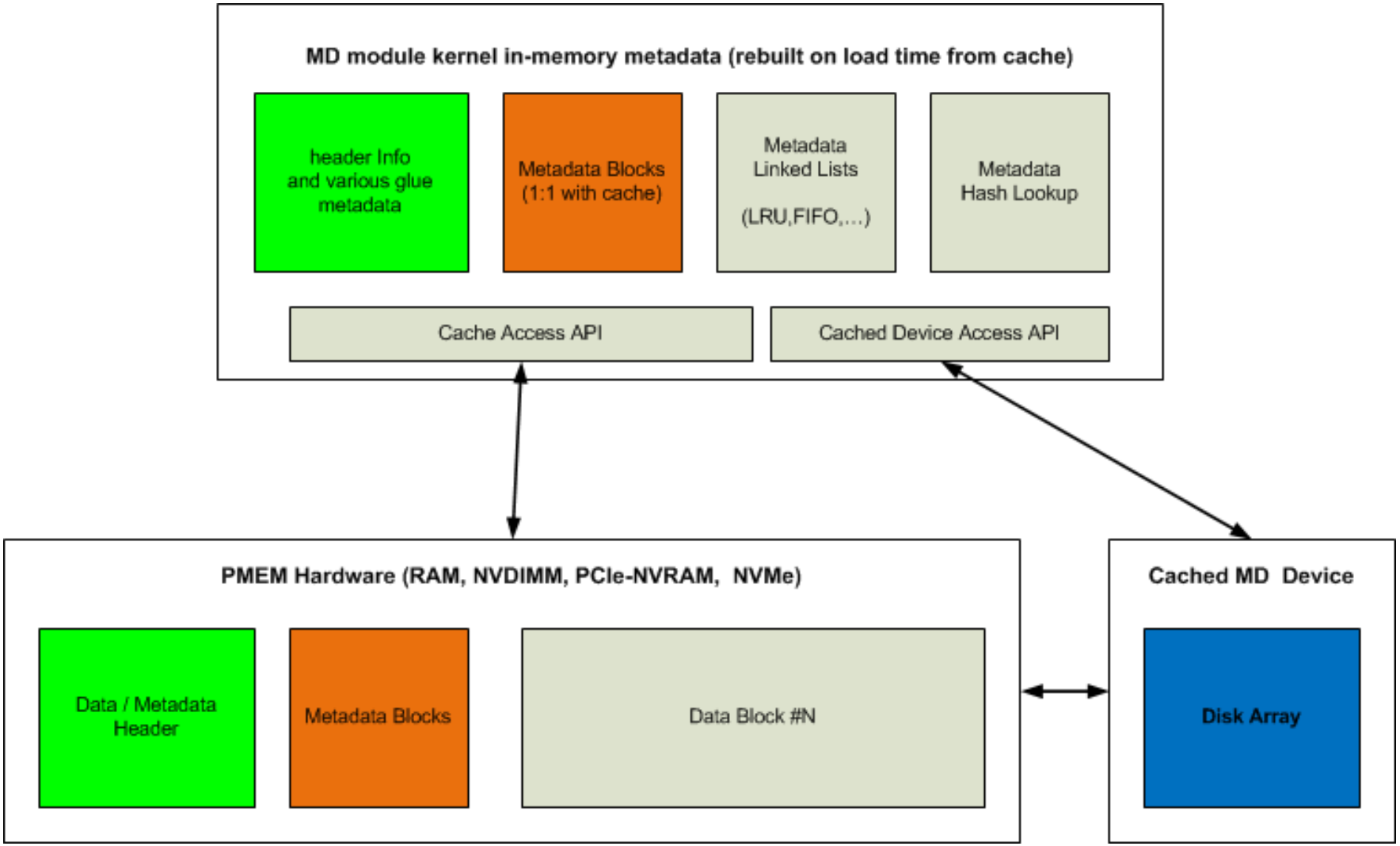
PMEM Hardware



TwitterCache

- A software solution to replace HW RAID
  - HW independent MD layer loadable module
  - PMEM hardware abstraction layer
  - Uses standard Linux RAID stack
  - PMEM commodity hardware for writeback cache

Current Architecture



TwitterCache core module

- HW independent cache implementation
- Supports double-buffering when needed
  - Minimizes number of data copies
- All metadata replicated in core for fast access
- State and perf metrics observable via /sys/fs/
- Tunable params changeable at run-time

HW abstraction API

- HW is abstracted via a Persistent Memory (“PMEM”) API layer.
- NVDIMM and/or PCIe NVRAM drivers register themselves with the generic PMEM provider
- PMEM layer provides
  - Memory copy APIs
  - Async transfer APIs w/ i/o done callbacks
  - Various configuration parameters and capabilities

PMEM API example

- get_page_read()
  - gets a page for read access
  - a callback is called when the page is ready
  - if PMEM is exposed as memory, its address is returned
  - If PMEM is exposed as PCIe-NVRAM, the data is transferred via DMA to a local buffer
- put_page_read()

PMEM API example, cont'd

- get_page_write()
  - gets a page for write access
- put_page_write()
  - releases the page buffer
  - for DMA implementations, data is copied to the cache block via DMA
  - also updates metadata if necessary
  - callback is called on completion

Cache atomicity and integrity

- Atomicity guaranteed by operation order
  - First update data, then update metadata
  - Dirty data written back to storage before replacement
  - Dirty write hits uses “dirty cloning”
- No dependency of hardware atomicity
- Integrity protections
  - crc32c checksums for metadata and data

Dirty Write Cloning

- Special care must be taken for rewriting a dirty block
  - In-place update risky, can lead to data corruption
- Dirty Write Cloning
  - Metadata block is first duplicated
  - New data is written to a new cache block
  - Temporal order during crash recovery guaranteed by logical timestamps

Single request latencies

	Read Hit	Write Miss	Write Hit
NVDIMM	7 µs	9 µs	9 µs
PCIe NVRAM	10 µs	11 µs	10 µs

PMEM vs SSD caching

- PMEM much faster but also much smaller
  - Implies lower cache_size / cached_device_size ratio
  - Implies lower hit ratio
  - Implies the need to tune the cache to operate mostly with dirty data (write buffer)

Development

- Working with HW prototypes of NVDIMM and PCIe-NVRAM from a few vendors
- Currently being developed and tested with NVDIMM and PCIe-NVRAM
  - NVMe support being added
  - Testing with Twitter workloads
- First production deployment in early 2015

Why not Bcache?

- Bcache general purpose and designed to cache HDD with SSD
- TwitterCache designed to use cache at DRAM or near-DRAM speeds
  - Can cache both HDD and SSD
  - Designed to minimize intermediate copies
  - Optimized and targeted towards fast PMEM technologies

Thank you for attending

@JoinTheFlock

@TwitterOS

@BitternCache

Extras

Some details

- Block size is PAGE_SIZE
- Fully associative
- Replacement: Random, 2Q*, ARC*, FIFO, LRU
- Sequential Read bypass
- Extensive tunable options, performance counters and timers