

Exploring Technologies for HPSS Disk Caches

Dorin Lobontu, Preslav Konstaninov, Andreas Petzold, Doris Ressmann



Why do we use the disk cache?

- Disk cache required for
 - Aggregation when writing
 - Transparent to client
 - Pack files into ~300GB aggregates
 - reduce number of tape marks
 - ~380MB/s write per drive
 - Full Aggregate Recall (FAR)
 - Request for one file triggers recall of full aggregate
 - ~400MB/s read per drive

Workload on Cache

■ Tier-1 writing to tape

- Write from client + read from client for checksum
- Writing to tape: read ~same as write from client



2:1 read:write

■ Tier-1 reading from tape

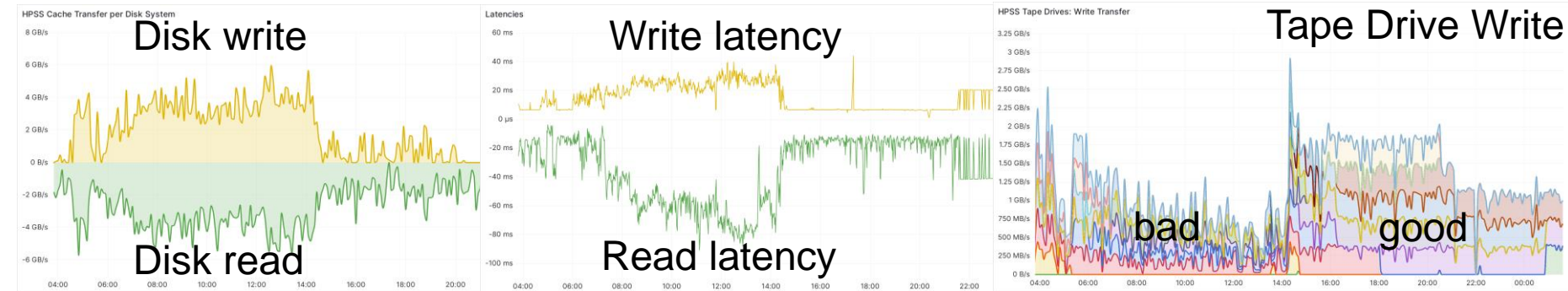
- Read from tape: write on cache one stream per drive
- Read from client: read from cache







1:1 read:write

HDD-based Setup

- 2 NetApp E5700 w/ 120 8TB drives each (~1.4PB usable)
 - Expect ~12GB/s per system with 70% read workload
- Observations
 - Never close to 24GB/s
 - System prioritizes writes → reads starved → writing to tape slow
 - DDP vs. RAID6 vs. RAID10 no big difference



Workload on Cache

- Tier-1 writing to tape
 - Write from client + read from client for checksum
 - Writing to tape: read ~same as write from client 2:1 read:write
- Tier-1 reading from tape
 - Read from tape: write on cache one stream per drive
 - Read from client: read from cache 1:1 read:write
- Random I/O to/from clients
- Sequential I/O to/from tape drives  Random IO only
- Streams to tape drives need to be stable  more IOPS help

Technology Options

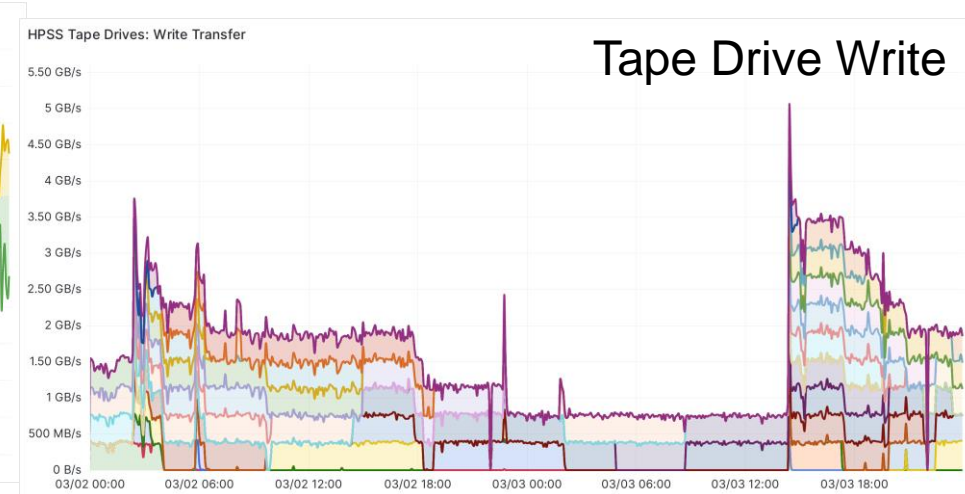
- More HDDs
 - JBODs + ZFS
 - More NetApp extension enclosures

- GPFS (22PB LSDF cluster right next to HPSS racks)
 - Integrate new HPSS movers into IB fabrics and GPFS clusters
 - Use files in GPFS as block devices for HPSS

- Flash

SSDs

- Added 2 Dell ME5024 + Extension Enclosures with 2x 48 3.84TB SSDs
 - ~250TB usable space
- Better latencies → much improved tape write rates
- Limited throughput of ME5024 controllers → isn't there something better?



Cache Requirements

- Low latencies → Flash/NVMe
- High throughput → AFA+NVMe or NVMe in server
- Storage redundancy → AFA or other NVMe-optimized RAID

- Big vendor AFA way too expensive
- NVMe-optimized RAID solutions
 - GRAID SupremeRAID: GPU-accelerated RAID
 - Xinnor xiRAID: software RAID

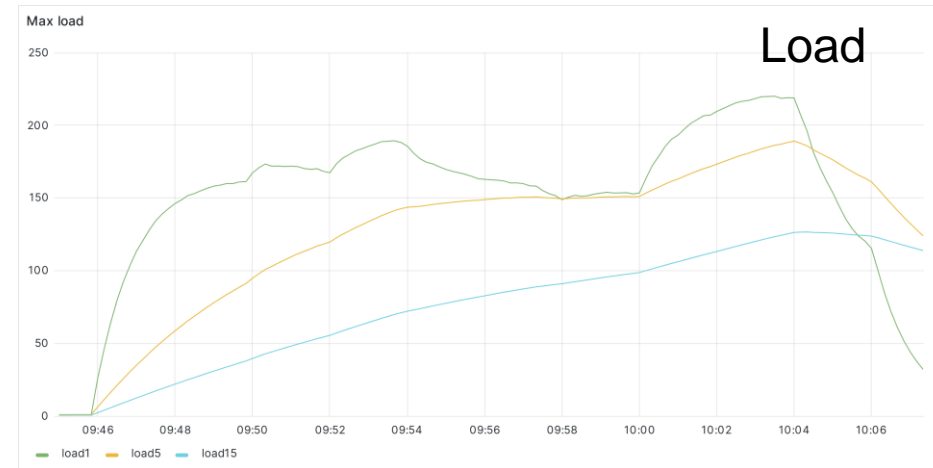
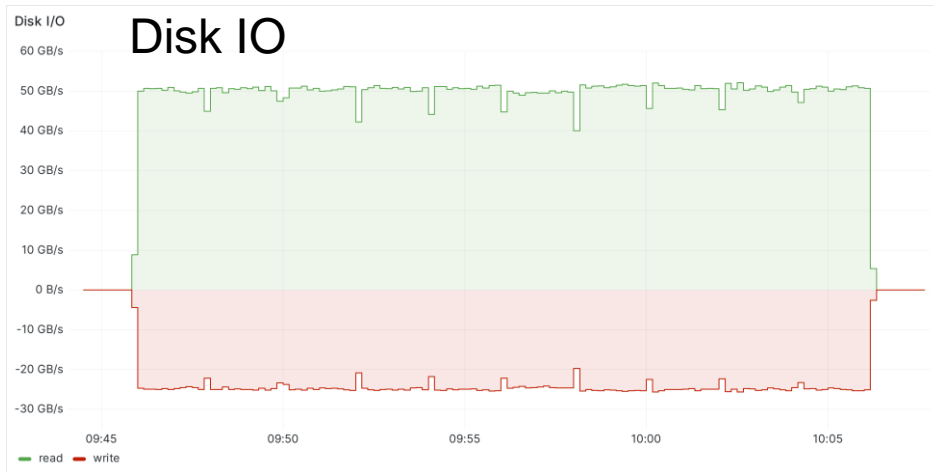
Setup (2023)

- 2U Supermicro AS-2015CS-TNR
 - Single AMD EPYC 9554P 64-Core 3.1GHz
 - 512GB RAM
 - 10x 30TB Micron 9400 NVMe devices (7GB/s)
 - 4x 100Gbit/s Ethernet
-
- xiRAID licensed per device used in RAID6
 - ➔ additional NVMe name spaces have to be licensed too :-)
 - Single xiRAID6 with several regular LVs on top
 - ~240TB usable space
 - LVs needed due to HPSS IO connection limits per disk device



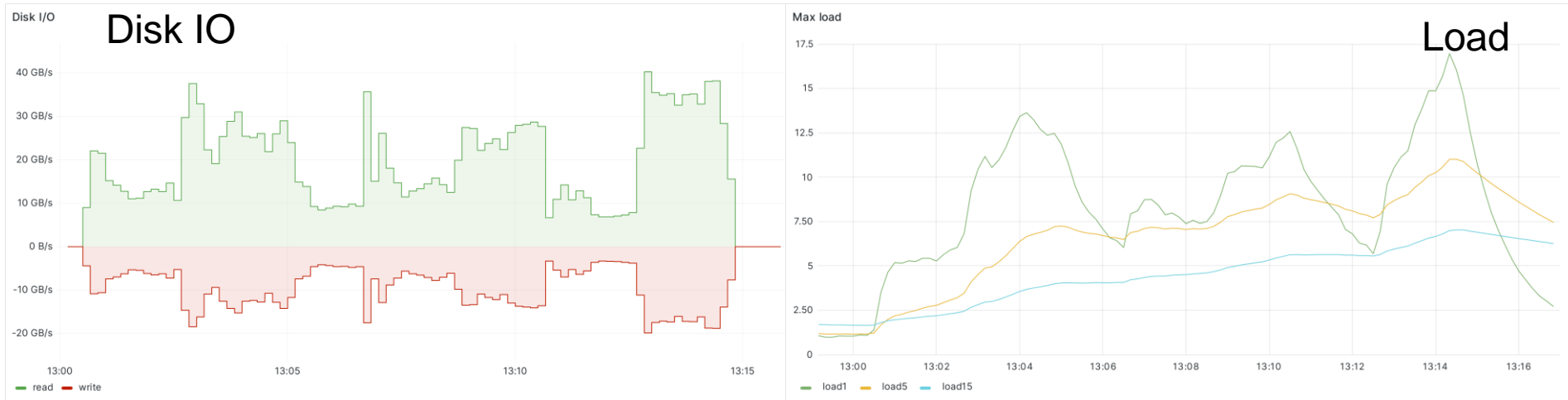
Benchmarks 1

- fio benchmarks with different block sizes/file sizes/number of clients
always with 2:1 read:write ratio
- Monitor throughput, CPU load



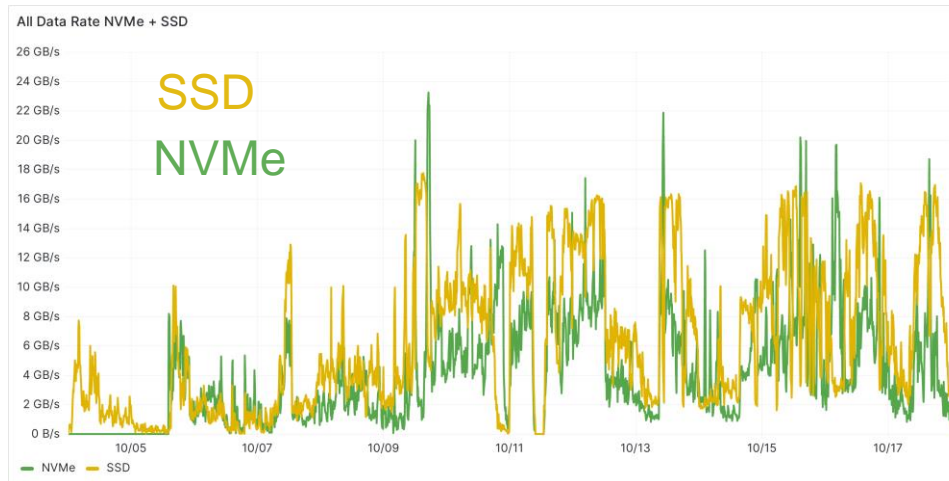
Benchmarks 2

- Same benchmark runs, but limit xiraid to 64 threads



In production

- Since October 2023, next to SSD systems
- No interference between xiRAID and HPSS mover process at current workload level
- SSDs still receive larger share of IO → HPSS tuning required



Next Steps (2024)

- Purchased 3 more servers
- Slightly different NVMe devices
 - 12 3 DWPD instead of 10 1 DWPD devices
 - same usable RAID6 capacity
- GridKa will use ~1PB NVMe-based buffer on 4 servers

Summary and Outlook

- Workload on tape system cache requires lots of IOPS
 - Many disks or flash
- Servers with large local NVMe storage powerful and cost effective
 - Excellent latencies and throughput
- Redundancy requirement
 - Proprietary RAID solutions still necessary – doesn't solve problem of broken server
- Would like to test PCIe connected storage enclosures
 - Decouple NVMe devices from servers
 - Hard to come by