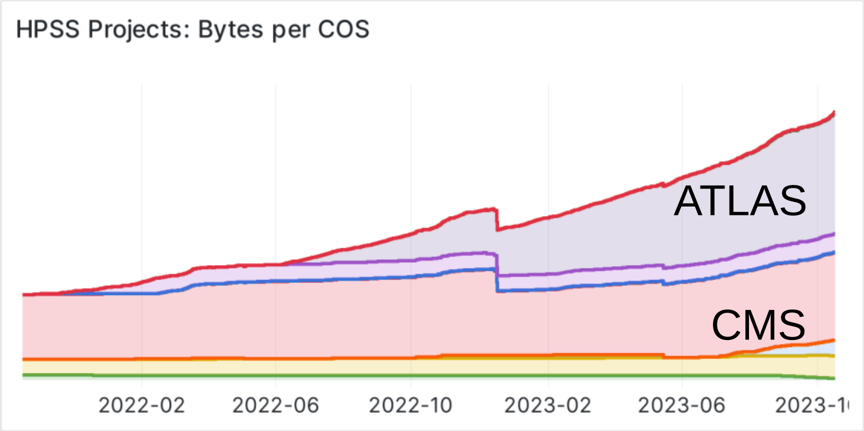
Exploring Technologies for HPSS Disk Caches

Dorin Lobontu, Preslav Konstaninov, Andreas Petzold, Doris Ressmann

Karlsruhe Institute of Technology

HPSS at KIT

- Serves as tape system platform for Tier-1 and bwDataArchive
- ~100PB used today
- 4 tape libraries
- two sites ~8x J-E lengths apart



GridKa

BWDATA ARCHIVE

HPSS Tier-1 Integration

- Migration TSM → HPSS since 2020
  - Outside of dCache/xrootd
  - 20-40 streams (write+read for checksum)
- Writing to tape
  - dCache/xrootd pools call HPSS client locally
  - 40 streams per pool
- Reading from tape
  - dCache ENDIT provider + new HPSS-specific scheduling backend per VO
  - Backend sorts files per aggregate, triggers stage, copies files into pools
  - Plan to replace file-based provider-backend communication

Why another disk cache?

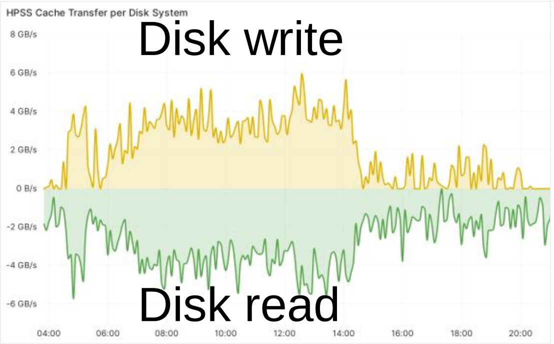
- Disk cache required for
  - Aggregation when writing
    - Transparent to client
    - Pack files into ~300GB aggregates
    - reduce number of tape marks
    - 380MB/s write speed per drive
  - Full Aggregate Recall (FAR)
    - Request for one file triggers recall of full aggregate
    - 400MB/s read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

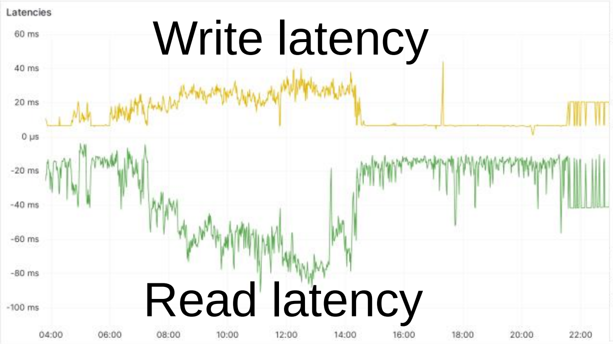
Workload on Cache

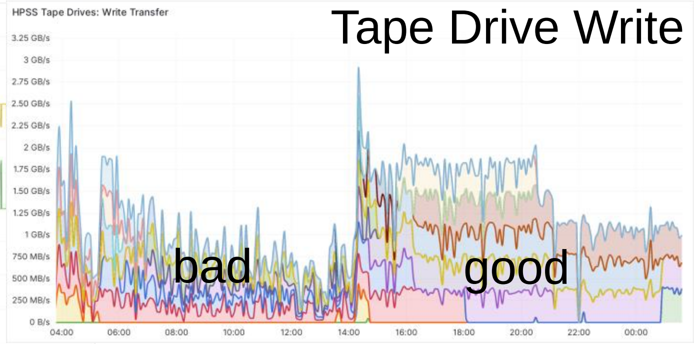
- Tier-1 writing to tape
  - Write from client + read from client for checksum
  - Writing to tape: read ~same as write from client → 2:1 read:write
- Tier-1 reading from tape
  - Read from tape: write on cache one stream per drive
  - Read from client: read from cache → 1:1 read:write

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

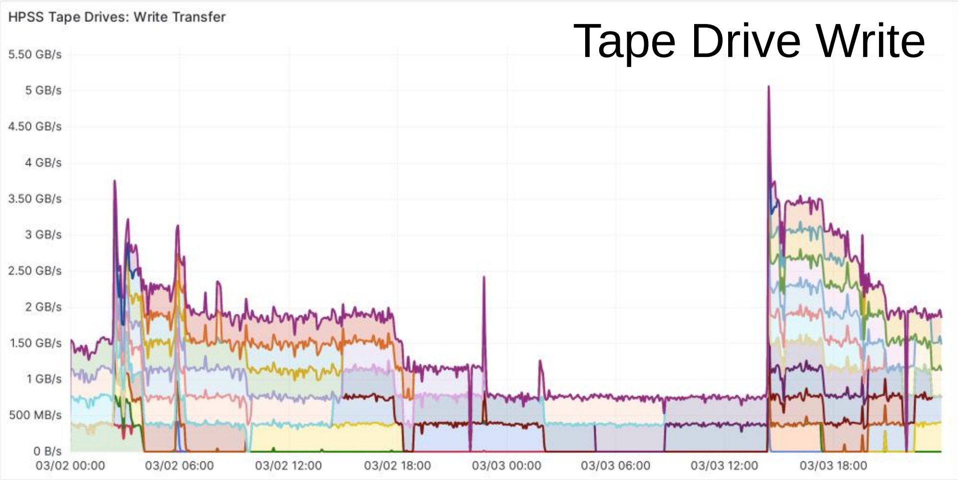
  → Random IO only

- Streams to tape drives need to be stable

  → more IOPS help

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+NVMoF 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

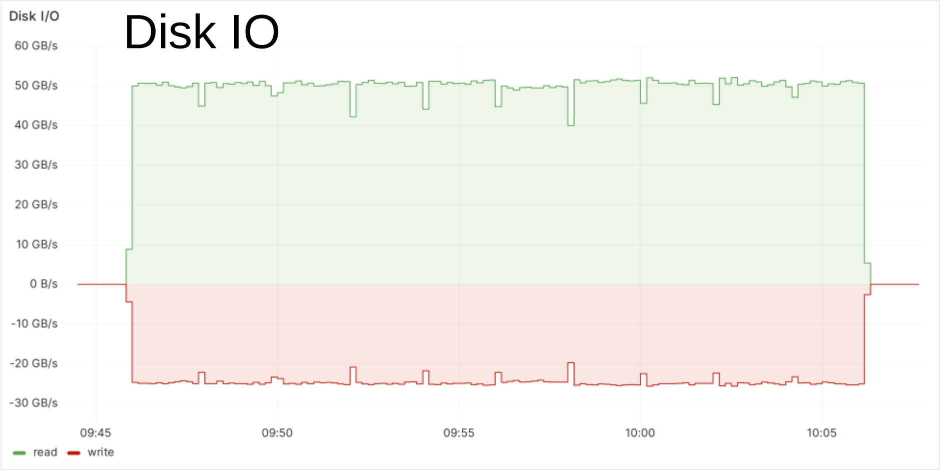
Setup

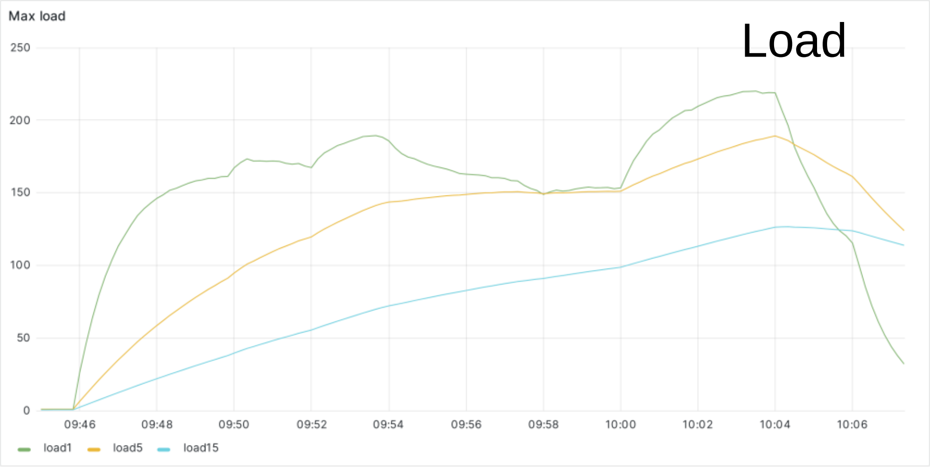
- 2U Supermicro AS-2015CS-TNR
- Single AMD EPYC 9554P 64-Core 3.1GHz
- 512GB RAM
- 10x 30TB Micron 9400 NVMe devices (7GB/s)
- 4x 100Gbit/s Ethernet

- xiRAID licensed per device used in RAIDs
  ➔ 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

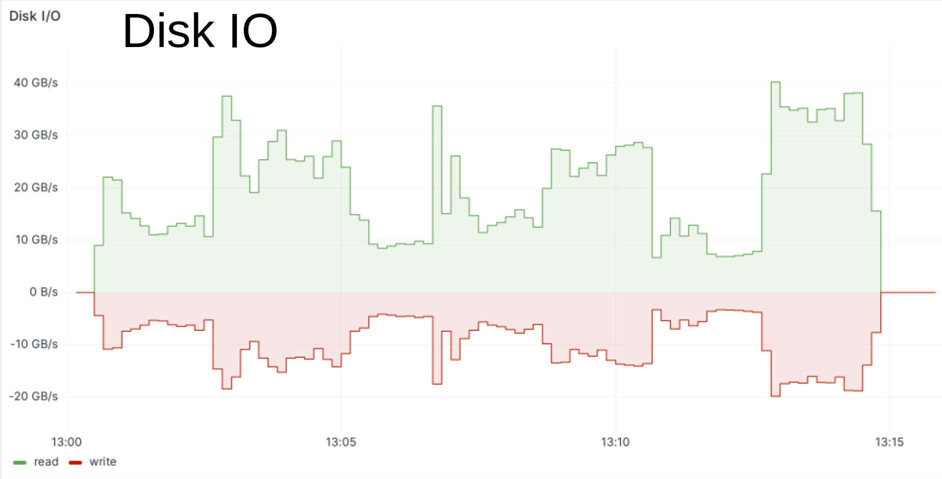
- fio benchmarks with different block sizes/file sizes/number of clients always with 2:1 read:write ratio
- Monitor throughput, CPU load

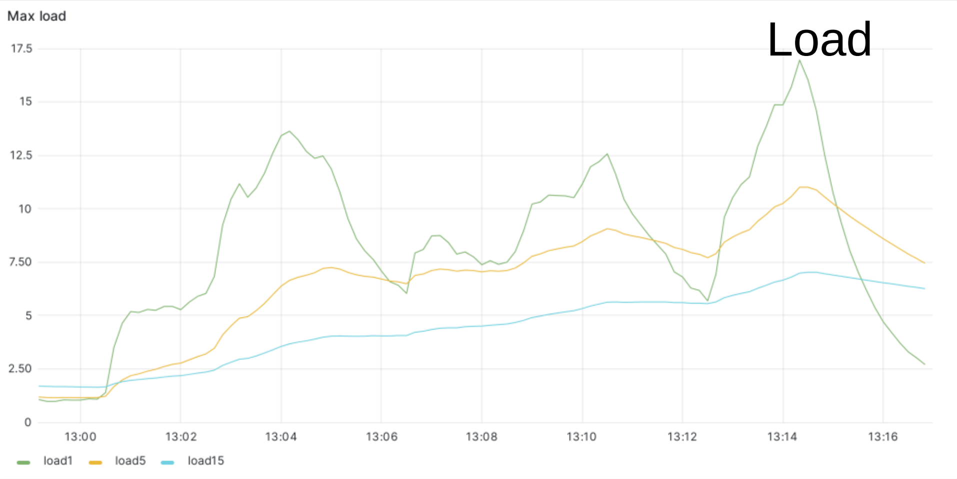




Benchmarks 2

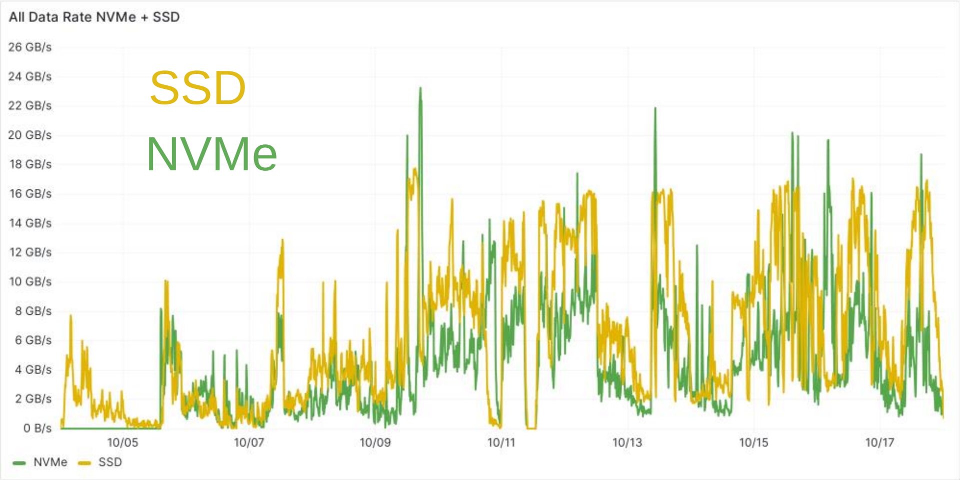
- Same benchmark runs, but limit xiraid to 64 threads





In production

- Since two weeks, 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



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 throuput
- Redundancy requirement
  - Propriatary RAID solutions or simple mdraid RAID1?
- Would like to test PCIe connected storage enclosures
  - Decouple NVMe devices from servers