The Use of Flash in Large-Scale Storage Systems

Nathan.Rutman@Seagate.com
Seagate ❤️s Flash!

Seagate acquired LSI’s Flash Components division May 2014

Selling multiple formats / capacities today

Nytro XP6500

Sandforce SF3700 flash controller

Seagate Deskod SSHD

1200.2 SSD SAS

Nytro™ XF1440 SSD NVMe
Why flash?

Advantages
■ Flash is faster in latency and throughput
■ Flash is much faster for seeks

Disadvantages
■ Flash is more expensive, less dense, limited write cycles

If it wasn’t, would we put flash everywhere

Tradeoffs
■ Performance
■ Cost
■ Durability
■ System Complexity
Architectures
Flash and Lustre

Where can we use flash in our Lustre systems?

- Flash on MDT
- Flash on OSS servers
- Flash on OST devices
- Flash in front of Lustre
Flash SSDs on the MDT

Assumed to be a perfect candidate
- Small, random IO
- High IOPS

But: a large MDS can use RAID to bundle spindles
We sell a 7+7 RAID 10 of 10K drives; mdtest with SSHD and SSDs drives shows no improvement
- Even for WIBs and journals
- Disks are cheaper, much better durability
- Some of our customers demand SSDs anyhow

Conclusion:
SSDs make sense for small MDTs, but for larger RAID pools the MDS should be your bottleneck, not the drives
Flash on OSTs

- SSHD
- Local metadata (journal device)
- Flash pool
- Flash cache
**SSHD**

Solid State Hybrid Drive

**Basics:**
- HDD magnetic media operates with the same characteristics as the HDD drive
- Persistent backup of DRAM-based write cache
  - All writes to DRAM, coalesced to disk
  - Back-EMF powers writes DRAM->flash
- NAND flash is used for read caching
  - Read data moved to flash by popularity

**Neutral:**
- Streaming performance same as HDD
- Other caching layers (OST, client) limits usefulness of read cache

**Positive:**
- Double random IO write/rewrite perf
- Remove effects of local fs metadata updates
- Host-pinning - journal, block bitmaps, MMP
OST MD on flash devices

- SSHD for local filesystem operations
  - block bitmaps
  - write-intent bitmaps (wibs)
  Conclusion: should be a good use case for SSHD

- External MDRAID WIBs
  - speeds RAID rebuilds
  - is an optimization, not critical
  Conclusion: put it on an SSD

- External EXT4 journals
  - frequent writing makes flash ʰ
  - are sequential anyhow
  Conclusion: use fast HDD for reliability
Flash OST pool

Basics:
- Pools of SSD and HDD OSTs
- Set striping per dir / usecase

Neutral:
- No automatic migration
  - Can use HSM policy engine and special data mover to automate tiering
  - Simpler than new burst buffer software

Positive:
- High random-IO r/w
- Easy to adjust sizes

Conclusion:
May make sense, depending on use case
OSS flash cache

separate flash layer between OSTs and HDDs

- All IOs flow through a large local flash cache device
- Writethrough or writeback

Positive:
- Coalesce random writes
- Cache reads
  - Bypass for cache miss - no penalty
- Large capacity - may help sequential to a point
- Transparent to upper layers
- Smart caching algorithms, LRUs
- SCSI on PCIe or NVMe

Conclusion:
Good choice for improving random & cached IO transparently
Interposing flash layer in front of Lustre

separate flash and Lustre systems; burst buffer

Basics
■ New software layer in front of Lustre
■ All IO written to this new interface
■ Layered tiers: primary to flash, secondary to Lustre

Negative
■ Additional read latency if strictly tiered (stage-in, stage-out)
■ Complexity: more layers, new API, HSM, failure handling/reliability, HA/dual-porting, RDMA/0-copy

Neutral
■ New frontend software stack
■ New frontend semantics

Positive
■ Accelerates random and sequential
■ No Lustre overhead
■ Restricted interface with backend
Use Cases

When should we use flash?
Use cases

- Defensive IO
- Job-based staging
- Random IO
- Streaming IO
- Capability duty cycle (all IO, all the time?)
- Read-heavy loads
Defensive IO (checkpoint-restart)

To BB or Not to BB?

Use flash as fast temp cache
Snapshot all memory in 5 min
Spool off to disk (sequential) in 55 min

1PB, 5 min to flash:
3.3TB/s / 500MB/s = 6600 SSD (@1TB)
1PB, 55 min to disk:
300GB/s / 100MB/s = 3000 HDD (@8TB)

and backend capacity 50PB: 6250 HDDs = 27 min
But wait a minute, we have 6600 extra SSDs in the system, let’s buy 6600 Lustre HDDs instead:
12900 HDDs = 12.9 min & 103PB

So for the cost delta of flash over HDD, you save 8 mins IO and lose 53PB of capacity.
What if we 2x the speed of flash and HDD? 8.7 min for HDD alone.

Conclusion: defensive IOs with a BB can buy some compute time, for a cost - but **DO THE MATH**.
Job-based staging

Job scheduler pre-stages all job data into flash
Job does IO to flash
Scheduler destages on completion

- Lustre Flash Pool or Burst Buffer
- Double buffers can hide stage time
- Read and write
- Good for fixed dataset sizes and distinct filesets
- Bad for unknown sizes (e.g. searches)
- Requires scheduler knowledge of filesets
- Everything is written twice
- Beware flash write cycles

Conclusion: like defensive IO, do the math. Relative sizes means disk-tier performance is free
Random IO

- OST SSHD or OSS flash cache
  - accelerate small IO
  - limited cache size

- Flash pool
  - accelerate specified IO
  - manual direction

- BB
  - accelerate all IO
  - new frontend software now must include cache handling, consolidation

Hard drives aren’t great at random IO

Conclusion: flash pool good for known random jobs, OSS flash cache good for unknown
Streaming IO

- OST SSHD or OSS flash cache
  - limited cache size, not generally useful

- Flash pool
  - large cache size

- BB
  - add destaging time

Operating on the data, or archiving it?
How continuous / large are your streams?
Hard drives are good at this already

Conclusion: flash pool works if you can age it out
Capability duty cycle

How much of the time do you need max bandwidth?

Analysis of a major HPC production storage system*
- 99% of the time, storage BW utilization < 33% of max
- 70% of the time, storage BW utilization < 5% of max

Conclusion: buys minor (~5%) time impact

Assume:
Flash 5x disk speed
Even distribution of load between 70-99%

*DDN, MSST15 http://storageconference.us/2015/Presentations/Vildibill.pdf
Read-heavy loads

If working set is large, flash doesn’t help
  big data, data mining

If working set fits into a flash buffer
  still have to stage-in
  or readthrough cache and hope for repeat reads
## Summary

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Performance</th>
<th>Cost</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>MDT flash</td>
<td>no gain</td>
<td>$$$</td>
<td>-</td>
</tr>
<tr>
<td>OST SSHD</td>
<td>▶️</td>
<td>$</td>
<td>-</td>
</tr>
<tr>
<td>OST MD flash</td>
<td>▶️</td>
<td>$</td>
<td>Θ</td>
</tr>
<tr>
<td>OST pool</td>
<td>▶️▶️▶️▶️ if perm</td>
<td>$$$</td>
<td>-</td>
</tr>
<tr>
<td>OSS cache</td>
<td>▶️▶️▶️ size dep</td>
<td>$</td>
<td>ΩΩ</td>
</tr>
<tr>
<td>BB</td>
<td>? size dep</td>
<td>$$$</td>
<td>ΩΩΩ</td>
</tr>
</tbody>
</table>
What does Seagate do with flash?

- Seagate makes both HDD and full range of flash devices
- All the systems that Seagate Systems Group sells contain flash in some form
- Both are getting faster and bigger
- MDS: RAID10 10K fast and durable
- OST: SSD for WIBs and journals
- OSS flash cache: will be offering SAS and NVME based systems over the next 12 months
- Flash as an interposing layer in front of Lustre: instead, flash and HDD characteristics should be treated quantitatively, not qualitatively (no separate systems)
Thanks