Building a Flash Optimized Storage System

Software Approach

Saumyabrata Bandyopadhyay and Ravi Dronamraju
Flash: The change agent in Storage Software Design

- **Performance**
  - Enables 4-5 times improvement in throughput at low latency
  - Sustained read and write rate
  - Scale up vs. Scale out

- **Physical Limits and Goals of Media**
  - Finite life span of the media
    - Every potential reuse of the block result in media life loss
  - Better power usage
  - Smaller Form Factor and Weight

- **Cost**
  - Higher upfront cost
  - Should integrate seamlessly with lower endurance Flash
  - Offset with better TCO
Performance: Reads

Techniques:

- IO path with few delay centers
- Use all the channels on the SSD
- Different priority for
  - Data being read by host vs. system
  - Data being read vs. written
- Software needs to be able to take advantage of all the CPU
  - MP safe data structures
Data ONTAP IO Path: Reads

Optimized read path to take advantage of device latencies.

Host

Protocol Layer

File / LUN

FlexVol (Indirection / Data management layer)

Aggregate / Physical layer with Software RAID

Storage

Physical Storage
Data ONTAP Read Path for Flash: Impact on performance

4x Improvement on Random Read Operations for the same latency
Performance: Writes

Techniques:

- Journal changes or log the operation to persistent lowest latency media
  - NVRAM/NVMem or battery backed DRAM

- Perform CPU intensive operations and metadata updates in the background instead of the IO path

- Writes to flash batched and sequential to reduce WAF

- Frees / overwrites batched to reduce random metadata updates
Data ONTAP IO Path: Writes

- Writes logged to NVRAM
- Data pushed to persistent storage periodically in the background
Performance Trade Off: Scale out vs. Scale up

Performance / Application requirements dictate architecture trade off

Scale Up

- No fixed Storage to DRAM and CPU ratio
- Metadata must be searchable on disk for performance
- Applications with Small working sets but large storage footprint

Scale Out

- Fixed Storage to DRAM and CPU ratio
- Metadata normally in memory
- Performance scales with nodes
- Performance sensitive applications
Data ONTAP: Scale Up and Scale out

- We need both

- Cost, performance and data center footprint dictate scale up vs. scale out

- Ideal performance when DRAM to Storage ratio around 1%
  - Not every workload / Application requires the same level of performance

- Scale up
  - Metadata for the working set fits in DRAM
  - Applications can tolerate occasional latency spikes during workload shifts

- Scale out
  - Consistent latency
  - Performance scales with storage
  - Large working sets
Flash Physical limits: Layout

- Log structured layout
  - No writes in place
  - Results in fragmentation – addressed using other means

- Indirection layer to address physical blocks
  - Ability to defrag & create clean segments
  - Reduces random overwrite, which in turn reduces WA
  - Allows for additional functionality that reduces the overall TCO
    - deduplication
    - cloning
    - thin provisioning

- Log On Log
  - Mitigated by writing sequentially to large segments using a single stream
  - Data segregation with multi-stream capabilities on next version SSD firmware
Data ONTAP – Layout Engine

- Write Anywhere File Layout (WAFL)
  - Log structured file system
  - Already optimized for write
  - NVRAM for staging writes

- Metadata searchable both in-memory as well as persistent storage

- Metadata updates optimized by batching updates

- FlexVols provide an indirection layer

- Generic File system instead of optimized for SAN

- Format that supports Storage Efficiency features

- Media aware physical layout engine – allows for support of different types of persistent media
Reduce the TCO by enabling Storage efficiency features

- Log structured file layout
  - Create space efficient snapshots with very little overhead

- Indirection layer allows for Storage efficiency features like
  - Deduplication
  - Clones
  - Thin provisioning
  - Writeable snapshots
  - Segment cleaning / Defrag – which reduces the Write Amp on the SSDs
    - Allowing for reduction of Overprovisioning on the SSDs

- Format that supports compression at various levels
Conclusion

- Software design requires efficient use of CPU and fewer delay centers
- Software needs to customized for media writes
- Layout and format is important to reduce TCO by enabling storage efficiency features
- Application performance requirements dictate architecture