Practical Applications of Lustre/ZFS Hybrid Systems

LUG 2014 – Miami FL

Q2-2014

Josh Judd, CTO

Agenda

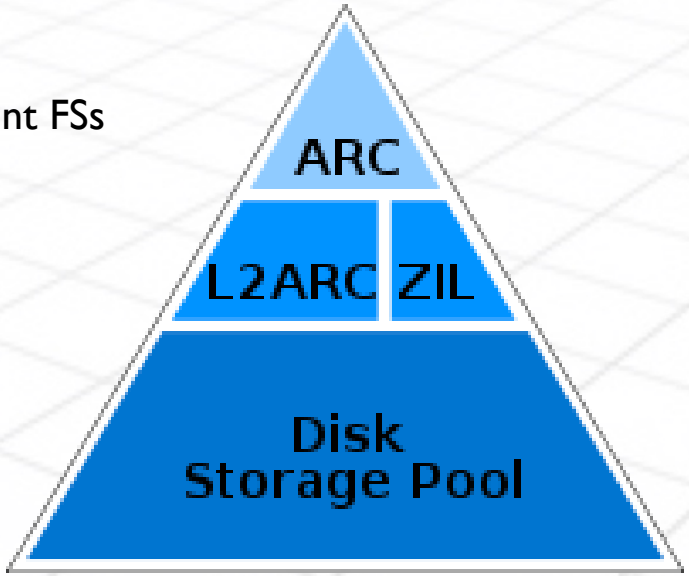
- Brief Review: Luster over ZFS
- Brief Overview: platforms used in example solutions
- Discuss three cases for SSD Hybrids in HPC and HPBC shops
- Summarize findings

Brief Review of Lustre/ZFS

- ZFS replaces ldiskfs/ext4 for backing FS
- It also provides a complete, feature-rich RAID solution
- ZFS also supports SSD/HDD hybrid modes natively
- LLNL developed this and uses it in systems like Sequoia –successfully
- Many other HPC shops have adopted the approach
- WARP provides commercial support and enhancements
  - Historically, using separated ZFS RAIDs connected to OSSs via RDMA
  - Now, using a fully integrated ZOL approach – Code from LLNL
  - Current-model WARP system has ZFS and Lustre running on a single controller
  - No more need for legacy RAIDs
  - Reduces cost and rack footprint; improves performance and MTBF

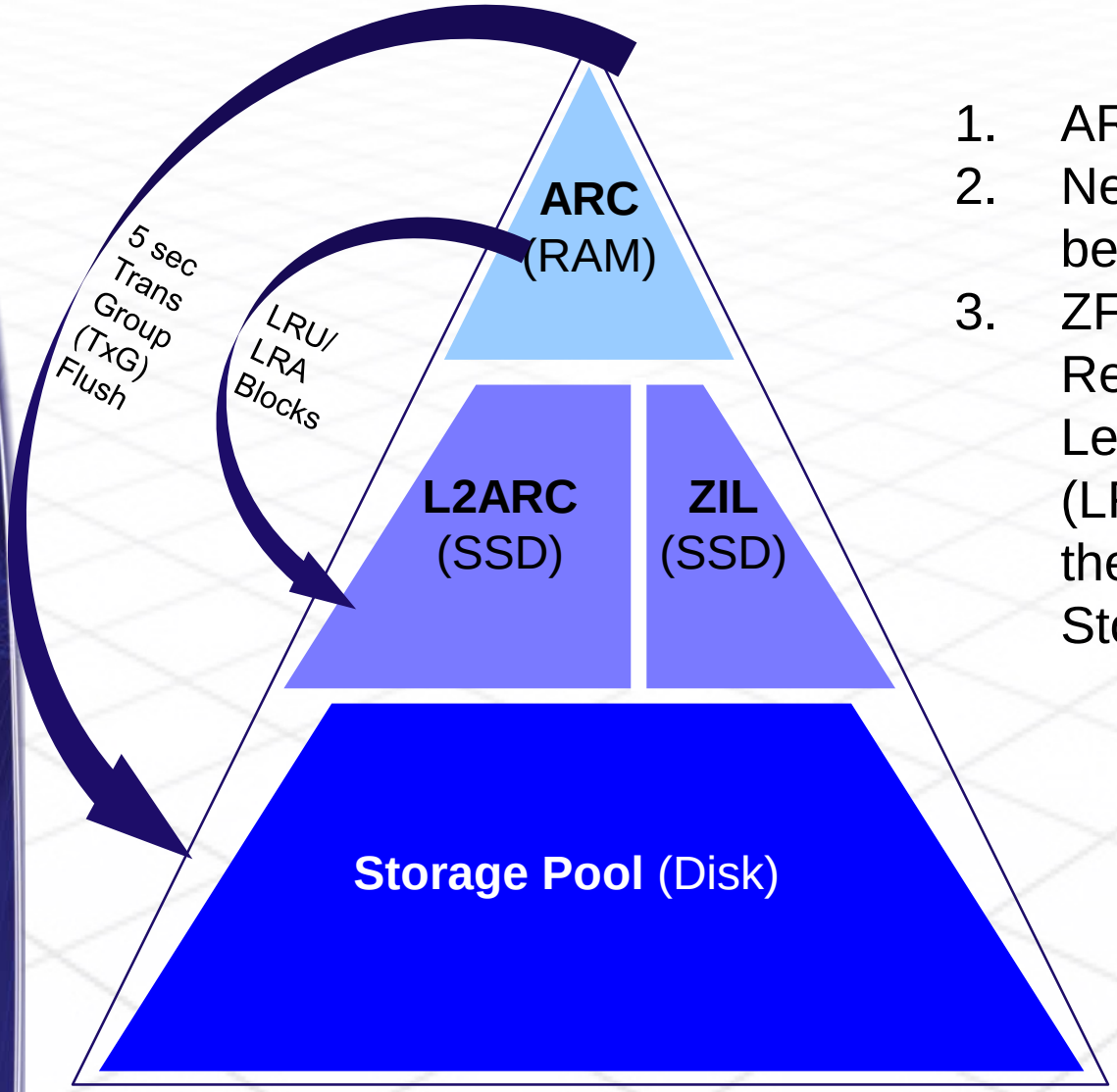
Brief Review of Lustre/ZFS (cont.)

- Read-optimized SSD – low cost
- 2nd location for read cache: if data is not in ARC, ZFS looks in L2ARC
- If L2ARC drives fail, data is secure on main storage
- Useful ratios:
  - Small, to cache just metadata
    - This is actually even useful for checkpoint FSs
  - Medium, for general workloads
  - Large (maybe 1:1), for analytics workloads



ZFS Model

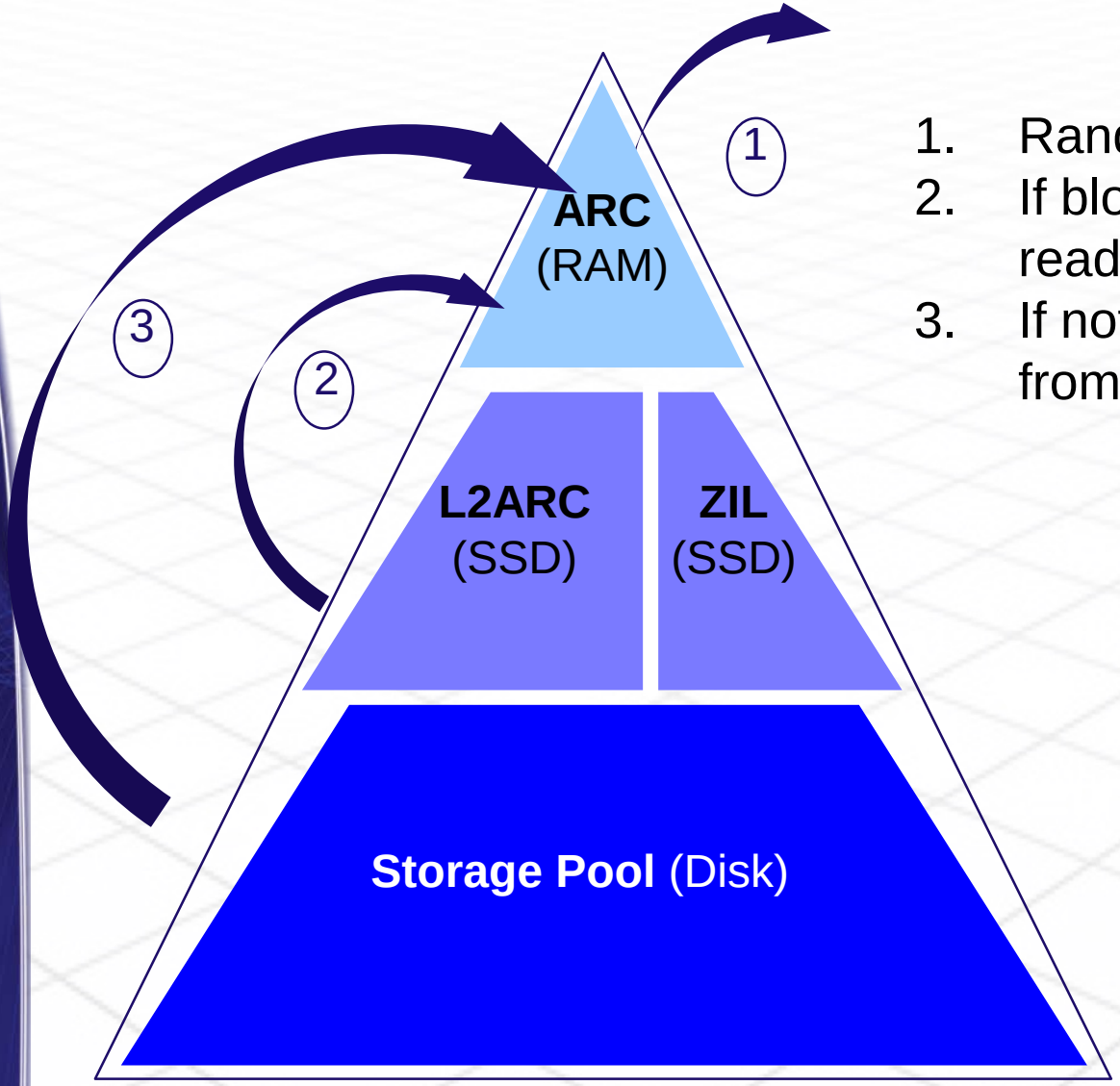
Brief Review of Lustre/ZFS (cont.)



1. ARC becomes full
2. New blocks arrive to be written
3. ZFS may move Least Recently Used (LRU) or Least Recently Accessed (LRA), or discard them as they already exist in the Storage Pool.

WARP MECHANICS®

Brief Review of Lustre/ZFS (cont.)



1. Random read from ARC
2. If block not in ARC, then read from L2ARC to ARC
3. If not in L2ARC, then read from Storage Pool to ARC

WARP WDS-8460 – primary building block

- 60x 4TB 6Gbps SAS drive modules per 4U enclosure (240TB)
  - Moving to 6TB drives (360TB/enclosure; up to 3.6PB/rack)
- Optional use of SSDs and NVRAM to create a hybrid HDD/SSD system
- Dual I/O controllers for redundancy and performance
- Hot-pluggable drives, I/O controllers, and power/cooling modules
- Dual sandy bridge CPU-based controllers or SAS JBOD modules
  - Upgradable to Ivy Bridge near term; dual Haswell mid term

WARP MECHANICS®

WARP WDS-8260 – SSD-optimized system

- 60x 6Gbps 2.5” SAS SSD or HDD modules per 2U enclosure
- Same dual I/O controllers as in 4u enclosure
- Up to 120TB pure SSD or HDD
- Up to 100TB as hybrid

- Use as turn-key HA MDS/MGS for Lustre
- Use as pure SSD system for IOPS intensive HPC or analytics workloads

Solution #1:

Separate Checkpoint vs. General Data

- Scenario:
  - Parallel FS is being used for multiple tasks
  - Read/write mix is 40/60 to 60/40
  - Reads are *not* from checkpoint or other classic HPC data, and may be interfering with it
  - Analysis shows that *some significant portion* of the reads are hitting the same data multiple times
- Solution:
  - Provide an "accelerated read" mount point for users with such workloads
  - Attach SSDs to the associated OSSs as L2ARC – ZFS read cache
- Benefit:
  - ZFS will selectively cache data onto the SSDs if there is a reason to suspect it will be read again
  - Doesn't just act like write through or back cache, or cache most recent writes
  - Allows using inexpensive, large, read-optimized SSDs without wear issues

Solution #2:

Analytics (Genomics / Life Sciences)

- Scenario:
  - Parallel FS is being used for single purpose which involves reading and re-reading files
  - Read/write mix is heavily weighted to reads... But...
  - After a time, data becomes inactive, yet must still be retained
  - So there is a known(ish) "working data set", with bulk storage being more like online archive
- Solution:
  - Add L2ARC slightly larger than the typical-case working data set (medium)
  - Bump up the fill rate, such that virtually all writes are cached
- Benefit:
  - ZFS will aggressively cache data onto the SSDs
  - This is sort of like a write through or back cache... But size is equal to entire working data set
  - Therefore typical case analysis jobs get ~100% SSD cache hits
  - Data gets "archived" when no longer active... Without even needing to move it!

Solution #3:

Analytics - HPBC / Financial

- Scenario:
  - Parallel FS is being used for single purpose: analyze market data for HFT
  - Read/write mix is almost read only, and all data is always active
  - RRDB style process is used to age out data, so there will be deletes but not changes
- Solution:
  - Add L2ARC equal to entire data set (large)
  - Bump up the fill rate to absolute max, such that all writes are cached
- Benefit:
  - ZFS will cache *all* written data onto the SSDs so analysis jobs get 100% cache hits
  - No write wear issue because ingest rate is low
  - Faster than a pure SSD solution with no added cost:
    - HDDs act as protection instead of having extra SSDs for parity
    - Reading from L2ARC is faster than reading from an SSD RAID

WARP MECHANICS®

Analytics - HPBC / Financial (cont.)

Interesting side note on this configuration re: SMB vs. Lustre:

- Customer initially had a layer of SMB gateways connected to Lustre
- This had performance and stability issues
  - Specific to Samba, not Lustre
  - "What I wouldn't give for a Windows Lustre driver!"
- Tried running Samba on OSSs; tried running it on ZFS systems without Lustre

Solution:

- Create CentOS hypervisor on bare metal – connect to Lustre as client
- Single Windows VM connects to hypervisor Samba, which shares Lustre
- Read only, so don't bother with CTDB etc.
- Now Samba has a "one client" scale instead of "thousands"

Findings

- Agree with ORNL: SSDs are not a “be all end all” for HPC
- If you can “steer” analytics workloads to separate FSs, then pure SSD may be fine
- But the killer app for [exascale | hyperscale] SSD systems seems hybrid
- Use SSDs to cache data which is expected to be “read mostly” or “RO”
- This allows using large commodity SSDs, without burning them out
- ZFS L2ARC can be combined with intelligent FS layout to do this economically

Thanks!

Q2-2014

Josh Judd, CTO