Lustre* I/O Performance on ZFS

Jinshan Xiong

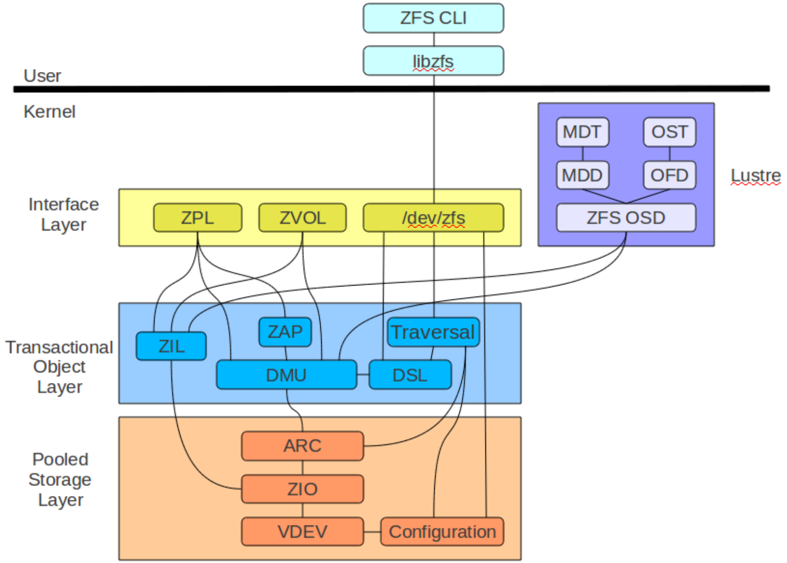
April, 2016

Agenda

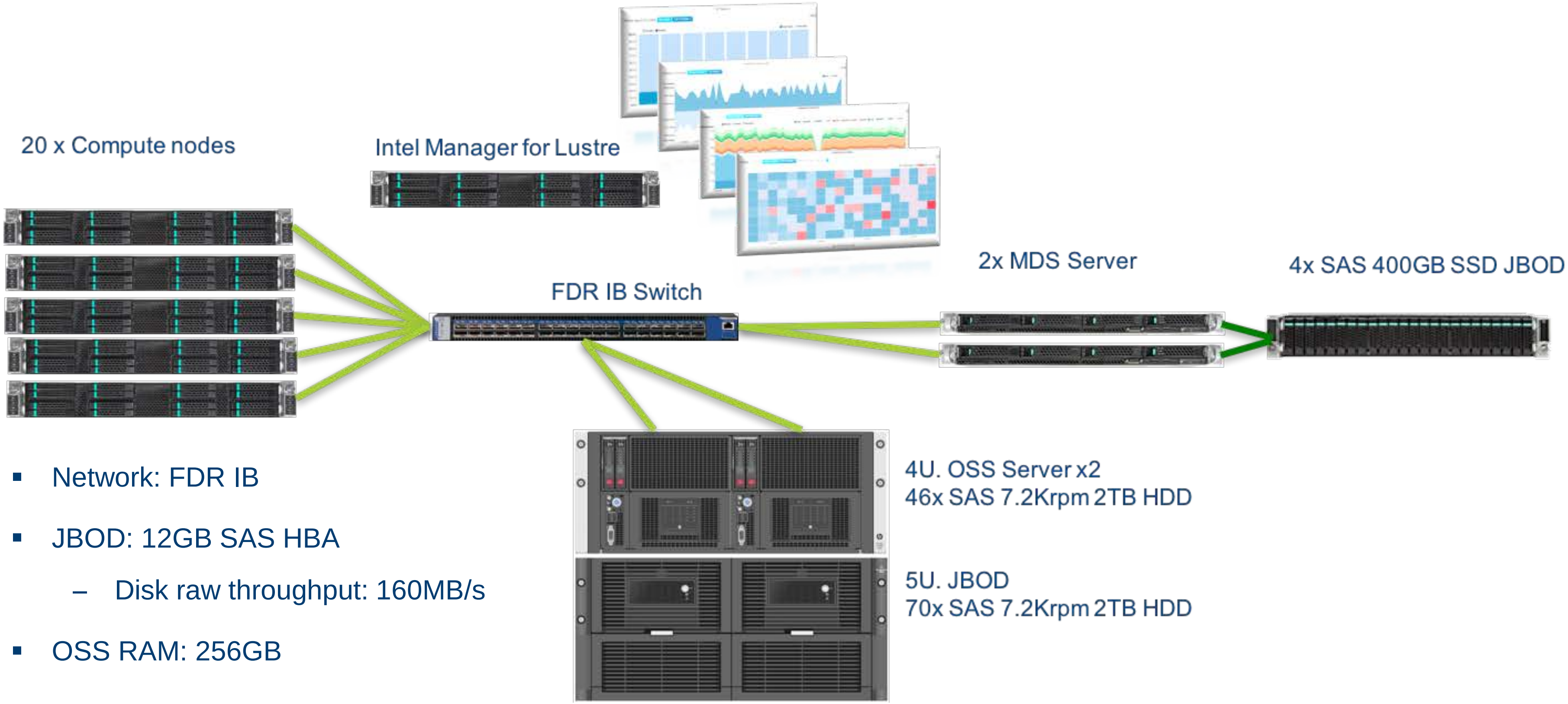
- Lustre* on ZFS
- Lustre performance on ZFS updates
- Review ZFS I/O Performance
  - Follow up ZFS slides from SDSC last year
- Future work

Lustre* on ZFS

- Why ZFS?
  - Superb write performance; writes are always sequential in ZFS
  - Always on-disk persistent
  - Built-in disks management
    - RAIDZ, mirror, etc.
  - Built-in block checksum
  - Built-in data scrub support
  - Metadata are duplicated for redundancy
  - …



Latest ZFS I/O Performance – Hardware

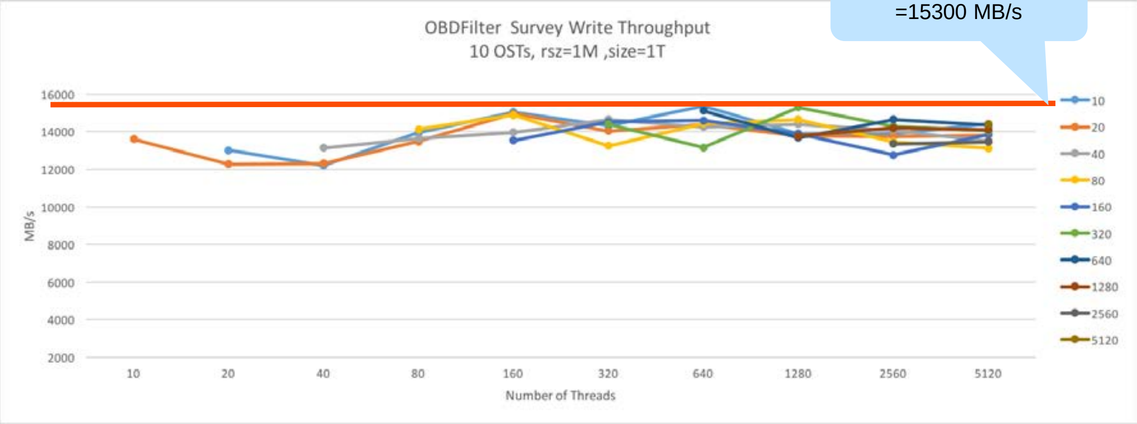


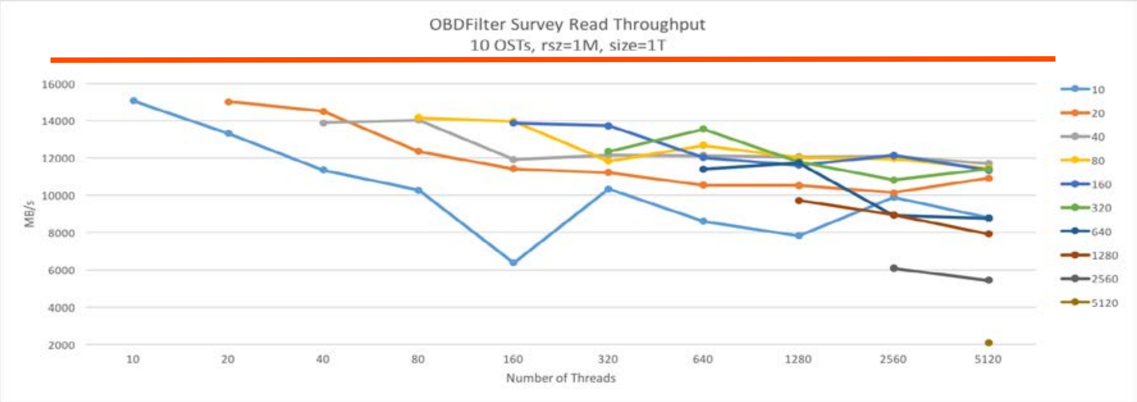
- Network: FDR IB
- JBOD: 12GB SAS HBA
  - Disk raw throughput: 160MB/s
- OSS RAM: 256GB

Latest ZFS I/O Performance – OBDFilter

- 10 OSTs - 9+2 RAIDZ2
- Single disk raw throughout
  - Write: ~170 MB/s
  - Read: ~190 MB/s
- Community release 2.8
- ZFS-0.6.4-92; record size: 1M
- RHEL 7.2
- Results
  - Write: 90 data disks deliver ~13GB/s

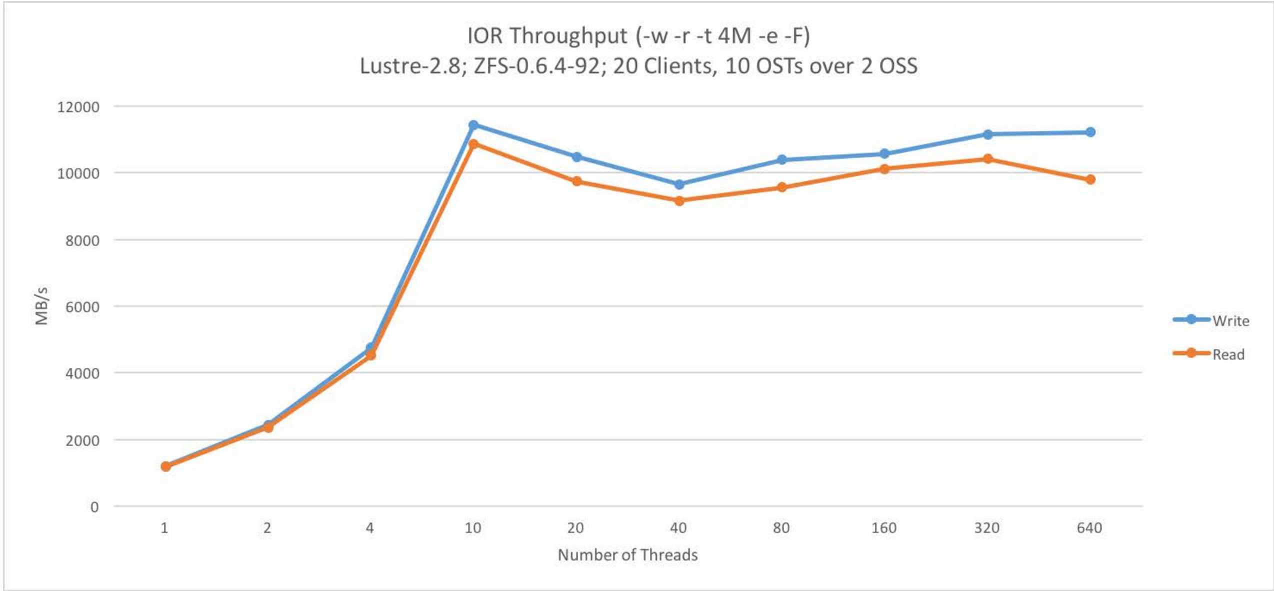
Theoretical Max Write =15300 MB/s


OBDFilter Survey Write Throughput
10 OSTs, rsz=1M ,size=1T


OBDFilter Survey Read Throughput
10 OSTs, rsz=1M, size=1T

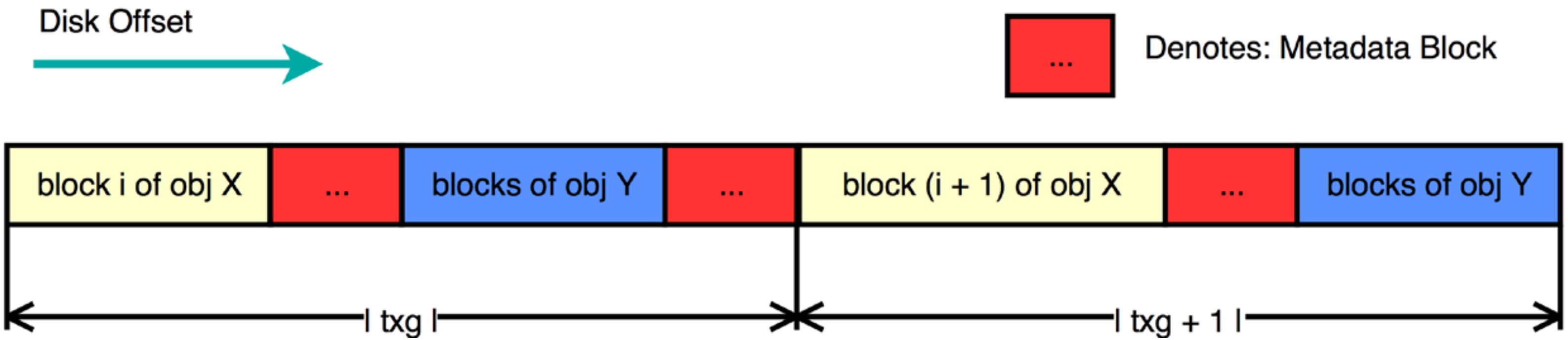
Latest ZFS I/O Performance - Lustre* Clients

- 10 OSTs, 9+2 RAIDZ
  - 110 disks in total, 90 data disks deliver 11GB/s
- ZFS 0.6.4-92
  - 1MB record size
  - 4KB sector size
  - Why? LU-7404
- Lustre 2.8



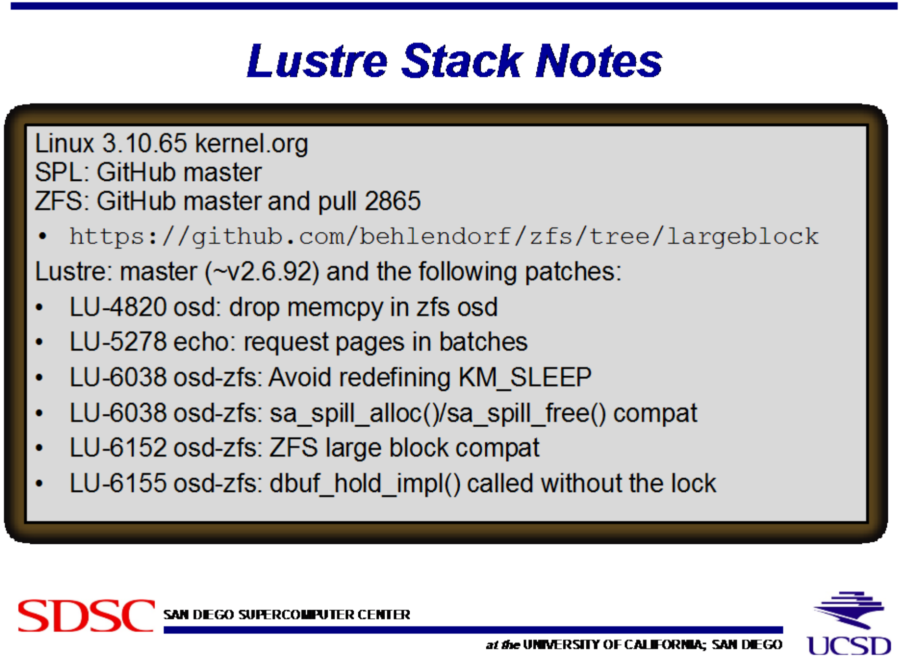
ZFS Read Problems

- No file aware block allocation
  - Blocks written sequentially may spread around the whole pool
  - Lots of disk seek to read them back
- This is why read is usually slower than write
- Bigger block size would mitigate this problem



Tickets Status Review

- Patches that have been landed into 2.8
  - LU-4820, LU-5278, LU-6038, LU-6152, LU-6155
- In progress: LU-7404
  - Identified commit 'Illumos 5497 - lock contention on arcs_mtx' caused I/O timeout problem
  - Still work with upstream developers
  - This is why 2.8 used ZFS-0.6.4.2



Lustre Stack Notes

Linux 3.10.65 kernel.org
SPL: GitHub master
ZFS: GitHub master and pull 2865

- `https://github.com/behlendorf/zfs/tree/largeblock`

Lustre: master (~v2.6.92) and the following patches:

- LU-4820 osd: drop memcpy in zfs osd
- LU-5278 echo: request pages in batches
- LU-6038 osd-zfs: Avoid redefining KM_SLEEP
- LU-6038 osd-zfs: sa_spill_alloc()/sa_spill_free() compat
- LU-6152 osd-zfs: ZFS large block compat
- LU-6155 osd-zfs: dbuf_hold_impl() called without the lock

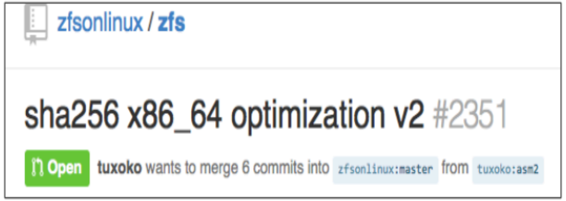
SDSC SAN DIEGO SUPERCOMPUTER CENTER
at the UNIVERSITY OF CALIFORNIA; SAN DIEGO
UCSD

Fast Checksum Computation

- Use AVX2 to compute Fletcher-4 checksum
- Compute RAIDZ parity with AVX2 is also in progress

Help is on the way!

- Work started on AVX(2) optimizations for checksums
- Hoping to see this extended to parity

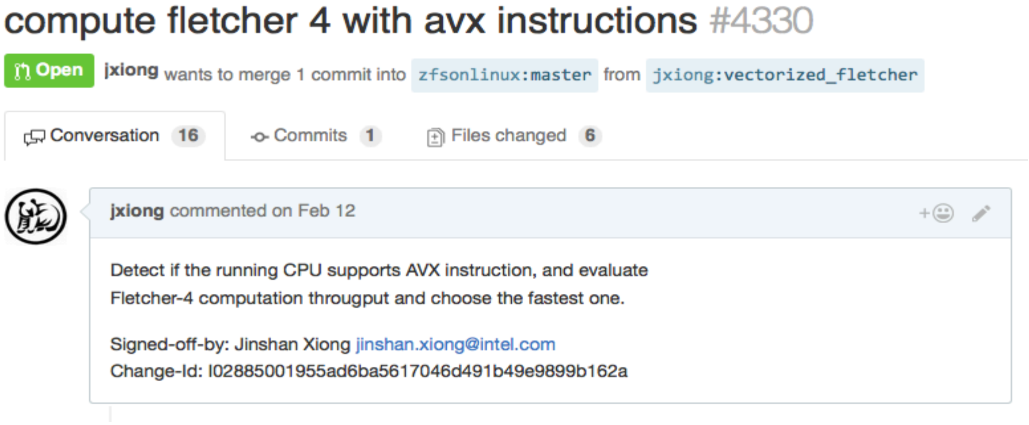

zfsonlinux / zfs

sha256 x86_64 optimization v2 #2351

Open tuxoko wants to merge 6 commits into zfsonlinux:master from tuxoko:asm2

https://github.com/zfsonlinux/zfs/pull/2351

SDSC SAN DIEGO SUPERCOMPUTER CENTER
at the UNIVERSITY OF CALIFORNIA; SAN DIEGO
UCSD

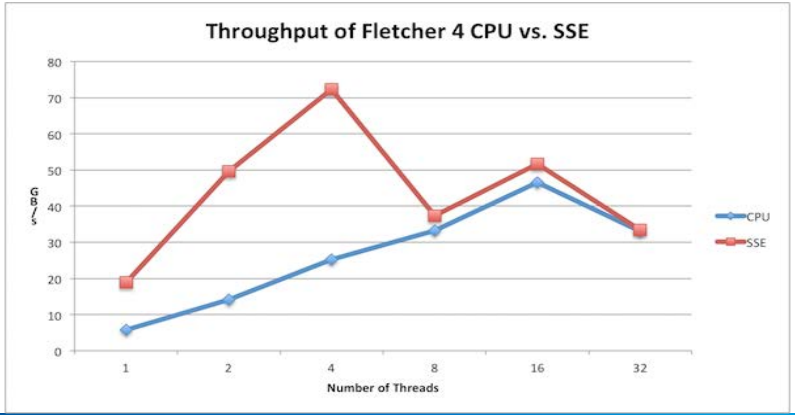

compute fletcher 4 with avx instructions #4330

Open jxiong wants to merge 1 commit into zfsonlinux:master from jxiong:vectorized_fletcher

jxiong commented on Feb 12

Detect if the running CPU supports AVX instruction, and evaluate Fletcher-4 computation througput and choose the fastest one.

Signed-off-by: Jinshan Xiong jinshan.xiong@intel.com
Change-Id: I02885001955ad6ba5617046d491b49e9899b162a



Work in Progress

- Development in progress for CORAL project
  - Large block size
  - Parity Declustered RAID - dRAID
  - Separate MD Allocation Class
- All work being upstreamed to ZFS-on-linux project when completed
  - Will likely become available Lustre* ~2.11 Community Release

ZFS 16M Block Size

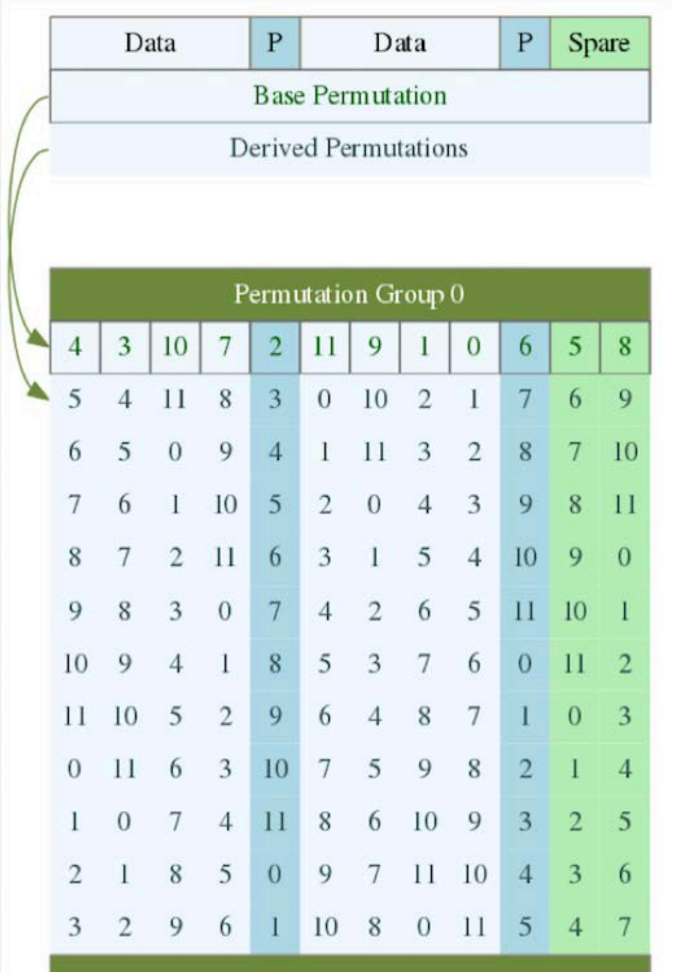
- ZFS now supports up to 16MB block size
  - Lustre* will support 16M RPC size to ensure large block size for ZFS
  - Problems with ZFS memory management
    - Large ARC data buffers are vmalloc() based slabs
    - Use scatter/gather page list to store ARC data
    - Compressed ARC buffer may help a little bit
- Dynamic block OSD-ZFS size is necessary to reduce overhead on small files
  - Avoid the overhead of read-modify-write
  - Application can set block size
  - OSD-ZFS can choose block size by I/O pattern

Why Large Block Size?

- Considering a 8+2 RAIDZ2 again
  - For a 1MB block size, every data drive will store 128KB data
    - Small I/O hurts performance
  - With 16MB block size, we can guarantee 2MB data on each drive
- Deliver better read performance

ZFS dRAID

- Faster rebuild/resilver time
  - Spare blocks are distributed over all disks
  - Short time leads to less risk on data loss
    - 2nd or 3rd disk failure during rebuild time
- Reasonable throughput in degraded mode
  - Lost one disk -> lose 1/N disk bandwidth
- Permutation development based on randomly generated initial permutation



Base Permutation

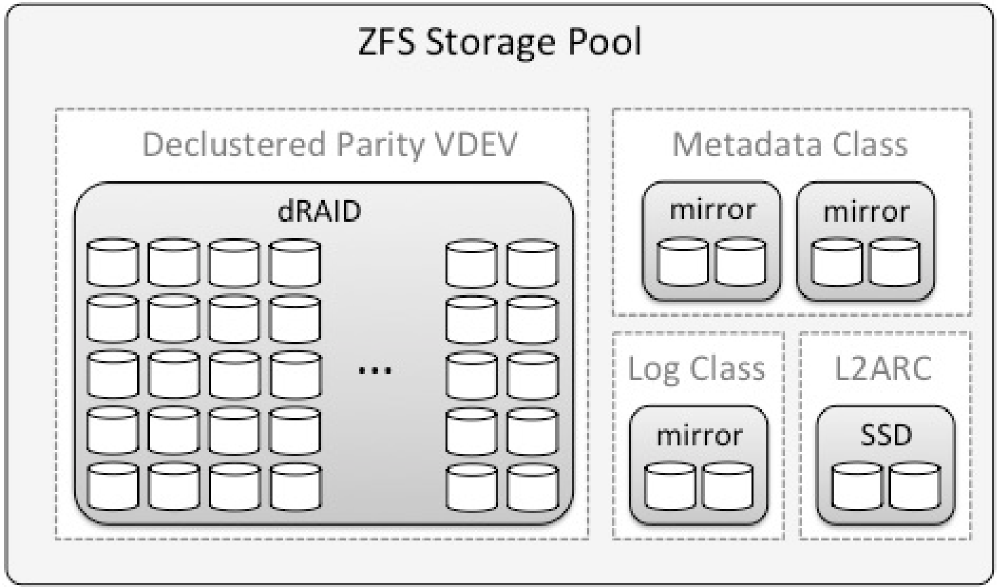
Derived Permutations

Permutation Group 0

Data				P	Data				P	Spare	
4	3	10	7	2	11	9	1	0	6	5	8
5	4	11	8	3	0	10	2	1	7	6	9
6	5	0	9	4	1	11	3	2	8	7	10
7	6	1	10	5	2	0	4	3	9	8	11
8	7	2	11	6	3	1	5	4	10	9	0
9	8	3	0	7	4	2	6	5	11	10	1
10	9	4	1	8	5	3	7	6	0	11	2
11	10	5	2	9	6	4	8	7	1	0	3
0	11	6	3	10	7	5	9	8	2	1	4
1	0	7	4	11	8	6	10	9	3	2	5
2	1	8	5	0	9	7	11	10	4	3	6
3	2	9	6	1	10	8	0	11	5	4	7

Separate MD Allocation Class

- Metadata blocks are with smaller size, and accessed more frequently
- A dedicated VDEV with high IOPS drives to store metadata
  - SSD or NVRAM
  - Mirrored for redundancy
- Better use of SSD than L2ARC



Why Separate MD Class?

- Loading metadata faster helps deliver better I/O performance
  - Lower read latency
  - Faster scrub/resilver
- Considering a 8+2 RAIDZ2 device
  - Metadata block size varies from 512B to 16KB in ZFS
  - For a 16KB metadata block, 8 data disks will store 2KB on each
  - Small I/O hurts read perf due to 2KB read from each disk for a data buffer

Legal Information

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps

Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit http://www.intel.com/performance.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at http://www.intel.com/content/www/us/en/software/intel-solutions-for-lustre-software.html.

Intel technologies may require enabled hardware, specific software, or services activation. Check with your system manufacturer or retailer.

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

Intel may make changes to specifications and product descriptions at any time, without notice. Designers must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined". Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The information here is subject to change without notice. Do not finalize a design with this information.

Intel, the Intel logo and Intel® Omni-Path are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

intel®

intel®

experience
what's inside™