dRAID: Parity Declustered RAID for ZFS

Isaac Huang

The Problem

RAIDz has many great features, but resilver is slow:

- No write hole, self healing, no special hardware

Pool config	Age	Used	Avail	Resilvered	Time	Speed
3x 12-drive RAIDz2	2 Y	120T	23.6T	4.24T	27h2m	45.68 MB/s
1x 4-drive RAIDz1	4 Y	6.42T	1.23T	2.29T	45h45m	14.58 MB/s
3x 6-drive RAIDz1		8.47T		560G	6h32m	24.00 MB/s
5x 6-drive RAIDz1		109T		3.24T	44h52m	21.03 MB/s

RAIDz Resilver Time

The Causes

Why is RAIDz resilver slow?

- Random IO due to block pointer tree traversal
  - Some mitigation with “sequential resilver” work
- Write throughput of a single replacement drive is a bottleneck
- A single VDEV doesn’t scale to a large number of child VDEVs
  - Aggregated read throughput for reconstruction is limited

The “Solutions”

Best “solutions” available today

- Reduce the number of block pointers by using large (1MB+) blocks
- Increase available total IOPS
  - Use narrow RAIDz (for example: 4-wide RAIDz1 or mirroring)
  - Use lots of small disks or SSDs
- None of these solutions address:
  - Single replacement drive bottleneck
  - Limited aggregate read throughput from child VDEVs

dRAID: No BP Tree Traversal

RAIDz1

LBA \ Disk	A	B	C	D	E
0	P_0	D_0	D_1	D_2	D_3
1	P_1	D_4	D_5	D_6	D_7
2	P_0	D_0	D_1	D_2	P_0
3	D_0	P_0	D_0	D_1	X

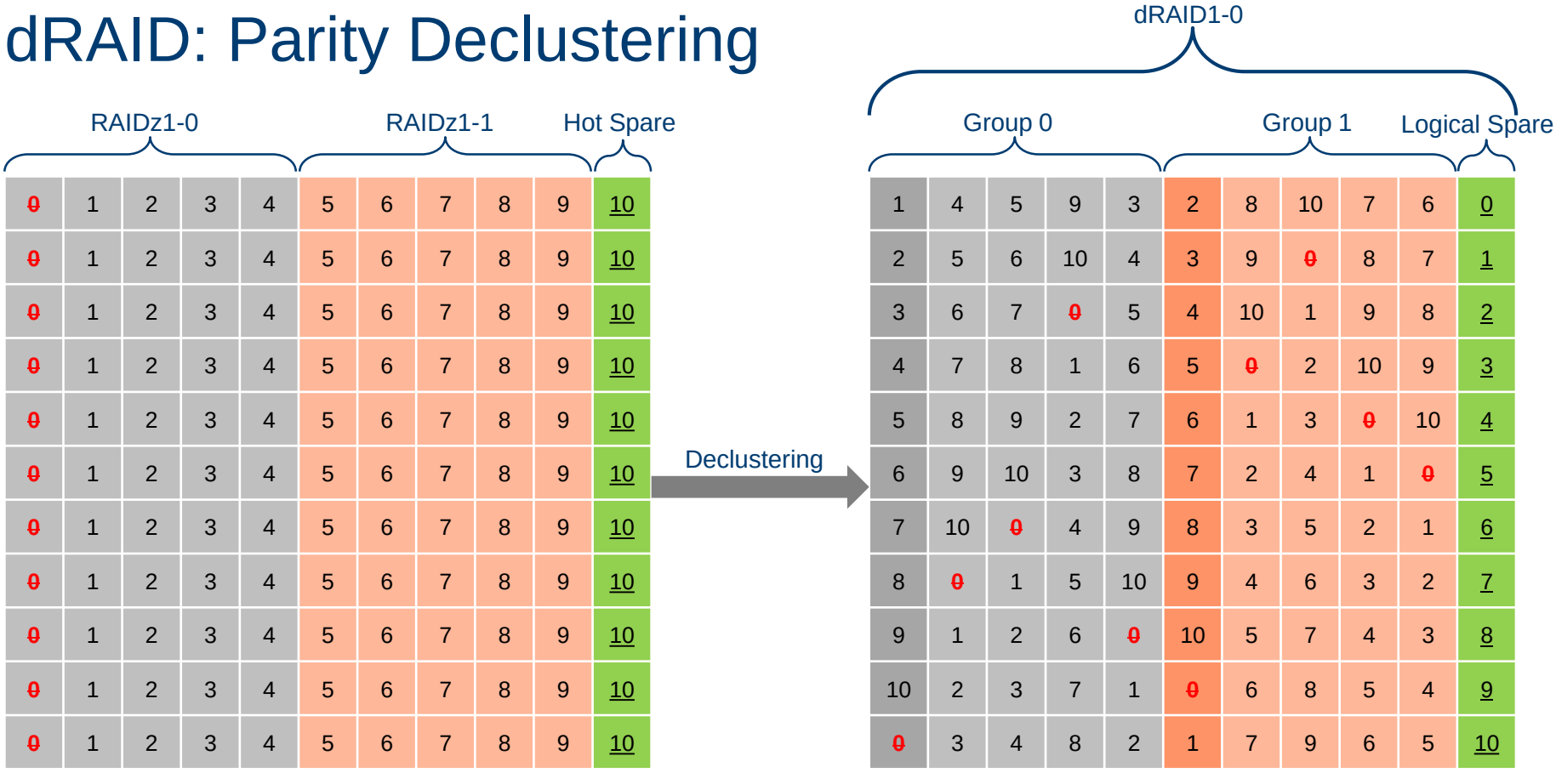
dRAID1

LBA \ Disk	A	B	C	D	E
0	P_0	D_0	D_1	D_2	D_3
1	P_1	D_4	D_5	D_6	D_7
2	P_0	D_0	D_1	D_2	X
3	P_0	D_0	X	X	X
4	P_0	D_0	D_1	X	X

dRAID: Sequential Rebuild

- No block pointer tree traversal
  - Completely sequential like traditional RAID-4/5/6
  - Skips free space by scanning spacemap objects
  - Large IO: not limited by block boundary
- No write hole, like RAIDz-1/2/3
  - RAIDz: variable stripe width
  - dRAID: fixed stripe width, but always allocate full stripes

dRAID: Parity Declustering



dRAID: Parity Declustering

Rebuild scales to a large number of drives

- Decouple redundancy group (P+D) from the number of child drives
- Write: spare blocks are rotated among all drives
- Read: shared evenly among all drives

IO \ Drive	1	2	3	4	5	6	7	8	9	10
Read	4	4	4	4	4	4	4	4	4	4
Write	1	1	1	1	1	1	1	1	1	1

Rebuild IO Distribution

dRAID: Demo

Live demo of dRAID rebuild:

- 43-drive dRAID2: 4 x (8D + 2P) groups, 3 distributed spares
  - 601G used out of 46T
  - Each drive capable of 150 MB/s
- Rebuild 1 failed drive to a distributed spare
  - Read 157.2G, write 17.5G

dRAID: Demo Results

Rebuild completed in 37 seconds:

- Aggregate throughput: read 4350.6 MB/s, write 484.3 MB/s
  - 3x faster than RAIDz resilver
  - Scales to more drives
- Combined read/write of a single drive at 115.12 MB/s

dRAID Downsides: Space Inflation

RAIDz1

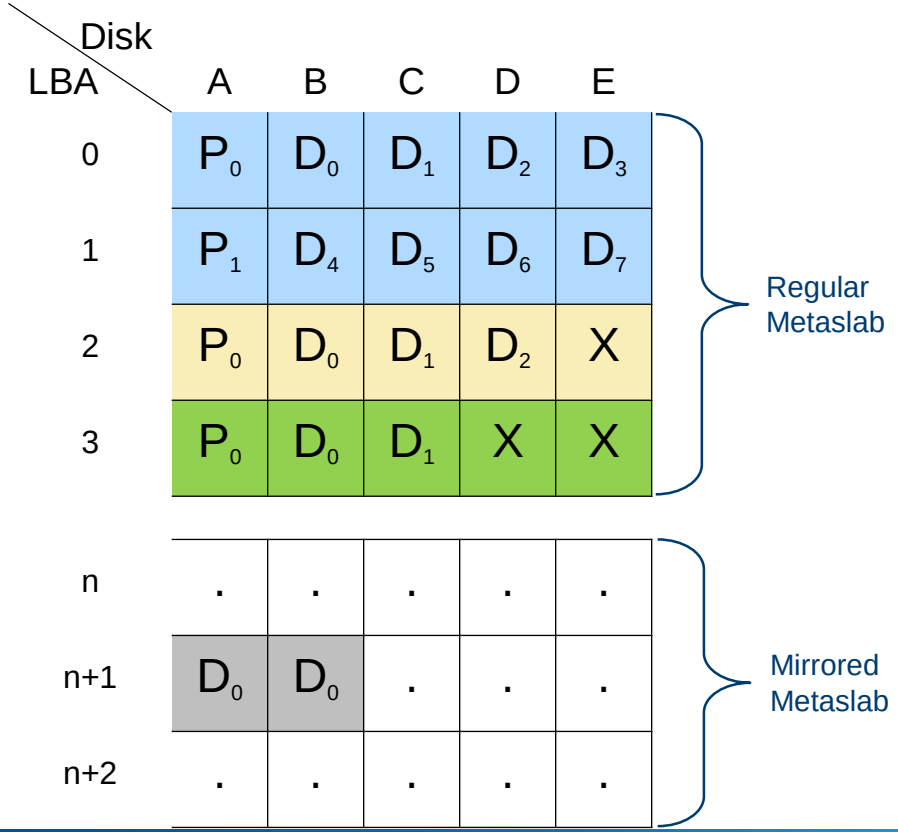
LBA \ Disk	A	B	C	D	E
0	P_0	D_0	D_1	D_2	D_3
1	P_1	D_4	D_5	D_6	D_7
2	P_0	D_0	D_1	D_2	P_0
3	D_0	P_0	D_0	D_1	X

dRAID1

LBA \ Disk	A	B	C	D	E
0	P_0	D_0	D_1	D_2	D_3
1	P_1	D_4	D_5	D_6	D_7
2	P_0	D_0	D_1	D_2	X
3	P_0	D_0	X	X	X
4	P_0	D_0	D_1	X	X

dRAID Space Inflation: Mirrored Metaslab

LBA \ Disk	A	B	C	D	E
0	P_0	D_0	D_1	D_2	D_3
1	P_1	D_4	D_5	D_6	D_7
2	P_0	D_0	D_1	D_2	X
3	P_0	D_0	X	X	X
4	P_0	D_0	D_1	X	X



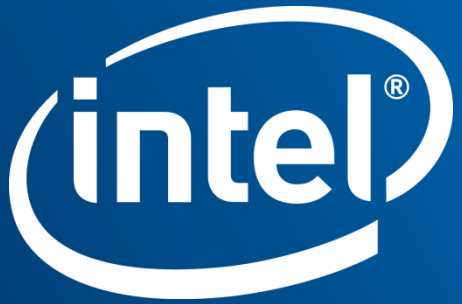
dRAID: Downsides

- Allocated block size inflation
  - Doesn’t work well for small blocks
  - Mitigated by mirrored metaslabs
- Rebuild cannot verify block checksum:
  - Still need to verify checksums by traversing BP tree
- Parity group and spare capacity chosen at VDEV creation

dRAID: Project Status

Feature complete:

- Code: https://github.com/zfsonlinux/zfs/pull/5841
- Document: https://github.com/zfsonlinux/zfs/wiki/dRAID-HOWTO
- Need community help: reviewing, testing, patching, porting.


intel®

dRAID: Demo

```
NAME          STATE     READ WRITE CKSUM
tank          ONLINE       0     0     0
  draid2-0    ONLINE       0     0     0
    sdb       ONLINE       0     0     0
    sdd       ONLINE       0     0     0
    sde       ONLINE       0     0     0
    ……
    sdar      ONLINE       0     0     0
spares
  $draid2-0-s0  AVAIL
  $draid2-0-s1  AVAIL
  $draid2-0-s2  AVAIL
```

dRAID: Demo

```
# zpool offline tank sde

# zpool replace tank sde '$draid2-0-s1

# zpool status

 scan: rebuilt 17.4G in 0h0m36s with 0 errors

 tank                DEGRADED     0     0     0

         draid2-0          DEGRADED     0     0     0

           sdb             ONLINE       0     0     0

           sdd             ONLINE       0     0     0

           spare-2         DEGRADED     0     0     0

             sde           OFFLINE      0     0     0

             $draid2-0-s1  ONLINE       0     0     0
```

dRAID: Demo


Read Bandwidth
10/9/2017
9:56:25 PM
200
180
160
140
120
100
80
60
40
20
0
21:54:50
21:54:55
21:55:00
21:55:05
21:55:10
21:55:15
21:55:20
21:55:25
21:55:30
21:55:35
21:55:40
21:55:45
21:55:50
21:55:55
21:56:00
21:56:05
21:56:10
21:56:15
21:56:20
sdb
sdc
sdd
sde
sdf
sdg
sdh
sdi
sdj
sdk
sdl
sdm
sdn
sdo
sdp
sdq
sdr
sds
sdt
sdu
sdv
Write Bandwidth
10/9/2017
9:56:25 PM
140
120
100
80
60
40
20
0
21:54:50
21:54:55
21:55:00
21:55:05
21:55:10
21:55:15
21:55:20
21:55:25
21:55:30
21:55:35
21:55:40
21:55:45
21:55:50
21:55:55
21:56:00
21:56:05
21:56:10
21:56:15
21:56:20
sdb
sdc
sdd
sde
sdf
sdg
sdh
sdi
sdj
sdk
sdl
sdm
sdn
sdo