RAID-Z

Adam Leventhal, Delphix

RAID-Z: RAID-5 For ZFS

Adam Leventhal, Delphix

RAID-Z: RAID-5 For ZFS

(Sort of)

Adam Leventhal, Delphix

Everything you could possibly want to know about RAID-Z and probably quite a bit more if you'll indulge me.

Adam Leventhal, Delphix

Everything you always wanted to know about RAID-Z*

*But were afraid to ask

Adam Leventhal, Delphix

What Is RAID?

- Redundant Array of Inexpensive Disks

  or

- Redundant Array of *Independent* Disks

- Coined in 1988
  - Descriptive rather than prescriptive
  - Changed when “inexpensive” became too hilarious

Several Different RAID Levels

RAID-0	striping (no actual redundancy)
RAID-1	mirroring
RAID-4	multiple blocks in a stripe share a parity block
RAID-5	same as RAID-4, but parity is rotated between disks
RAID-6	same as RAID-5, but with double parity

Several Different RAID Levels

RAID-0	striping (no actual redundancy)
RAID-1	mirroring
RAID-2	DRAM-style ECC (K data disks + log(K) parity disks)
RAID-3	blocks are carved up and written to multiple disks in a parity-protected stripe
RAID-4	multiple blocks in a stripe share a parity block
RAID-5	same as RAID-4, but parity is rotated between disks
RAID-6	same as RAID-5, but with double parity
RAID-7.N	RAID with N parity disks
RAID-7	generalized M+N RAID

Why RAID-Z?

- Software RAID-5 stinks
- "RAID-5 write hole" when rewriting a stripe:
  - Read existing parity
  - Write new data
  - Write updated parity
- Special hardware required: NV-DRAM
- Software RAID-5 is slow or unsafe
- ZFS is designed to need no special hardware

What is RAID-Z?

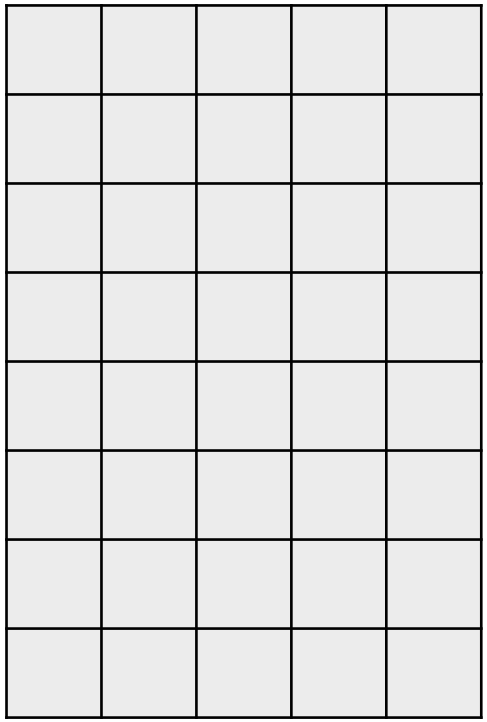
- No in-place modifications
- Variable-width stripes / full-stripe writes
- Distributed parity like RAID-5
- Three flavors
  - Single-parity (2005): like RAID-5
  - Double-parity (2006): like RAID-6
  - Triple-parity (2009): RAID-7.3

RAID-Z Idiosyncrasies

- Space accounting
- Skipped sectors v. performance
- Resilvering
- Random IOPS

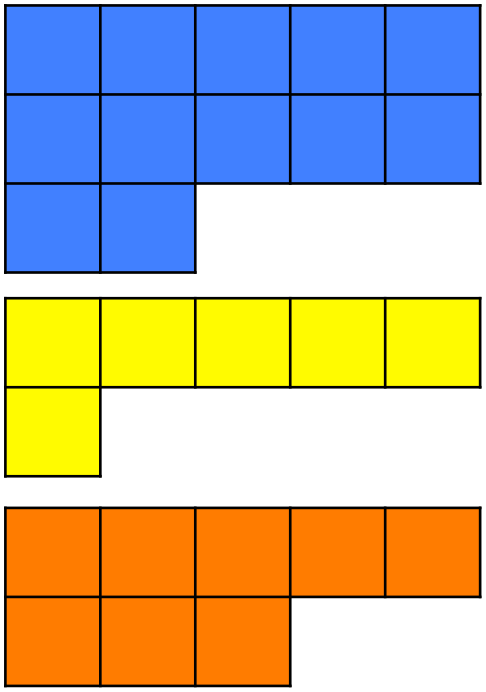
Space Accounting

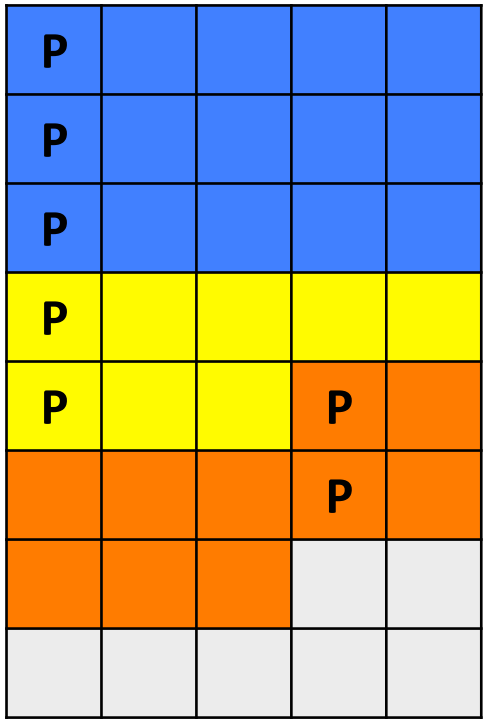
- Disks are divided into sectors
- Columns represent different disks
- Rows represent different sectors



Space Accounting

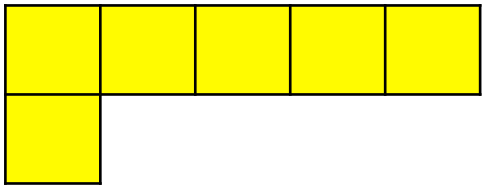
- Write:

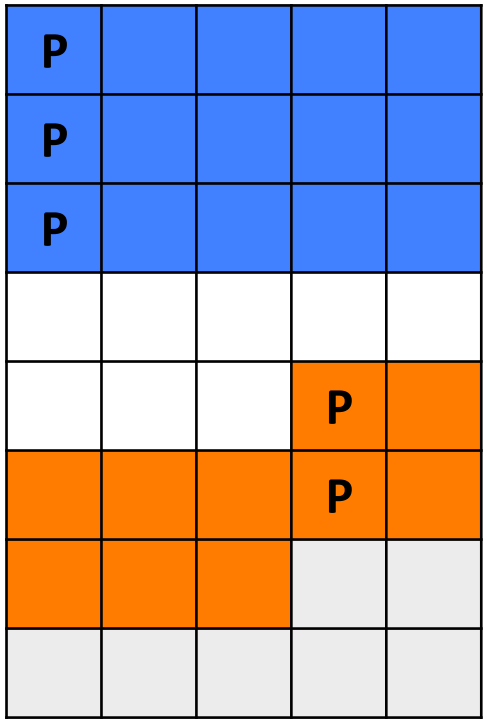




Space Accounting

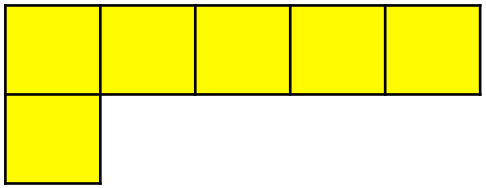
- Free:



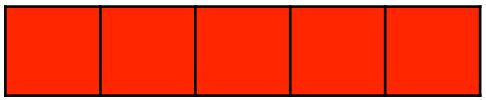


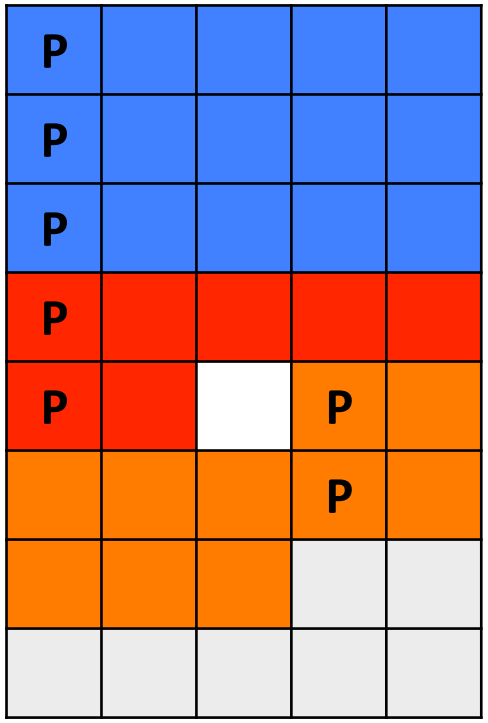
Space Accounting

- Free:



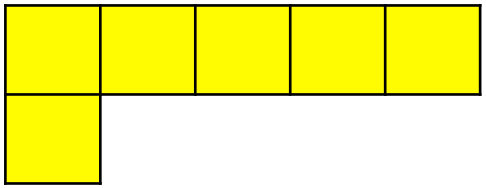
- Write:



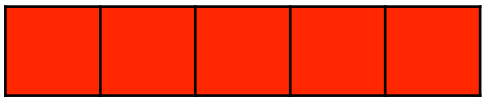


Space Accounting

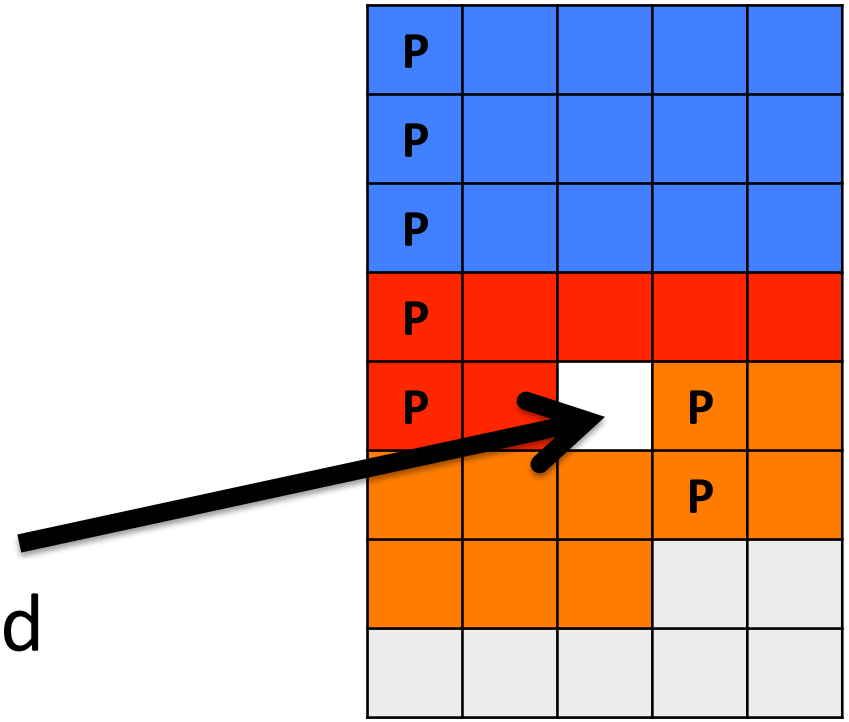
- Free:



- Write:

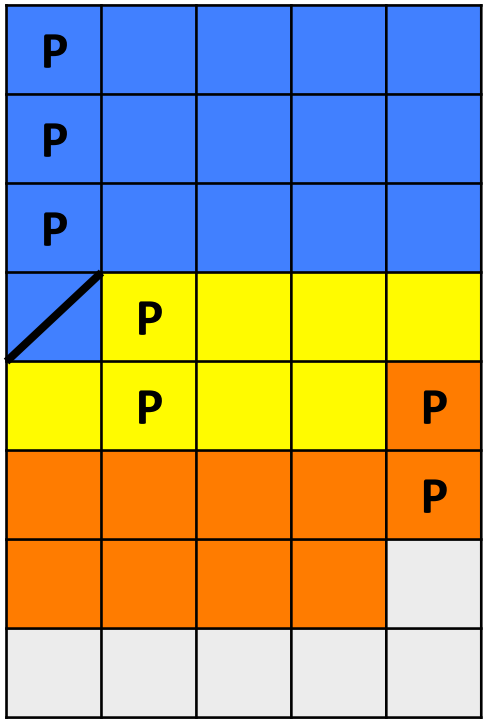


- This sector is “free”, but can never be used



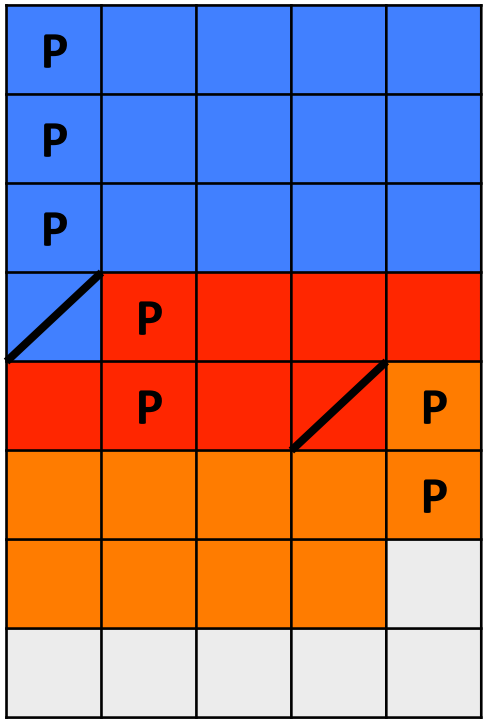
Space Accounting

- Solution: round up to nearest (nparity + 1) and skip unused sectors
- Skipped sectors ensure there are never free sectors that can never be used



Space Accounting

- Solution: round up to nearest (nparity + 1) and skip unused sectors
- Skipped sectors ensure there are never free sectors that can never be used



Space Accounting

- Skipped sectors are important so that we don't "lose" space
- Variable width stripes are needed to avoid the RAID-5 write hole
  - How many parity blocks per row?

- 4 + 1 RAID-Z x 1T HDD = ???
- Well, that depends on ***how*** you write

Skipped Sectors v. Performance

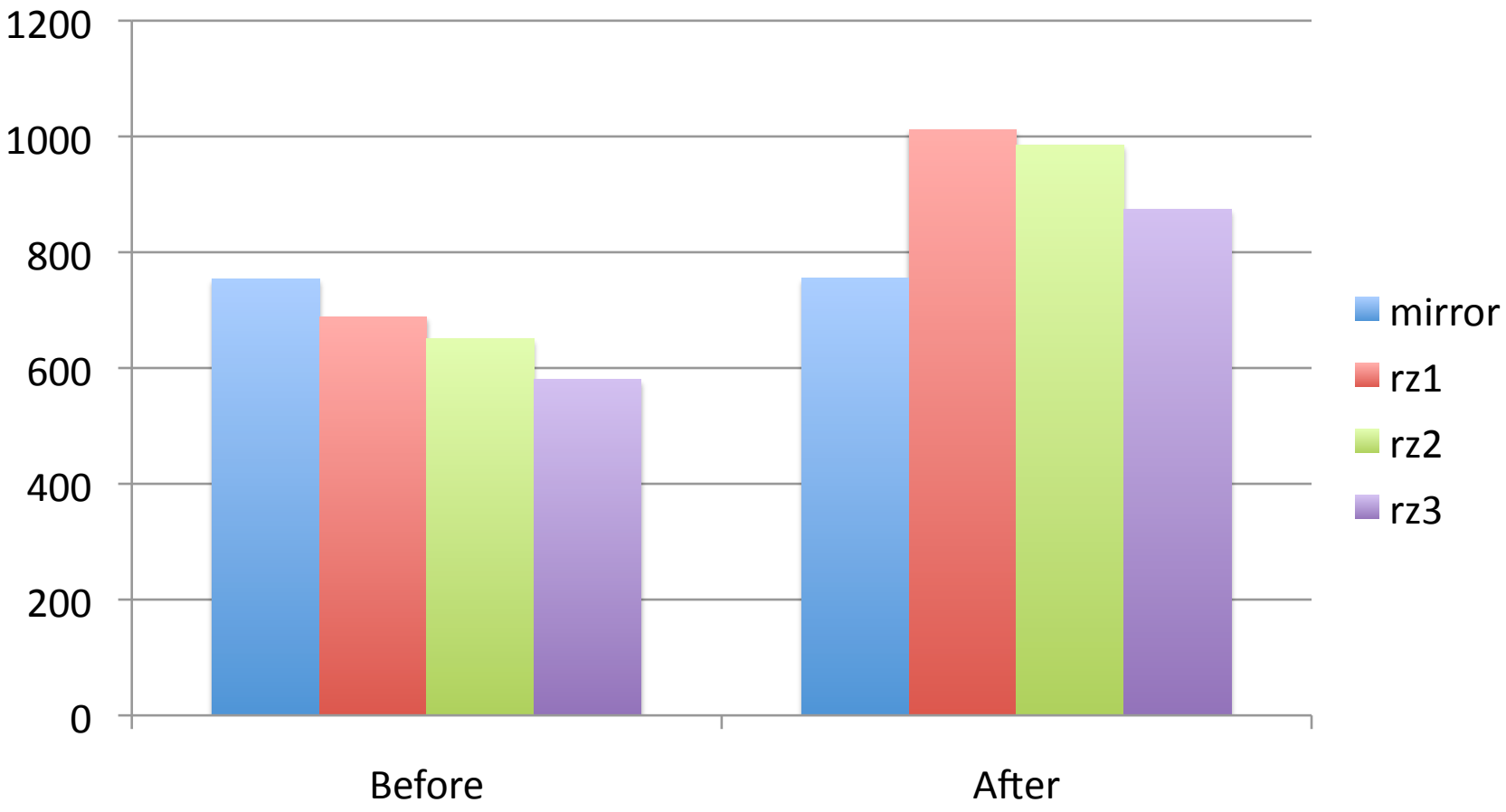
- Skipped sectors for space accounting create a new problem
- Data on an individual disk looks like this:

- Reads and writes are small (random v. stream)
- Impedes ZFS IO aggregation

Skipped Sectors v. Performance

- Reads: just read more than we needed if it helps create big, contiguous chunks
  - "mind the gap"
- Writes: a little trickier
  - Can't just overwrite – those sectors might be in use!
  - But we know when we skip a sector
  - Generate ***optional IOs*** to aid aggregation

Skipped Sectors v. Performance



Sun Storage 7410, 48 x 1T 7200 RPM SATA (2009)
Multi-threaded streaming write workload (MB/s)

Resilvering

- Traditional RAID: blithely XOR drives together
- RAID-Z: walk metadata to discover layout

- Pros: don't have to touch free sectors
  great for less-full storage pools
- Cons: many random IOPS to read metadata
  O(total metadata) not O(data to resilver)

Random IOPS

- RAID-3 spread a block between disks
  - Each read or write touches all disks in a stripe
- RAID-4 improved upon RAID-3
  - Writing a block modifies one disk, updates parity
  - Reading a block accesses just one disk
- RAID-Z is closer to RAID-3 than to RAID-4
- For stripe width ***N***, a RAID-Z stripe has ***1/N*** as many IOPS as RAID-5

Do Random IOPS Matter?

2001

200 IOPS

Do Random IOPS Matter?

2001
200 IOPS

2010
35,000 IOPS

Flash and NV Storage

- Flash has many many more random read IOPS
- … but we move to flash because we want to use them, not waste them!
- … but can we take advantage of many IOPS x many SSDs?
- … and how does the L2ARC change the random IOPS load on our disks?

Summing Up

- RAID-Z is not exactly RAID-5 (or RAID-6)
- Some gotchas to keep in mind when deploying RAID-Z or analyzing performance
- Flash may change the picture for you

- Would ubiquitous flash or NV-DRAM eliminate the need for RAID-Z?

Questions?

Links:
http://blogs.sun.com/bonwick/entry/raid_z
http://dtrace.org/blogs/ahl/2006/06/18/double-parity-raid-z
http://dtrace.org/blogs/ahl/2009/07/21/triple-parity-raid-z
http://dtrace.org/blogs/ahl/2009/12/21/acm_triple_parity_raid/
http://blogs.sun.com/bonwick/entry/space_maps
http://dtrace.org/blogs/ahl/2010/07/21/what-is-raid-z
http://queue.acm.org/detail.cfm?id=1317400

Adam Leventhal, Delphix
ahl@delphix.com
twitter: @ahl
blog: dtrace.org/blogs/ahl