

Why are my (ZFS) disks so noisy?



(and how to build your own Caddy on the cheap)

Hi! I'm Jim Salter.

- Mercenary sysadmin since the 90s
- ZFS guy since 2008
(FreeBSD 7.0-RELEASE)
- ZFS on Linux guy since 2010
(Ubuntu Lucid)
- Written for Ars Technica,
Wirecutter, Smallnetbuilder, etc
- “The Sanoid guy” / “The biggest
ZFS stan on the planet”



Why so much platter chatter?

- Chassis model?
- Drive model?
- txg_sync_interval?
- Prefetch?
- Phase of the moon?
- Host operating system?
- Hypersensitive ears?



simple answers only!



One piece at a time

- Simple questions don't always have simple answers
- Simple problems don't always have simple solutions
- Complex questions don't always need complex answers
- Complex problems don't always need complex solutions







“How does the TV work?”




The Problem Space

- OpenZFS
- 12 mechanical drives
- Unspecified chassis
- Proxmox (unspecified, likely 7.x or 8.0)
- Unspecified VM workload
- Audible drive chatter at least once a second, all day long

(Protip: platter chatter is a function of mechanical drive head seeks!)

OpenZFS fundamentals

- Writes commit to disk each *txg_sync_interval* seconds
- Fragment Differently<tm> thanks to Copy on Write
- Topology == > fragmentation  
- Blocksize ==> fragmentation  
- Kernel tunables ==> fragmentation  

- Fragmentation  ==> seeks  ==> noise 

Fragment Differently

- Copy-on-Write filesystems do not overwrite blocks in-place, they write new blocks, then link them in place of the old blocks
- Modify one block in the middle of a large file? That's a new frag
- Tend to congruously read blocks that were written congruously? That's **less** fragmentation

Blocksize, frags, and you

- 1MiB “random” I/O is nearly indistinguishable from sequential
- 64KiB random I/O is a pretty good generic workload model
- 16KiB random I/O is a pretty good database workload model

- “Sequential” operations rely on no fragmentation and no competition

About RAIDz topology

- RAIDz vdevs stripe each block to the vdev
- Blocks aren't split into pieces *a la* mdraid, they're split into multiple sector-wide stripes
- You only get the “naive efficiency” of a vdev if all blocks written are evenly divisible by $(n-p)$
- Awkward vdev widths == one under-width stripe at the end of every block
- Block too small for vdev width == lack of efficiency + lower performance than more, smaller vdevs on same drives










About mirror topology

- Mirror vdevs write each block in its entirety onto each member disk of the vdev
- Blocks aren't split into pieces OR stripes, period
- Full size block == larger read/write ops == higher performance
- Write IOPS of one drive per vdev
- Read IOPS of ALL drives per vdev
- Write width of 1 == more contiguous ops == fewer seeks

Kernel tunables

- Not a txg_sync_interval problem (why not?)
- Not a prefetch/readahead problem (why not?)
- CACHE vdev aka L2ARC? Consider l2arc_mfuonly, l2arc_noprefetch
- SPECIAL vdev? Consider special_small_blocks (dataset tunable, not kernel)

Bagging on Proxmox

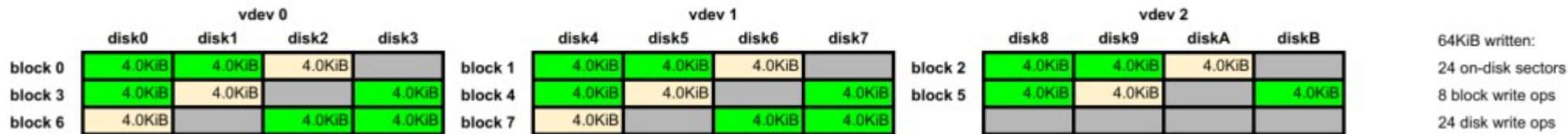
- Proxmox REALLY wants you to use ZVOLs, not filesystem datasets
- Proxmox defaults to volblocksize=8K (<=7.x) or volblocksize=16K (8.x)
-  blocksize ==>  fragmentation ==>  seeks ==>  noise
-   blocksize ==>  partial stripe writes ==>   noise

Putting it all together

- 3x 4-wide RAIDz1 + volblocksize=8K ==> ALL stripe writes partial
- 3x 4-wide RAIDz1 + volblocksize=16K ==> MANY stripe writes partial
- Partial stripe writes == single-sector ops, now AND later

Visualizing: 4w RAIDz1/8K

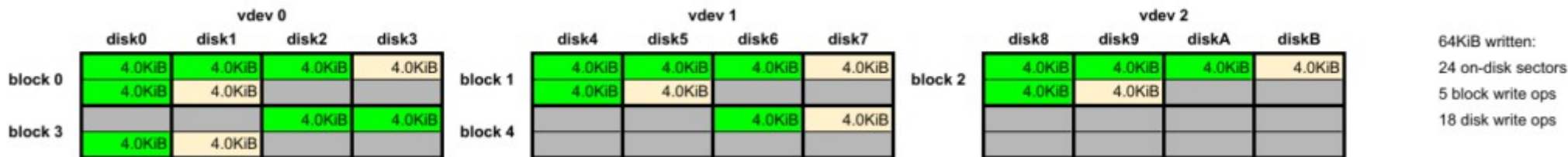
Three 4-wide Z1 vdevs, volblocksize=8KiB



- Green sector: data
- Beige sector: parity
- Grey sector: not written to on THIS block write (NOT a hole!)
- EVERY write is only one sector wide!
- 24 disk write ops to commit 64KiB of data

Visualizing: 4w RAIDz1/16K

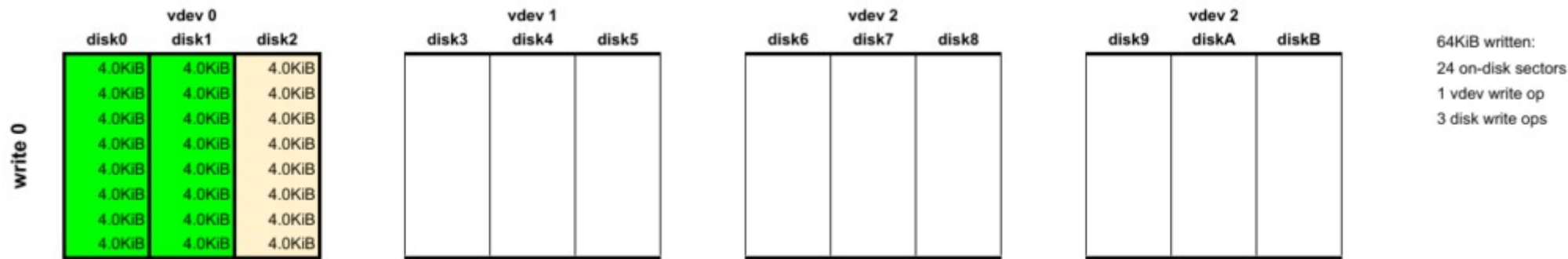
Three 4-wide Z1 vdevs, volblocksize=16KiB



- Green sector: data
- Beige sector: parity
- Grey sector: not written to on THIS block write (NOT a hole!)
- Every OTHER write is only one sector wide
- 18 disk write ops to commit 64KiB of data

Visualizing: 3w RAIDz1/64K

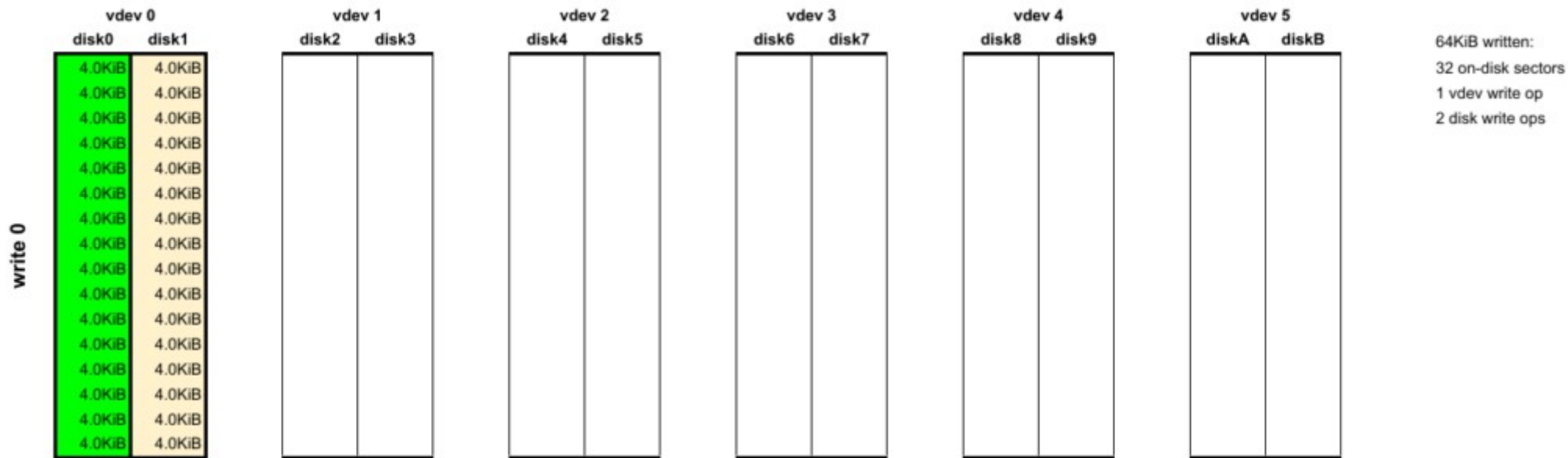
Four 3-wide Z1 vdevs, volblocksize=64KiB



- NO partial stripes
- NO single-sector writes—ALL writes are eight sectors wide!
- 3 disk write ops to commit 64KiB of data

Visualizing: 2w mirrors/64K

Six 2-wide mirror vdevs, volblocksize=64KiB



- NO partial stripes, all writes SIXTEEN sectors wide, 2 write ops!
- Double read IOPS per vdev! Easy expansion! Free kittens!

Consider the Stone Axe

- If an HDD makes a noise in a heavy steel case with rubber grommets and no silly glass window, can you hear it?
- If you can't hear an HDD clicking, is it really making a sound?
- If a philosopher's stone turns lead into gold, what does a philosopher's NAS do?



Time for Questions!

You may find this (and all my other public presentation slide decks) at:

<https://jrs-s.net/presentations>

License: CC-NC-BY-SA 3.0 unported; for commercial or other uncovered use, please contact me directly. I'm not hard to find.

