DigitalOcean

Discards in Ceph

Ceph Day NYC 2024

Matt Vandermeulen, Storage Systems

Contents

- The problem
- The need to enable discards
- First attempted solution (firmware)
- Second attempted solution (software)
- Other possibilities
- Reflection

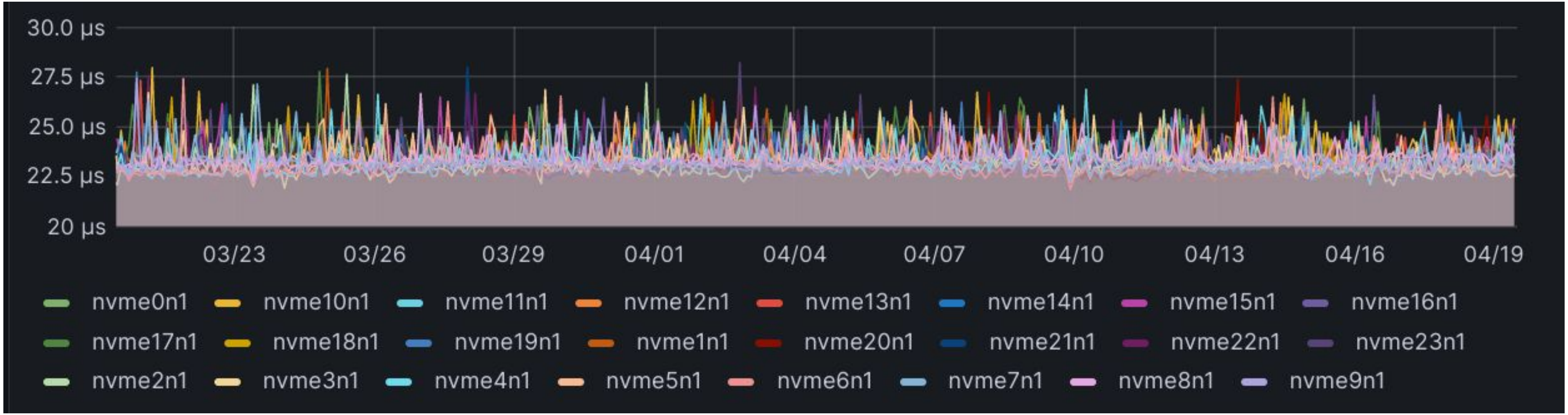
The Problem

- **We have a couple dozen drive models in our fleet today**
- **Probably half a dozen vendors**
- **We’ve never had to worry about enabling discards in the past**
- **We have a single drive model that shall not be (n|sh)amed**
- **This drive hit a write cliff in production during an RBD workload**
    - Disk write performance suffered significantly in this case
    - This started when most of the writes were now overwrites on the disk

Enter: Async Discards

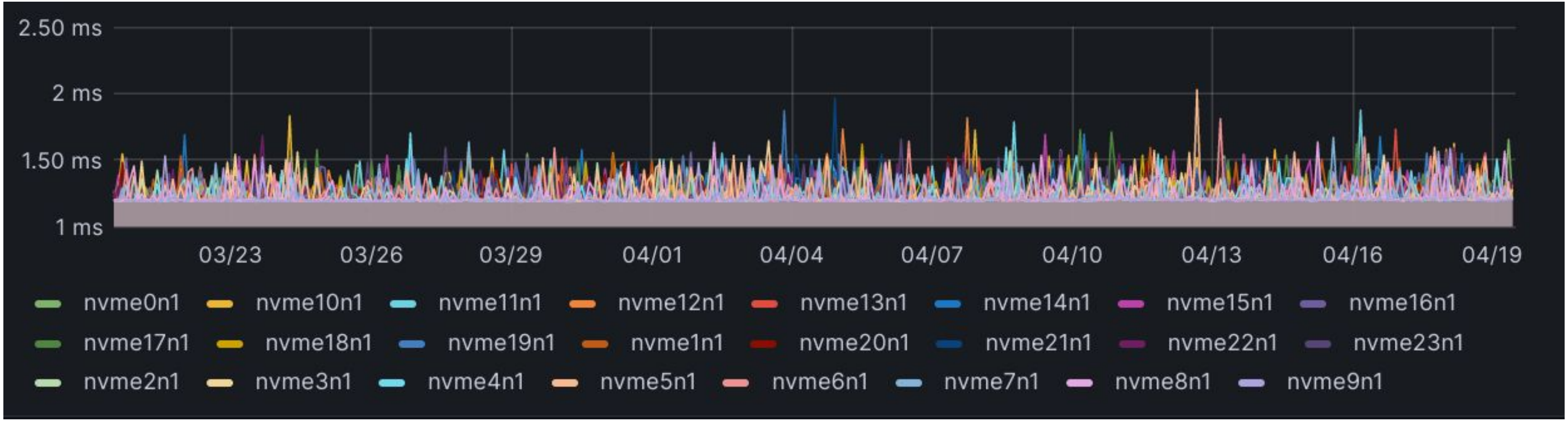
- **Discards were not enabled anywhere on our fleet**
- **We enabled async discards on a small number of drive models**
  - This was across different vendors
  - Discards helped significantly with performance on one model, only slightly on another
  - On other models, it hurt performance, so we don’t use it
- **Now we started to observe high discard latency**
  - But only on the one specific model
- **Disk performance is (mostly) restored and maintained**
- **Let’s look at discard latencies…**

Reference Discard Time



```
irate(node_disk_discard_time_seconds_total[5m]) /
irate(node_disk_discards_completed_total[5m])
```

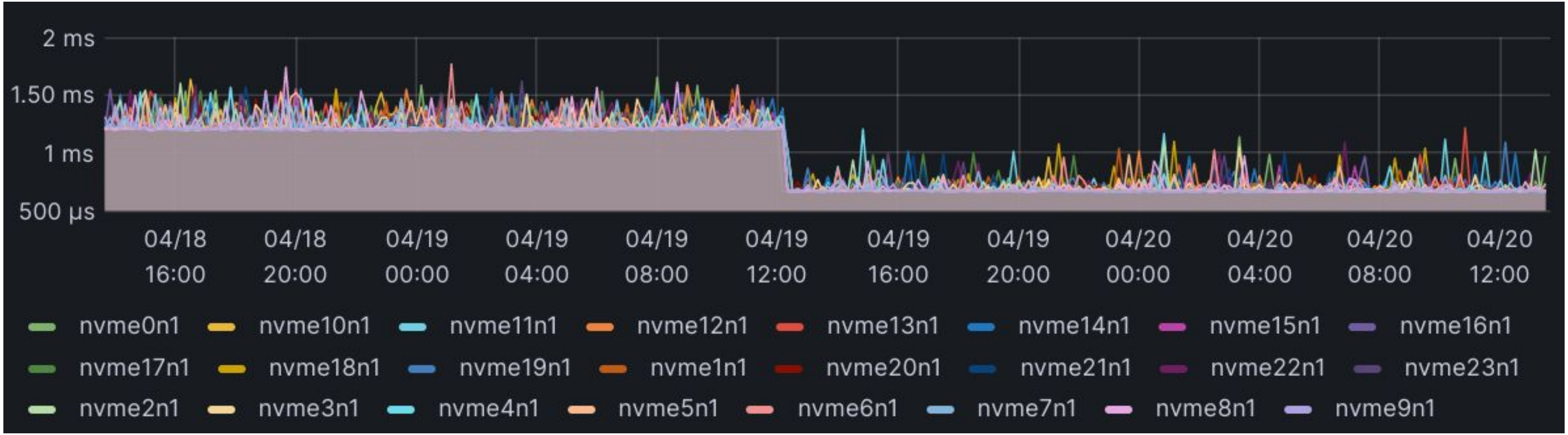
Problematic Discard Time



First Solution: Firmware

- The first thing we did was engage the vendor
- They worked with us to understand our workload
- They provided us with a new firmware version to try out

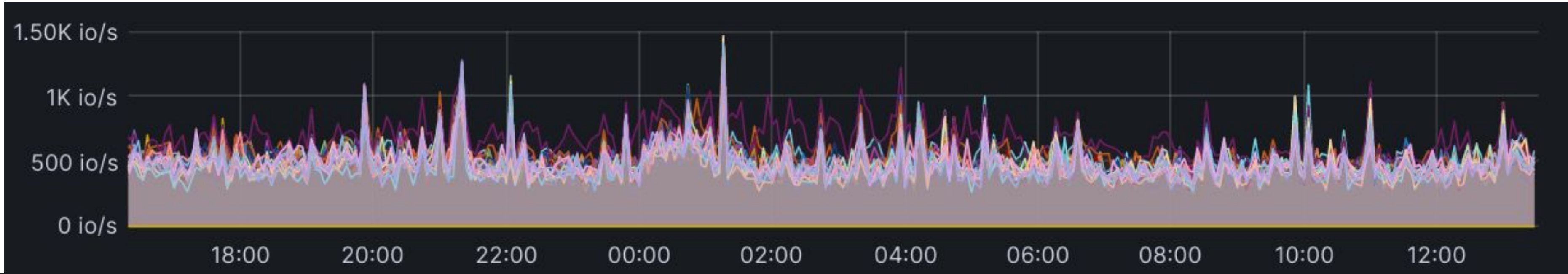
First Solution: Firmware

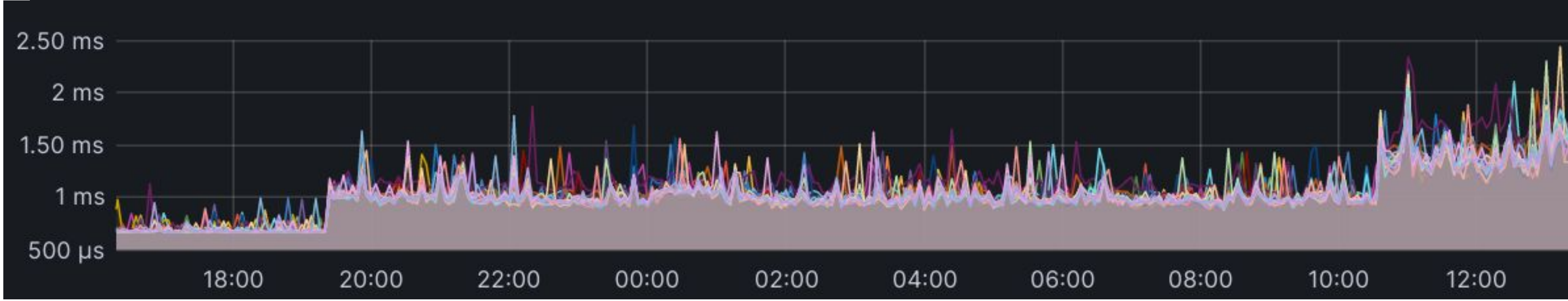


Second Solution: Software

- Can we work around the latency issue in software?
- Learn how the discard mechanism works in Ceph today
- We know that it is (or, can be) async, handled in another thread
- Can we get more parallelism out of it?
- Yup! #55469
- Did that help?

Second Solution: Software





Other Possibilities

- **Matt H on Slack has done some work with switching allocators**
- **Hybrid allocator may be inefficient at finding allocations for SSDs**
  - The hybrid allocator wants to keep recent allocations pretty close together, which is beneficial for HDDs
  - This means that we use lots of different LBAs, which means lots of flash is allocated, making overwrites slower
- **The AVL allocator may be better at re-using blocks**
  - This implies that blocks will no longer be sequential, flash media doesn't care too much about that
  - This needs some more testing
  - It's possible that we might be able to replace discards with switching to the AVL allocator

DigitalOcean

Reflection

Looking back

Reflection

- In house: Identifying issues
- External: Working with vendors, waiting on firmware updates
- Community: Able to work on things in-house, sharing with the community
- Open source software continues to allow us to move quickly
- Input from the community gives us extra eyes on our approaches

DigitalOcean

Thank You!

Hiring plug http://do.co/jobs

Q&A?