

Distributed Storage (Ceph)

...

Krishnan

Motivation

1. Redundancy / Backups

What if a drive fails?

Backblaze Hard Drive Failure Rates for 2024

Reporting period 1/1/2024 - 12/31/2024 inclusive

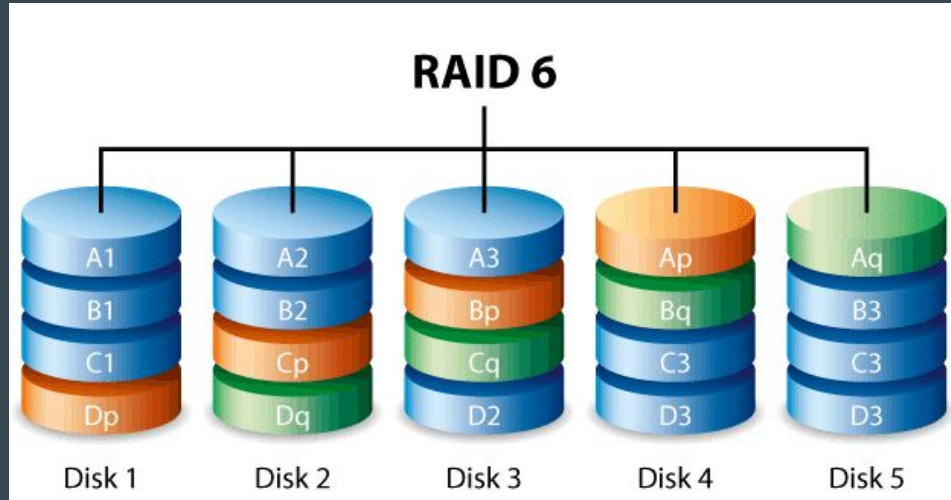
Drive models with drive count > 250 as of 12/31/2024 and drive days > 50,000 in 2024.

MFG	Model	Size (TB)	Drive Count	Avg Age (months)	Drive Days	Drive Failures	AFR
HGST	HMS5C4040ALE640	4	263	95.5	327,114	4	0.45%
HGST	HMS5C4040BLE640	4	4,120	95.9	3,078,618	18	0.21%
HGST	HUH728080ALE600	8	1,084	79.8	400,697	18	1.64%
HGST	HUH721212ALE600	12	2,606	61.8	949,117	32	1.23%
HGST	HUH721212ALE604	12	13,313	43.3	4,786,933	435	3.32%
HGST	HUH721212ALN604	12	10,191	66.5	3,814,797	590	5.65%
Seagate	ST8000DM002	8	9,078	98.6	3,355,725	162	1.76%
Seagate	ST8000NM0055	8	13,534	87.0	5,020,845	356	2.59%
Seagate	ST10000NM0086	10	1,034	84.4	391,133	58	5.41%
Seagate	ST12000NM0007	12	1,038	61.3	400,953	125	11.38%
Seagate	ST12000NM0008	12	19,134	56.2	7,060,183	502	2.60%
Seagate	ST12000NM000J	12	723	8.5	172,556	16	3.38%
Seagate	ST12000NM001G	12	13,184	46.8	4,814,820	166	1.26%
Seagate	ST14000NM001G	14	10,589	46.3	3,863,921	175	1.65%
Seagate	ST14000NM0138	14	1,321	48.9	498,291	73	5.35%
Seagate	ST16000NM001G	16	33,600	28.6	11,737,654	226	0.70%
Seagate	ST16000NM002J	16	461	25.8	168,170	1	0.22%
Toshiba	MG07ACA14TA	14	37,703	49.7	13,729,393	414	1.10%
Toshiba	MG07ACA14TEY	14	742	36.4	230,649	10	1.58%
Toshiba	MG08ACA16TA	16	40,185	19.1	14,089,908	484	1.25%
Toshiba	MG08ACA16TE	16	5,912	38.4	2,173,570	68	1.14%
Toshiba	MG08ACA16TEY	16	5,163	36.9	1,889,434	82	1.58%
Toshiba	MG10ACA20TE	20	5,943	3.8	631,665	15	0.87%
WDC	WUH721414ALE6L4	14	8,542	47.8	3,110,146	72	0.84%
WDC	WUH721816ALE6L0	16	3,016	36.2	1,103,215	37	1.22%
WDC	WUH721816ALE6L4	16	26,475	21.1	9,365,201	94	0.37%
WDC	WUH722222ALE6L4	22	30,000	5.8	4,741,582	139	1.07%
Totals			298,954		101,906,290	4,372	1.57%

2. High-Availability

What if a drive fails, AND you don't want stuff to break until it's replaced?

We want the backups to immediately fill the role of the primaries when needed.



4. High-Availability Part 2

What if a server fails?

What if the whole rack loses power?



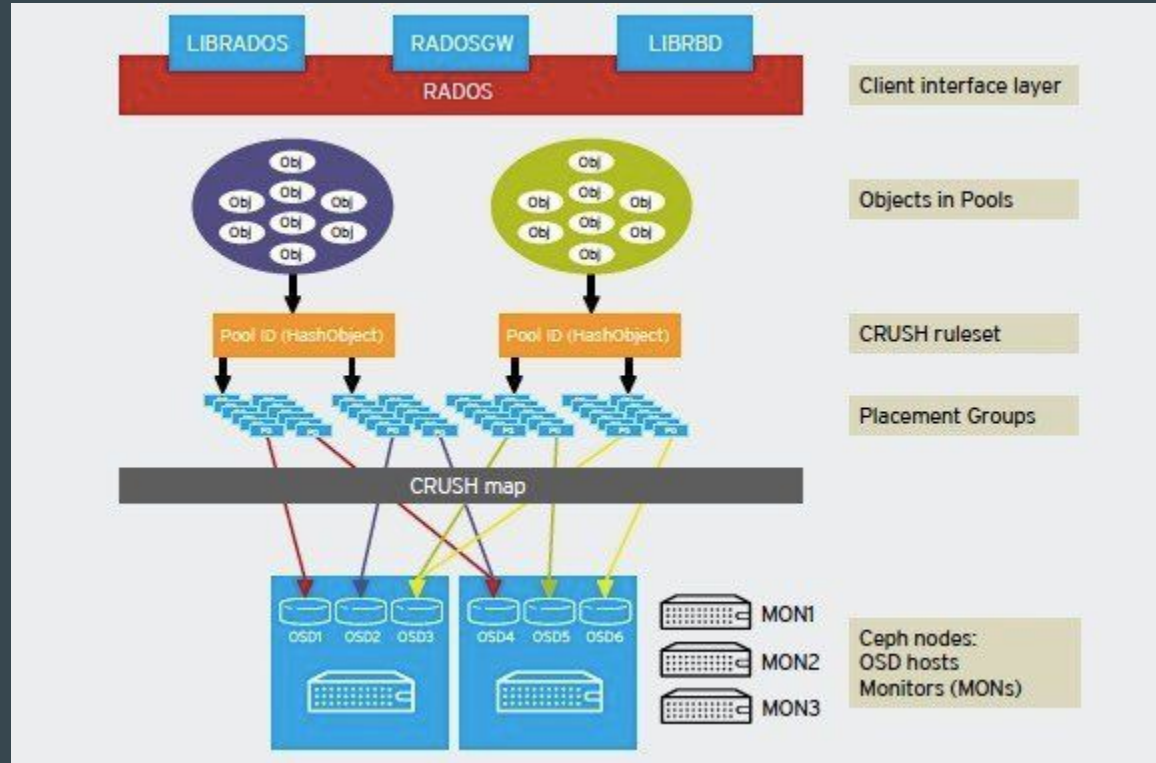


What is Ceph?

TL;DR

ZFS, but the drives are split among multiple servers (over the network).

Ceph Architecture






Ceph from a Hardware POV

We want three copies of data

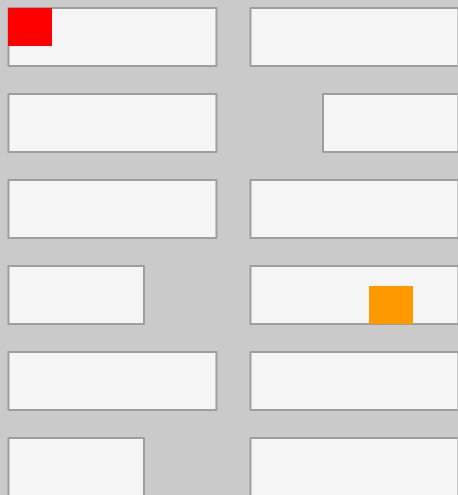
Why?

- Sometimes, you're in a situation where a drive doesn't *fail*, but a few bits flip (see: bit rot)
- The concept of **quorum**: If two copies of the data agree on something, that's considered "truth"

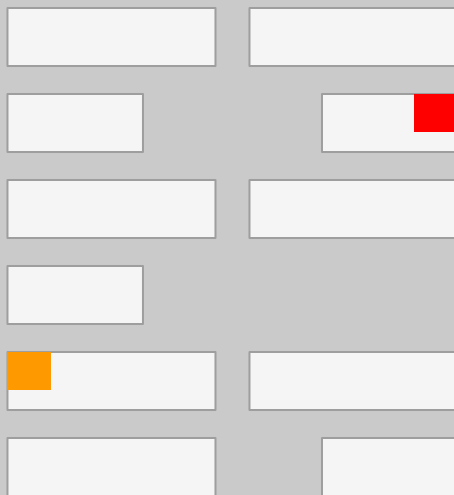
	101010111 1 1010010 0 0010101111010
	1
	101010111 0 1010010 0 0010101111010
	1
	101010111 1 1010010 1 0010101111010
	1

**How do we account for
both drive failures and
server failures?**

Server A



Server B



Server C






Quick Terminology Note



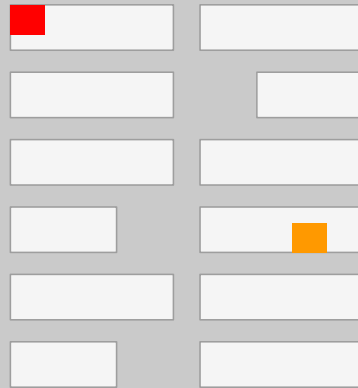
- Individual drives are wrapped in OSDs (Object Storage Daemons)
- Each “block” of data is called a PG (Placement Group)
- The random-ish algorithm that assigns PGs to OSDs is called CRUSH (Controlled Replication Under Scalable Hashing)
- There’s a [paper](#) about this!

We Still Need...

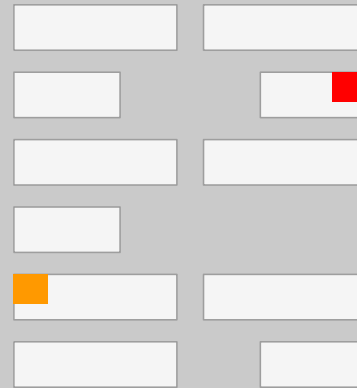
- A way to route data requests to the right drives
 - If I want to read from  , Ceph needs to find one of the copies and give it to me
 - If I want to write to  , Ceph needs to write to all three copies of 

Server D - Monitor

Server A



Server B



Server C



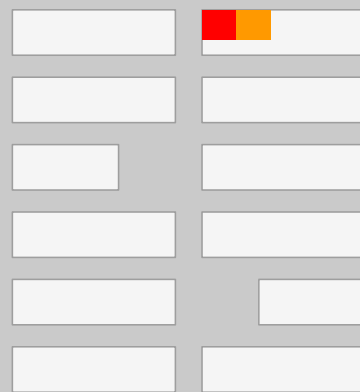
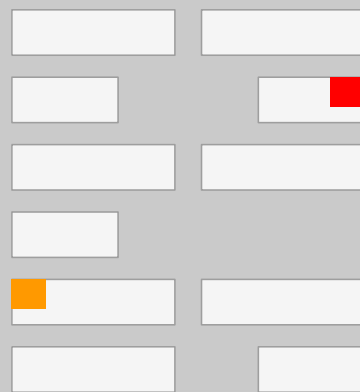
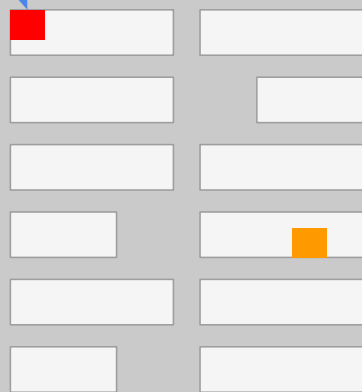
Read 

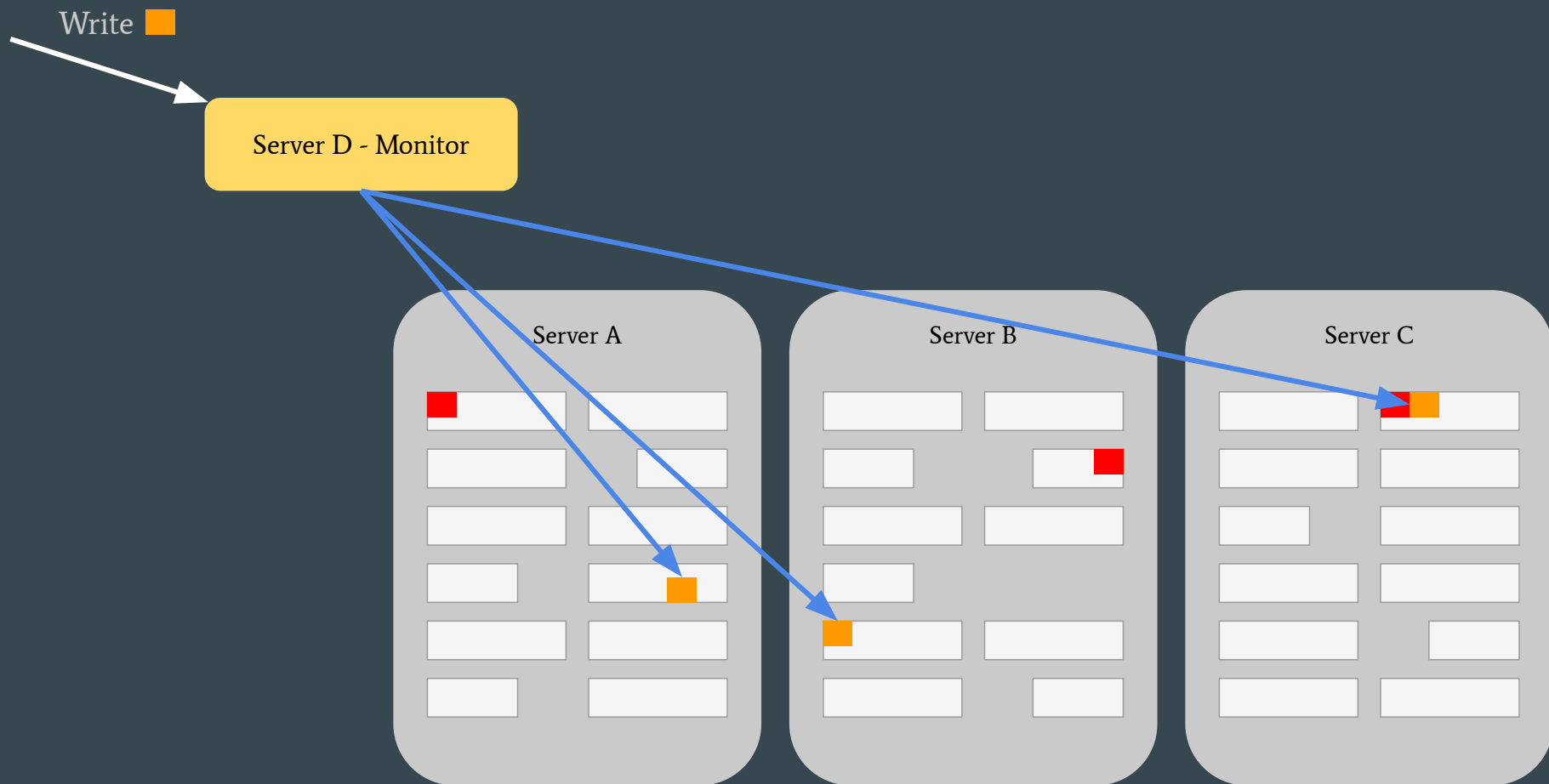
Server D - Monitor

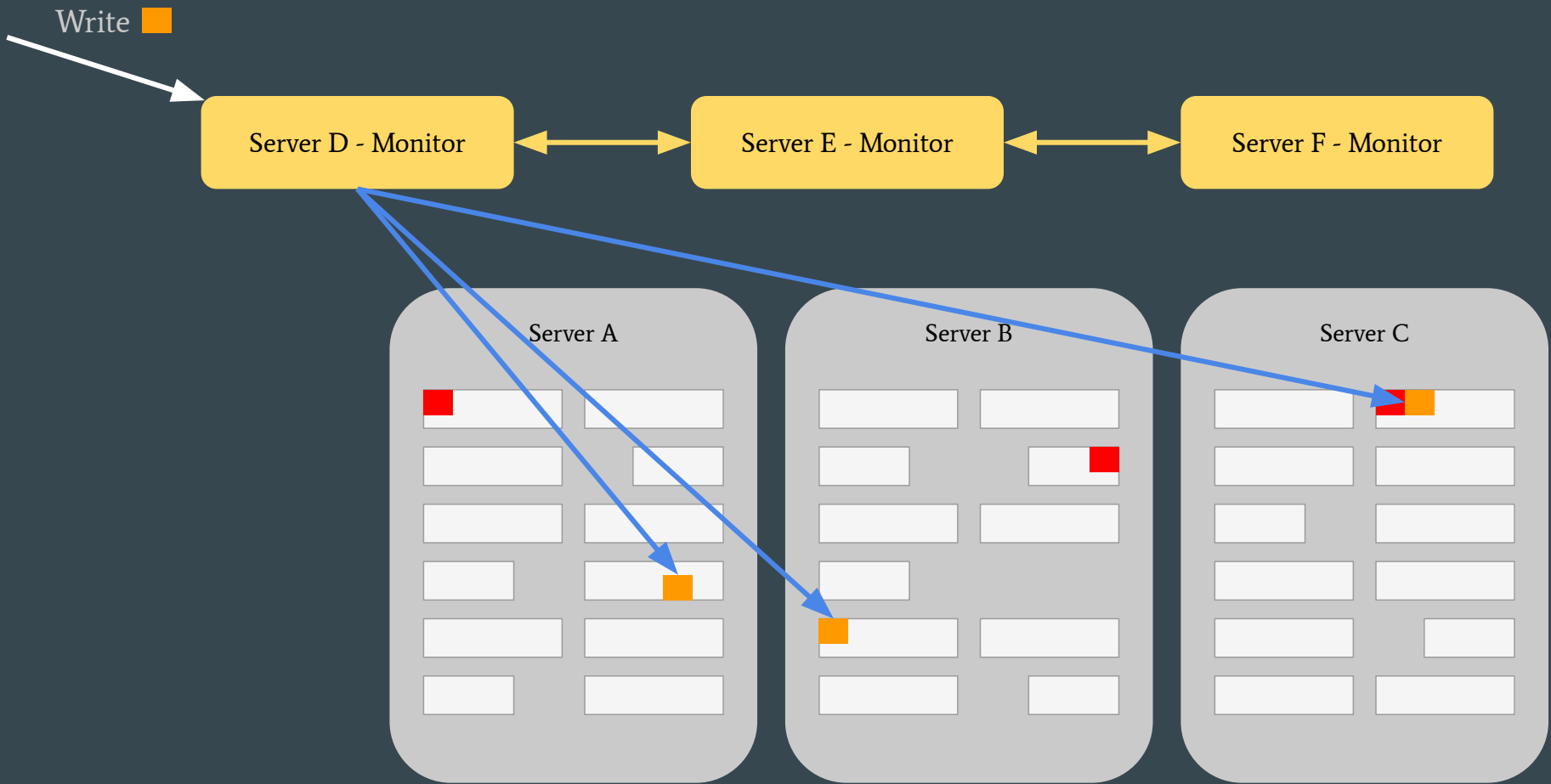
Server A

Server B




Server C













We Still Need...

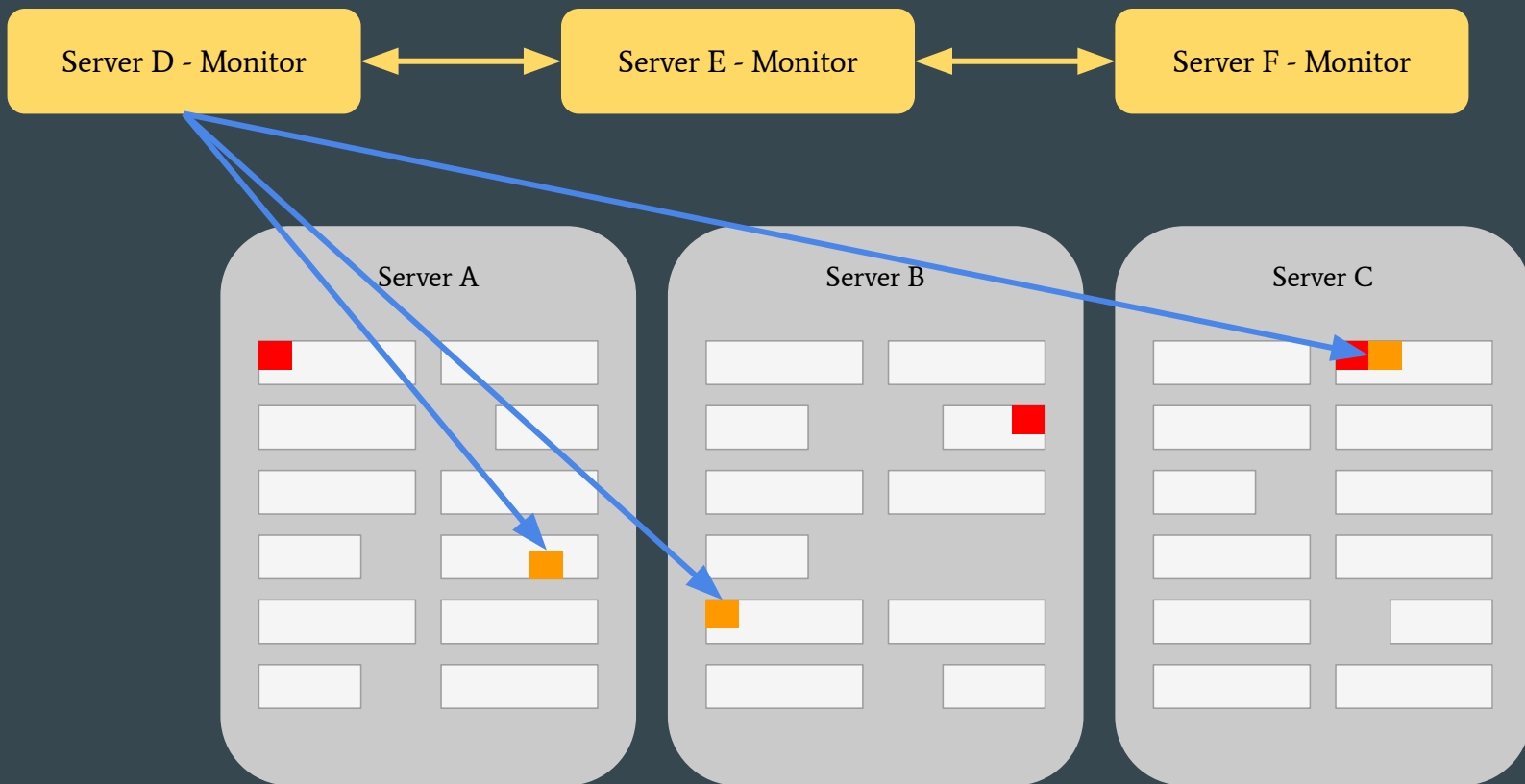
- ✓ A way to route data requests to the right drives
 - If I want to read from  , Ceph needs to find one of the copies and give it to me
 - If I want to write to  , Ceph needs to write to all three copies of 

We Still Need...

- ✓ A way to route data requests to the right drives
 - If I want to read from , Ceph needs to find one of the copies and give it to me
 - If I want to write to , Ceph needs to write to all three copies of 
- A way to scan all copies of data to find inconsistencies

	101010111110100100001010111010
	1
	101010111010100100001010111010
	1
	101010111110100101001010111010
	1




Scrubbing



Scrubbing

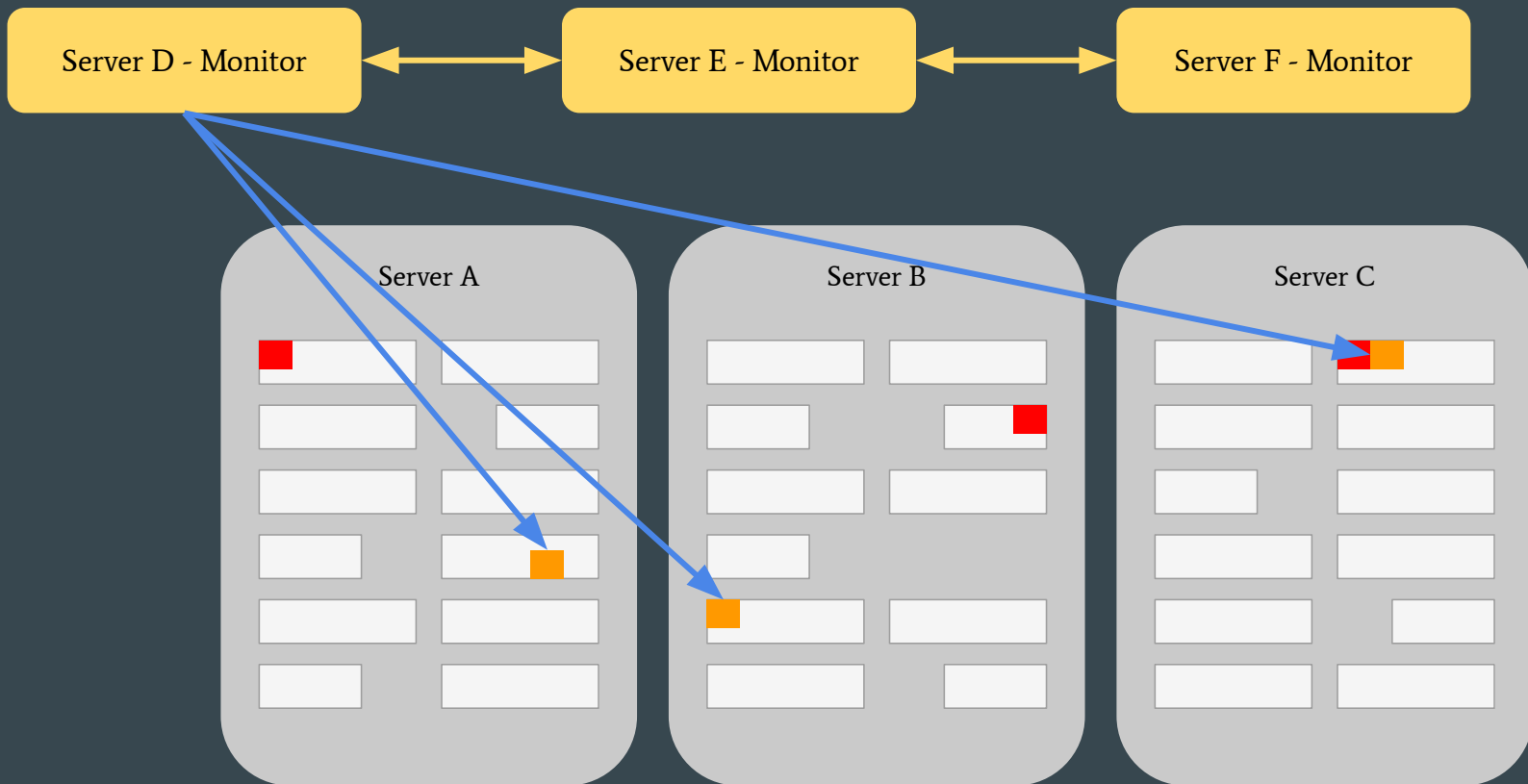
- Think of it like `fsck`
- Light Scrubbing:
 - Happens daily (per PG)
 - Compare overall characteristics (object size, attributes, etc.)
- Deep Scrubbing:
 - Happens weekly (per PG)
 - Compare data checksums
 - If they don't match, compare data bit-by-bit

We Still Need...

- ✓ A way to route data requests to the right drives
 - If I want to read from  , Ceph needs to find one of the copies and give it to me
 - If I want to write to  , Ceph needs to write to all three copies of 
- ✓ A way to scan all copies of data to find inconsistencies

Ceph from a Software POV

Ceph Architecture (So Far)



What can we build on top of this architecture?

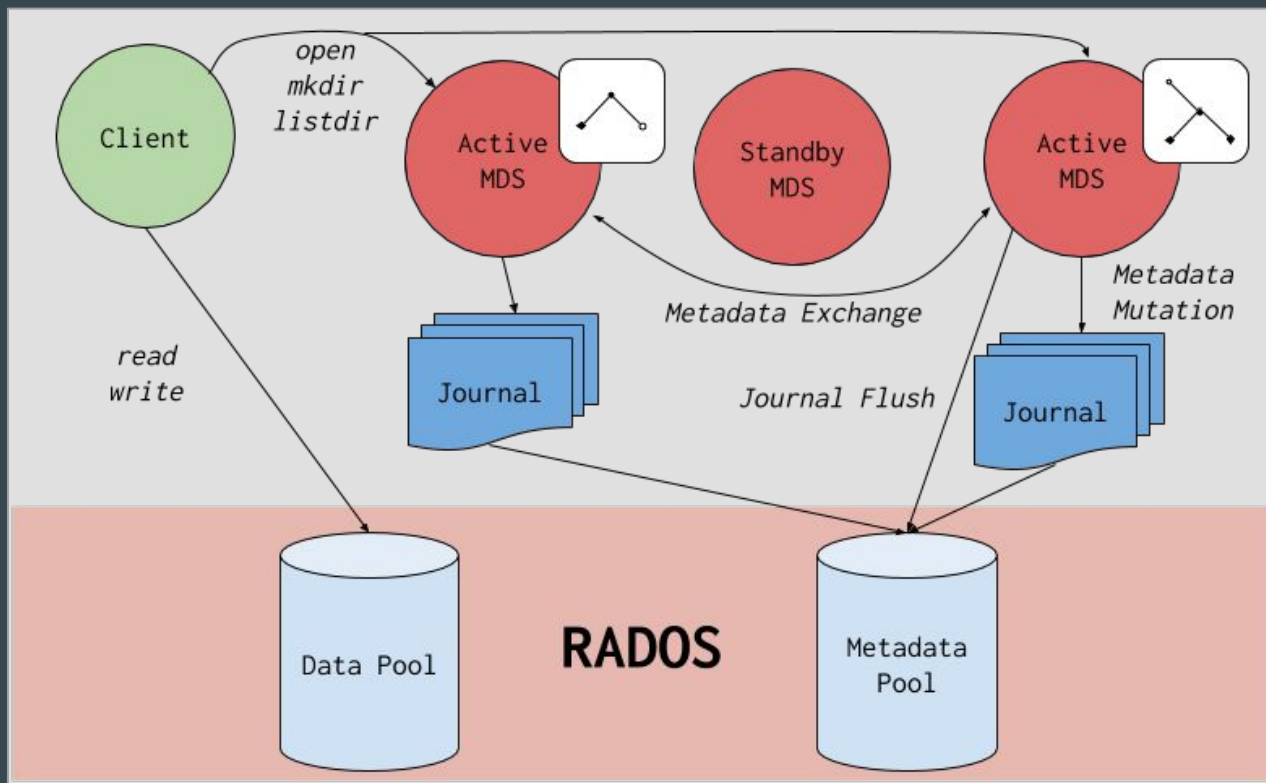
A lot.

- **CephFS** - A journaling POSIX filesystem
 - It's very complicated
 - There's another great [paper](#) on this!
- **RBD** - Block Device
 - Literally just splits data into blocks
 - Very powerful, especially for running VMs (great QEMU support)
- **RGW** - Object Gateway
 - AWS S3-like Buckets of data
 - Provides an S3-compatible API

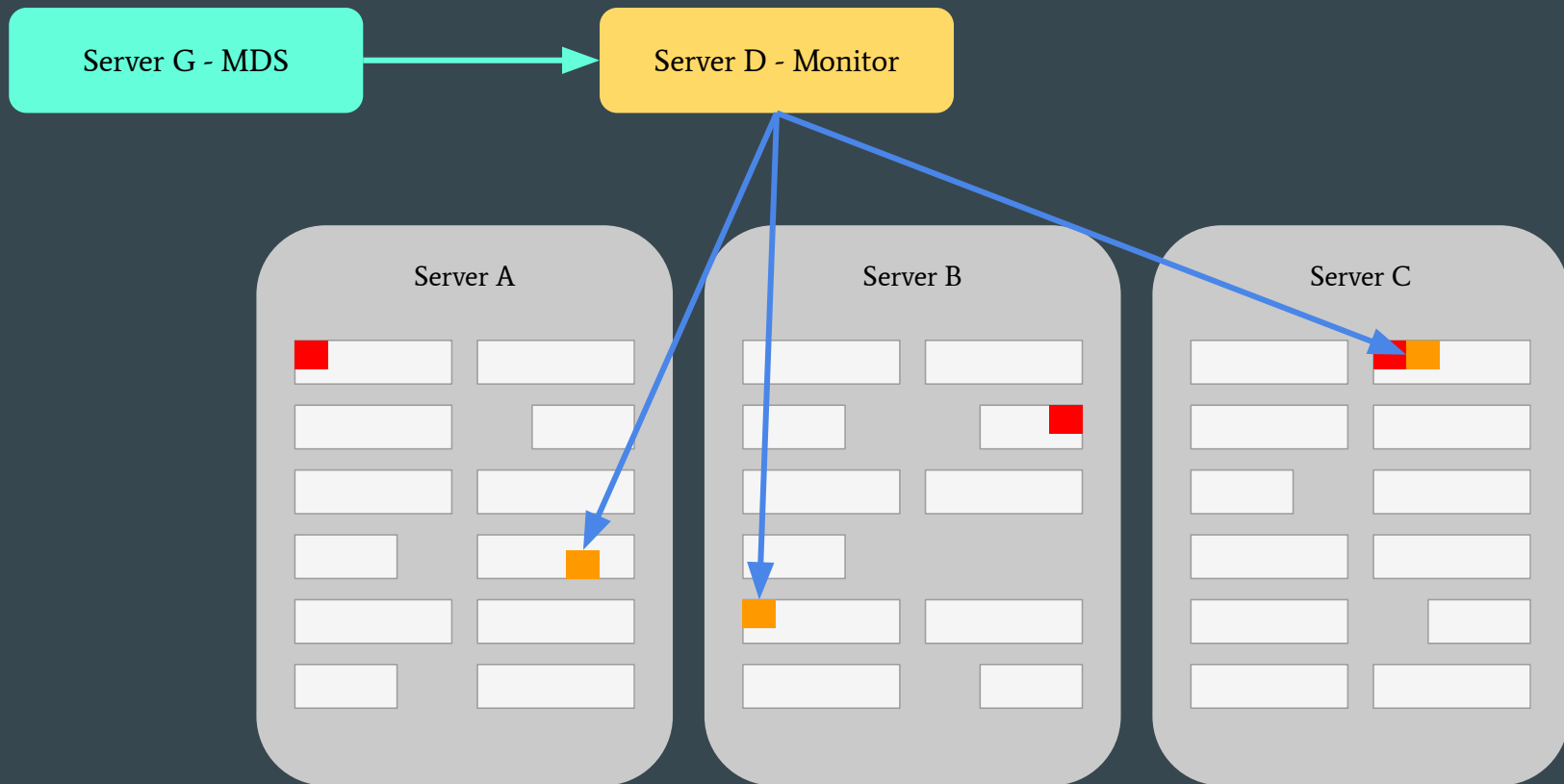
Ceph Pools

- A high-level group of data
- You could have a pool for:
 - Filesystem (CephFS)
 - VMs
 - Less important VMs
 - Logs
 - ...
- Main advantage: per-pool replication configuration
 - **Filesystem** should be replicated 5x
 - **VMs** should be replicated 5x
 - **Less important VMs** should be replicated 3x
 - **Logs** should be replicated 1x

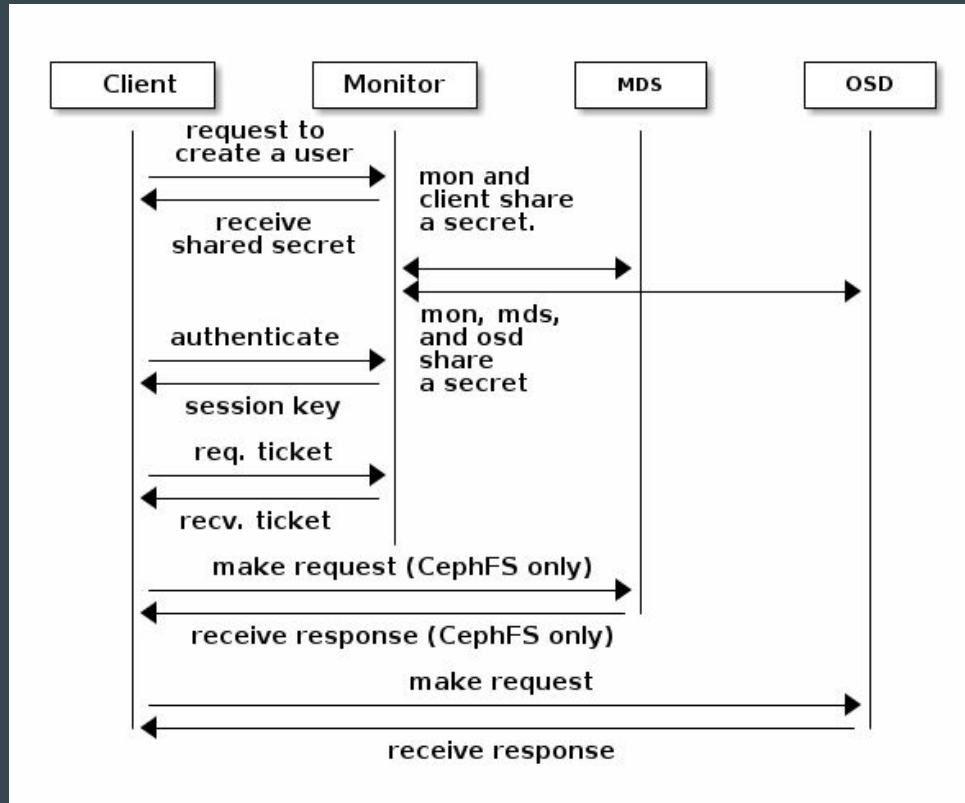
CephFS Architecture



MDS: Metadata Server



Authentication



Other Random (Potentially Cool) Things

- CephFS works really well with NFS (The Network File System Protocol)
 - You can share specific directories within CephFS
 - These will generally be public-facing directories, as opposed to internal ones
 - This bypasses Ceph-based authentication, but allows for your own implementation (e.g. Kerberos)
- Snapshots
 - Supported by both CephFS and RBD
 - These don't store the entire state of your filesystem or VM (as that's expensive...)
 - They use Copy-on-Write (COW) for future modifications
 - Surprisingly efficient

Demo

Thank You!

Questions? Comments?