9 years of Ceph at Walmart

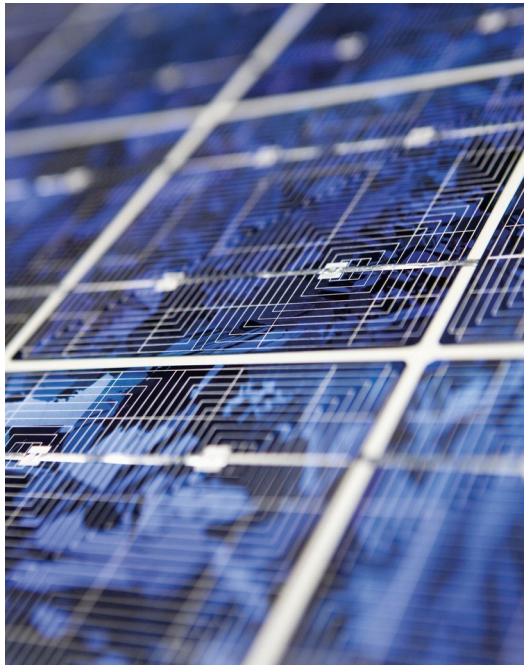
Pavan Rallabhandi
Anton Thaker

How it started

- Ceph block (librbd) integrated with OpenStack Cinder.
  - Started with sub 1TB flash drives.
  - Single 10GbE networking.
  - 1U density less than half of one OSD today.
- Ceph object (swift) used to power big data workloads.
- Ceph Hammer
- Ceph upgrades:
  - H - > J -> L
- Conversions from LevelDB to RocksDB on Filestore

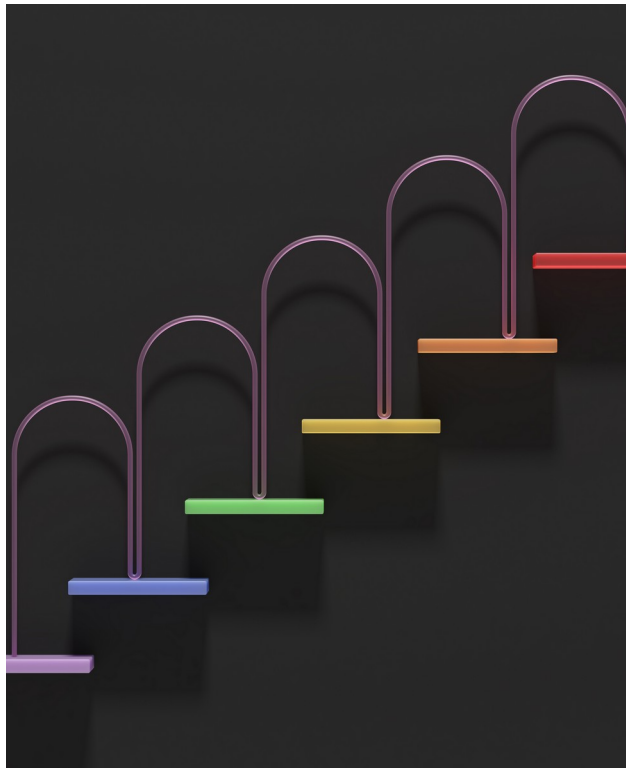
Old problems we ran into

- RGW crashes and bugs
  - Swift object expiration
  - OOM
- RGW leaking data
  - Swift copy object semantics
- RGW performance issues
  - Civetweb
  - Thread pool size; memory hog
- RGW fixes by Walmart
  - https://github.com/pulls?q=author%3Aprallabh+is%3Apublic



How it's going

- Clusters on Quincy
  - Upgraded from N -> P - > Q
- Block storage latency monitoring framework
- Performance limits of a Ceph cluster
  - Multiple OSDs per drive vs multiple drives per OSD
- Boot from Volume
- TRIM/Discard
- High-Throughput Object workloads
- Running OSDs on top of existing LVM volumes
- WPQ Scheduler
- 1U density ~ 20x higher than our first servers
- Industry Swings: Block vs Object / Public vs Private



Current challenges

- Cephadm scaling issues
- Managed vs unmanaged OSDs
- Ceph-mgr anomalies
  - Ceph-exporter missing metrics at load
  - Ceph performance metrics are inaccurate
  - We shouldn't need to fail MGRs
  - rbd perf metrics at large volume scale
- Thundering herd problems
  - Dealing with Large VM app clusters
  - TRIM/Discard
- Device health metrics
- Large Ceph clusters starving ephemeral ports on hypervisors
- RGW still crashes and leaks after 9 years
- Latency with large number of OSDs
- Latency during failures

RGW still crashes and leaks after 9 years

- RGW should not be crashing from client requests
- TempURL bug example on 17.2.7:
  - wget https://ceph/swift/v1/bucket?temp_url_sig=0123456789abcdef0123456789abcdef0123456789abcdef0123456789abcdef&temp_url_expires=1234567890
- Object expiration leak on resharded buckets
- Object expiry changes are not honored
- Crashes on heavy GC loads
- Object lock broken on multi-part uploads

Latency with large number of OSDs

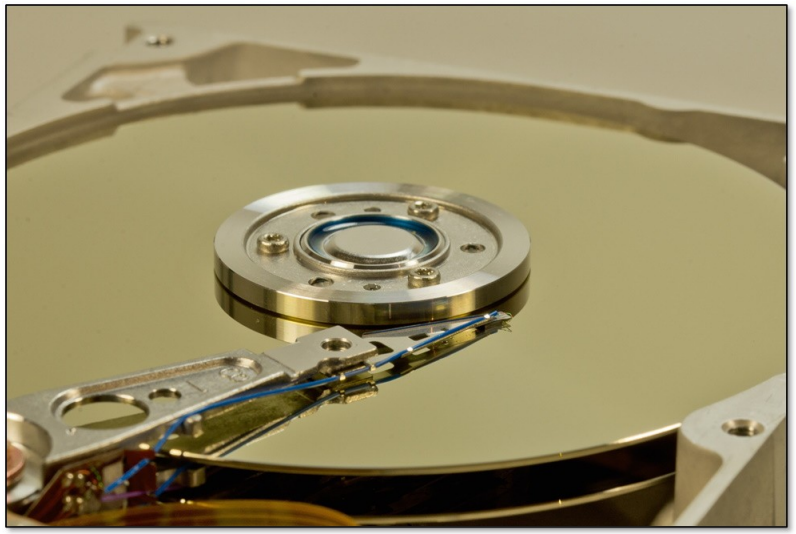
- As number of OSDs scales up, so does the latency:

- 48 OSDs, 48 hosts ~ 0.5 ms latency
- 96 OSDs, 48 hosts ~ 0.6 ms latency
- 480 OSDs, 48 hosts ~ 0.8 ms latency
- 960 OSDs, 48 hosts ~ 1.1 ms latency

Latency during failures

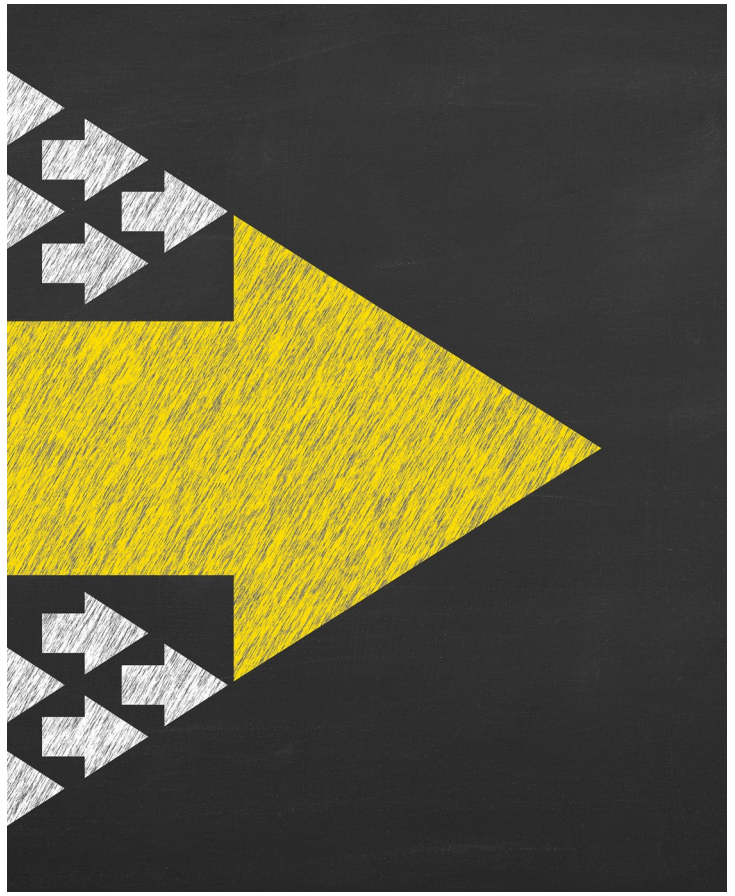
- Latency during graceful OSD stops/restarts ~ 1-2 s
- Latency during ungraceful OSD failures ~ 20 s
- Latency on colocated MON / OSD host failures ~ 40 s
- Latency during ungraceful hardware failures ~ ?? m

- Tuning for lower latency during recovery
- osd_max_backfills
- osd_recovery_sleep 0 ←→ 0.05 ←→ 0.1
- norebalance flag



Looking forward

- Low latency for Ceph block
- Stability of Ceph-mgr & cephadm
- Stability of RGW
- E2E latency view for Ceph block IO performance metrics
- Ceph-csi
- NVMe-oF
- Erasure Coding for Block
- Multi-device classes for tiering/cost savings
- Another 20x density increase???



Thank You!

Questions ???