Google

Google’s Journey to Adopt DAOS

DAOS Virtual DUG 2025

jeffolivier@google.com

May 22, 2025

Agenda

01 What is unique about DAOS in the cloud?

02 Networking Challenges

03 Selected Deep Dives

04 DAOS Development Challenges

05 Q&A

01

What is unique about DAOS in the cloud?

What is unique about DAOS in cloud?

- Benefits
  - Automation
    - Any manual intervention is expensive
    - Restoring service is priority
    - Debugging should be done post mortem
  - One click instance creation
    - GKE offers similar ease of client setup
  - Optimized import/export of data
  - Full replacement VMs can be swapped in when something fails
    - Sometimes, the infrastructure just does it automatically
  - Tools for capturing and viewing logs and metrics
- Challenges
  - Placement of VMs relative to compute
  - TCP only, no RDMA
  - Limited VM shape options
  - Client compatibility

02

Networking Challenges

Networking challenges

A seemingly unending stream of challenges...

- Firewall rules (explained on future slide) - **HANG**
- Connection tracking blocking connections - **HANG**
- Exhausting open file descriptors - **HANG**
- Docker reserved ip range 172.17.0.0/16 - **HANG**
- libfabric tcp provider issues (and sometimes mercury/cart)
    - Not enough multi-receive buffers - **HANG**
    - Erroneous handling of multi-receive buffers on errors in libfabric and mercury - **HANG**
    - Wrong error causing endpoints to get into bad state - **HANG**
    - Stale replies cause endpoint to choke - **HANG**
    - Disabled endpoint on retryable failures - **HANG**
    - Keepalive needed to detect engine fails (upstream patch pending) - **HANG**
    - Race causing dfuse crash (investigating)

Libfabric expertise has been a challenge

03

Selected Deep Dives

Poor Object Placement

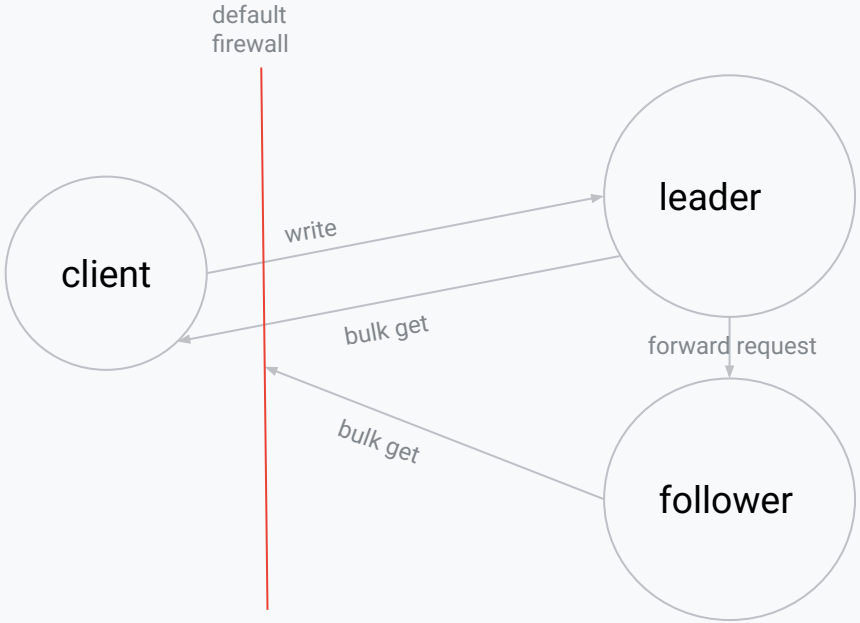
- **Problem:** With BALANCED file config (EC_2P1G8), writing 8GiB files was hitting -DER_NOSPACE between 30-50% total capacity due to imbalance
- **Root cause:** Allocating user portion of oid->hi sequentially does not work well
- **Solution:** OID Cycling using large prime

```
static inline void
daos_obj_oid_cycle(daos_obj_id_t *oid)
{
     /** Uses a large prime number to guarantee hitting every
unique value */
     oid->hi = (oid->hi + 999999937) & UINT_MAX;
}
```

About 95% utilization on average with this technique

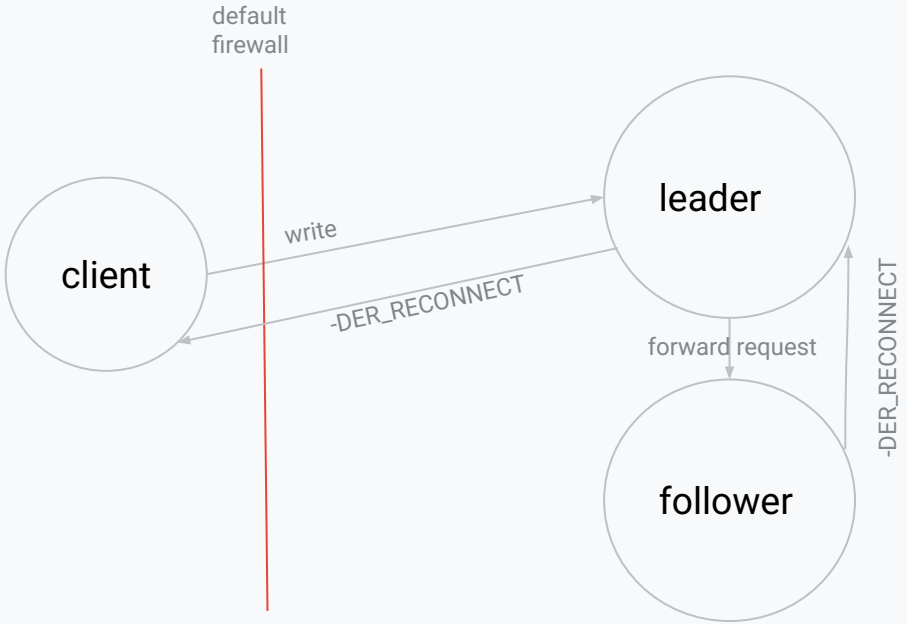
The Firewall Problem

- By default, a Virtual Private Cloud network will allow connections to Parallelstore server ports but not the other way around
- **Problem:** If a user doesn't open all client ports, the client will hang



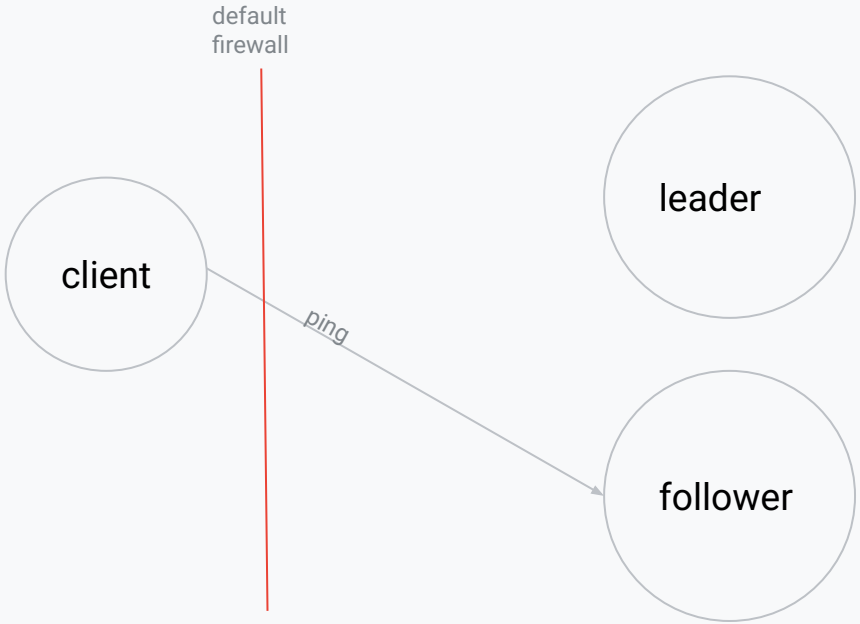
The Firewall Solution

- Implemented in (feature/firewall) branch
- Solution: Set fully compatible flag in libfabric indicating the client endpoint is behind a firewall. If a server doesn't have a connection, reply with an error.



The Firewall Solution

- Client will ping follower targets, establishing a connection.
- Bulk transfer will work on retry of original write

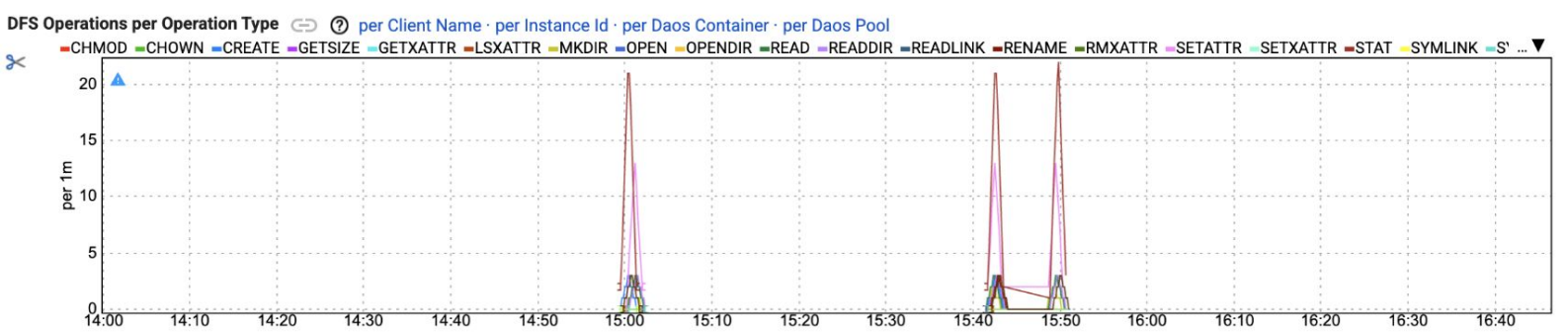


Repair Issues

- SSD uncorrectable bit error rate in local SSD makes EC_2P1 impractical
  - Needs checksum scrubber to repair extents with errors
  - Scrubber has some stability issues, occasional crashes
- We can’t replace single targets
  - Single target failures should be upgraded to server failures
- We disable degraded mode rebuilds (delay_rebuild)
  - Avoids -DER_NOSPACE
  - VM replacement can be fast so avoid two rebuilds

What is happening in a DAOS instance?

- Metrics are generally preferable to logs
  - Counter metrics are preferable to gauges for graphs
- Logs are also useful but are often overly chatty
- Client side visibility is a challenge
  - Very little control or visibility
  - Client telemetry will be helpful but user has to enable it
- Hopefully these tools will reduce manual intervention to help customers



04

DAOS Development Challenges

DAOS development challenges

- Google model - Rapid release
  - We try to release a new version of DAOS every two weeks
  - Google has a nice pipeline to catch regressions
    - But DAOS testing is hard internally
    - Upstream tests make a lot of assumptions
      - systemd assumptions
      - matching clients/servers
      - Machine shapes and capabilities
      - Internal network, repos, etc
    - Automated perf and longevity testing
- RPM builds outside of HPE are a pain
- Lack of CI visibility
- Difficult to setup testing externally
  - Internal tooling is easier to use
- Length of time from PR creation to submission is days long
  - Internal changes have very fast turnaround time
- New developer onboarding is time consuming

05

Q/A