DAOS Data Protection and Fault Recovery

DUG’24 – Michael Hennecke

https://daos.io/

DAOS Data Protection

- DAOS provides data protection and data availability through replication and/or Erasure Coding across the network (to multiple servers)
- Minimum level of data protection, aka “redundancy factor” (RF, or rd_fac):
  - The ***default*** rd_fac for new *containers* can be set on the *pool* level:

    ```
    dmg  pool create --properties=rd_fac:2 ... mypool
    ```
  - The ***actual*** *container* rd_fac (set explicitly, or from pool default) will be ***enforced***:

    ```
    daos cont create --properties=rd_fac:2 ... mypool mycont
    ```
- daos cont create options -o, -d, -f are only setting *defaults* for new objects in that container (they cannot be weaker than the rd_fac of the container)
  - The –o –d –f defaults are ***not enforced***; each object can set its OCLASS at object creation time
  - When a fault occurs, the DAOS system *cannot* rely on the –o –d –f oclass defaults, e.g. a container with rd_fac:0 and –d RP_3G1 –f RP_3GX will still be treated as RF0 and will go offline

What happens when a fault occurs

For a single fault (one network link, one NVMe SSD, ...), and RF >=1:

- Affected engine (or targets for one SSD fault) gets excluded
- Pool **State** changes from “Ready” to “Degraded” (will be renamed to avoid confusion)
  - Pool **Disabled** column shows the number of missing targets (e.g. 24/2016)
- Pool **Rebuild State** changes from “idle” or ”done” to “Busy”
- Automatic rebuild is started, pool data stays available
- When rebuild has completed, **Rebuild State** changes to “done”

After the problem has been resolved, and engine rank got started again:

- Manually run `dmg pool reintegrate --rank=<N>  mypool` (for each pool)
  - This changes pool **State** to “Ready” again, and **Disabled** target count gets reduced

Multiple Failures

Before DAOS 2.6.1, each failure is treated as an individual event

- Rebuild is immediately started for each fault
- When # of faults exceeds RF:
  - Rebuild is halted
  - Containers are marked “Unhealthy”, and access to the container is blocked
  - Manual intervention is needed after faults are resolved, to get back to “healthy” state

With DAOS 2.6.1, DAOS will “correlate” failures that happen at the same time

- Needs **DAOS_POOL_RF=2** setting in engine environment
- Will *not* start rebuild when # of faults is higher than DAOS_POOL_RF
- Easier recovery when failures got resolved (e.g. after a switch outage)

DAOS 2.6.2 (and 2.8) further improve recovery from such “mass failures”

Questions?

d a o s

Cont with RF2, trying to set weaker –d –f defaults fails

```
dmg pool create -u daosperf -g users --size=100T \
  --properties=rd_fac:2   daosperf_pool01

# cont will inherit the default rd_fac:2 from pool level...
daos cont create --type posix -d S1 -f SX \
  daosperf_pool01 cont01
dfs  ERR  src/client/dfs/cont.c:120 dfs_cont_create() File object class
cannot tolerate RF failures
ERROR: daos: failed to create container: DER_INVAL(-1003): Invalid parameters
```

Cont with RF0, setting stronger –d –f defaults works (1/2)

```
dmg pool create -u daosperf -g users --size=100T \
  --properties=rd_fac:0   daosperf_pool02

# cont will inherit the default rd_fac:0 from pool level...
daos cont create --type posix -d RP_3G1 -f RP_3GX \
  daosperf_pool02 cont02
```

- The –d –f is only setting **defaults**, you can create objects with higher or lower levels of data protection, e.g. using
  - `daos fs set-attr --oclass=SX --path=...`

Cont with RF0, setting stronger –d –f defaults works (2/2)

```
# “touch” a file with specific OCLASS weaker than the –f default:
daos fs set-attr --path=/tmp/daosperf/scratchfile --oclass=SX
daos fs get-attr --path=/tmp/daosperf/scratchfile
Object Class = S2016

-rw-rw-r-- 1 daosperf users 0 Nov 18 16:02 /tmp/daosperf/scratchfile

echo "not protected!" > /tmp/daosperf/scratchfile

daos fs get-attr --path=/tmp/daosperf/scratchfile
Object Class = S2016
```