Cephalautopsy - When Erasure Coding Goes Wrong

Jamie Pryde

Contents

1. Data redundancy and inconsistency
2. EC consistency checker 1 (Offline OSD checker)
3. Offline consistency checker demo
4. EC consistency checker 2 (Online OSD checker)

Data redundancy and inconsistency

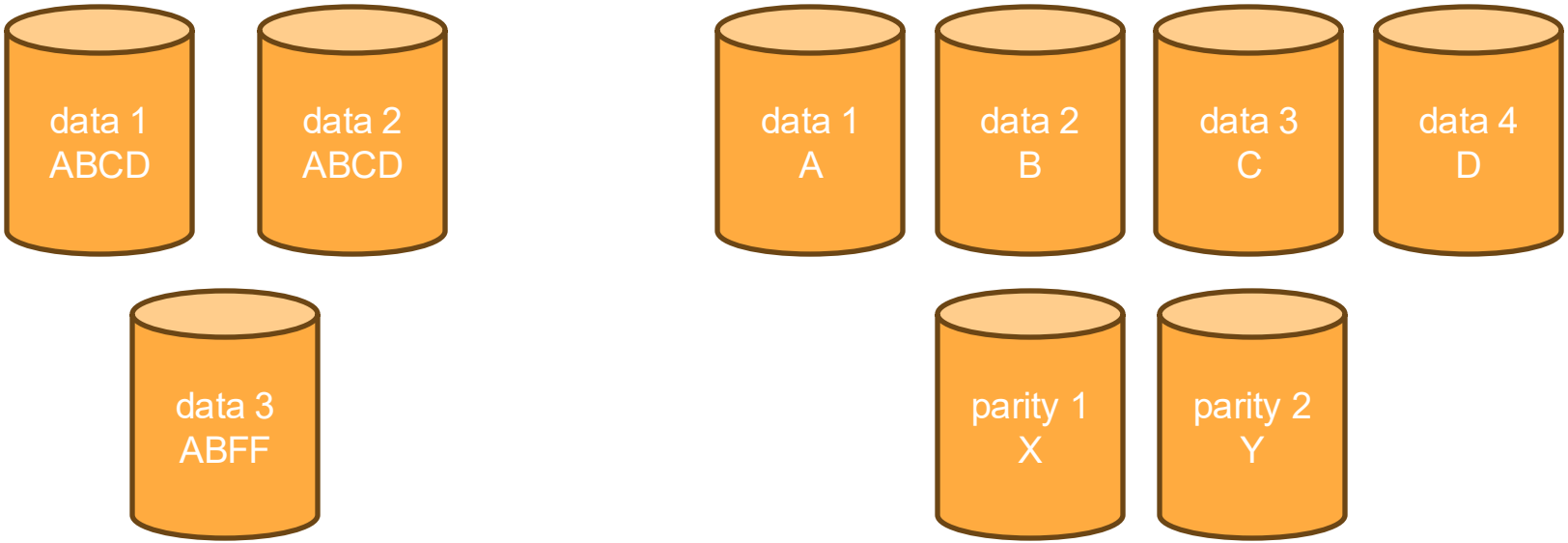
It is important to check consistency when storing redundant data (using replication or erasure coding).

A failing OSD (due to hardware failure / firmware bug etc) may fail a write and then return the wrong data when read.

A code bug could cause an OSD to write corrupt data.

We might not detect any problems until an application crashes or corrupt data is read from an OSD.

Inconsistencies in EC pools



3x replication

EC 4 + 2

How do we find an inconsistency in the erasure coded data?

Offline EC consistency checker

Tool that reads data from OSDs and recalculates the parity to check for inconsistencies across all the OSDs.

Takes OSDs offline and uses ceph-objectstore-tool to read data.

Available to end users to check consistency in their EC pools.

Use case 1: Can be used after an application crashes or there are signs of data corruption in an EC pool to help identify the bad OSD(s).

Use case 2: Can be used to scan a cluster's EC pools for consistency before bringing the cluster back online to check that it is safe to do so.

Offline EC consistency checker usage

```
python3 ../qa/tasks/vstart_runner.py --config-mode /work/ceph/qa/tasks/ec_parity_consistency.yaml

# cat /work/ceph/qa/tasks/ec_parity_consistency.yaml

roles:

- - mon.a

  - mgr.x

  - osd.0

  - osd.1

tasks:

  - ec_parity_consistency:
```

Offline EC consistency checker - no errors

2025-05-21 11:47:36,159.159 INFO:tasks.ec_parity_consistency:Consistent objects counted: 13

2025-05-21 11:47:36,159.159 INFO:tasks.ec_parity_consistency:Inconsistent objects counted 0

2025-05-21 11:47:36,159.159 INFO:tasks.ec_parity_consistency:Objects skipped: 0

2025-05-21 11:47:36,159.159 INFO:tasks.ec_parity_consistency:Total objects checked: 13

2025-05-21 11:47:36,160.160 INFO:tasks.ec_parity_consistency:Consistent objects:
['rbd_data.5.10aff15ad10.0000000000000000_-2','rbd_data.5.10aff15ad10.0000000000000003_-2','rbd_data.5.10aff15ad10.0000000000000006_-2','rbd_data.5.10aff15ad10.000000000000000a_-2','rbd_data.5.10aff15ad10.000000000000000b_-2','rbd_data.5.10aff15ad10.0000000000000001_-2','rbd_data.5.10aff15ad10.0000000000000009_-2','rbd_data.5.10aff15ad10.0000000000000004_-2','rbd_data.5.10aff15ad10.0000000000000007_-2','rbd_data.5.10aff15ad10.000000000000000c_-2','rbd_data.5.10aff15ad10.0000000000000002_-2','rbd_data.5.10aff15ad10.0000000000000008_-2','rbd_data.5.10aff15ad10.0000000000000005_-2']

Offline EC consistency checker - one error

2025-05-21 22:16:58,121.121 INFO:tasks.ec_parity_consistency:Consistent objects counted: 12

2025-05-21 22:16:58,121.121 INFO:tasks.ec_parity_consistency:Inconsistent objects counted 1

2025-05-21 22:16:58,122.122 INFO:tasks.ec_parity_consistency:Objects skipped: 0

2025-05-21 22:16:58,122.122 INFO:tasks.ec_parity_consistency:Total objects checked: 13

2025-05-21 22:16:58,122.122 INFO:tasks.ec_parity_consistency:Consistent objects: ['rbd_data.5.10af35bb3a53.0000000000000004_-2','rbd_data.5.10af35bb3a53.0000000000000005_-2','rbd_data.5.10af35bb3a53.000000000000000a_-2','rbd_data.5.10af35bb3a53.0000000000000002_-2','rbd_data.5.10af35bb3a53.000000000000000c_-2','rbd_data.5.10af35bb3a53.0000000000000003_-2','rbd_data.5.10af35bb3a53.0000000000000007_-2','rbd_data.5.10af35bb3a53.0000000000000001_-2','rbd_data.5.10af35bb3a53.0000000000000009_-2','rbd_data.5.10af35bb3a53.000000000000000b_-2','rbd_data.5.10af35bb3a53.0000000000000000_-2','rbd_data.5.10af35bb3a53.0000000000000006_-2']

2025-05-21 22:16:58,122.122 INFO:tasks.ec_parity_consistency:Objects with a mismatch: ['rbd_data.5.10af35bb3a53.0000000000000008_-2']

Online EC consistency checker

Alternative version of the consistency checker that can be used with the new EC I/O exerciser to find inconsistencies.

Intended for use during development rather than by end users.

Finds inconsistencies quickly rather than having to wait for a reconstruct or decode that fails later. e.g. start test, inject error, check consistency, continue test.

Gives us confidence that the new fast EC code works!

Code location

Being developed by Connor Fawcett

https://github.com/ceph/ceph/pull/59903 - offline checker

https://github.com/ceph/ceph/pull/62170 - online checker

Will be finished later this year

Thanks!

jamiepry@uk.ibm.com