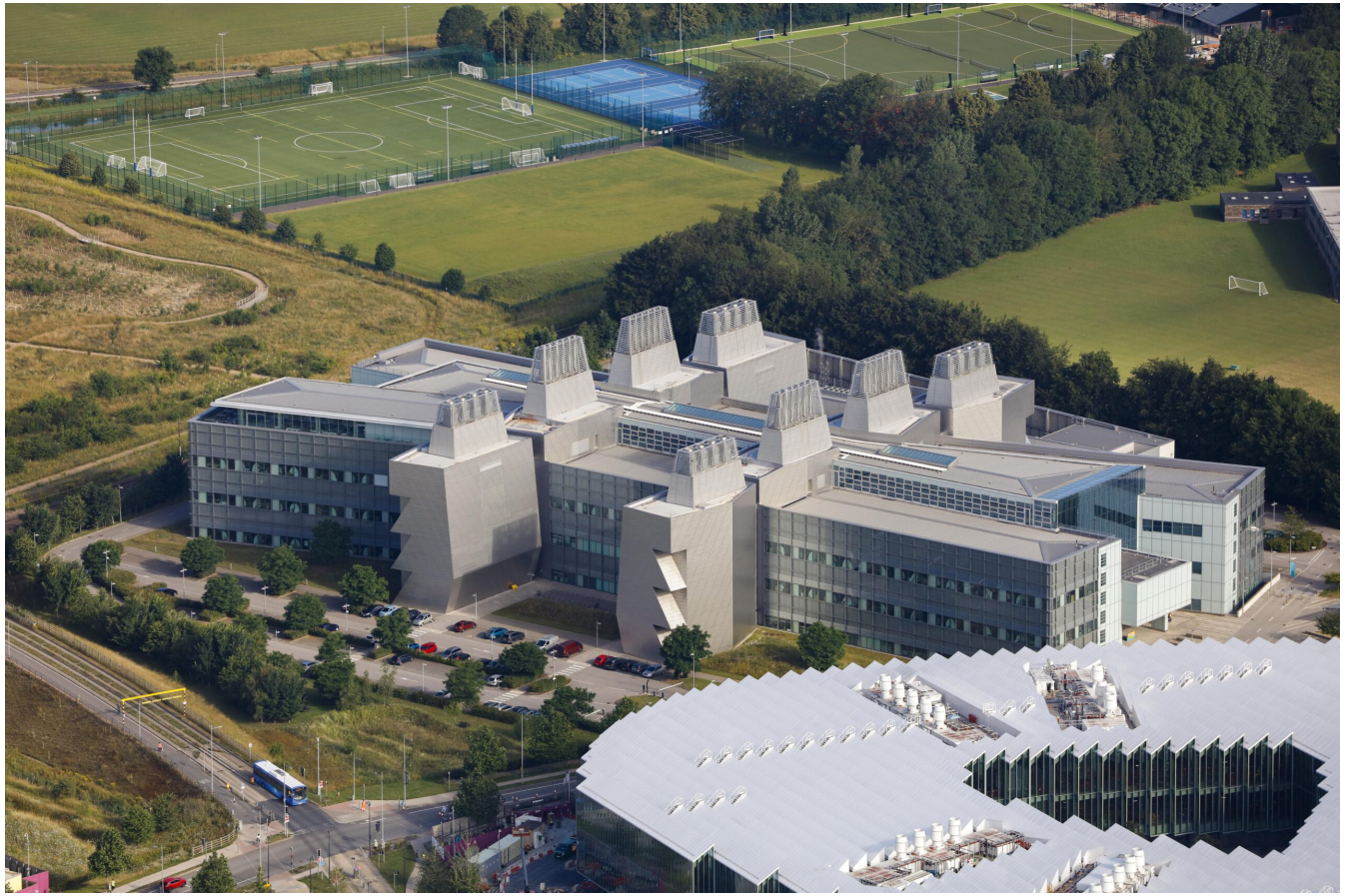
MRC Laboratory of Molecular Biology

The anatomy of a CephFS metadata disaster

A case study of our CephFS disaster recovery

MRC Laboratory of Molecular Biology

- National research institute in the UK focusing on biological and medical research
- Established in 1947 to initially use x-rays to investigate molecular structures and now researches molecular biology in a range of biological disciplines:
  - Structural biology
  - Neuroscience
  - Protein & nucleic acid chemistry
  - Genetics & genomics
- Awarded 12 Noble Prizes



MRC Laboratory of Molecular Biology

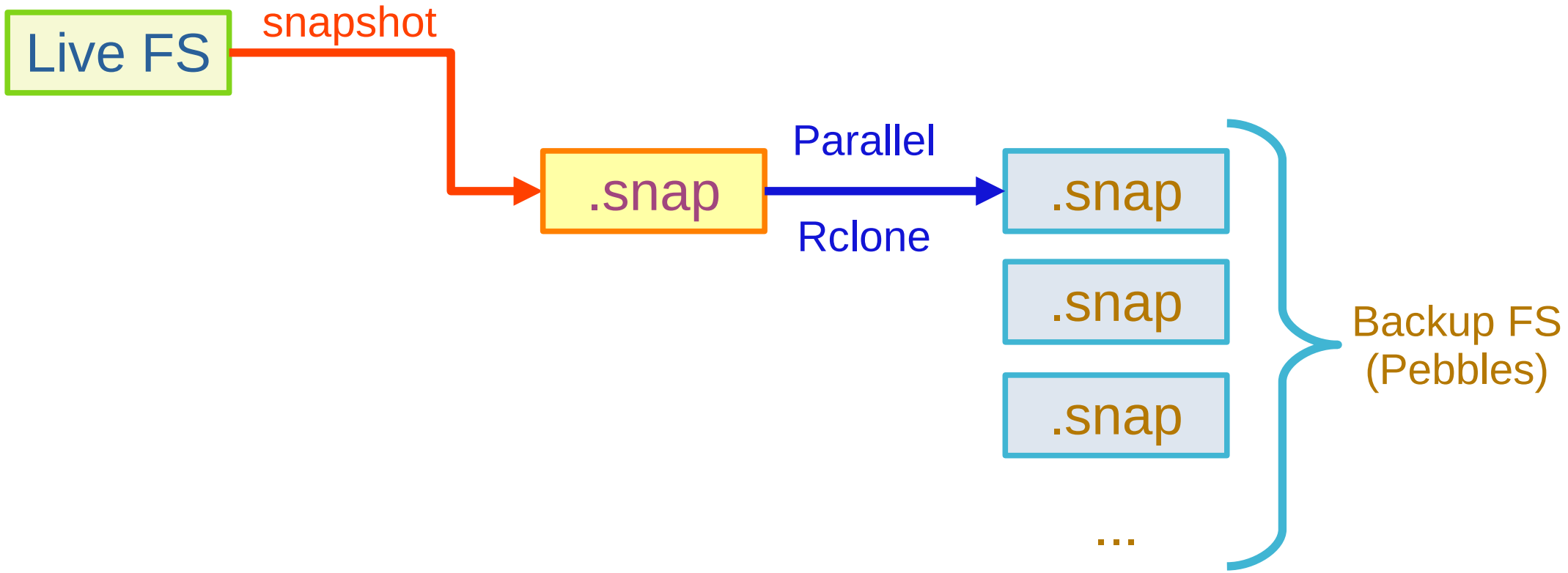
Ceph at the LMB

- Ceph is the main storage system used at the LMB where we have 3 user-facing clusters totalling:
  - 40-node: 4.6 PiB
  - 29-node: 5.7 PiB
  - 36-node: 7.0 PiB

- Our data backup is managed with a single CephFS cluster
  - 36-node: 22 PiB
  - 32 bulk data nodes with 16x 108-bay JBODs on HDDs
  - 4 metadata nodes with metadata on NVMe array

UKRI MRC Laboratory of Molecular Biology

CephFS backup solution



MRC Laboratory of Molecular Biology

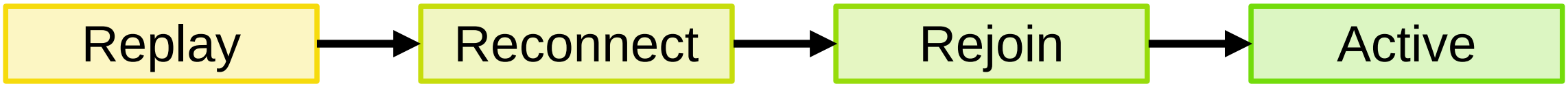
```
root@pebbles-n4 09:25 [~]: ceph -s
  cluster:
    id:     e3f7535e-d35f-4a5d-88f0-a1e97abcd631
    health: HEALTH_WARN
            1 filesystem is degraded
...
```

```
root@pebbles-n4 09:25 [~]: ceph -s
  cluster:
    id:     e3f7535e-d35f-4a5d-88f0-a1e97abcd631
    health: HEALTH_WARN
            1 filesystem is degraded
root@pebbles-n4 09:25 [~]: ceph health detail
…
[WRN] FS_DEGRADED: 1 filesystem is degraded
    fs ceph_backup is degraded
[WRN] MDS_INSUFFICIENT_STANDBY: insufficient standby MDS daemons
available
    have 0; want 1 more
...
```

```
root@pebbles-n4 09:26 [~]: ceph fs status
ceph_backup - 0 clients
===========
RANK      STATE         MDS      ACTIVITY   DNS    INOS   DIRS   CAPS
 0    replay(laggy)  pebbles-s4               0      0      0      0
        POOL           TYPE     USED  AVAIL
   mds_backup_fs      metadata  1256G  2771G
ec82_primary_fs_data    data       0   2771G
      ec82pool          data    8440T  3110T
```

MRC Laboratory of Molecular Biology

Typically MDS failover procedure for single MDS FS



MDS Rank

For a filesystem with multiple MDS daemons, each MDS is assigned a **rank** and some subset of the metadata workload associated with that **rank**. For 1 MDS filesystems, there is only 1 rank which is the **0th rank**.

MRC Laboratory of Molecular Biology

Typically MDS failover procedure for single MDS FS

root@pebbles-s1 11:37 [ceph]: ceph tell mds.pebbles-s4 config set debug_mds 10/20

```
    -5> 2024-06-24T11:38:48.867+0100 7fa352637700 10 mds.0.journal EMetaBlob.replay noting opened inode [inode 0x1005cd4fe35
[539,head] /cephfs-users/afellows/Ferdos/20210625_real_DDFHFKLMT_KriosIII_K3/cryolo/test_micrographs/FoilHole_2764
9821_Data_27626128_27626130_20210628_005006_fractions_ave_Z124.mrc.teberet7.partial auth v436384 DIRTYPARENT s=0
n(v0 1=1+0) (iversion lock) cr={99995144=0-4194304@538} | dirtyparent=1 dirty=1 0x561e08074000]
    -4> 2024-06-24T11:38:48.867+0100 7fa352637700 10 mds.0.journal EMetaBlob.replay inotable tablev 3112837 <= table 3112837
    -3> 2024-06-24T11:38:48.867+0100 7fa352637700 10 mds.0.journal EMetaBlob.replay sessionmap v 1560540883, table
1560540882 prealloc [] used 0x1005cd4fe35
    -2> 2024-06-24T11:38:48.867+0100 7fa352637700 20 mds.0.journal  (session prealloc
[0x10051731516~0x28,0x1005174147c~0x121,0x10051741792~0xa5,0x100559851c5~0x1,0x100597d4b49~0x1,0x1005c20be47~0x
1,0x1005c604e72~0x1,0x1005c89f398~0x1,0x10
05cb53aeb~0x1,0x1005cd4b17c~0x386])
    -1> 2024-06-24T11:38:48.868+0100 7fa352637700 -1
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos8/DIST/
centos8/MACHINE_SIZE/gigantic/release/17.2.7/rpm/el8/BUILD/ceph-17.2.7/src/in
clude/interval_set.h: In function 'void interval_set<T, C>::erase(T, T, std::function<bool(T, T)>) [with T = inodeno_t; C = std::map]'
thread 7fa352637700 time 2024-06-24T11:38:48.869662+0100
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos8/DIST/
centos8/MACHINE_SIZE/gigantic/release/17.2.7/rpm/el8/BUILD/ceph-17.2.7/src/include/interval_set.h: 568: FAILED ceph_assert(p-
>first <= start)
```

MRC Laboratory of Molecular Biology

The EmetaBlob class

This is part of the metadata journalling system

```
mds.0.journal EMetaBlob.replay noting opened inode
```

- Found an open inode

```
mds.0.journal EMetaBlob.replay inotable tablev



3112837 <= table 3112837
```

- Get which version of the inotable we are on
  - Inotable is a table which lists the inodes that we are able to allocate
- The current inotable version is the same as the one the MDS has stored

The EmetaBlob class

This is part of the metadata journalling system

```
mds.0.journal EMetaBlob.replay sessionmap



v 1560540883, table 1560540882


prealloc [] used 0x1005cd4fe35
```

- Get which version of the sessionmap we are on
  - Sessionmap is the table of the client sessions which includes what their caps are, if and which preallocated inodes they have, etc.
- The live version of the sessionmap is 1 step ahead of the stored sessionmap
- The system lists the inodes that have been preallocated for the clients

The EmetaBlob class

This is part of the metadata journalling system

```
 /home/jenkins-build/…In function 'void
interval_set...FAILED ceph_assert(p->first <= start)
```

- We have crash for this assert for the interval_set function where

  ceph_assert(p->first <= start)

  Failed!

MRC Laboratory of Molecular Biology

The EmetaBlob class

This is part of the metadata journalling system

```
/home/jenkins-build/…In function 'void
interval_set...FAILED ceph_assert(p->first <= start)
```

- We have crash for this assert for the interval_set function where

  ceph_assert(p->first <= start)

  Failed!

Thus there is some corruption/ error with this sessionmap in the journal!

MRC Laboratory of Molecular Biology

Warning

If you do not have expert knowledge of CephFS internals, you will need to seek assistance before using any of these tools.

The tools mentioned here can easily cause damage as well as fixing it.

It is essential to understand exactly what has gone wrong with your file system before attempting to repair it.

If you do not have access to professional support for your cluster, consult the ceph-users mailing list or the #ceph IRC/Slack channel.

https://docs.ceph.com/en/reef/cephfs/disaster-recovery-experts/#advanced-metadata-repair-tools

cephfs-journal-tool

```
root@pebbles-s1 13:57 [~]: cephfs-journal-tool --rank=ceph_backup:0 journal export
↳ceph_backup_journal.bin
journal is 90131465778558~553606233
wrote 553606233 bytes at offset 90131465778558 to ceph_backup_journal.bin
NOTE: this is a _sparse_ file; you can
        $ tar cSzf ceph_backup_journal.bin.tgz ceph_backup_journal.bin
      to efficiently compress it while preserving sparseness.
root@pebbles-s1 13:58 [~]: ls -lrth
...
-rw-r--r--  1 root root  82T Jun 24 13:58  ceph_backup_journal.bin



# cephfs-journal-tool  --rank=${my_cephfs}:0 journal import ...
```

```
root@pebbles-s1 15:42 [~]: cephfs-journal-tool --rank=ceph_backup:0 event
↳recover_dentries summary
Events by type:
  OPEN: 334
  PURGED: 1
  SESSION: 2
  SUBTREEMAP: 134
  UPDATE: 70424
Errors: 0
```

```
root@pebbles-s1 15:42 [~]: cephfs-journal-tool --rank=ceph_backup:0 event
↳recover_dentries summary
Events by type:
  OPEN: 334
  PURGED: 1
  SESSION: 2
  SUBTREEMAP: 134
  UPDATE: 70424
Errors: 0
root@pebbles-s1 15:44 [~]: cephfs-journal-tool --rank=ceph_backup:0 journal reset
old journal was 90131465778558~553606233
new journal start will be 90132019412992 (28201 bytes past old end)
writing journal head
writing EResetJournal entry
done
```

MRC Laboratory of Molecular Biology

```
root@pebbles-s1 11:57 [~]: cephfs_failed=ceph_backup
root@pebbles-s1 11:57 [~]: cephfs_recovery=ceph_backup_recovery
root@pebbles-s1 11:57 [~]: ceph fs fail ${cephfs_failed}
ceph_backup marked not joinable; MDS cannot join the cluster. All MDS ranks
marked failed.
```

```
root@pebbles-s1 11:57 [~]: cephfs_failed=ceph_backup
root@pebbles-s1 11:57 [~]: cephfs_recovery=ceph_backup_recovery
root@pebbles-s1 11:57 [~]: ceph fs fail ${cephfs_failed}
ceph_backup marked not joinable; MDS cannot join the cluster. All MDS ranks
marked failed.
root@pebbles-s1 11:59 [~]: ceph fs new ${cephfs_recovery}
↳${cephfs_recovery}_meta
root@pebbles-s1 12:00 [~]: ceph fs add_data_pool  ${cephfs_recovery} ec82pool
root@pebbles-s1 12:02 [~]: ceph fs dump  | less
...
Filesystem 'ceph_backup' (1)
data_pools      [6,3]
metadata_pool   2
...
Filesystem 'ceph_backup_recovery' (3)
data_pools      [6,3]
metadata_pool   8
...
```

```
root@pebbles-s1 12:04 [~]:  cephfs-table-tool ${cephfs_recovery}:0 reset session

root@pebbles-s1 13:00 [~]:  cephfs-table-tool ${cephfs_recovery}:0 reset snap

root@pebbles-s1 13:00 [~]:   cephfs-table-tool ${cephfs_recovery}:0 reset inode

root@pebbles-s1 13:01 [~]: cephfs-journal-tool --rank ${cephfs_recovery}:0 journal
↳reset --force
writing EResetJournal entry
root@pebbles-s1 13:01 [~]: ceph fs status ${cephfs_recovery}
ceph_backup_recovery - 0 clients
=====================
RANK  STATE   MDS  ACTIVITY  DNS  INOS  DIRS  CAPS
 0    failed
         POOL             TYPE     USED  AVAIL
ceph_backup_recovery_meta  metadata  60.0k  3066G
   ec82_primary_fs_data      data       0   3066G
        ec82pool           data    8085T  4734T
```

cephfs-data-scan

```
cephfs-data-scan init --force-init --filesystem ${cephfs_recovery} --alternate-pool
↳${cephfs_recovery}_meta
```

```
cephfs-data-scan scan_extents --worker_n $1 --worker_m ${worker_max} --alternate-pool
↳${cephfs_recovery}_meta --filesystem ${cephfs_recovery} ${bulk_data_pool}
```

- Scan all the objects to determine the timestamps and sizes of all the inodes

```
cephfs-data-scan scan_inodes --worker_n $1 --worker_m ${worker_max} --alternate-pool
↳${cephfs_recovery}_meta --filesystem ${cephfs_recovery} ${bulk_data_pool}
```

- Scan the first object of all the inodes to rebuild the metadata

```
cephfs-data-scan scan_links --filesystem ${cephfs_recovery}
```

- Scan the inode linkages and fix any errors

cephfs-data-scan

```
HOSTS=( my_host1 my_host2 … )
worker_max=${#HOSTS[@]}
i=0
while [[ $i -lt ${max_worker} ]]; do
        HOST=${HOSTS[i]}
        ssh HOST screen -d -m
        “cephfs-data-scan scan_extents –worker_n $i  --worker_m $worker_max
        ↳--alternate-pool ${cephfs_recovery}_meta
        ↳--filesystem ${cephfs_recovery} ${bulk_data_pool}”
        ((i++))
done
```

```
root@pebbles-s1 12:04 [~]:  cephfs-journal-tool –rank ${cephfs_recovery}:0 event
↳recover_dentries list --alternate-pool ${cephfs_recovery}_meta

# ceph config rm mds mds_verify_scatter

# ceph config rm mds mds_debug_scatterstat
```

```
root@pebbles-s1 08:44 [~]: ceph fs set ${cephfs_recovery} joinable true
root@pebbles-n4 08:44 [~]: ceph fs status
ceph_backup_recovery - 0 clients
==========
RANK  STATE     MDS        ACTIVITY    DNS    INOS   DIRS   CAPS
 0    active  pebbles-s2  Reqs:  0 /s  1950k  1942k  66.2k  0
         POOL             TYPE     USED  AVAIL
     ceph_backup_recovery_meta        metadata  1018G  3512G
ec82_primary_fs_data    data       0   3512G
     ec82pool          data    7084T  7792T
STANDBY MDS
 pebbles-s4
 pebbles-s1
root@pebbles-n4 08:46 [~]: ceph tell mds.${cephfs_recovery}:0 scrub start
↳/ recursive,repair,force
```

MRC Laboratory of Molecular Biology

Clear disaster recovery procedure

- Relatively robust
- However scan_extents and scan_inodes take a very long time
  - ~ 1 month in our case
- Long recovery time may not be appropriate for your situation and potential loss of the data may be preferable to prolonged down-time

```
    -5> 2024-10-02T15:20:52.494+0100 7fa8b6d95640 10 mds.0.journal EMetaBlob.replay noting opened inode [inode 0x1002c5f1d10
[126,head] /cephfs2-users/crusso/cjrLabArchives/knayde/Data2020/2020March04_LHC2C7_SmallHole_F4MSM_Pu_Process/Polish/
job030/MotionCorrectedMovies/
FoilHole_4616489_Data_3448109_3448111_20200306_014931_Fractions_cor_stack_tracks_plot_0012.ef980630.partial auth
v283166 DIRTYPARENT s=0 n(v0 1=1+0) (iversion lock) cr={159821641=0-4194304@125} | dirtyparent=1 dirty=1 0x564f21a00000]
    -4> 2024-10-02T15:20:52.494+0100 7fa8b6d95640 10 mds.0.journal EMetaBlob.replay inotable tablev 1481899 <= table 1481899
    -3> 2024-10-02T15:20:52.494+0100 7fa8b6d95640 10 mds.0.journal EMetaBlob.replay sessionmap v 746010300, table 746010299
prealloc [] used 0x1002c5f1d10
    -2> 2024-10-02T15:20:52.494+0100 7fa8b6d95640 20 mds.0.journal  (session prealloc
[0x1002bdf9f74~0x22,0x1002bdfb51d~0x7c,0x1002c59def8~0xfb,0x1002c5eec0c~0x86,0x1002c5eee87~0x1f5])
    -1> 2024-10-02T15:20:52.494+0100 7fa8b6d95640 -1
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos9/DIST/
centos9/MACHINE_SIZE/gigantic/release/18.2.4/rpm/el9/BUILD/ceph-18.2.4/src/include/interval_set.h: In function 'void interval_set<T,
C>::erase(T, T, std::function<bool(T, T)>) [with T = inodeno_t; C = std::map]' thread 7fa8b6d95640 time 2024-10-
02T15:20:52.495403+0100
/home/jenkins-build/build/workspace/ceph-build/ARCH/x86_64/AVAILABLE_ARCH/x86_64/AVAILABLE_DIST/centos9/DIST/
centos9/MACHINE_SIZE/gigantic/release/18.2.4/rpm/el9/BUILD/ceph-18.2.4/src/include/interval_set.h: 568: FAILED ceph_assert(p-
>first <= start)
```

July 24th

2024-06-24 mds.0.journal EMetaBlob.replay noting opened inode …

2024-06-24 mds.0.journal EMetaBlob.replay inotable tablev …

2024-06-24 mds.0.journal EMetaBlob.replay sessionmap
... prealloc [] used 0x1005cd4fe35

2024-06-24 mds.0.journal …

2024-06-24 … interval_set.h: In function
… 'void interval_set<T, C>::erase(T, T, std::function<bool(T, T)>)
… interval_set.h: 568: FAILED ceph_assert(p->first <= start)

October 2nd

2024-10-02 mds.0.journal EMetaBlob.replay noting opened inode …

2024-10-02 mds.0.journal EMetaBlob.replay inotable tablev …

2024-10-02 mds.0.journal EMetaBlob.replay sessionmap
... prealloc [] used 0x1002c5f1d10

2024-10-02 mds.0.journal …

2024-10-02 … interval_set.h: In function
'void interval_set<T, C>::erase(T, T, std::function<bool(T, T)>)
… interval_set.h: 568: FAILED ceph_assert(p->first <= start)

Second cephfs disaster

- MDS error log tells us that we have a corrupt journal
  - Issue with the sessionmap potentially
- The underlying metadata store should be fine
- Everything in the journal before this should be recoverable though

Second cephfs disaster

- MDS error log tells us that we have a corrupt journal
  - Issue with the sessionmap potentially
- The underlying metadata store should be fine
- Everything in the journal before this should be recoverable though

Thus recovering the journal and resetting the various cephfs tables should allow replay to finish

Second cephfs disaster

```
root@pebbles-s1 08:28 [~]: rados -p  ${meta_pool} export ${meta_pool}_pool.bak

root@pebbles-s1 13:36 [~]: cephfs-journal-tool --rank=${my_cephfs}:0 journal export
↳${my_cephfs}.bak.jrnl
```

Second cephfs disaster

```
root@pebbles-s1 13:37 [~]: cephfs-journal-tool --rank=${my_cephfs}:0 event
↳recover_dentries summary
...
2024-10-06T09:14:45.693+0100 7f8e33a50280  1 recover_dentries: frag
10004904022.00000000 is corrupt, overwriting
Events by type:
  FRAGMENT: 10482
  OPEN: 1358423
  SESSION: 8
  SUBTREEMAP: 103078
  TABLECLIENT: 8
  TABLESERVER: 22
  UPDATE: 65991730
Errors: 0

root@pebbles-s1 15:46 [~]: cephfs-journal-tool --rank=${my_cephfs}:0 journal reset
```

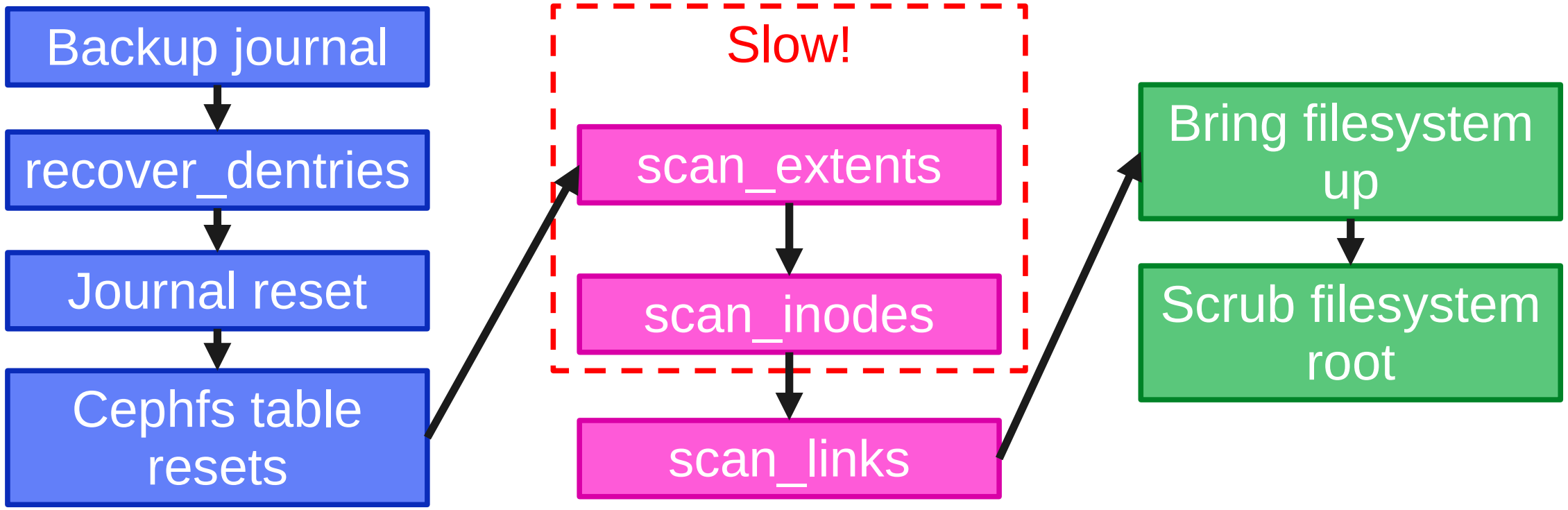
Second cephfs disaster

```
root@pebbles-s1 15:46 [~]: cephfs-table-tool ${my_cephfs}:0 reset session

root@pebbles-s1 15:46 [~]: cephfs-data-scan scan_links --filesystem ${my_cephfs}
```

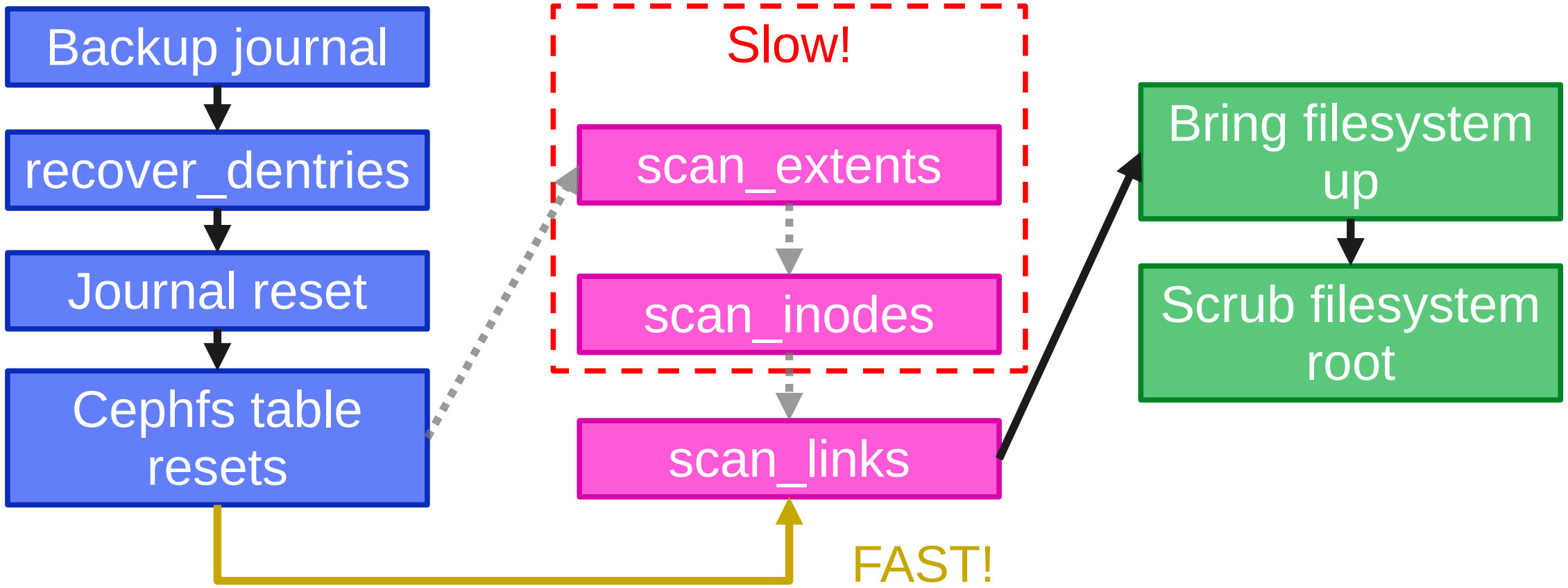
Second cephfs disaster

```
root@pebbles-s1 08:16 [~]:  ceph fs set ${my_cephfs} joinable true
ceph_spare marked joinable; MDS may join as newly active.

root@pebbles-s1 08:16 [~]: ceph tell mds.${my_cephfs}:0 scrub start
↳/ recursive,repair,force
...
{
    "return_code": 0,
    "scrub_tag": "1528de35-5508-4636-841e-3d0a030b7b10",
    "mode": "asynchronous"
}
```

Second cephfs disaster recovery



Second cephfs disaster recovery



MRC Laboratory of Molecular Biology

Second cephfs disaster recovery

- We assumed that the metadata pool was undamaged and could be brought back “as is” following a simple repair or rebuild
- We assumed that transactions associated with the corrupt parts of the journal were unsuccessful and made no underlying changes to any of the data pools
- Potentially lost or corrupted data would be overwritten with correct data when a new backup session started
- Our object pool was too large to recover in a feasible amount of time and no alternative backup cluster was available to continue the backup process during recovery

Postmortem – why did this happen?

- Still unclear why this occurred
- Suspect that asynchronous writes with aggressive parallel rclone jobs can lead to write issues
- We've set all our backup mounts to do synchronous writes and we've not had this sessionmap error again
  - Compared with getting 2 errors in the space of 3 months
- This was on Reef (version 18.2.4) and we have moved to Squid (version 19.2.2)

MRC Laboratory of Molecular Biology

Acknowledgements

Jake Grimmett

Toby Darling

MRC Laboratory of Molecular Biology

MRC Laboratory
of Molecular
Biology

Thank you!

www2.mrc-lmb.cam.ac.uk

ivan@mrc-lmb.cam.ac.uk