HANDLING A FULL CEPH FILESYSTEM

When a RADOS cluster reaches its `mon_osd_full_ratio` (default 95%) capacity, it is marked with the OSD full flag. This flag causes most normal RADOS clients to pause all operations until it is resolved (for example by adding more capacity to the cluster).

The filesystem has some special handling of the full flag, explained below.

HAMMER AND LATER

Since the hammer release, a full filesystem will lead to ENOSPC results from:

- Data writes on the client
- Metadata operations other than deletes and truncates

Because the full condition may not be encountered until data is flushed to disk (sometime after a `write` call has already returned 0), the ENOSPC error may not be seen until the application calls `fsync` or `fclose` (or equivalent) on the file handle.

Calling `fsync` is guaranteed to reliably indicate whether the data made it to disk, and will return an error if it doesn't. `fclose` will only return an error if buffered data happened to be flushed since the last write – a successful `fclose` does not guarantee that the data made it to disk, and in a full-space situation, buffered data may be discarded after an `fclose` if no space is available to persist it.

Warning: If an application appears to be misbehaving on a full filesystem, check that it is performing `fsync()` calls as necessary to ensure data is on disk before proceeding.

Data writes may be cancelled by the client if they are in flight at the time the OSD full flag is sent. Clients update the `osd_epoch_barrier` when releasing capabilities on files affected by cancelled operations, in order to ensure that these cancelled operations do not interfere with subsequent access to the data objects by the MDS or other clients. For more on the epoch barrier mechanism, see Background: Blacklisting and OSD epoch barrier.

LEGACY (PRE-HAMMER) BEHAVIOR

In versions of Ceph earlier than hammer, the MDS would ignore the full status of the RADOS cluster, and any data writes from clients would stall until the cluster ceased to be full.

There are two dangerous conditions to watch for with this behaviour:

- If a client had pending writes to a file, then it was not possible for the client to release the file to the MDS for deletion: this could lead to difficulty clearing space on a full filesystem
- If clients continued to create a large number of empty files, the resulting metadata writes from the MDS could lead to total exhaustion of space on the OSDs such that no further deletions could be performed.