CONFIGURING DIRECTORY FRAGMENTATION

In CephFS, directories are *fragmented* when they become very large or very busy. This splits up the metadata so that it can be shared between multiple MDS daemons, and between multiple objects in the metadata pool.

In normal operation, directory fragmentation is invisbible to users and administrators, and all the configuration settings mentioned here should be left at their default values.

While directory fragmentation enables CephFS to handle very large numbers of entries in a single directory, application programmers should remain conservative about creating very large directories, as they still have a resource cost in situations such as a CephFS client listing the directory, where all the fragments must be loaded at once.

All directories are initially created as a single fragment. This fragment may be *split* to divide up the directory into more fragments, and these fragments may be *merged* to reduce the number of fragments in the directory.

SPLITTING AND MERGING

An MDS will only consider doing splits if the allow_dirfrags setting is true in the file system map (set on the mons). This setting is true by default since the *Luminous* release (12.2.X).

When an MDS identifies a directory fragment to be split, it does not do the split immediately. Because splitting interrupts metadata IO, a short delay is used to allow short bursts of client IO to complete before the split begins. This delay is configured with `mds_bal_fragment_interval`, which defaults to 5 seconds.

When the split is done, the directory fragment is broken up into a power of two number of new fragments. The number of new fragments is given by two to the power `mds_bal_split_bits`, i.e. if `mds_bal_split_bits` is 2, then four new fragments will be created. The default setting is 3, i.e. splits create 8 new fragments.

The criteria for initiating a split or a merge are described in the following sections.

SIZE THRESHOLDS

A directory fragment is elegible for splitting when its size exceeds `mds_bal_split_size` (default 10000). Ordinarily this split is delayed by `mds_bal_fragment_interval`, but if the fragment size exceeds a factor of `mds_bal_fragment_fast_factor` the split size, the split will happen immediately (holding up any client metadata IO on the directory).

`mds_bal_fragment_size_max` is the hard limit on the size of directory fragments. If it is reached, clients will receive ENOSPC errors if they try to create files in the fragment. On a properly configured system, this limit should never be reached on ordinary directories, as they will have split long before. By default, this is set to 10 times the split size, giving a dirfrag size limit of 100000. Increasing this limit may lead to oversized directory fragment objects in the metadata pool, which the OSDs may not be able to handle.

A directory fragment is elegible for merging when its size is less than `mds_bal_merge_size`. There is no merge equivalent of the “fast splitting” explained above: fast splitting exists to avoid creating oversized directory fragments, there is no equivalent issue to avoid when merging. The default merge size is 50.

ACTIVITY THRESHOLDS

In addition to splitting fragments based on their size, the MDS may split directory fragments if their activity exceeds a threshold.

The MDS maintains separate time-decaying load counters for read and write operations on directory fragments. The decaying load counters have an exponential decay based on the `mds_decay_halflife` setting.

On writes, the write counter is incremented, and compared with `mds_bal_split_wr`, triggering a split if the threshold is exceeded. Write operations include metadata IO such as renames, unlinks and creations.

The `mds_bal_split_rd` threshold is applied based on the read operation load counter, which tracks readdir operations.

By the default, the read threshold is 25000 and the write threshold is 10000, i.e. 2.5x as many reads as writes would be required to trigger a split.

After fragments are split due to the activity thresholds, they are only merged based on the size threshold (`mds_bal_merge_size`), so a spike in activity may cause a directory to stay fragmented forever unless some entries are unlinked.