Splitting Dir HLD

Author WangDi

Aug,25,2006

1 Introduction

In CMD, when the dir entries exceed some limit number, the dir will be split and distributed over serval MDSes for load balancing. Reading readdir HLD, understanding iam and CMD are assumed for this HLD. Note: this splitting is just a feature for this cycle, and it will be removed after this cycle.

2 Requirements

- Split handling should be constrained to lmv and cmm, which should be transparent to other layers.
- The dir should be split according to some rule, then client(lmv) could know which mds it should go when it met such split objects.

3 Functional specification

Splitting dir API will discuss the following issue:

- In CMM layer, checking whether the dir need to be split, if it is, just split it.
- When client met such objects, it should help to find the right mds.

4 Use cases

4.1 Splitting dir

- Before creating objects in cmm layer, it will check whether the dir need to be split, if it is, it will call the spitting API to split the dir.
- After the dir is split, the client will be notified and redo this create.

4.2 Access to split dir

- When handling a request in client(LMV layer), before looking up the right mds, it should check whether the object is split.
- If it is split, it should find the right MDS according to some rule.
- After that, the req will be sent to to that MDS.

4.3 readdir in a split dir

- Client: send readdir req to the first MDS according to the hash interval.
- Server: execute readdir in hash order and fill the request buffer with the dir entries.
- Client: get all the entries of the MDS, the client will resort to the next MDS in the hash interval, Until it get all the entries of the dir.

5 Logic specification

5.1 Splitting objects

5.1.1 splitting rules

According to HLD of readdir, when splitting, we will define several equal segments to distribute the objects.

- Checking the dir to make sure whether it should be split. Only when the count of the entries reached the upper limit number, it could be split.
- Divide total _hash_ range into equal intervals and assign each interval to separate MDS. Dir hash value will have 30bits, since the lowest and biggest bits are clear.
- Iterate over the index entries of the dir and scatter index entries to each MDS.

5.1.2 splitting objects in cmm layer

When splitting objects in cmm layer, the process should be

- Before creating the objects, checking whether the parent of should be split. Note: there are some cases, the object should not be split, even the dir entries count already reach the upper limit count.

  - Root object should not be split.
  - Object with unsplit flags should not be split.
  - Only dir could be split.

- Creating slave objects in other MDS, then get split EA and set it to the dir EA.
- Scan the entry of this dir object by the object index api, and split it.
  - calculate each segment width. (0x3fffffff/mds_count, see 5.1.1)
  - scan the entry of the object, since entries are stored according to hash value, so we can
    * retrieve the entries from the bottom device, and allocate new fid for each entry.
    * then the remote MDS unpack them and insert them to the slave object, so a new writepage handler need to be added in mdt readpage service.
    * The bulk page transfer RPC mechanism would be implemented between MDS to transfer these split dir entries. (Note: This bulk page rpc mechanism is already implemented in cmd2, we just need port it to new CMD layer).
- After splitting the objects, the MDS should notify client(lmv), then client will recreate the object under the split parent dir.

After MDS splits the objects, client will retrieve the striped info from MDS, after that, it will create the split object in client cache for later access, which has been implemented in lmv module now.

5.2 Access split dir

5.2.1 Access split object in lmv

Currently all of this has been implemented in lmv module as originally, we will keep it untouched in this cycle.

5.2.2 Access split dir in readdir

This is discussed in another HLD (readdir HLD).

5.2.3 Access split object in CMM layer

CMM layer will handling those split objects in the new stack layer, which is already implemented in cmm layer, and discussed in other HLD.

6 State management

6.1 Split locking

While split will only happen in creating process, so the split parent object will be protect by EX lock in the whole split process.

6.2 Recovery changes

Since splitting will be only used for some performance test in this cycle, and will be removed after this cycle, so recovery of it will not be needed. Note: in recovery verification test, there will be no split.