SUBTREE EXPORTS

NORMAL MIGRATION

The exporter begins by doing some checks in export_dir() to verify that it is permissible to export the subtree at this time. In particular, the cluster must not be degraded, the subtree root may not be freezing or frozen (ie already exporting, or nested beneath something that is exporting), and the path must be pinned (ie not conflicted with a rename). If these conditions are met, the subtree freeze is initiated, and the exporter is committed to the subtree migration, barring an intervening failure of the importer or itself.

The MExportDirDiscover serves simply to ensure that the base directory being exported is open on the destination node. It is pinned by the importer to prevent it from being trimmed. This occurs before the exporter completes the freeze of the subtree to ensure that the importer is able to replicate the necessary metadata. When the exporter receives the MExportDirDiscoverAck, it allows the freeze to proceed.

The MExportDirPrep message then follows to populate a spanning tree that includes all dirs, inodes, and dentries necessary to reach any nested exports within the exported region. This replicates metadata as well, but it is pushed out by the exporter, avoiding deadlock with the regular discover and replication process. The importer is responsible for opening the bounding directories from any third parties before acknowledging. This ensures that the importer has correct dir_auth information about where authority is delegated for all points nested within the subtree being migrated. While processing the MExportDirPrep, the importer freezes the entire subtree region to prevent any new replication or cache expiration.

The warning stage occurs only if the base subtree directory is open by nodes other than the importer and exporter. If so, then a MExportDirNotify message informs any bystanders that the authority for the region is temporarily ambiguous. In particular, bystanders who are trimming items from their cache must send MCacheExpire messages to both the old and new authorities. This is necessary to ensure that the surviving authority reliably receives all expirations even if the importer or exporter fails. While the subtree is frozen (on both the importer and exporter), expirations will not be immediately processed; instead, they will be queued until the region is unfrozen and it can be determined that the node is or is not authoritative for the region.

The MExportDir message sends the actual subtree metadata to the importer. Upon receipt, the importer inserts the data into its cache, logs a copy in the EImportStart, and replies with an MExportDirAck. The exporter can now log an EExport, which ultimately specifies that the export was a success. In the presence of failures, it is the existence of the EExport that disambiguates authority during recovery.

Once logged, the exporter will send an MExportDirNotify to any bystanders, informing them that the authority is no longer ambiguous and cache expirations should be sent only to the new authority (the importer). Once these are acknowledged, implicitly flushing the bystander to exporter message streams of any stray expiration notices, the exporter unfreezes the subtree, cleans up its state, and sends a final MExportDirFinish to the importer. Upon receipt, the importer logs an EImportFinish(true), unfreezes its subtree, and cleans up its state.

PARTIAL FAILURE RECOVERY

RECOVERY FROM JOURNAL