HLD for CMD3 Stability Fixing

Huang Hua

2007-03-15

1 Introduction

This task mainly focuses on some stability fixes, specifically fixes for issues that were found during cmd3 time-frame, but weren't fixed due to time constraints. These fixes should give us a more stable baseline code, more reasonable implementation.

2 Requirements

The main goal of this task is to prepare a stable and clean code baseline for release b1_8. There are some problems and issues existing in current code due to time constraints. These problems are supposed to be resolved in this cycle as soon as possible. We will classify those fixes into three categories: mandatory fixes, optional fixes, and optimizations, from high priority to low priority. Mandatory fixes must be finished in this cycle. Optional fixes and optimizations will be implemented only if we have enough time.

2.1 Mandatory Fixes

1. Separate portal for mds-mds requests.
2. MDS originated destroy.
3. removal of lustre_shrink_reply().
4. osd_create().
5. join-file feature.
6. Reconnection refusal in recovery.
7. Remove unused kernel patches.
8. Revert back changes in recovery due to multipoint failures.
9. readdir issues.

2.2 Optional Fixes

1. Open Lock should be added.
2. Fixing all FIXME and TODO.

2.3 Optimizations

1. symlink in inode.
2. last_rcvd enhancement.

3 Functional specification

3.1 Separate portal for mds-mds requests

Separate portal for mds-mds requests is needed for recovery purposes to avoid deadlock. If some MDS is stuck-ed due to failure of another MDS, it can not handle import connect because its all thread are occupied. This is only for CMD feature. Release b1_8 will not support CMD feature officially. This has been done by tappro, and need to apply patch only.

3.2 MDS originated destroy

Being able destroy OST objects from MDS is needed for some cases like orphan handling, cross-ref destroy, error handling and etc.

3.3 removal of lustre_shrink_reply()

req_capsule interface needs an equivalent of lustre_shrink_reply(). We will design a new function to replace it, and also do not use segment offset as a parameter. After this no explicit buffer offsets would be present in cmd3 mds code.

3.4 osd_create()

ldiskfs uses parent directory as an argument to ino allocation algorithm as a hint. Our code always supplies inode of top level directory instead. This results in sub-optimal inode allocation. We need to pass some allocation hint to it.

3.5 join-file feature

Join-file feature is not supported currently in CMD3 code. This feature need to be added.

3.6 Reconnection refusal in recovery

class_export_rpc_put/get() can lead to 'deadlock' situation after disconnection if client's request is in progress and server is waiting for the same client to sent AST. Server refuses client to reconnect now and evict it.

3.7 Remove unused kernel patches

review all kernel patches, maybe we don't need some of them which were added just for performance but have bad impact in other areas if any.

3.8 Revert back changes in recovery due to multi-point failures

revert back changes in recovery due to multi-point failures (that was hacks, they will remain in b_new_cmd so we can re-design them when it will be needed. But now we need clear recovery code even if it can't handle multi-point issues.

3.9 readdir issues

There are some corner cases in readdir handling.

3.10 Fixing all FIXME and TODO

Overview all FIXME's, and TODO's from the sources, and try to fix them.

4 Use cases

Most parts of this task are stability fixes, and the regular release tests should pass. There are also some tests against join file and other features, and all these tests should pass.

4.1 Reconnection refusal in recovery

This situation should be thoroughly tested in various recovery circumstances: such as network failure between different nodes, a reboot event of some node, and so on.

4.2 readdir issues

Corner cases should be tested:

- Hash collision;
- File name length from 1 to 255 (that is max).
- sub-items from 1 to vary large. up to more than 10,000,000 sub-items in a directory.
- Concurrent operation in a directory from a single node and multiple node. After operations, readdir() should give correct results.

5 Logic specification

This task includes code cleanup and feature porting. Some design and implementation already exist, and only need to be modified/ported. So, please refer to the original design HLD/DLD.

5.1 osd_create()

A new data type named “struct lu_allocation_hint” will be introduced. This is a hint where and how to create a new object. This hint can be generated from parent and child relationship.

5.2 req_capsule_shrink_reply()

This function will be added to do reply message buffer shrinking if some segments of reply is smaller than planned, and move the higher segment forward if needed and applicable. It first decides the offset of the specified field in reply message, and then calls lustre_shrink_reply() to complete its functionality. The users of req_capsule_*() will not care about request/reply buffer offset any more.

5.3 Reconnection refusal in recovery

Server need to send AST to disconnected client using old export.

5.4 MDS originated destroy

Sometimes, MDT need to destroy some objects when doing rollback, orphan handling, and etc. This can do done by obd_destyoy() while passing corresponding osc as argument. Code should be ported from old MDS.

6 State management

6.1 Locking changes

Open lock will be added if we finish adding it.

6.2 Disk format changes

Disk format will not be changed.

6.3 API changes

1. The do_create() will be changed: allocation hint will be added to the parameter list.
2. The lustre_shirnk_reply() will be replaced by req_shrink_reply().

7 Focus for Inspection

1. Will the change to do_create() violate the object storage device interface design? In OSD, objects have no relation between each other (I mean the parent/sub-item relationship). The relationship between objects in file system forms a directory tree. If we add the parent object to do_create() interface, we maybe break it.
2. split issues. Directory split is a feature required by Hendrix project. Because the CMD feature is not supported officially in b1_8, this split feature will not be supported officially. So we only need to make it sure that it basically works, and does not break during normal operations.