Lustre* snapshot

fan.yong@intel.com

High Performance Data Division

Lustre* needs breakthrough on advance features!

Traditionally, Lustre scenarios were restricted in enterprise environment because of lacking advance features. Snapshot is one of them.

- Requirement source
  - Enterprise users
- Technical background
  - ZFS-based backend
- Funding resources
  - Intel self-sponsored

*Other names and brands may be claimed as the property of others.

How can Lustre* snapshot be used?

- Restore the old version file(s) from the snapshot
  - The file is removed by misoperation, no recycle bin.
  - Application failures cause the file data to be invalid.
- Restore the whole Lustre system from the snapshot
  - Some serious software bugs may cause the system corruption.
  - Upgrade/downgrade Lustre/kernel may hit some incompatible trouble.
- Create backup from the snapshot
  - It defines the time-point with that the backup tools can work deliberately.
  - Snapshot is NOT safe backup, it can NOT handle device-level corruption.

*Other names and brands may be claimed as the property of others.

Project overview

- Snapshot I: ZFS backend based snapshot
  - Relative simple and reliable solution
  - Only for Intel EE Lustre* release, IEEL3.0
  - Main components (not only for snapshot)
    - Global write barrier
    - Fork/erase Lustre configuration
    - Mount snapshot as read only
    - User space interfaces
- Snapshot II: Lustre native snapshot
  - More controllable and independent solution
  - The users’ feedback for Lustre snapshot I

*Other names and brands may be claimed as the property of others.

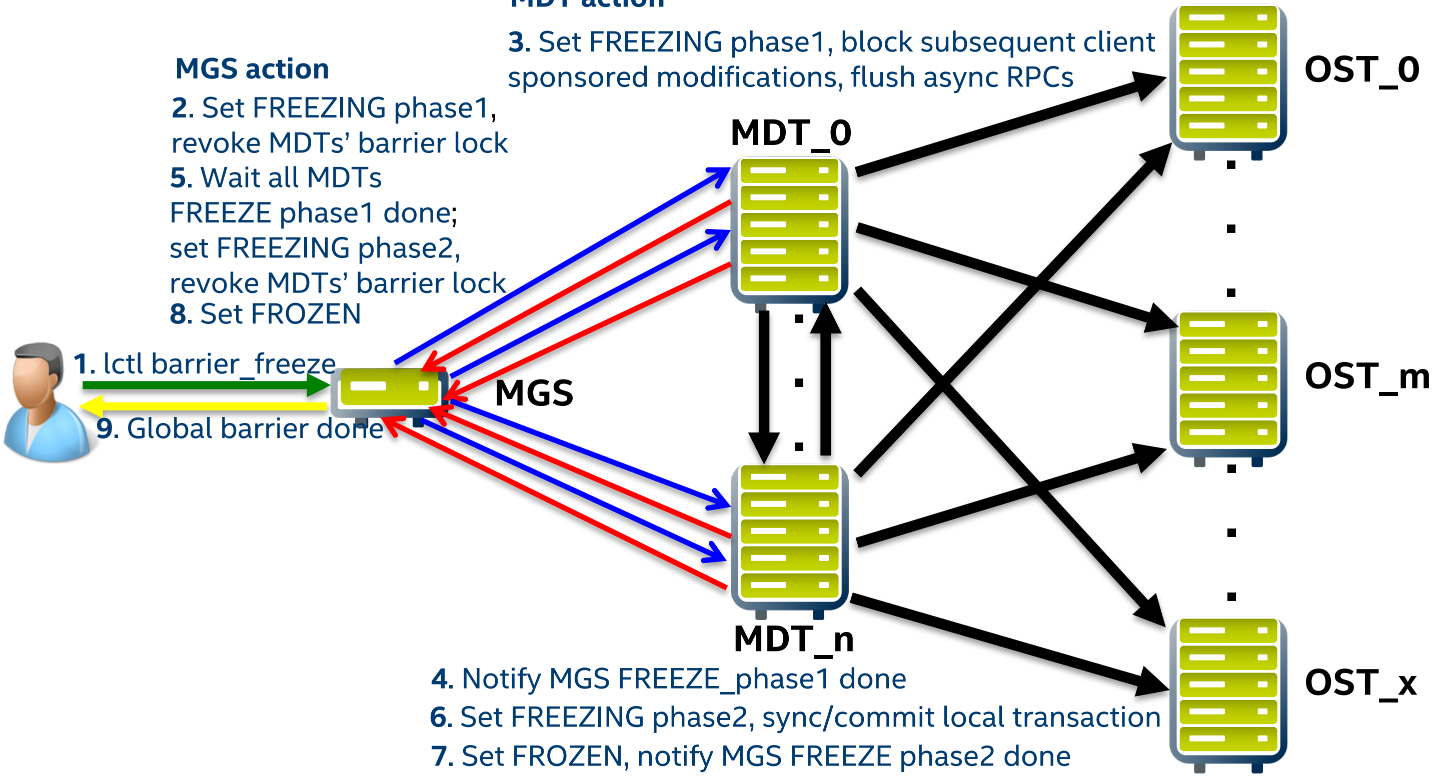
Global write barrier

"Freeze" the Lustre* system for a shot time during creating local snapshot pieces on every Lustre target (MGS/MDT/OST).

- Write barrier on MDTs only
- Each-MDT self-controlled timeout mechanism guarantees that the MDT/system will not be blocked for ever by malicious barrier owner or network/target trouble
- New lctl commands to control the global write barrier
  - lctl barrier_freeze <fsname> [timeout]
  - lctl barrier_thaw <fsname>
  - lctl barrier_status <fsname>

*Other names and brands may be claimed as the property of others.

Two phases to setup global write barrier



MDT action

3. Set FREEZING phase1, block subsequent client sponsored modifications, flush async RPCs

MGS action

2. Set FREEZING phase1, revoke MDTs' barrier lock

5. Wait all MDTs FREEZE phase1 done; set FREEZING phase2, revoke MDTs' barrier lock

8. Set FROZEN

1. lctl barrier_freeze

9. Global barrier done

4. Notify MGS FREEZE_phase1 done

6. Set FREEZING phase2, sync/commit local transaction

7. Set FROZEN, notify MGS FREEZE phase2 done

Fork/erase Lustre* config-logs

The snapshot needs new filesystem name for being mounted when the original Lustre system is running.

- The snapshot is independent from the original filesystem
- Filesystem name exists in per-target based Lustre config-logs
- Some filesystem parameters are in per-target based config-logs
- Fork/erase Lustre config-logs is based on original filesystem, not snapshot
- New lctl commands to fork/erase Lustre config-logs
  - lctl fork_lcfg <fsname> <new_name>
  - lctl erase_lcfg <fsname>

Mount snapshot as read only

ZFS snapshot is strictly readonly, any attempt to modify the snapshot will trigger backend failure or assertion, so must filter out all modifications.

- Open ZFS dataset as readonly mode
- NOT start cross-server sync thread
- NOT start pre-create thread
- NOT start quota thread
- Skip copy local config-logs
- Skip seq file initialization
- Skip orphan cleanup
- Skip recovery
- Ignore last_rcvd modification
- Ignore local sync operations
- Refuse to create transaction
- Forbid LFSCK

User space interfaces - lsnapshot

- Lustre* snapshot configuration: **/etc/lsnapshot.conf**
  - Format: *<host> <pool_dir> <pool> <local_filesystem> <role(,s)> <index>*
- Lustre snapshot logs: **/etc/lsnapshot.log**
- Setup global write barrier before creating snapshot
- Fork filesystem config-logs before creating snapshot
- Maintain local snapshot pieces on the every target (MGS/MDT/OST) with global Lustre snapshot interfaces
  - Executable binary tools set: create, destroy, list, modify, mount, umount, …

lsnapshot create

- Create snapshot for the given filesystem.

```
lsnapshot create <snapshot_name> [-b | --barrier [on | off]] [-c | --comment "comment"] [-h | --help] [-t | --timeout
time]
Options:
-b: set write barrier before creating snapshot, the default value is 'on'.
-c: describe what the snapshot is for, and so on.
-h: for help information.
-t: the life cycle (seconds) for write barrier, the default value is 60 seconds.
```

- Example 1: create snapshot “mysnapshot_1” with comment “This is a test”. Before that check lsnapshot.conf that will be used in subsequent tests also.

```
# cat /etc/lsnapshot.conf
rhel6  /tmp  lustre-mdt1  mdt1  MGS,MDT    0
rhel6  /tmp  lustre-mdt2  mdt2  MDT   1
rhel6  /tmp  lustre-ost1  ost1  OST   0
rhel6  /tmp  lustre-ost2  ost2  OST   1
root@RHEL6:~/
```

```
# lsnapshot create mysnapshot_1 –c “This is a test”
root@RHEL6:~/
# cat /etc/lsnapshot.log
Mon Aug 31 15:01:49 2015 (24010:main:2235): Create
snapshot mysnapshot_1 successfully with comment <This
is a test>, barrier <enable>, timeout <60>
```

lsnapshot list

- List the snapshot or all snapshots.

```
lsnapshot list [snapshot_name] [-d | --detail] [-h | --help]
Options:
-d: list every piece for the specified snapshot.
-h: for help information.
```

- Example 2: list the just created snapshot “mysnapshot_1”.

```
# lsnapshot list mysnapshot_1

filesystem_name: lustre
snapshot_name: mysnapshot_1
create_time: Mon Aug 31 15:01:25 2015
modify_time: Mon Aug 31 15:01:25 2015
snapshot_fsname: 19d4c4ef
comment: This is a test
status: not mount
```

- Example 3: list all the snapshots.

```
# lsnapshot create mysnapshot_2 -c “More snapshots”
root@RHEL6:~/
# lsnapshot list

filesystem_name: lustre
snapshot_name: mysnapshot_2
comment: More snapshots
create_time: Mon Aug 31 15:54:40 2015
modify_time: Mon Aug 31 15:54:40 2015
snapshot_fsname: 707a270e
status: not mount

filesystem_name: lustre
snapshot_name: mysnapshot_1
create_time: Mon Aug 31 15:01:25 2015
modify_time: Mon Aug 31 15:01:25 2015
snapshot_fsname: 19d4c4ef
comment: This is a test
status: not mount
```

lsnapshot mount/umount

- mount usage:

```
lsnapshot mount <snapshot_name>
```

- Example 4: mount “mysnapshot_1”.

```
# lsnapshot mount mysnapshot_1
root@RHEL6:~/
# lsnapshot list mysnapshot_1

filesystem_name: lustre
snapshot_name: mysnapshot_1
create_time: Mon Aug 31 15:01:25 2015
modify_time: Mon Aug 31 15:01:25 2015
snapshot_fsname: 19d4c4ef
comment: This is a test
status: mounted
root@RHEL6:~/
# mount -t lustre -o ro rhel6@tcp:/19d4c4ef /mnt/lustre2
root@RHEL6:~/
```

- umount usage:

```
lsnapshot umount <snapshot_name>
```

- Example 5: umount “mysnapshot_1”.

```
# lsnapshot umount mysnapshot_1
root@RHEL6:~/
# lsnapshot list mysnapshot_1

filesystem_name: lustre
snapshot_name: mysnapshot_1
create_time: Mon Aug 31 15:01:25 2015
modify_time: Mon Aug 31 15:01:25 2015
snapshot_fsname: 19d4c4ef
comment: This is a test
status: not mount
root@RHEL6:~/
```

lsnapshot destroy

- Destroy the specified snapshot.

```
lsnapshot destroy <snapshot_name> [-f | --force] [-h | --help]
Options:
-f: destroy the snapshot by force.
-h: for help information.
```

- Example 6: destroy “mysnapshot_1”.

```
# lsnapshot destroy mysnapshot_1

root@RHEL6:~/
# lsnapshot list mysnapshot_1
The target mysnapshot_1 is not Lustre snapshot or does not exists
Can't list the snapshot mysnapshot_1
```

- Example 7: forbid to destroy the snapshot if it is in using.

```
# lsnapshot mount mysnapshot_2

root@RHEL6:~/
# lsnapshot destroy mysnapshot_2
cannot destroy snapshot lustre-mdt1/mdt1@mysnapshot_2: dataset is busy
cannot destroy snapshot lustre-mdt2/mdt2@mysnapshot_2: dataset is busy
cannot destroy snapshot lustre-ost1/ost1@mysnapshot_2: dataset is busy
cannot destroy snapshot lustre-ost2/ost2@mysnapshot_2: dataset is busy
Can't destroy the snapshot mysnapshot_2
root@RHEL6:~/
# lsnapshot umount mysnapshot_2
```

lsnapshot modify

- Change the specified snapshot's name and/or comment.

```
lsnapshot modify <snapshot_name> [-c | --comment "new_comment"] [-h | --help] [-n | --name new_snapshot_name]
Options:
-c: update the snapshot's comment.
-h: for help information.
-n: rename the snapshot's name.
```

- Example 8: rename “mysnapshot_2”.

```
# lsnapshot modify mysnapshot_2 -n mysnapshot_3 -c “My original name was mysnapshot_2”
root@RHEL6:~/
# lsnapshot list mysnapshot_3

filesystem_name: lustre
snapshot_name: mysnapshot_3
comment: My original name was mysnapshot_2
create_time: Mon Aug 31 15:54:40 2015
modify_time: Mon Aug 31 17:10:02 2015
snapshot_fsname: 707a270e
status: not mount
```

intel®

experience
what's inside™