Fault-tolerant Service by Raft in Lustre

Lai Siyao, Zhang Hongchao
Oct. 2025



Agenda

- Overview
- Architecture
- Raft Implementation
- Fault-tolerant MGS (FTM)
- Future Plan

© DDN 2024

Overview

- Lustre Internal Services
  - Lustre Management Service (MGS) – mount, parameters, imperative recovery, nodemap
  - FID Location DataBase (FLDB) – manage the mapping of FID's sequence and the Targets (MDT, OST)
  - Quota Master Target (QMT) -- provides the Quota service to all Targets (MDT, OST)
- Fault-tolerant
  - Reliability: avoid configuration files loss/corruption
  - Availability: the service (MGS, FLDB, QMT) running on replicated nodes can be used
- Performance Improvement
  - Shortens mount time for large clusters: client mount requests handled by all MGS services
  - Improve the Quota performance, especially, if there are large number of Quota IDs
- Current Status
  - The first stage mainly focus on Fault-tolerant MGS (FTM)

© DDN 2024

Architecture

- Lustre Raft Implementation
  - Generic Raft protocol implementation from https://github.com/willemt/raft
  - Use the index of MDT as the node ID of the Raft node
  - Network (ptlrpc) and storage(LLOG) integration with Lustre
  - Raft initialization, handers, callback and cleanup
  - Each MDT can be regarded as a Raft Node if it is mounted with “-o raft” or “-o raft_nonvote”
- Multiple Service support
  - Start MGS/FLDB/Quota service on all the node with MDT mounted as Raft node
  - The Updates for each service will be propagated to other Raft nodes by the Leader node
- Service configuration update by Raft
  - Replicated Service configuration changes by Raft FSM (finite state machine)

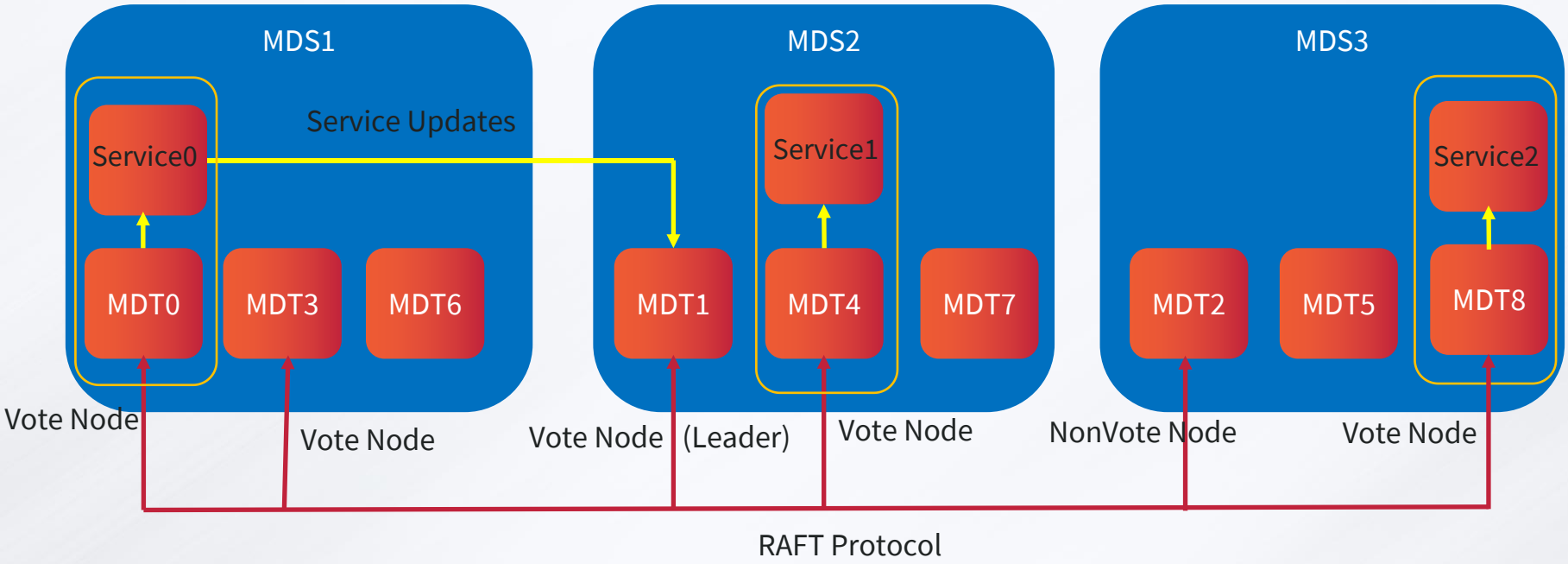
© DDN 2024

Raft Implementation

- Core components
  - New kernel module: lraft
  - Leader election: Automatic selection of Leader node
  - Raft Log replication: Consistent state across nodes
  - Fault tolerance: Survives (N-1)/2 node failures
- Deployment
  - Mount option ‘-o raft’ for voting nodes, ‘-o raft_nonvote’ for non-voting nodes
  - 5 voting nodes: Tolerate 2 node failures
  - 3 voting nodes: Tolerate 1 node failure
  - Additional non-voting nodes: Candidates for removed voting nodes

© DDN 2024

Raft Implementation



© DDN 2024

Raft Implementation: Requests

- RAFT_HANDSHAKE
  Each Target (MDT and OST) will process this request even even if it is ***NOT*** a RAFT node,
  Target will send it target info back to the caller, which will generate the MGS configuration by it.

- RAFT_REQUESTVOTE
  Raft (vote) node process this request to vote for the Leader Election

- RAFT_APPENDENTRIES
  Raft node (vote and nonvote) will process this request to receive the logs entries from Leader

- RAFT_SET_INFO
  Raft node use this request to forward request to Leader node

© DDN 2024

Raft Implementation: Raft Callback

- send_requestvote
  - Send request to peer Raft nodes to elect itself as Leader
- send_appendentries
  - Leader node sends log entries to other Raft nodes
- persist_vote, persist_term
  - Save the current vote/term result to disk
- log_offer, log_clear, applylog, log_pop
  - Process log entries (Raft FSM)
  - Log is allocated in log_offer and freed in log_clear
  - applylog will be called if Raft thinks the log entry has been committed
  - Log is deleted by calling log_pop

© DDN 2024

FTM

- Fault-tolarent MGS (FTM)
  - Part of the Lustre Metadata Redundancy (LMR) design to avoid dependence on MDT0000 for critical filesystem services
  - Improve reliability by avoiding loss/corruption of the filesystem configuration llog files
  - Improve availability by avoiding filesystem inaccessibility if the MGS service is failed over
  - Reduce recovery time, the MGS IR will be maintained in memory on all of the MGS replicas
  - The HA configuration can be simplified, MGS service will start on every node with an MDT

© DDN 2024

FTM: Multiple MGS

- Starter MGS (MGS0)
  - First MGS to bring MDTs/OSTs up
  - Can be an MDT or separate device
- Additional MGS
  - On other MDTs with ‘mgs’ mount option (-o raft, -o raft_nonvote)
  - Must be one Raft nodes
  - Config logs generated after receiving the Target registration info during Raft HandShake
  - Can be used for client mount

© DDN 2024

FTM: Configuration Files

- The Starter MGS
  - Other Target (OST and MDT) send its target info to register on the starter MGS in the old ways
  - It will not propagate the target info to other Raft nodes.
- Additional MGS
  - During processing RAFT_HANDSHAKE request, other MDT/OSTs send it target info back
  - The additional MGS use it to generate the MGS configuration files.
  - Client can connect to these MGS to mount by the generated configuration files.

© DDN 2024

FTM: Parameters Update

- Synchronize parameters update by Raft
  - MGS0 processes the parameters update
  - MGS0 sends the parameters change to Leader node if it is not the Leader node
  - Leader node replicates parameters changes to other nodes
  - All MGS process the parameters change after receive the message from Raft
- Configuration Regeneration
  - The configuration files will be regenerated after the replicated MGS is restarted
  - The parameters changes will be re-processed to apply on the newly created config files

© DDN 2024

FTM: Nodemap

- Synchronize nodemap update by Raft
  - There is only one Nodemap instance in one node
  - Nodemap update request will be forwarded to leader, then send it to other nodes by Leader
  - All MGS nodes re-process the Nodemap request to update its status

- Nodemap updates conflict between local MGS and MGC
  - Nodemap updates could be from local MGS (Raft) and the MGC (from the starter MGS)
  - In “mgs_raft_ioctl”, the Nodemap request will be skipped if there is local replicated MGS

© DDN 2024

FTM: Imperative Recovery

- Imperative Recovery
  - Clients use it to speed up the Recovery
  - No configuration files needed to save in storage, the Target Status Table will be in memory

- Synchronization of IR update
  - Target still notify the MGS0 for IR update in the old way
  - Using MGS iocontrol to pack the IR request and to forward it to other nodes by Raft
  - Replicated MGS process the IR request received from its Raft module
  - The Clients will use the IR notification from its connected MGS

© DDN 2024

Future Plans

- FTM Phase 1: has completed and will push to community version soon
- FTM Phase 2: Start MGS dynamically, Target use replicated MGS to mount, ...
- Deploy other services on Raft: FLDB, Quota
- Lustre Metadata Redundancy (LMR)

© DDN 2024

Q&A

© DDN 2024


ddn