DEVELOPERS SUMMIT

MDT multiple slots for reply reconstruction

2014/09/24

Grégoire PICHON

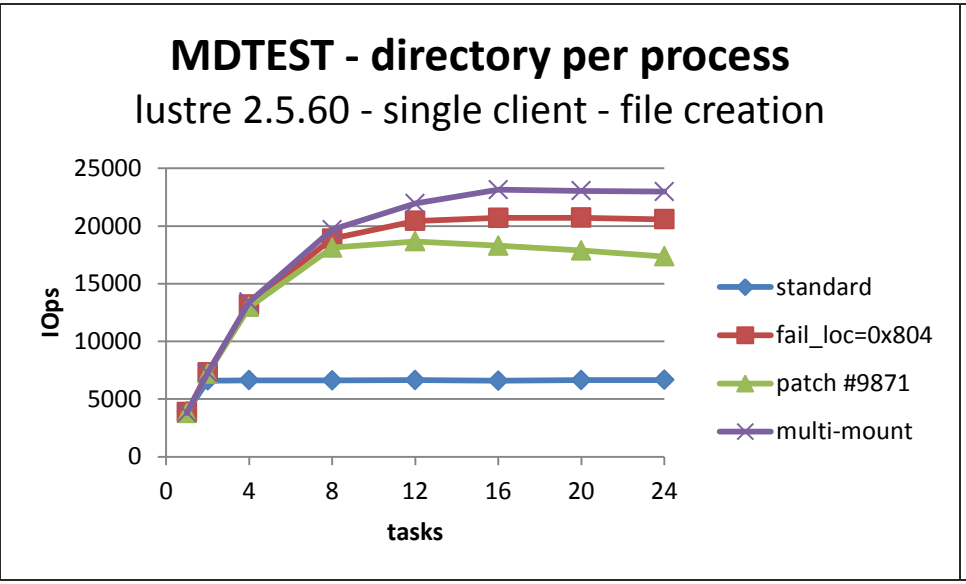
Parallel File Systems

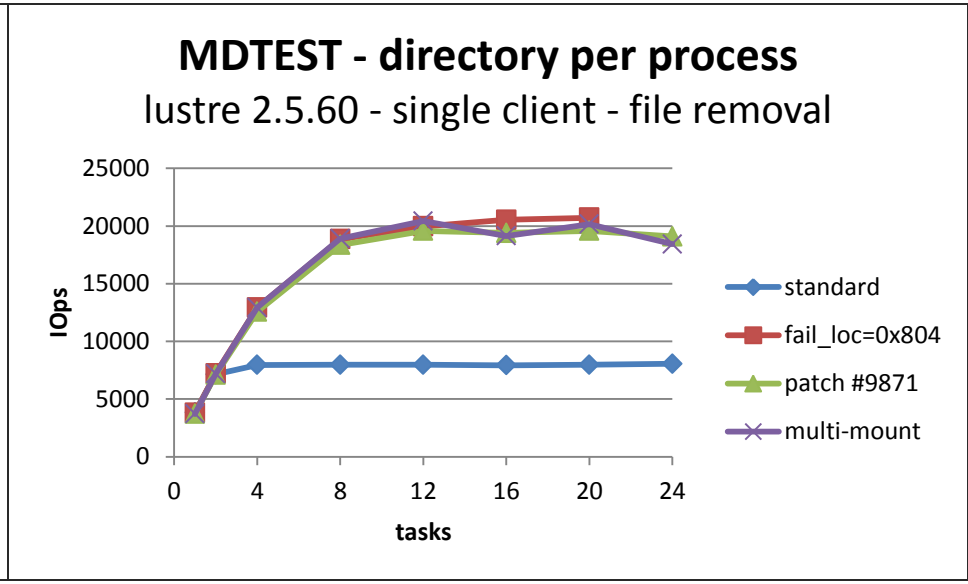
Extreme Computing R&D

Single client metadata performance issue

- MDT is able to handle one modify RPC at a time per client
  - one slot per client in the MDT last_rcvd file
  - slot is used to save the state of the last transaction
  - reply can be reconstructed in case of RPC resend
- MDC requests are serialized
- single client performance of metadata operations is low
  - all modify operations (creation, unlink, setattr, ...)
  - "read" operations are not concerned (stat, lookup, readdir, layout)
- tracked through LU-5319
- experimental patch from Alexey Zhuravlev

Single client metadata performance issue





fail_loc=0x804 : bypass the mdc request serialization

patch #9871 : Alexey Zhuravlev's experimental patch that support multiple MDT slots

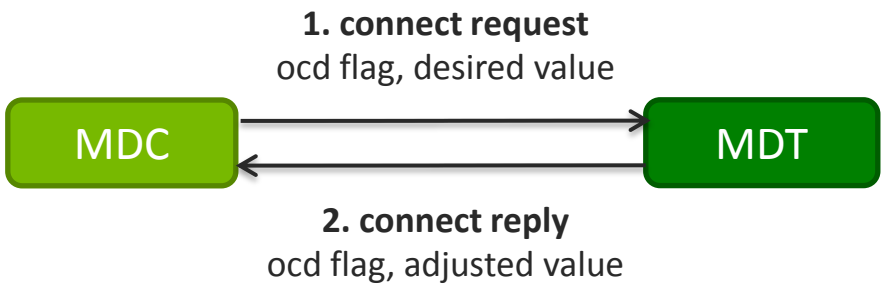
multi-mount : fs is mounted several times on the node

Solution Requirements

- as described in solution architecture document
  - improve single client metadata performance
  - allow MDT to handle several modify metadata requests per client in parallel
  - ensure consistency of MDT operations and reply data on disk
  - client/server full compatibility
    - nodes that support and do not support the feature
  - upgrade and downgrade support
    - for Lustre client
    - upgrade only for MDS

MDT connection

- obd connect data
  - **OBD_CONNECT_MULTIMDRPCS flag**
    indicates support of multiple modify metadata RPCs in parallel
  - **ocd_maxmdrpcs**
    specifies the maximum number of modify metadata RPCs in parallel



Client side

client obd

- cl_max_md_rpcs_in_flight
  - maximum number of modify metadata RPCs in parallel
  - tunable before connection using 'lctl set_param mdc.xxx.max_md_rpcs_in_flight=yyy'
  - cannot exceed cl_max_rpcs_in_flight
  - adjusted during connection phase with MDT
  - default value is 8
- cl_md_rpcs_in_flight

  current number of modify metadata RPCs in flight
- cl_md_rpcs_waitq

  wait queue for threads when max is reached
- cl_md_rpcs_bitmap

  if modify metadata RPCs needs to be tagged, bitmap of tags in use by in flight RPCs

allow 1 more RPC in flight above max for CLOSE request

a modify metadata request handled by the MDT might trigger lock cancellation. This can require a close request to be sent from the same client.

Server side : tunable

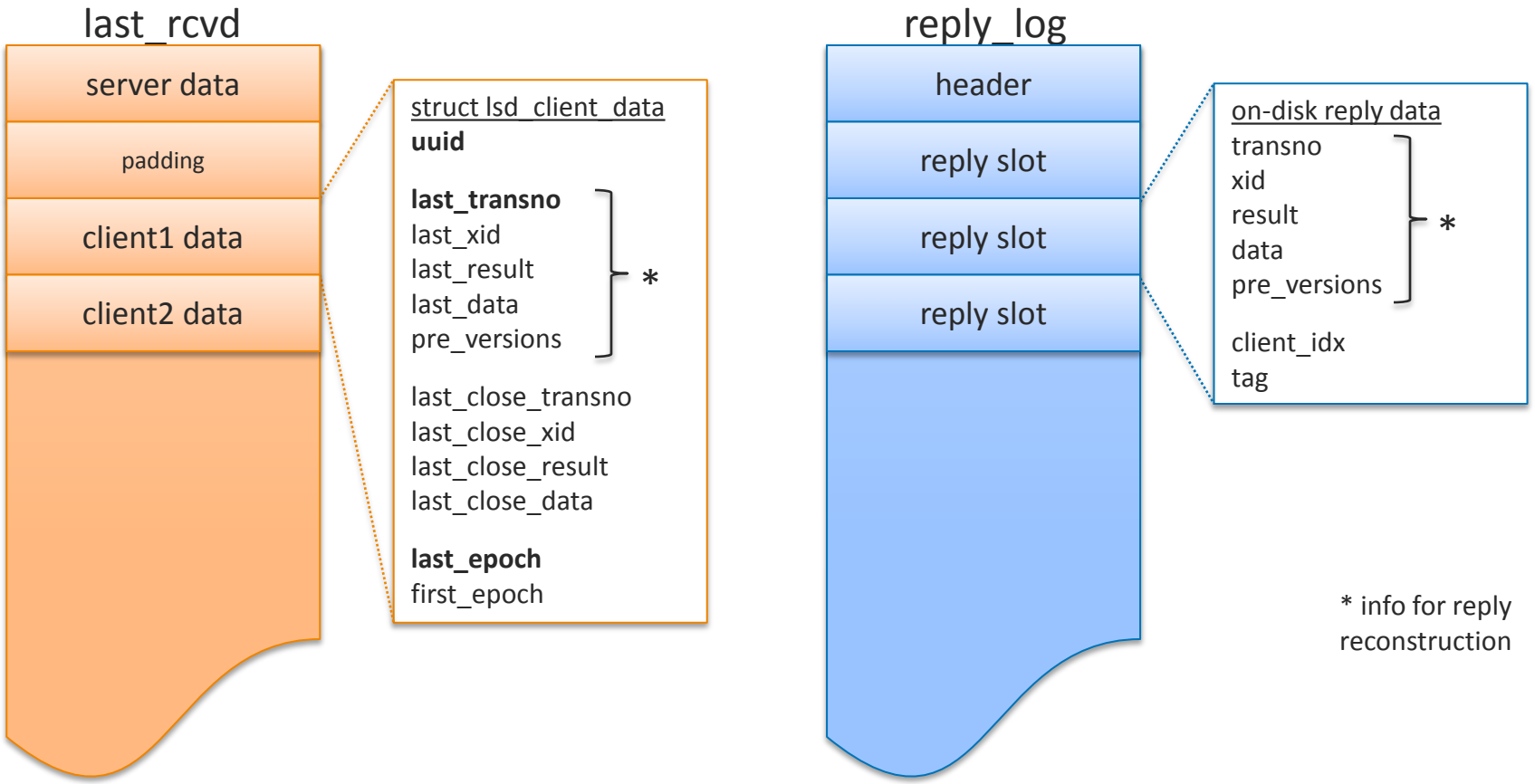
- limit maximum modify metadata RPCs in flight per client
  - to avoid a client to overflow the MDT
  - kernel module parameter: max_md_rpcs_in_flight_per_client

    read-write tunable

    effective for new client connections

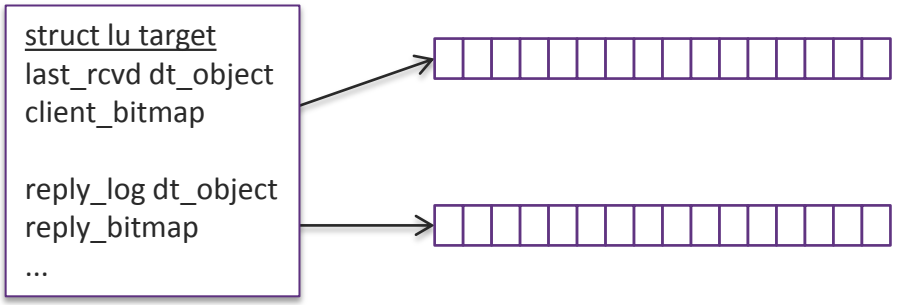
Server side : on-disk data

- reply_log file, in addition to last_rcvd file
  - does accessing continuously the same file at the same place could become a performance issue ?



Server side : in-memory data

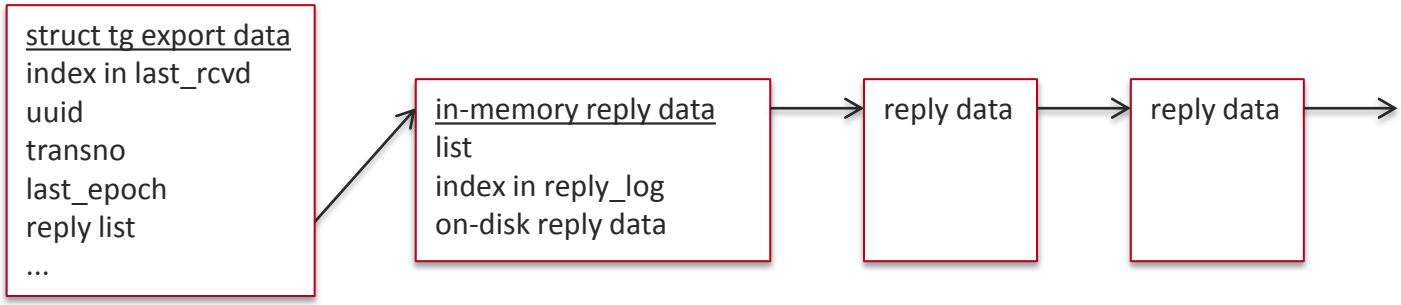
target



bitmap of used client area in last_rcvd file
- one area for each connected client

bitmap of used slots in reply_log file
- extends dynamically
- allocated by group of 1M slots

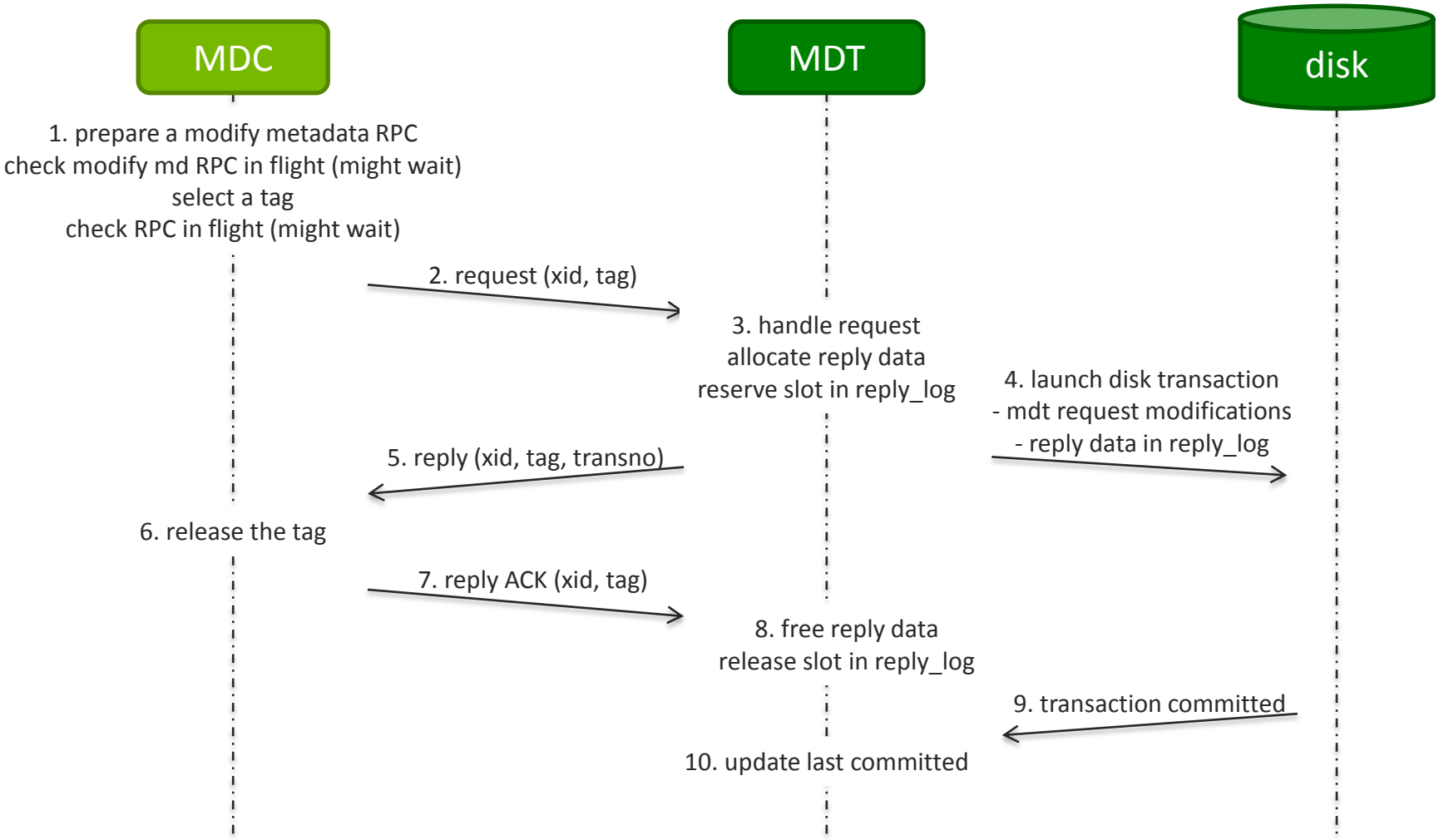
target export



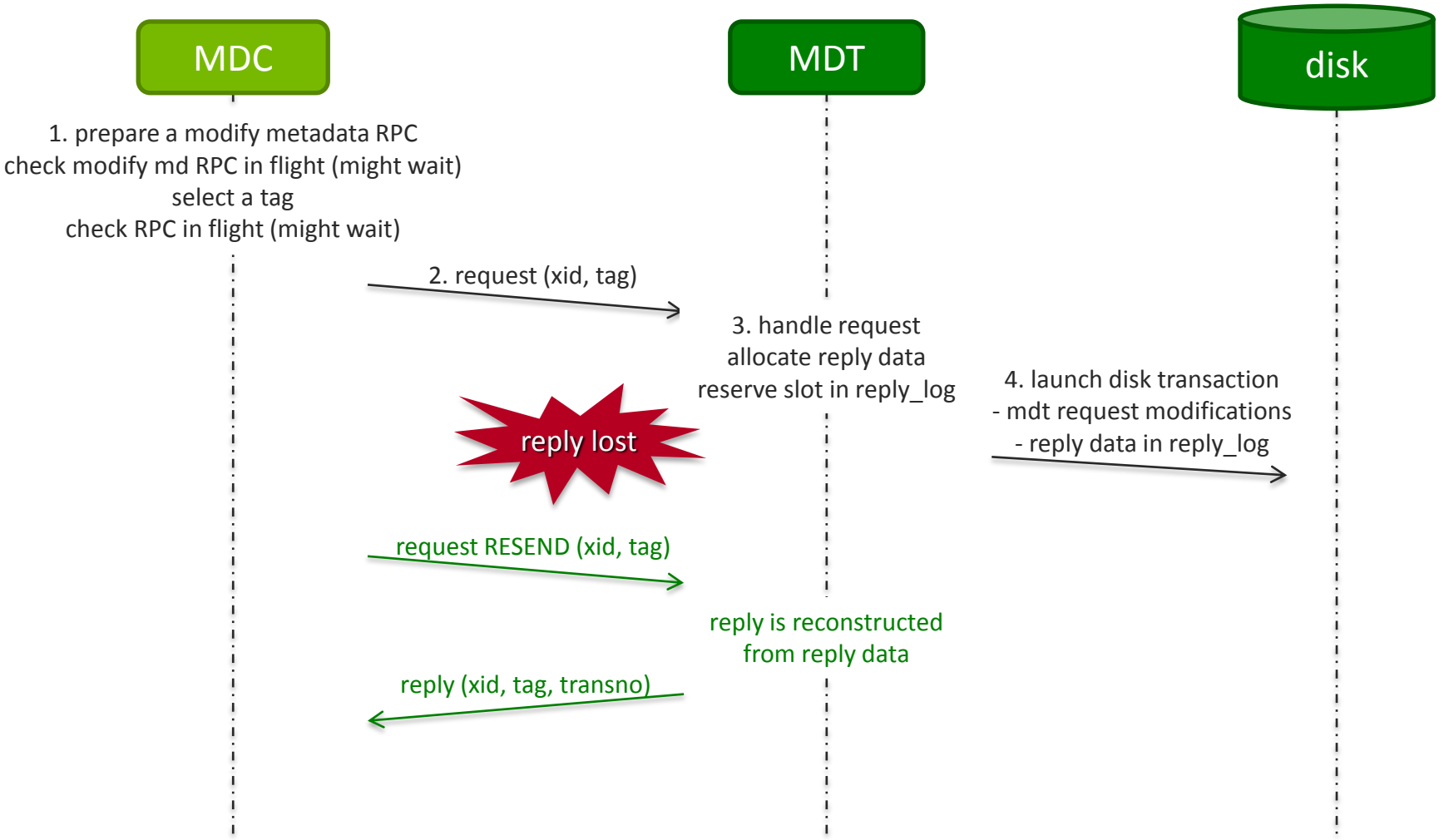
reply data

- allocated when MDT request is handled
- freed when server knows client received the reply
  - reply ACK is received from client
  - metadata RPC tag is reused by the client for another request
  - embed in messages the last xid of reply received by the client

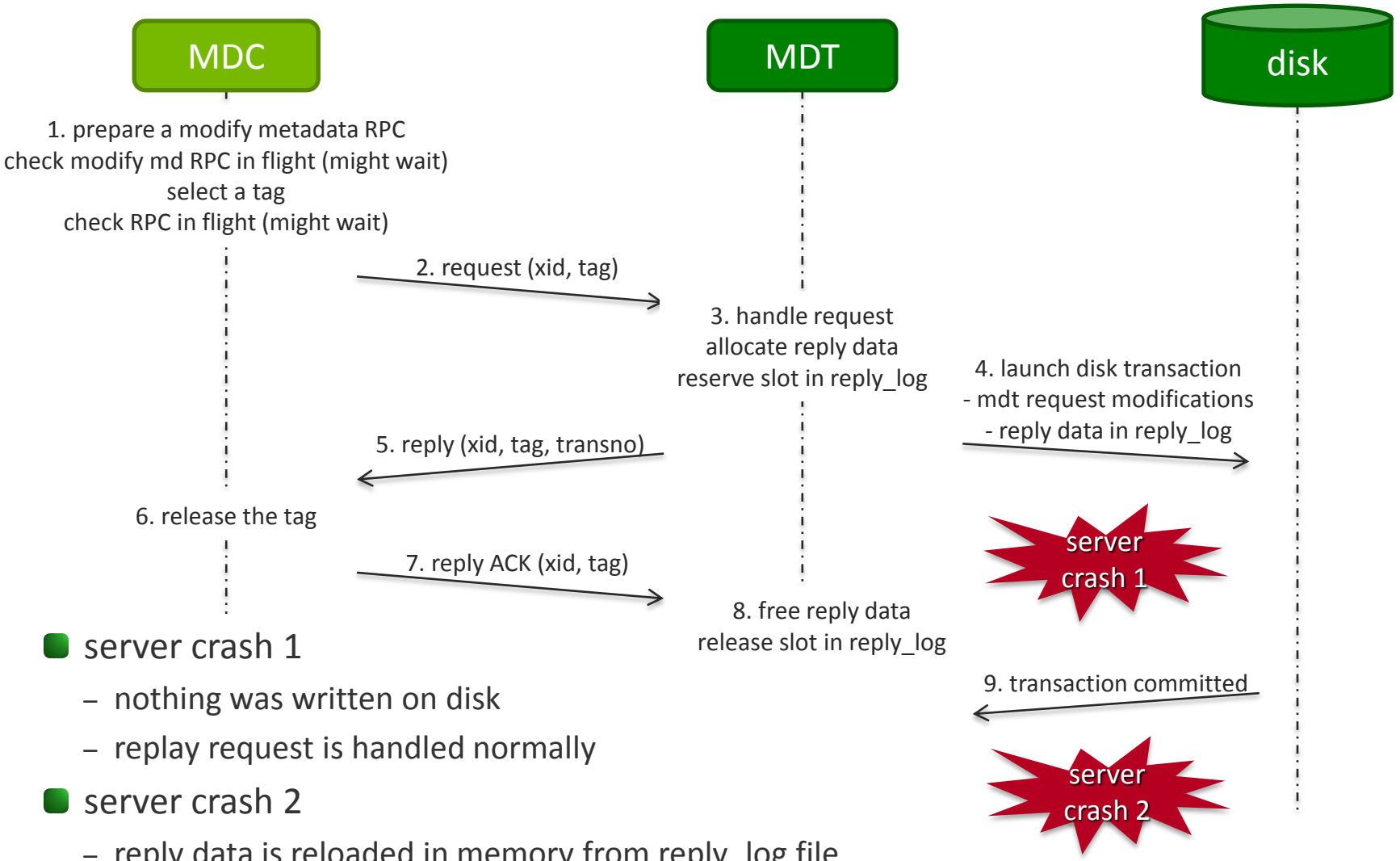
Metadata request flow



Metadata request flow: reply lost



Metadata request flow: server crash



- server crash 1
  - nothing was written on disk
  - replay request is handled normally
- server crash 2
  - reply data is reloaded in memory from reply_log file
  - replay request is handled with reply reconstruction