Lustre 2.8 feature : Multiple metadata modify RPCs in parallel

Grégoire Pichon
BDS R&D Data Management

23-09-2015

Bull
atos technologies

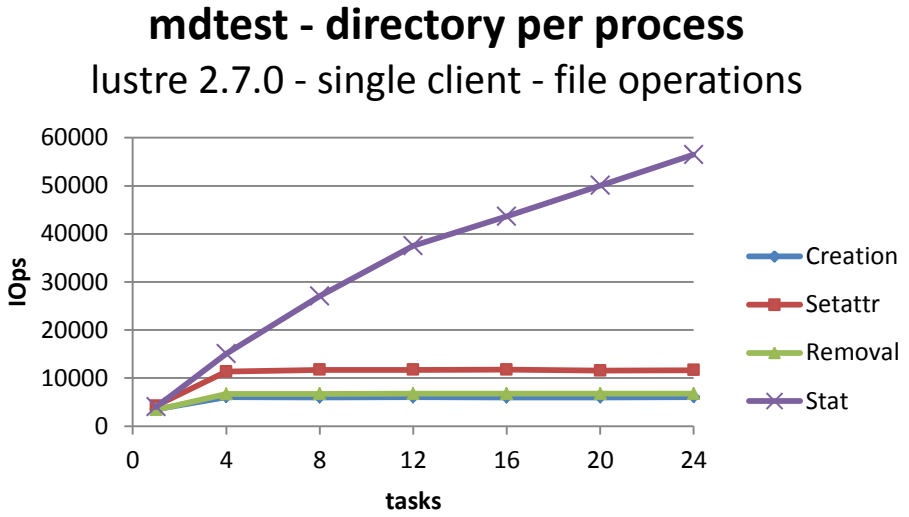
Agenda

- Client metadata performance issue
- Solution description
- Client metadata performance results
- Configuration parameters and runtime statistics
- Feature design and internals
- What next ?

Single Client Metadata Performance Issue

Lustre 2.7 and before

- Client performance does not scale
- MDC modify requests are serialized
  - modify operations
    - creation, unlink, setattr, …



- MDT limitation
  - MDT handles only one modify RPC at a time per client
  - one slot in the LAST_RCVD file per client
  - used to
    - save transaction result
    - reconstruct reply in case request is resent by client

LAST_RCVD file

Lustre 2.7 and before

- Lustre target internal file
  - server data
    - server uuid, target index, mount count, compatibility flags, …
  - per client data
    - client uuid, last transaction, last close transaction
    - xid, transno, operation data, result, pre-versions
- Used at target recovery
  - recreate exports of connected clients at time of crash
  - restore client last transaction
  - compute client highest committed transno

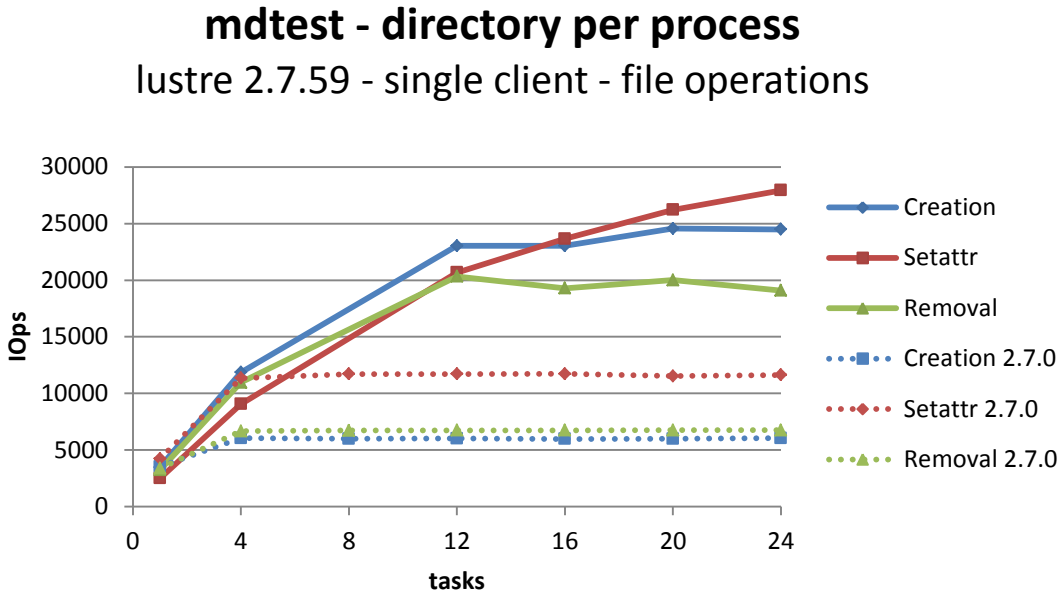
JIRA ticket LU-5319

https://jira.hpdd.intel.com/browse/LU-5319

- Solution goals
  - 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
  - Guarantee client/server full compatibility
  - Support upgrade and downgrade of client and server
- Bull/Atos development with Intel support
  - based on an experimental patch from Alexey Zhuravlev
  - targeted to Lustre 2.8
- Documents available
  - Solution architecture, Design, Test plan
- Funded by CEA

Performance Results – file operations

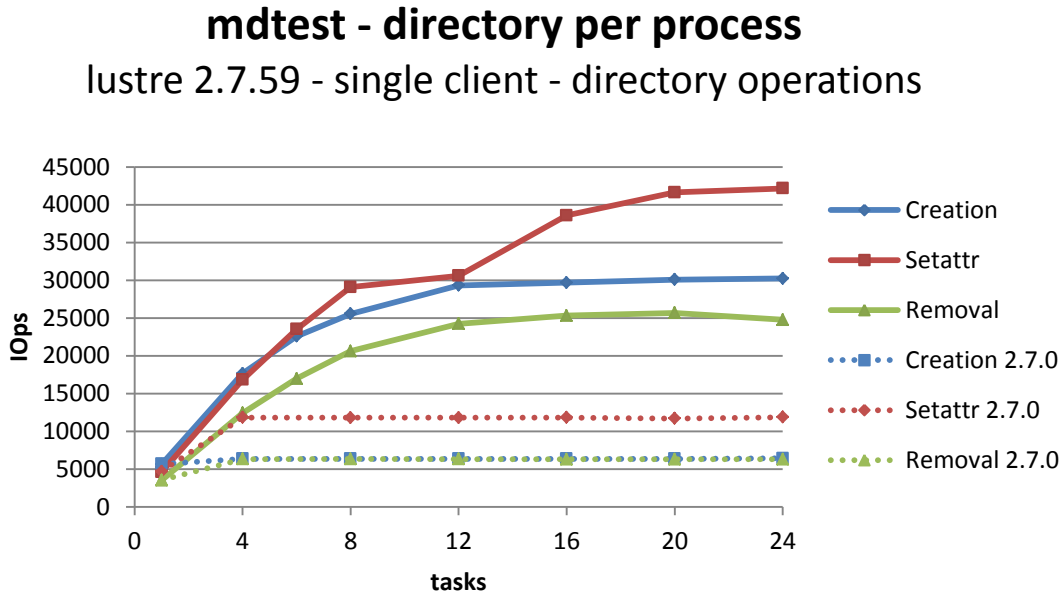
Lustre 2.8



- Client metadata performance significantly improved
  - file creation rate ×4
  - file removal rate ×3
  - file setattr rate ×2.5
- mdtest benchmark has been patched to measure setattr operations rate

Performance Results – directory operations

Lustre 2.8



- Client metadata performance significantly improved
  - directory creation rate ×5
  - directory removal rate ×4
  - directory setattr rate ×3

Configuration Parameters

for the administrator

- **MDC** max_mod_rpcs_in_flight
  - maximum number of modify RPCs sent in parallel
  - default value is 7
  - strictly less than MDC max_rpcs_in_flight value
  - less or equal to MDT max_mod_rpcs_per_client value

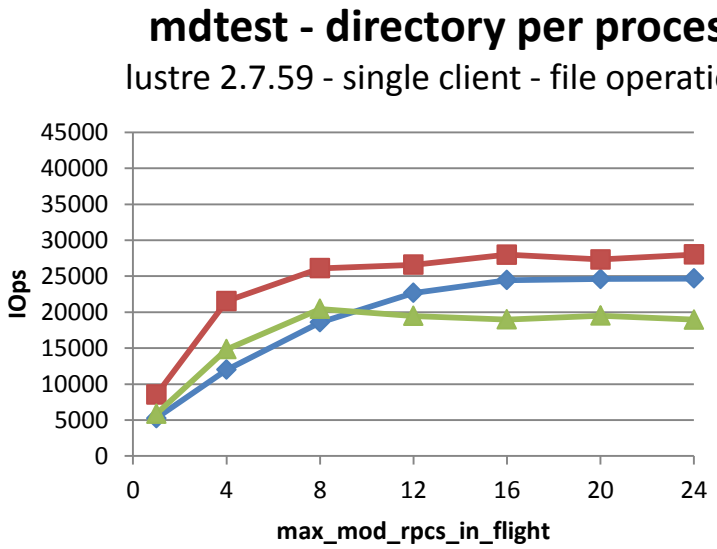
```
# lctl set_param mdc.fsname-MDT*-mdc-*.max_mod_rpcs_in_flight=12
```

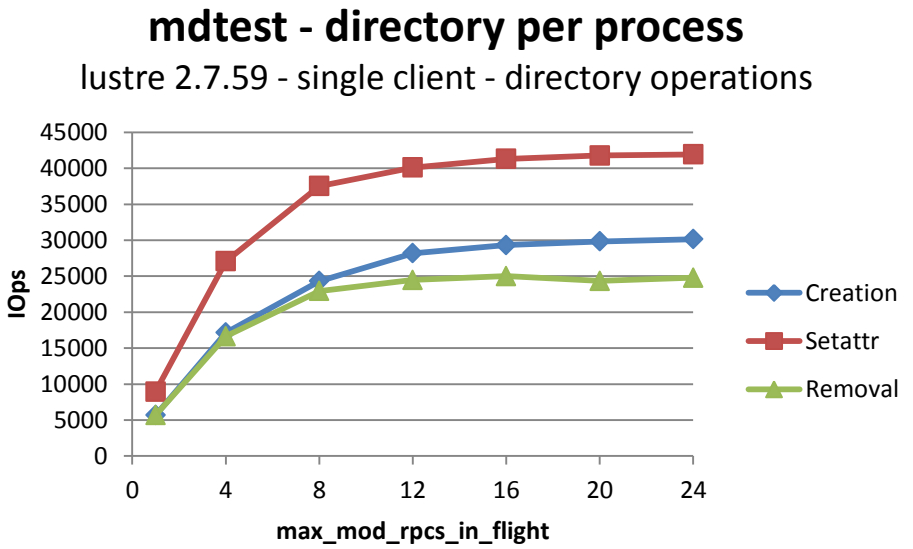
- **MDT** max_mod_rpcs_per_client
  - maximum number of modify RPCs in flight allowed per client
  - effective for new client connections
  - default value is 8

```
# echo 12 > /sys/module/mdt/parameters/max_mod_rpcs_per_client
```

Performance Results

Lustre 2.8





- Performance is no more limited by the number of RPCs in flight
- Limitation on server side

Runtime Statistics

- **MDC** rpc_stats

```
# lctl get_param mdc.fsms-MDT0000-mdc-*.rpc_stats
mdc.fsms-MDT0000-mdc-ffff88077fb3a000.rpc_stats=
snapshot_time:         1441876896.567070 (secs.usecs)
modify_RPCs_in_flight:  0

                        modify
rpcs in flight        rpcs   % cum %
0:                       0   0   0
1:                      56   0   0
2:                      40   0   0
3:                      70   0   0
4                       41   0   0
5:                      51   0   1
6:                      88   0   1
7:                     366   1   2
8:                    1321   5   8
9:                    3624  15  23
10:                   6482  27  50
11:                   7321  30  81
12:                   4540  18 100
```

Debug tool

- **MDT** lr_reader
  - displays content of MDT internal files LAST_RCVD and REPLY_DATA
  - supports only ldiskfs targets
  - YAML format

```
# lr_reader -c /dev/sdh
last_rcvd:
  uuid: fsms-MDT0000_UUID
  feature_compat: 0x8
  feature_incompat: 0x61c
  feature_rocompat: 0x1
  last_transaction: 4294967298
  target_index: 0
  mount_count: 1
  client_area_start: 8192
  client_area_size: 128

  79136f3b-7d85-e265-37aa-dbb40ec5a30c:
    generation: 2
    last_transaction: 0
    last_xid: 0
    last_result: 0
    last_data: 0
```

```
# lr_reader -r /dev/sdh
...
reply_data:
  0:
    client_generation: 2
    last_transaction: 4426736549
    last_xid: 1511845291497772
    last_result: 0
    last_data: 0

  1:
    client_generation: 2
    last_transaction: 4426736566
    last_xid: 1511845291498048
    last_result: 0
    last_data: 0
```

Design points

- Implementation details on :
  - MDT connection
  - Metadata request flow
  - MDC message tag
  - MDT reply data
  - Reply reconstruction
  - MDT recovery

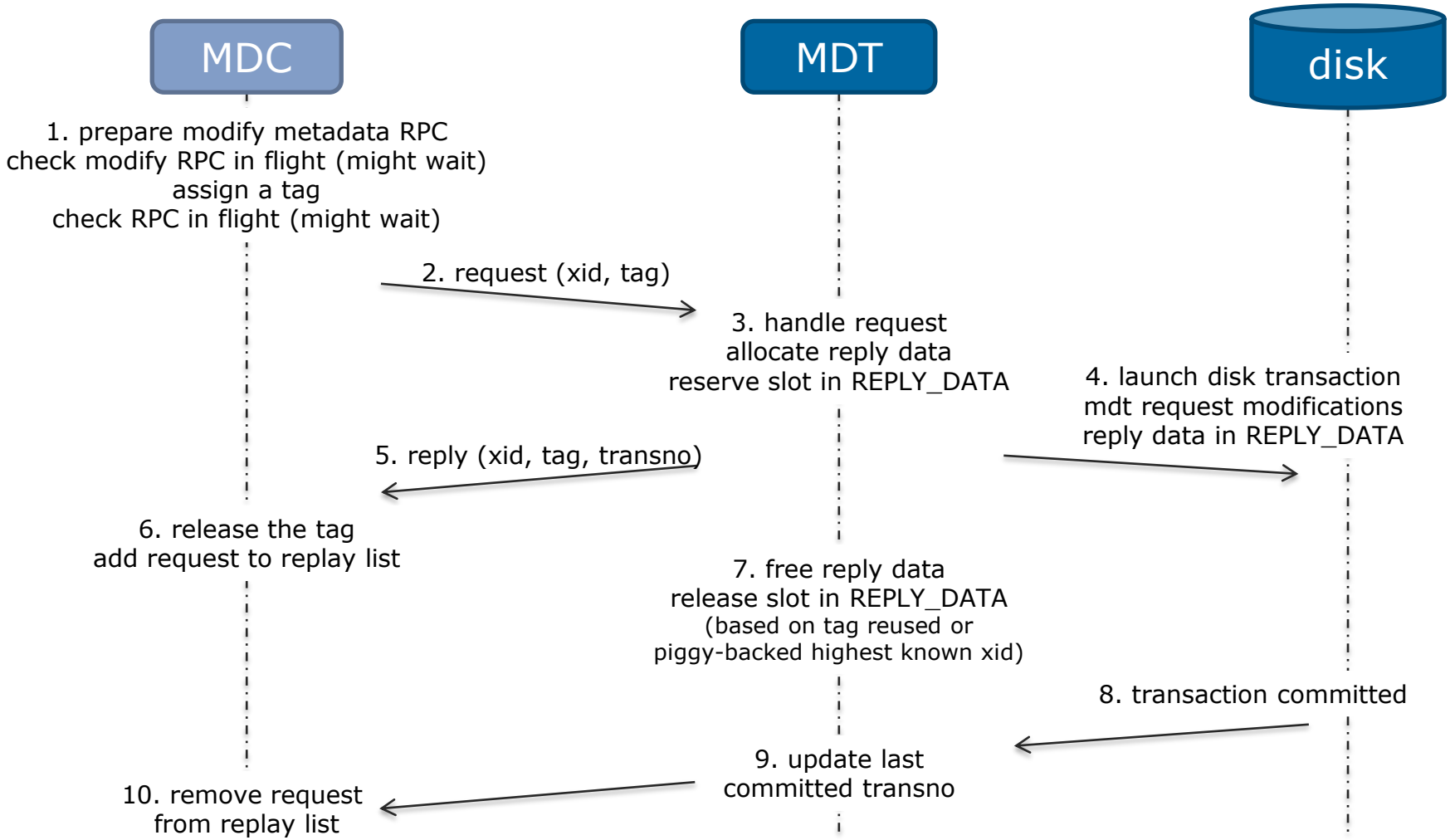
MDT connection

- Additional target connection data
  - OBD_CONNECT_MULTIMODRPCS flag
    - indicates support of the multiple modify metadata RPCs in flight feature
  - ocd_maxmodrpcs field
    - returned by server as the maximum number of modify RPCs allowed per client

1. connect request
OBD_CONNECT_MULTIMODRPCS flag

2. connect reply
OBD_CONNECT_MULTIMODRPCS flag
ocd_maxmodrpcs value

Metadata Request Flow

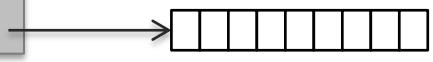


How does client assign message tag ?

- each modify request is assigned a tag
  - value between 1 and max_mod_rpcs_in_flight
  - 0 for non-modify request
- one extra CLOSE request is allowed above max
  - to prevent deadlock in case of lock cancellation
- client maintains a bitmap of in-use tag values

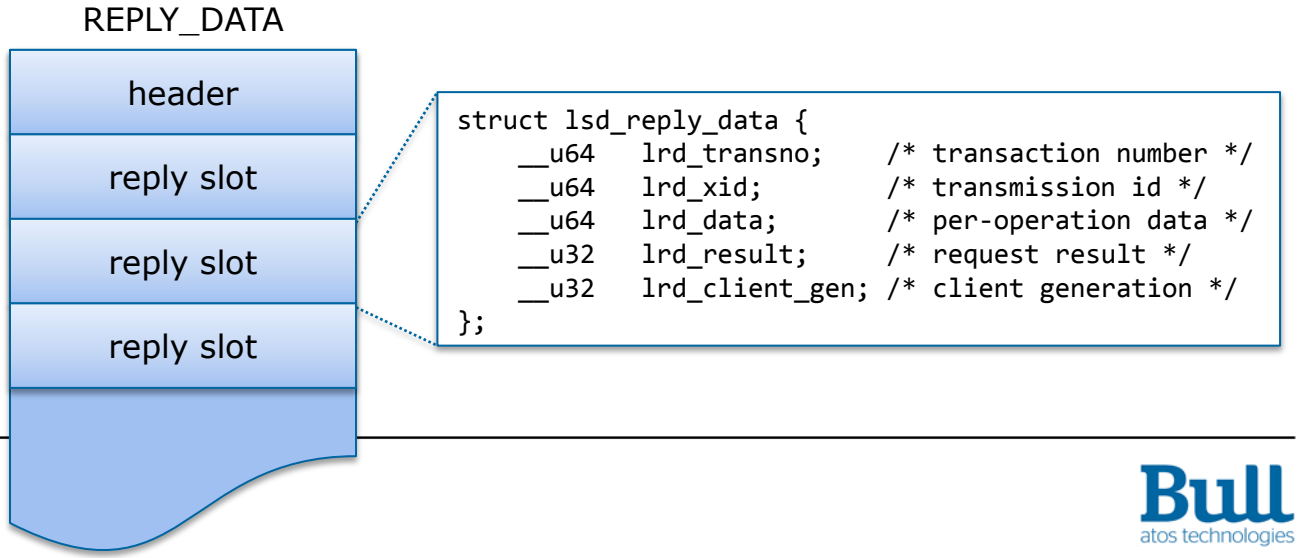
⇒ tag allows server to release reply data

```
struct client_obd
    ...
    spinlock_t           cl_mod_rpcs_lock;
    __u16                cl_max_mod_rpcs_in_flight;
    __u16                cl_mod_rpcs_in_flight;
    __u16                cl_close_rpcs_in_flight;
    wait_queue_head_t    cl_mod_rpcs_waitq;
    unsigned long       *cl_mod_tag_bitmap;
```

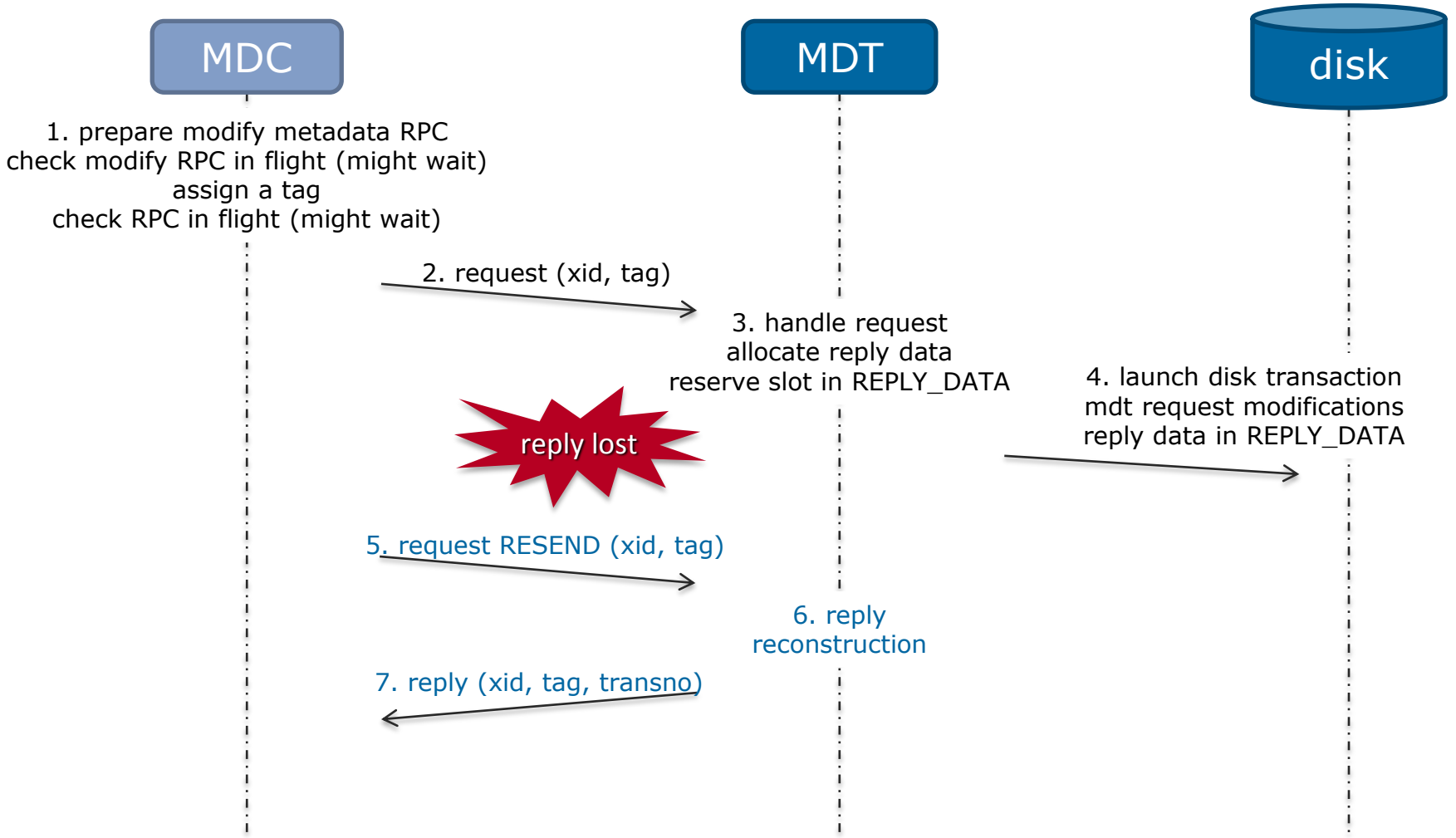


MDT reply data

- Allocated when MDT request is handled
  - list of reply data is anchored in target export
  - target maintains a bitmap of in-use REPLY_DATA slots
- Released as soon as MDT knows client received the reply
  - client embeds in each request the highest known xid
  - client reuses a tag
- The slot with each export's highest transno is never released
- New internal file



Metadata Request Flow: reply lost

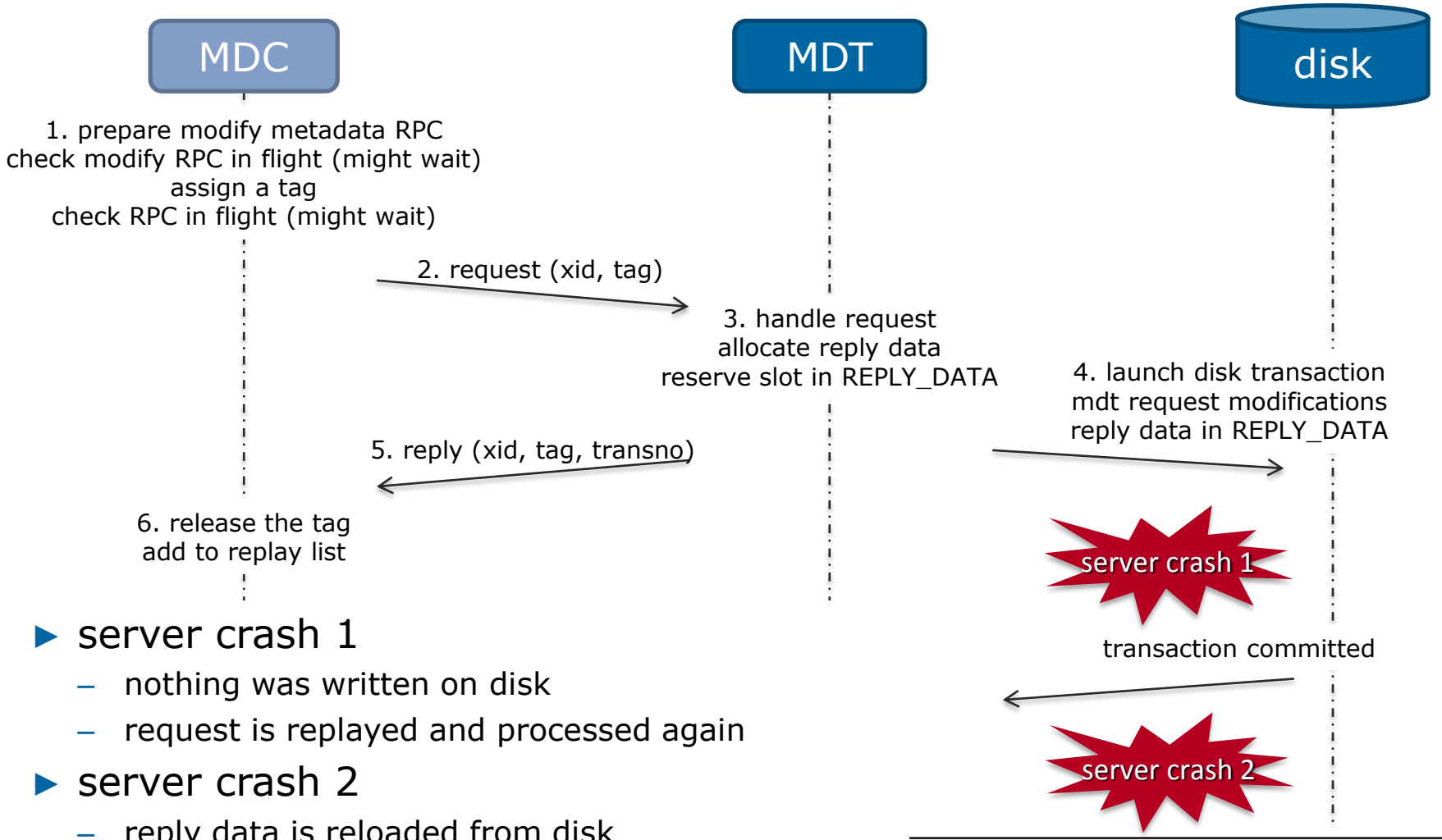


Target recovery

1. from LAST_RCVD file
   - recreate client exports

2. from REPLY_DATA file
   - restore in-memory reply data
   - compute client last committed transno

Metadata Request Flow: server crash

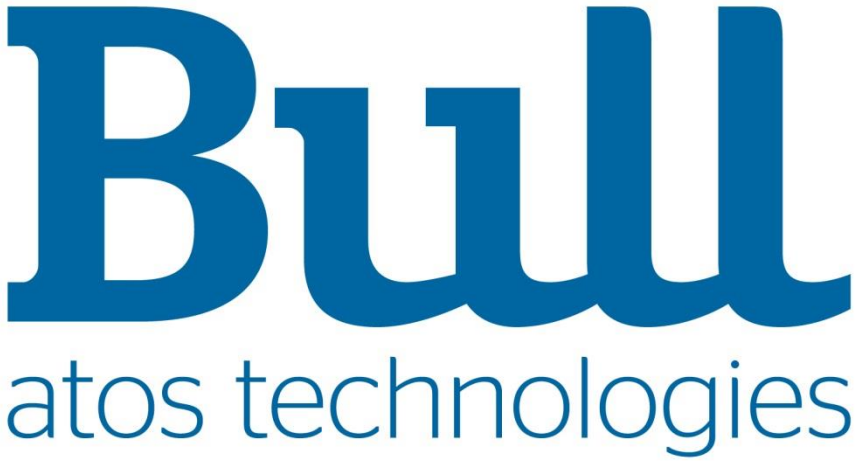


- server crash 1
  - nothing was written on disk
  - request is replayed and processed again
- server crash 2
  - reply data is reloaded from disk
  - request is not replayed

What next ?

- OSP support of multiple modify requests to MDT
  - LU-6864 "Support multiple modify RPCs in flight for MDT-MDT connection"
  - will improve performance of cross-MDT modify requests
    - remote directory creation, unlink


```
lfs mkdir -i1 dir0/dir1a
lfs mkdir -i1 dir0/dir1b
lfs mkdir -i1 dir0/dir1c
lfs mkdir -i1 dir0/dir1d
...
```


Bull
atos technologies

Patch list

Gerrit	Subject	Status
13960	LU-5319 ptlrpc: Add OBD_CONNECT_MULTIMODRPCS flag	Merged
14095	LU-5319 ptlrpc: Add a tag field to ptlrpc messages	Merged
14153	LU-5319 mdc: add max modify RPCs in flight variable	Merged
14374	LU-5319 mdc: manage number of modify RPCs in flight	Merged
14793	LU-5319 ptlrpc: embed highest XID in each request	Merged
14860	LU-5319 mdt: support multiple modify RCPs in parallel	Merged
14861	LU-5319 tests: testcases for multiple modify RPCs feature	Merged
14862	LU-5319 utils: update lr_reader to display additional data	Merged
15576	LU-6840 target: update reply data after update replay	Merged
15971	LU-6981 target: update obd_last_committed	Merged
16045	LU-7028 tgt: initialize spin lock in tgt_init()	Merged
15473	LU-5951 ptlrpc: track unreplied requests	In progress
16215	LU-7082 test: fix synchronization of conf_sanity test_90	In progress
16429	LUDOC-304 tuning: support multiple modify RPCs in parallel	In progress

Test Bed

- MDS
  - 2 sockets / 8 cores Intel Xeon Nehalem X5550
  - 36GiB memory
  - 1 Infiniband FDR adapter
  - MDT ram device 4GiB or 8GiB
- OSS
  - 2 sockets / 8 cores Intel Xeon Nehalem X5560
  - 36GiB memory
  - 1 Infiniband FDR adapter
  - 2 OSTs ram device 4GiB
- Client
  - 2 sockets / 24 cores Intel Xeon IvyBridge E5-2697v2
  - 64GiB memory
  - 1 Infiniband FDR adapter