



Whamcloud

Lustre quota aggregation (LQA)

LAD 2025

Sergey Cherementsev

scherementsev@ddn.com



Lustre Quotas Overview

▶ **User, group, and project quotas**

- Control space and inode usage on a per-identity basis

▶ **Helps prevent resource overuse**

- Ensures fair sharing in multi-user environments

▶ **Backend FS: Idiskfs/ZFS**

- Tracking/storage of inode/block usage per user/group/project ID

▶ **Quota master (QMT)**

- Aggregates usage and allocates remaining space to OST/MDT

▶ **Limits enforced at the OST and MDT level**

- Accurate space and inode tracking across all targets

Lustre quotas history

► 2007 (Lustre 1.4)

- Basic user/group quotas added, based on **ext4/ldiskfs** on-disk quotas

► 2012 (Lustre 2.3)

- Integration with **ZFS** on-disk quotas

► 2017 (Lustre 2.10)

- **Project quota** (aka "directory"), for research groups and multi-tenant systems

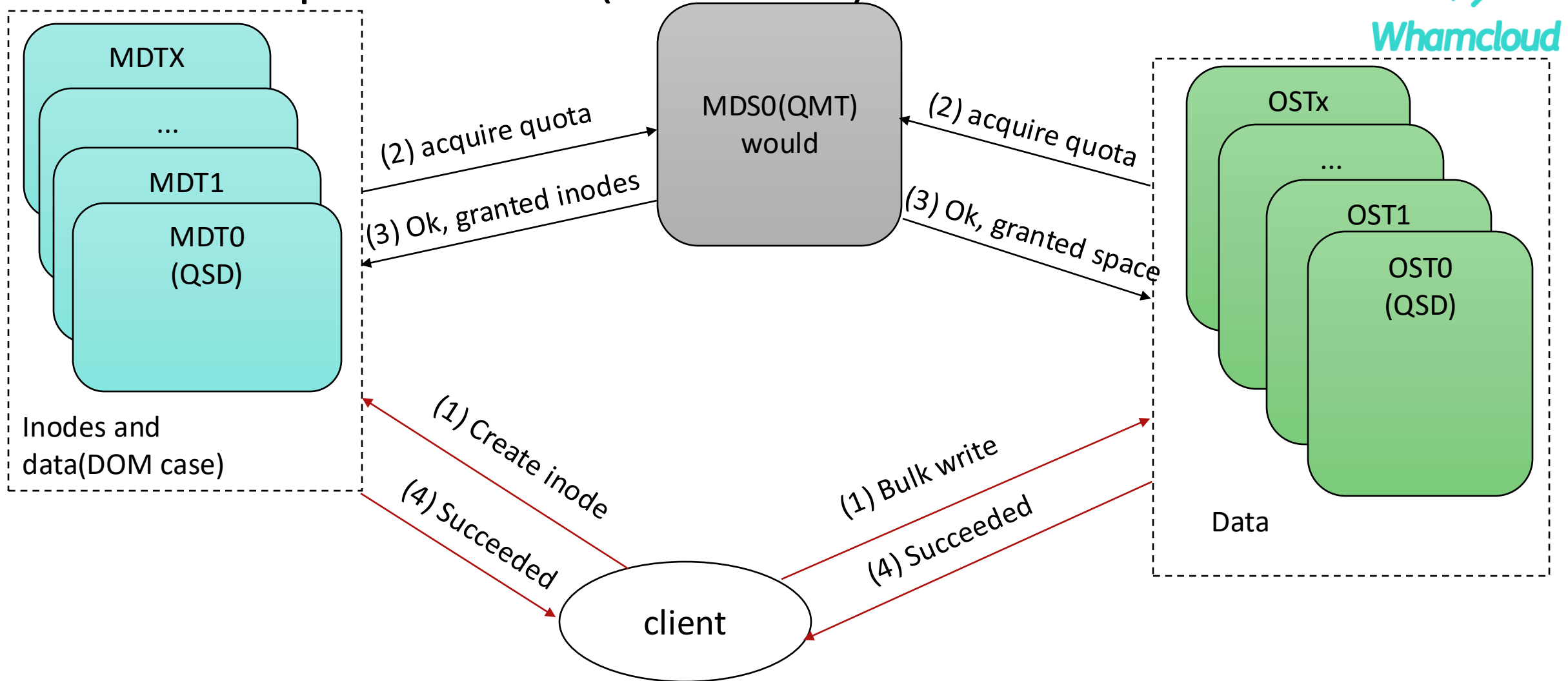
► 2019 (Lustre 2.14)

- Introduced **OST Pool Quotas**

► 2026 (Lustre 2.18)

- Introduces **Quota Aggregation**

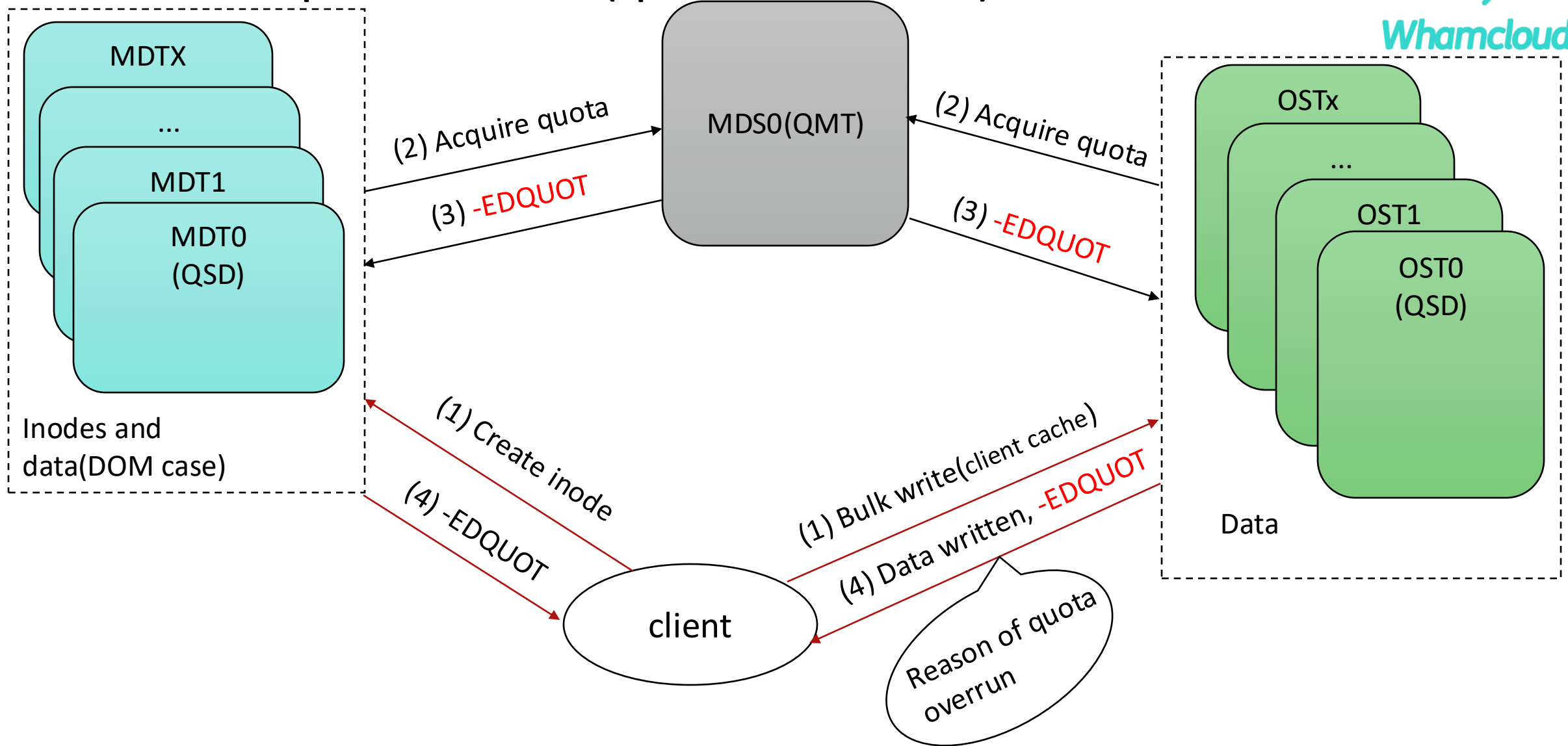
How do quotas work? (succeeded)



How do quotas work? (quota exceeded)



Whamcloud



Lustre Quota Aggregation or LQA

- ▶ **LQA** enables aggregating and limiting usage across multiple quota IDs
- ▶ Many distinct LQAs can be created in a single filesystem
- ▶ LQA consists of one or more non-overlapped ID ranges
- ▶ Quota limits are set per LQA for each quota type (usr/grp/prj)
- ▶ Managed on the QMT

Why LQA?



- ▶ Users have requested the ability to aggregate a range of quota IDs for a tenant into a single aggregate limit
- ▶ Must allow the tenant to manage the limits of individual quota IDs

Required Changes

- ▶ Server side needed most significant design and development effort:
 - QMT - main changes
 - QSD - no changes
 - Client - slightly changed
- ▶ Userspace needed some small modifications:
 - `lfs setquota --lqa, lfs quota --lqa`
 - `lctl lqa new/add/remove/destroy`

LQA management

► `lctl lqa` commands must be run on the MDS mounting MDT0000

```
# create a new LQA in testfs named tenant1
mds0# lctl lqa new --fsname testfs --name tenant1
# the same, but automatically find fsname if not given
mds0# lctl lqa new --name tenant1

# add some quota ID ranges to the tenant1 LQA
mds0# lctl lqa add --name tenant1 --range 1000-1999
mds0# lctl lqa add --name tenant1 --range 3000-3499

# List the configured LQAs, and details about the tenant1 LQA
mds0# lctl lqa list
name: tenant1, tenant2, tenant3
mds0# lctl lqa list --name tenant1
name: tenant1
ranges: 1000-1999, 3000-3499

# remove a quota ID range from the tenant1 LQA, or remove it completely
mds0# lctl lqa remove --name tenant1 --range 1000-1999
mds0# lctl lqa destroy --name tenant1
```

LQA Quota Operations

```
# set quota limits for newly-created LQA tenant1
client# lfs setquota --lqa tenant1 -U -B100M -b50M -I100K -i20K /mnt/testfs
client# lfs setquota --lqa tenant1 -G -B100M -b50M -I100K -i20K /mnt/testfs
client# lfs setquota --lqa tenant1 -P -B100M -b50M -I100K -i20K /mnt/testfs

# show user quota limits for LQA tenant1
client# lfs quota --lqa tenant1 -U /mnt/testfs
Disk quotas for lqa:tenant1:
    Filesystem    kbytes    bquota    blimit bgrace files  iquota  ilimit  igrace
/mnt/testfs      40960         0  102400      -   300       0    2000      -
```

Technical details

- ▶ LQA is a pool with a red-black tree
- ▶ LQA pool is handled along with the other pools
- ▶ LQA has only 2 LQEs (lustre quota entries) for each quota type:
 - ID:0 - to store the grace timeout
 - ID:1 – to store the limits and granted space

LQA and Pool Quotas (PQ)

- ▶ LQA supports both data and metadata -> PQ account only data today
- ▶ LQA accounts the data (inodes) for an aggregation of QIDs -> PQ for a range of OSTs per QID
- ▶ PQs can include only existing OSTs
- ▶ It is not required that all QIDs from LQA's ID range present in the system

LQA and nodemaps



Whamcloud

- LQA used together with nodemap to aggregate project limits for a tenant

```
# create a new tenant1 nodemap for an ID offset range
mgs# lctl nodemap_add tenant1
mgs# lctl nodemap_add_range tenant1 192.168.56.[100-200]@tcp
mgs# lctl nodemap_add_offset --name tenant1 --offset 200000 --limit 100000
mgs# lctl nodemap_activate 1

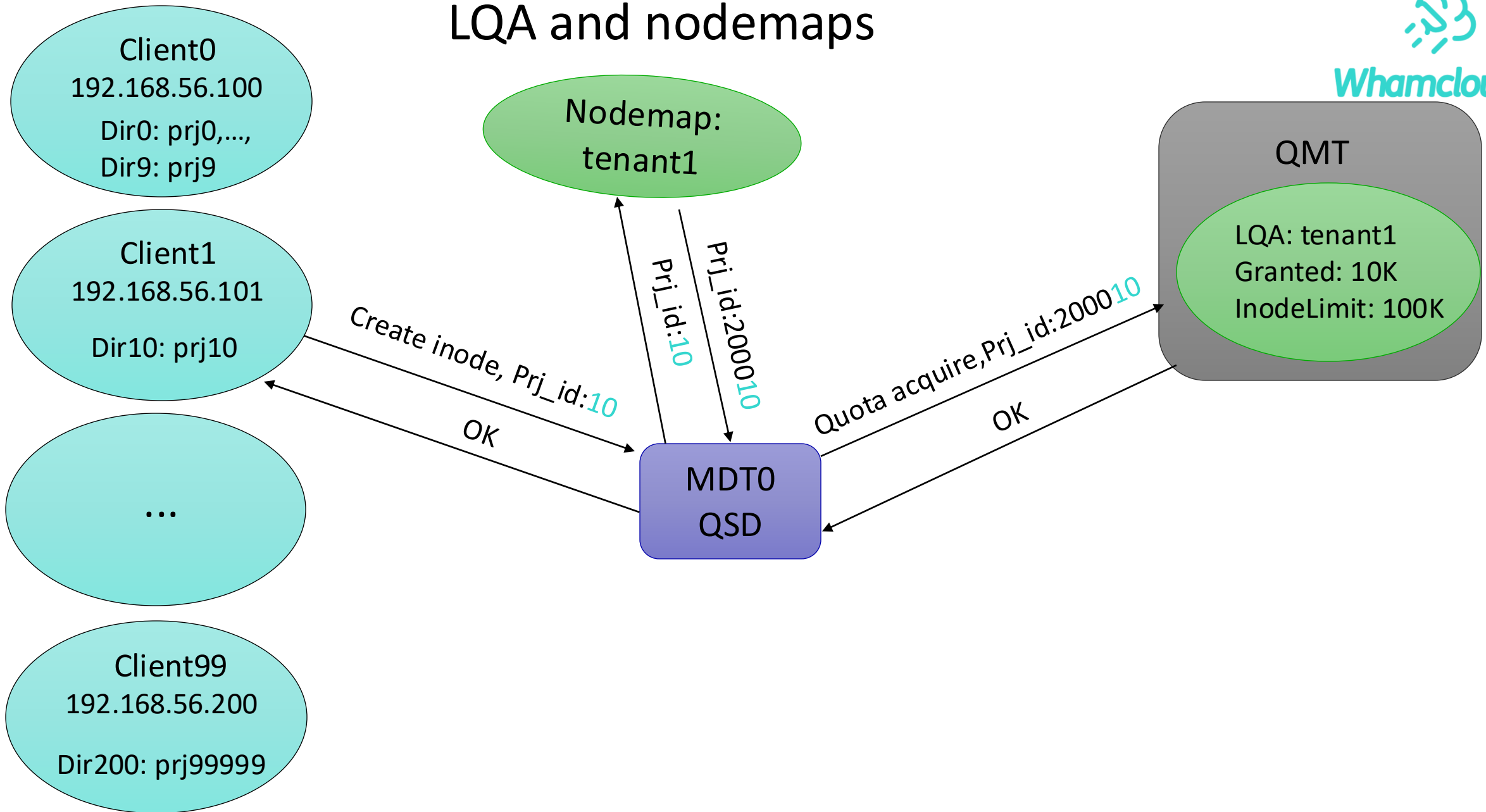
# create a new LQA with the same tenant1 name for the quota ID range
mds0# lctl lqa new --lqa tenant1
mds0# lctl lqa add --lqa tenant1 --range 200000-299999

# set quota limits to the LQA tenant1
client# lfs setquota -P --lqa tenant1 -B500G -b450G -I100K -i80K
/mnt/testfs
client# lfs project /mnt/testfs | tr '\n' ','
0 P /mnt/testfs/project0, ..., 99999 P /mnt/testfs/project99999
```



Whamcloud

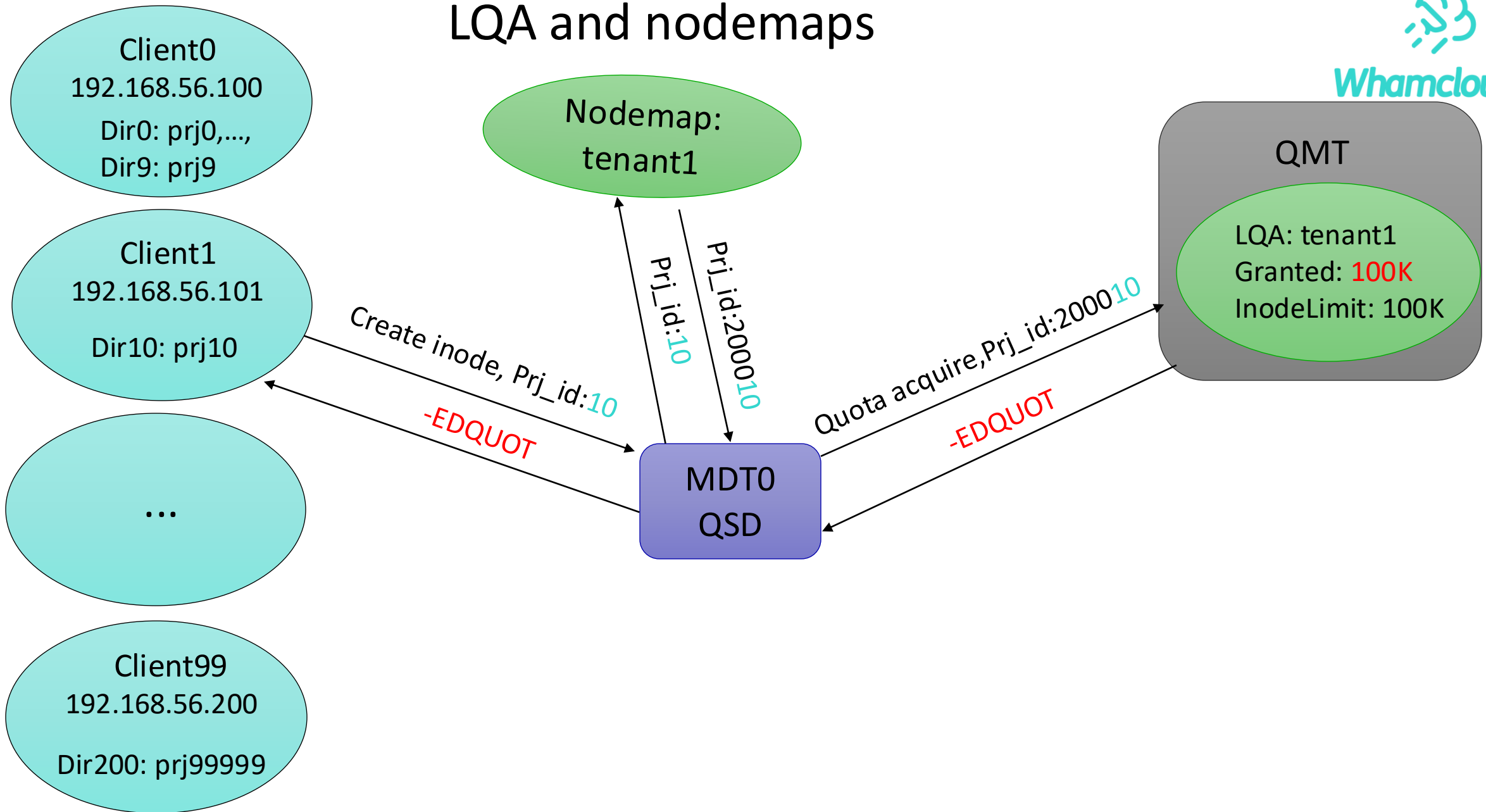
LQA and nodemaps





Whamcloud

LQA and nodemaps



LQA and nodemaps



Whamcloud

- Show LQA limit if smaller than FS size and dir1 project ID limit

```
client# lfs quota --lqa tenant1 -P -h /mnt/testfs
```

Disk quotas for lqa:tenant1:

Filesystem	kbytes	bquota	blimit	bgrace	files	iquota	ilimit	igrace
/mnt/testfs	45G	450G	500G	-	300	80K	100K	-

*# show LQA limit if smaller than FS size and **dir1** project quota ID limit*

```
client# df -h /mnt/testfs/
```

Filesystem	Size	Used	Avail	Use%	Mounted on
192.168.56.203@tcp:/lustre	450G	45G	405G	10%	/mnt/testfs

Resources:

► <https://jira.whamcloud.com/browse/LU-18222>

► Patches:

- lctl LQA commands <https://review.whamcloud.com/60247>
- LQA to quota pool <https://review.whamcloud.com/61351>
- LQA lfs setquota/quota cmds <https://review.whamcloud.com/61284>
- LQA directory-based management <https://review.whamcloud.com/61239>



Whamcloud

Thank you!

Sergey Cherementsev
scherementsev@ddn.com

