Lustre/ldiskfs Metadata Performance Boost

DataDirect Networks Japan, Inc.

Shuichi Ihara

Shilong Wang

2017/10/05

Why is metadata performance important?

- **Lustre is general purpose filesystem for Big data**
  - 1 Million files per job are quite common with life science application
  - AI/Machine learning type of workload requires small file access with low latency. Metadata performance is one of key factors of it.
  - Lustre metadata performance has been performing well.
- **Vertical and Horizontal scale**
  - 28 (and 32) CPU cores/socket is available Today.
  - DNE helps Horizontal scale out Metadata, but needs to understand your single MDS metadata performance first.

Challenges on Lustre metadata performance

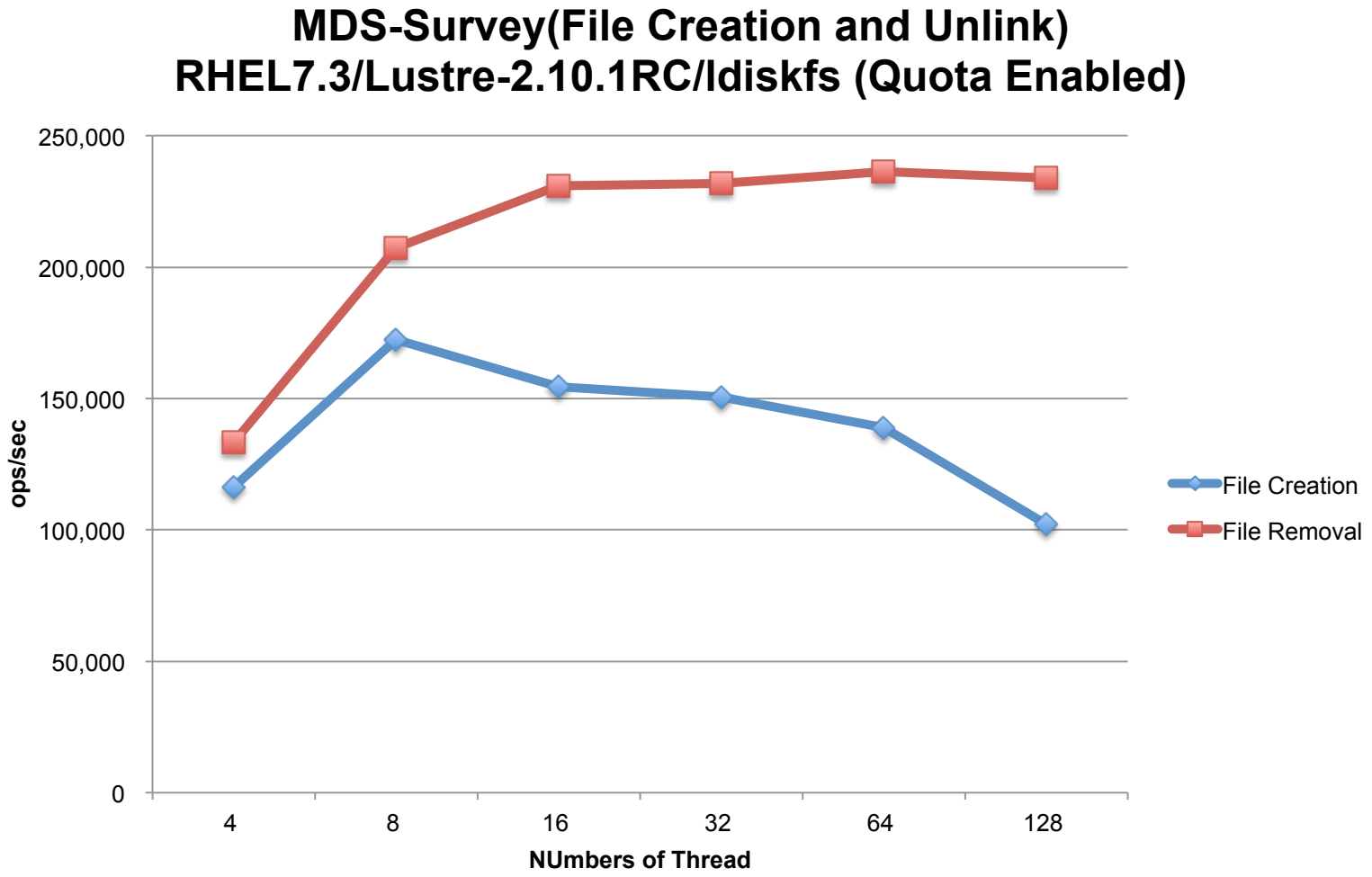
- **Lustre metadata is complex and very sensitive**
  - Latency on many layers (CPU, network, disks, kernel..) affects Lustre metadata performance.
  - Deep analysis and Maximizing single Metadata server performance is important for efficient MDS configuration.
- **Consistent Metadata benchmark is not simple**
  - There are many dependencies between each lustre version. (Kernel support changes, OFED, Hardware configuration, etc)
  - mdtest is common benchmark tool. MDS-survey is useful and good start, but it's limited layer to test. Eventually, End-to-End requires to understand full Lustre metadata performance.

What's Lustre metadata performance Today? Benchmark Configuration

Forces on single MDS and MDT testing

- **1 x MDS**
  - 1 x CPU Socket
    - Intel Skylake Processer (Platinum 8160, 2 1GHz, 24 CPU cores)
    - 96GB DDR4 Memory
    - FDR Infiniband
  - Lustre-2.10.1
- **1 x MDT**
  - 4 x Toshiba RI SSD(RAID10)
  - DDN SFA7700X
- **4 x OSS and 40 x OST with DDN SFA14KXE**
- **32 x Client**
  - 2 x E5-2650v4
  - 128GB DDR4 Memory
  - FDR Infiniband
  - Lustre-2.10.1

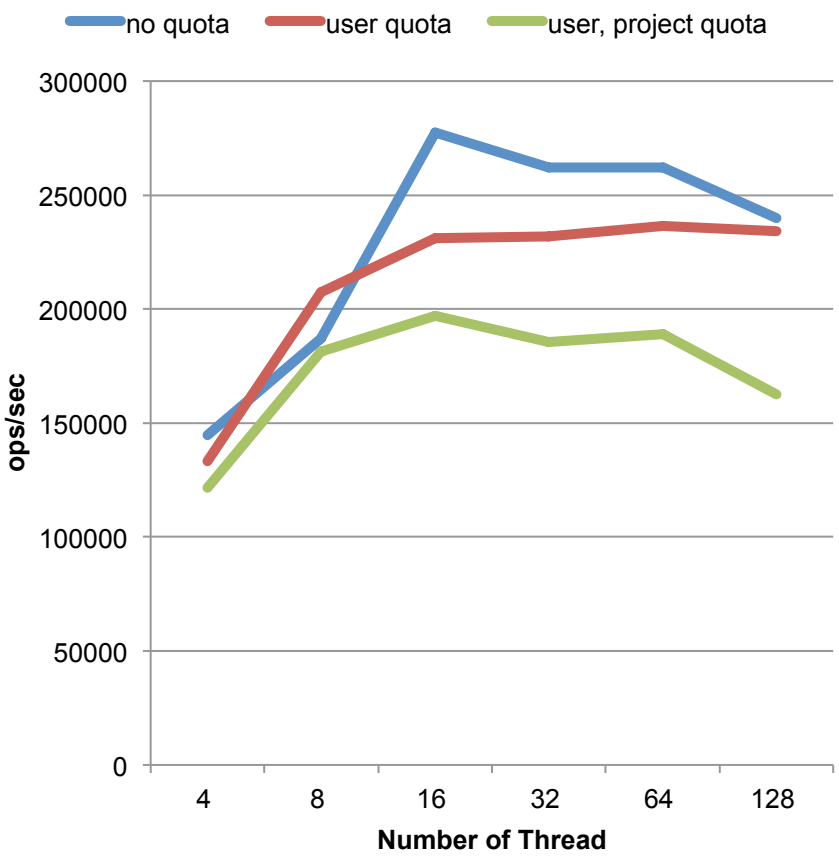
MDS-Survey
RHEL7.3/Lustre-2.10/ldiskfs



Metadata Performance impacts with Enabling Quota(RHEL7.3/Lustre-2.10.1)

mds-survey(creation)

mds-survey(unlink)



New metadata performance limit on Lustre/ldiksfs

Lustre/ldiskfs has been performing metadata rate, but new high-end CPUs expose next level performance limit.

- **File creations under heavy concurrency**
  - Many threads create files to a MDT simultaneously
  - Scalability problem on Many CPU core system
- **Quota scalability**
  - Lustre Quota scalability was hidden by other limitation
  - Will hit quota scalability issue when lustre metadata performance improves
  - New quota accounting (e.g. project quota) introduced additional performance impacts

A problem on File creation under concurrency

- **Profiled with perf-tools during mdtest to ldiskfs/ext4**
  - Collected CPU costs for all functions in ext4 and jbd2
  - Found heavy lock contentions on group spinlock

FUNC	TOTAL_TIME(us)	COUNT	AVG(us)
ext4_create	1707443399	1440000	1185.72
_raw_spin_lock	**1317641501**	**180899929**	**7.28**
jbd2__journal_start	287821030	1453950	197.96
jbd2_journal_get_write_access	33441470	73077185	0.46
ext4_add_nondir	29435963	1440000	20.44
ext4_add_entry	26015166	1440049	18.07
ext4_dx_add_entry	25729337	1432814	17.96
ext4_mark_inode_dirty	12302433	5774407	2.13

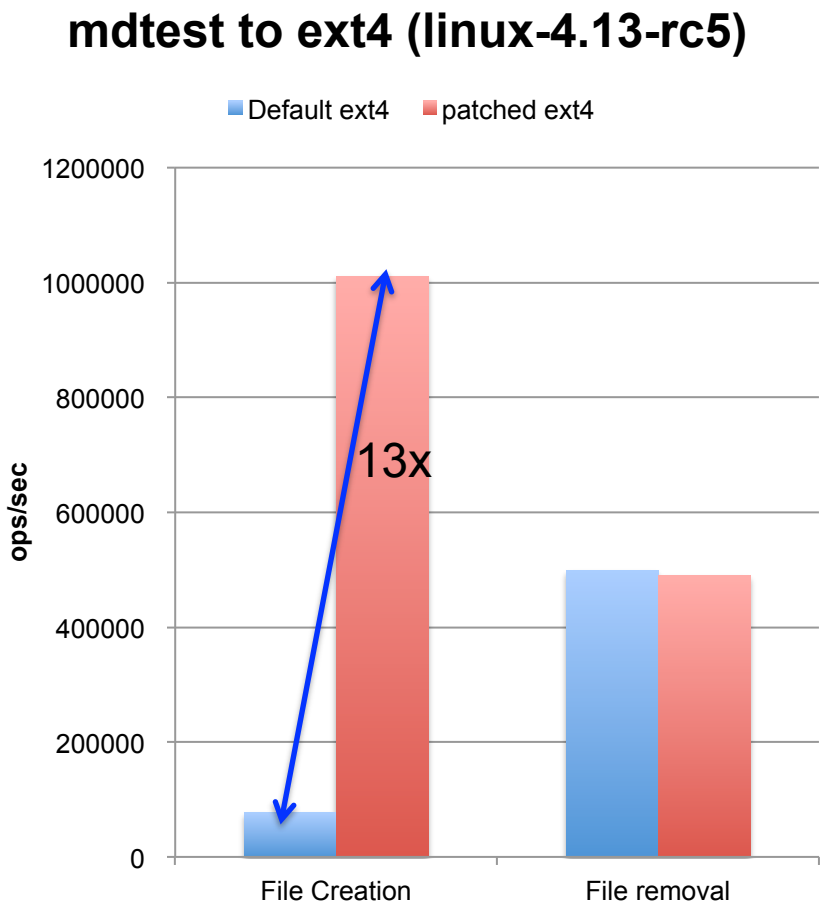
  - Same contentions exist in the upstream kernel

Fix lock contentions in upstream kernel

- **Fixed and merged upstream kernel (4.14)**

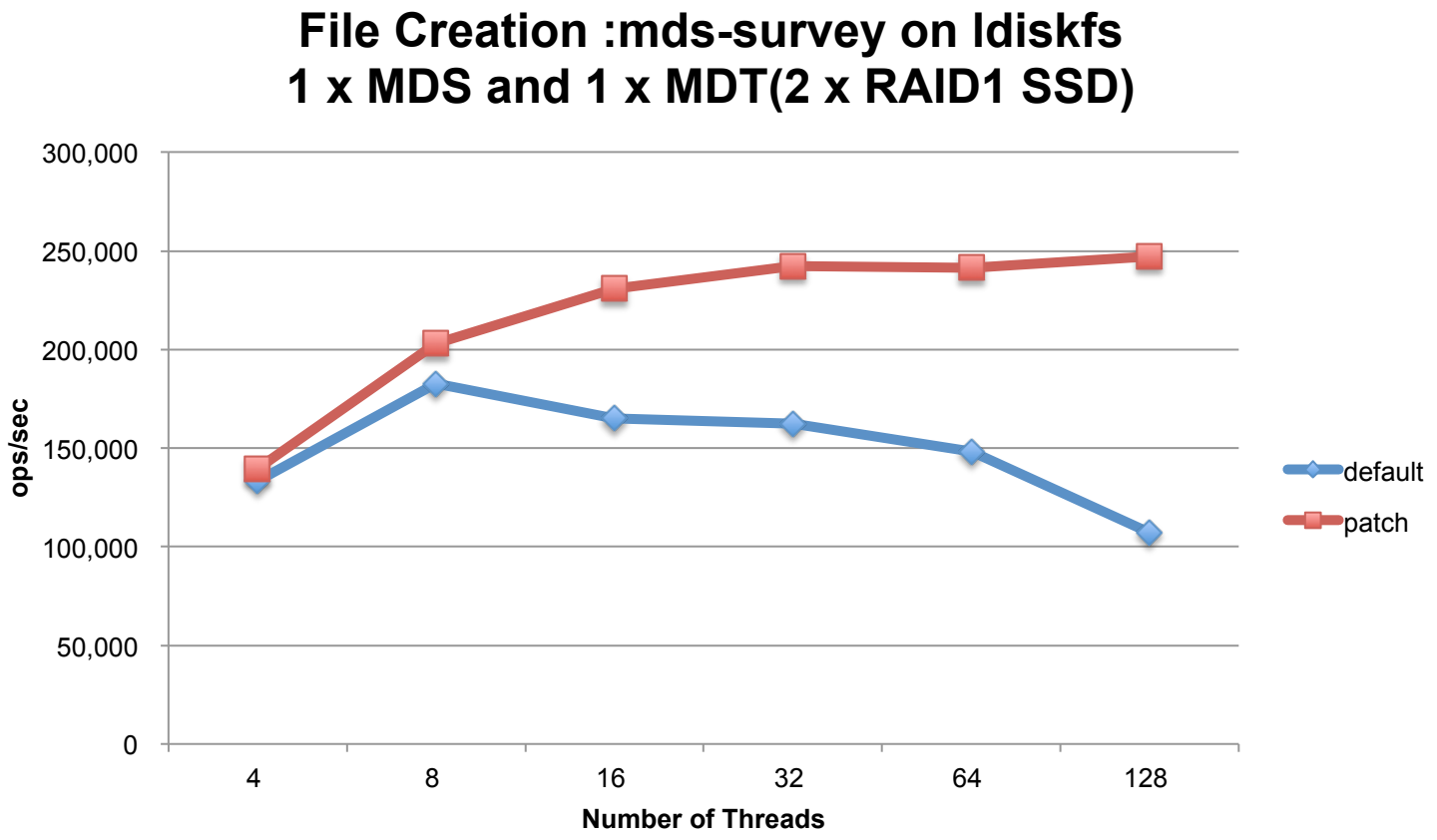
  Wang Shilong (2):
  ext4: cleanup goto next group
  ext4: reduce lock contention in __ext4_new_inode

- **13x performance improvement on file creation**
  - Run mdtest to ext4 directly
  - Unique directory operations
  - Quota disabled



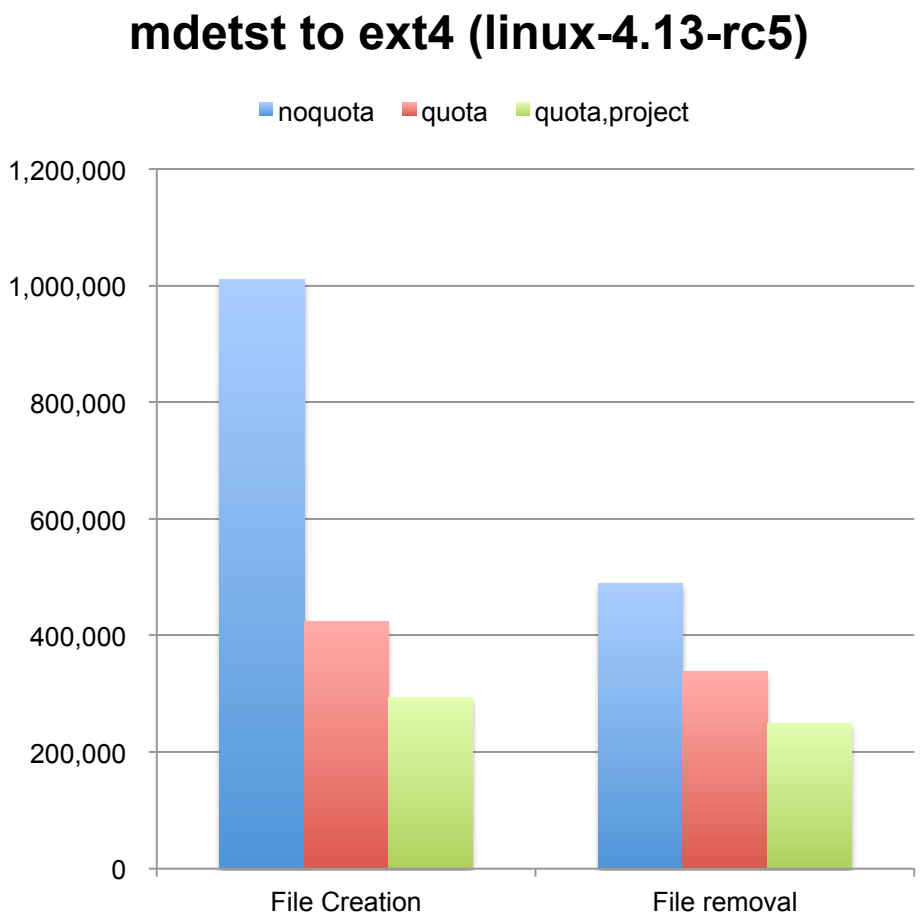
mds-survey on patched ldiskfs

- LU-9796: speedup file creation under heavy concurrency
- Ported patches to ldiskfs for RHEL7 kernel



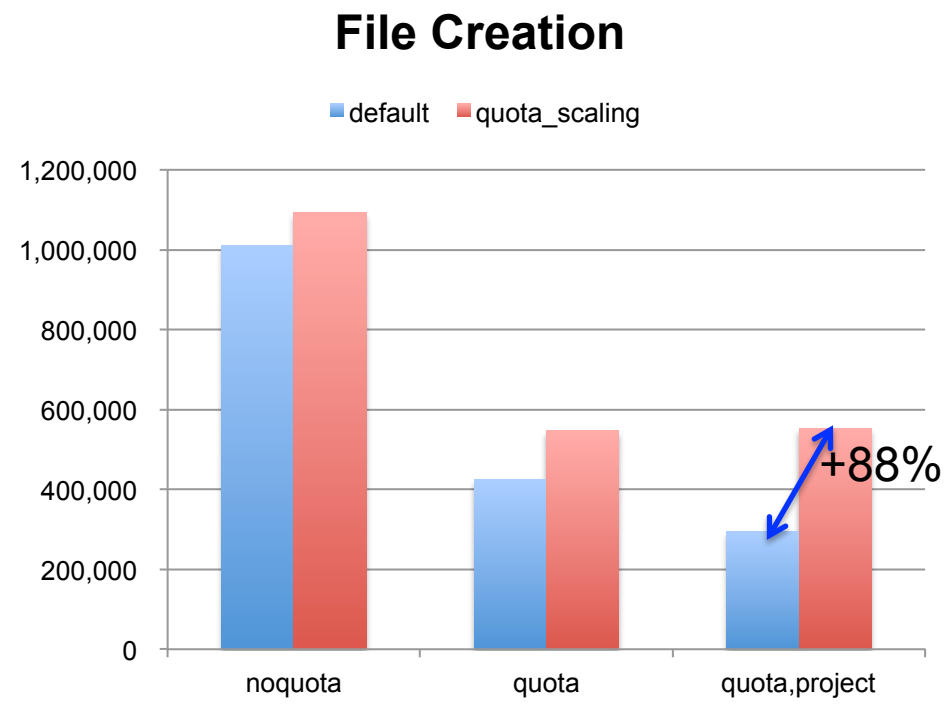
Quota scalability problem

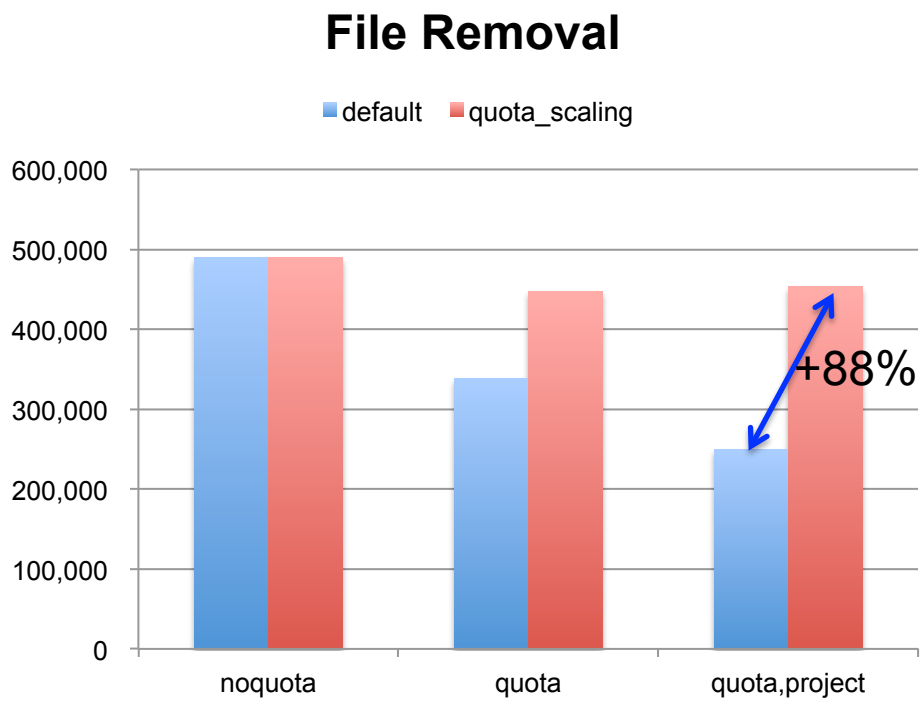
- **File creation/unlink affects performance when quota enabled**
  - Same behaviors on RHEL7 and upstream kernel
  - Project quota introduced additional performance penalty



Quota scalability improvements in Ext4

- **New quota scaling patch introduced in upstream kernel**
  - Tested new Jan Kara's quota scaling patches (merged in 4.14)
  - Huge performance gains when quota enabled

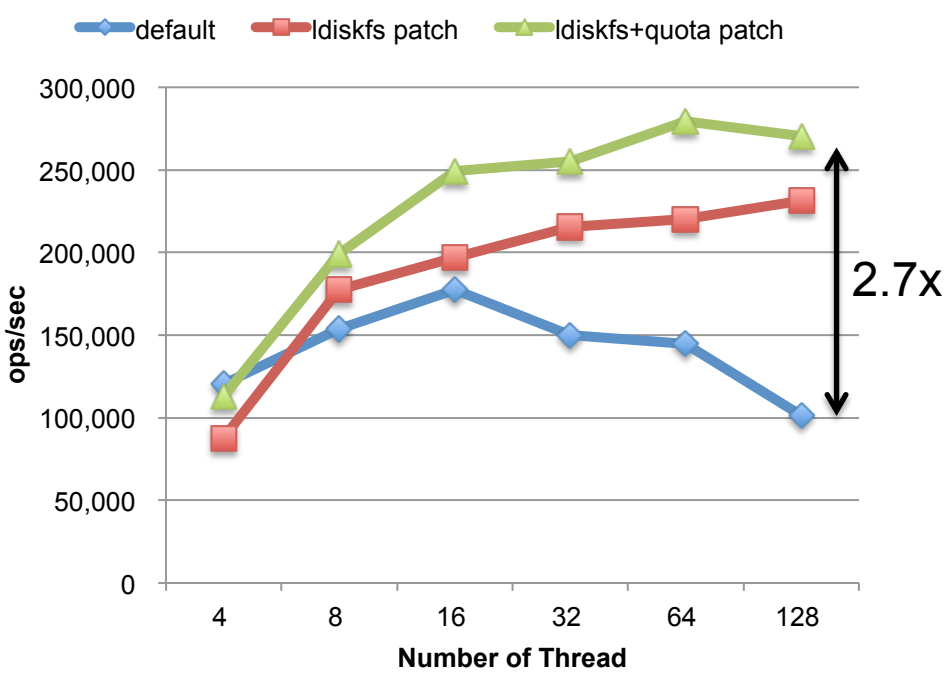




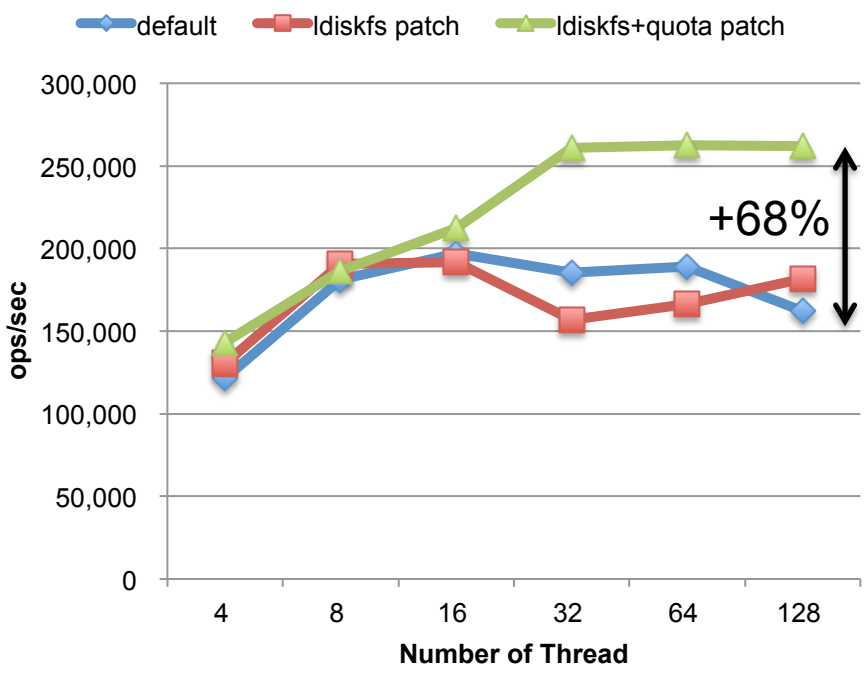
Performance results
Quota scaling for Lustre

- Experimentally ported quota patches to RHEL7 kernel for Lustre server (LU-10034: Quota scaling for Lustre)
- User/Group and Project quota enabled

mds-survey(File Creation)



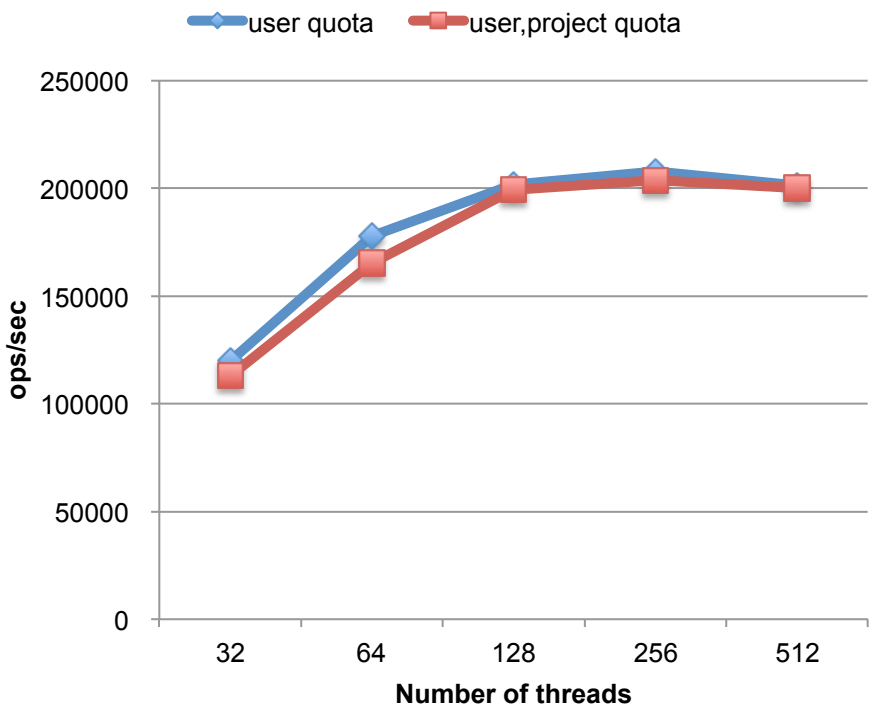
mds-survey(File removal)



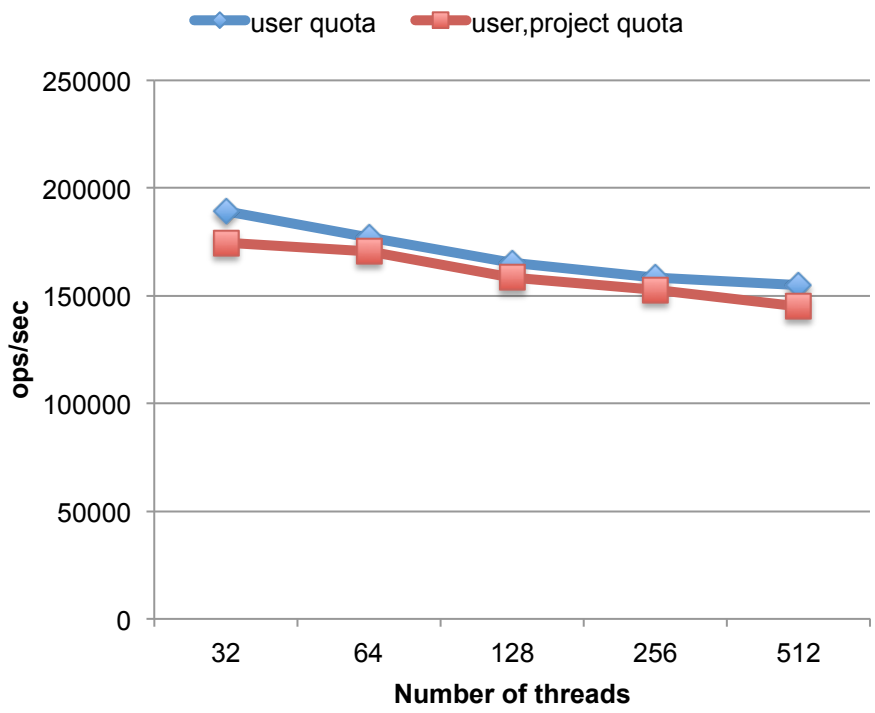
End-to-End Lustre Metadata Performance mdtest (1.28M files, unique directories)

- Applied ext4's contention fix and quota scaling patch against RHEL7 kernel
- mdtest from 32 clients to single MDS/MDT

mdtest(File creation)



mdtest(File removal)



Additional metadata performance efforts

- **LU-7251 osp: do not assign commit callback to every handle**
  - Reduction and optimization of cancel RPCs
  - Improved unlink operations
- **LU-9840 lod: add ldo_dir_stripe_loaded**
  - Performance improvements on file creation to single shared directory
- **LU-9972 performance regression on rmdir**
- **LU-10005 osp: cache non-exist EA**
  - Performance regression for non-root MDT

Conclusions

- Evaluated Lustre-2.10 metadata performance on new hardware and exposed new performance limits.
- Fixed contention problem at inode allocation and tested ext4 quota scaling patch.
- Demonstrated 200K file creation and 150K unlink per second on single MDT with full quota accounting.
- Need further investigation on Lustre unlink performance.
- Will test even more CPU cores to maximize single metadata performance.

Thank you!