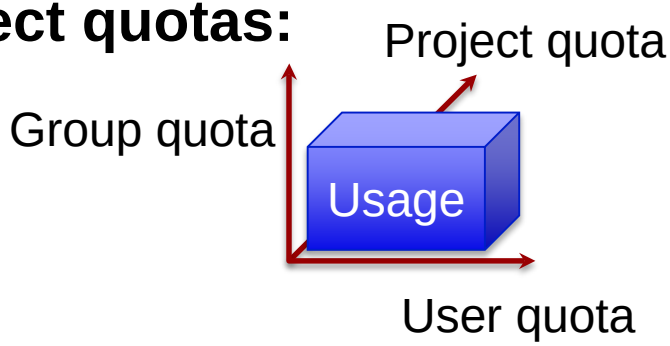
Project Quota for Lustre

Li Xi, Shuichi Ihara

DataDirect Networks Japan

What is Project Quota?

- **Project**
  - An aggregation of unrelated inodes that might scattered across different directories
- **Project quota**
  - A new quota type that supplements existing user/group quota types
- **File systems supporting project quotas:**
  - XFS
  - GPFS: “fileset” quota
  - New: Ext4
  - Coming soon: Lustre!



Project Quota of ext4

All project quota patches for ext4 have been merged into the kernel mainline. (Since linux-4.5)

8b4953e ext4: reserve code points for the project quota feature

```
--- a/fs/ext4/ext4.h
+++ b/fs/ext4/ext4.h
@@ -374,6 +374,7 @@ struct flex_groups {
 #define EXT4_EA_INODE_FL                0x00200000 /* Inode used for large EA */
 #define EXT4_EOFBLOCKS_FL               0x00400000 /* Blocks allocated beyond EOF */
 #define EXT4_INLINE_DATA_FL             0x10000000 /* Inode has inline data. */
+#define EXT4_PROJINHERIT_FL             0x20000000 /* Create with parents projid */
 #define EXT4_RESERVED_FL                0x80000000 /* reserved for ext4 lib */
```

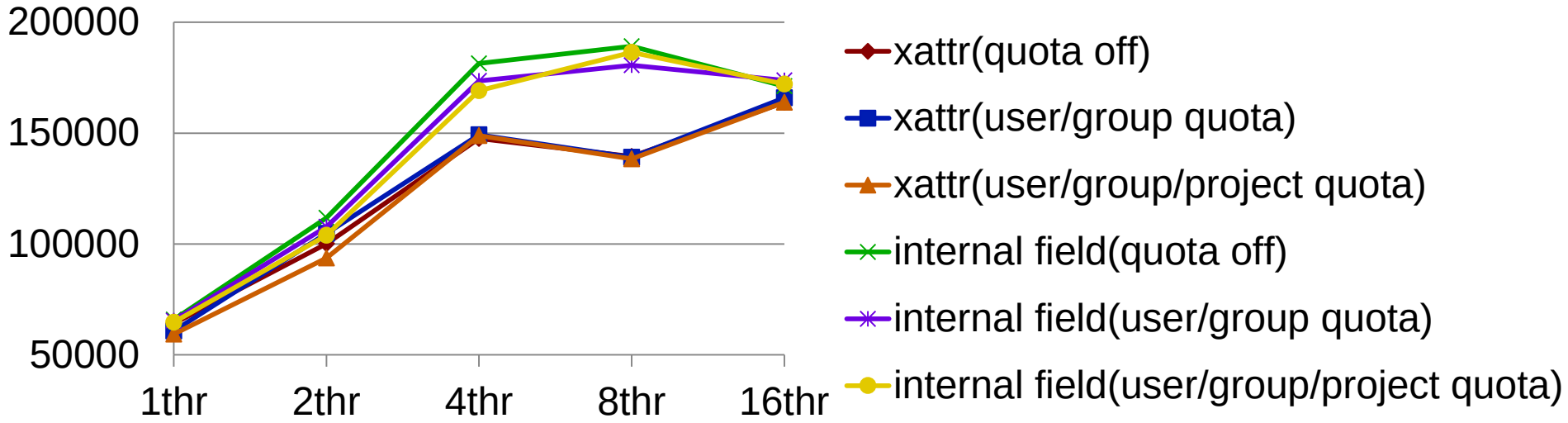
9b7365f ext4: add FS_IOC_FSSETXATTR/FS_IOC_FSGETXATTR interface support

689c958 ext4: add project quota support

040cb37 ext4: adds project ID support

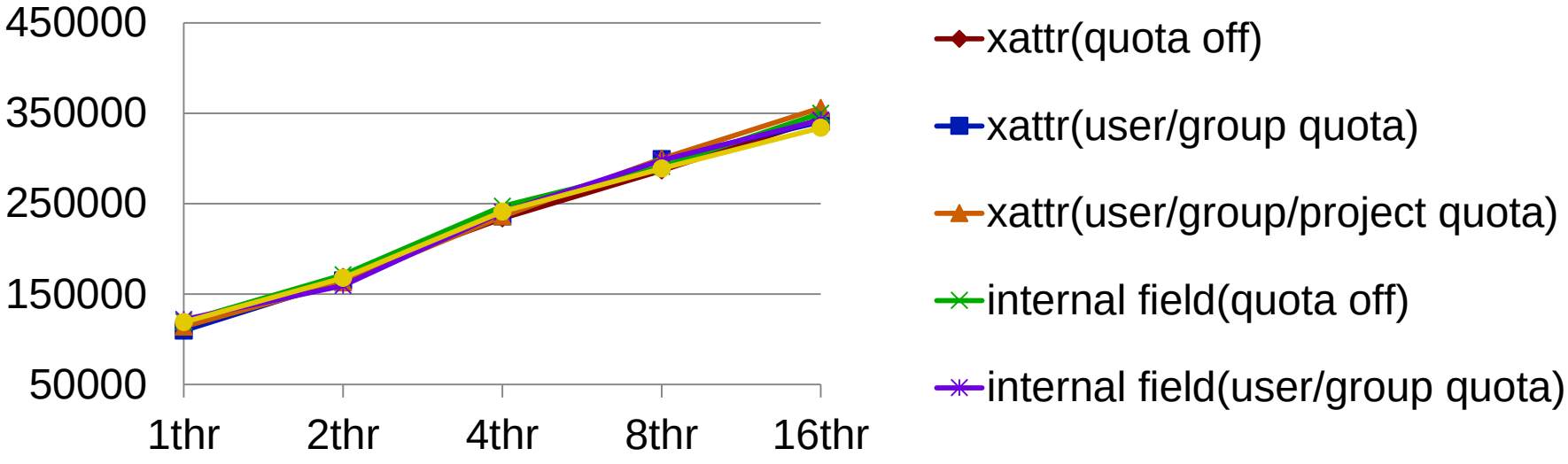
Performance: File creation on Ext4

Kernel: 3.16.0-rc5
Server: Dell R620 (2 x E5-2667 3.3GHz, 256GB memory)
Storage: 10 x 15K SAS disks(RAID10)
Test tool: mdtest-1.9.3.
Mdtest created 800K files in total.



Performance results: File removal on Ext4

Kernel: 3.16.0-rc5
Server: Dell R620 (2 x E5-2667 3.3GHz, 256GB memory)
Storage: 10 x 15K SAS disks(RAID10)
Test tool: mdtest-1.9.3.
Mdtest created 800K files in total.



Why Project Quota for Lustre?

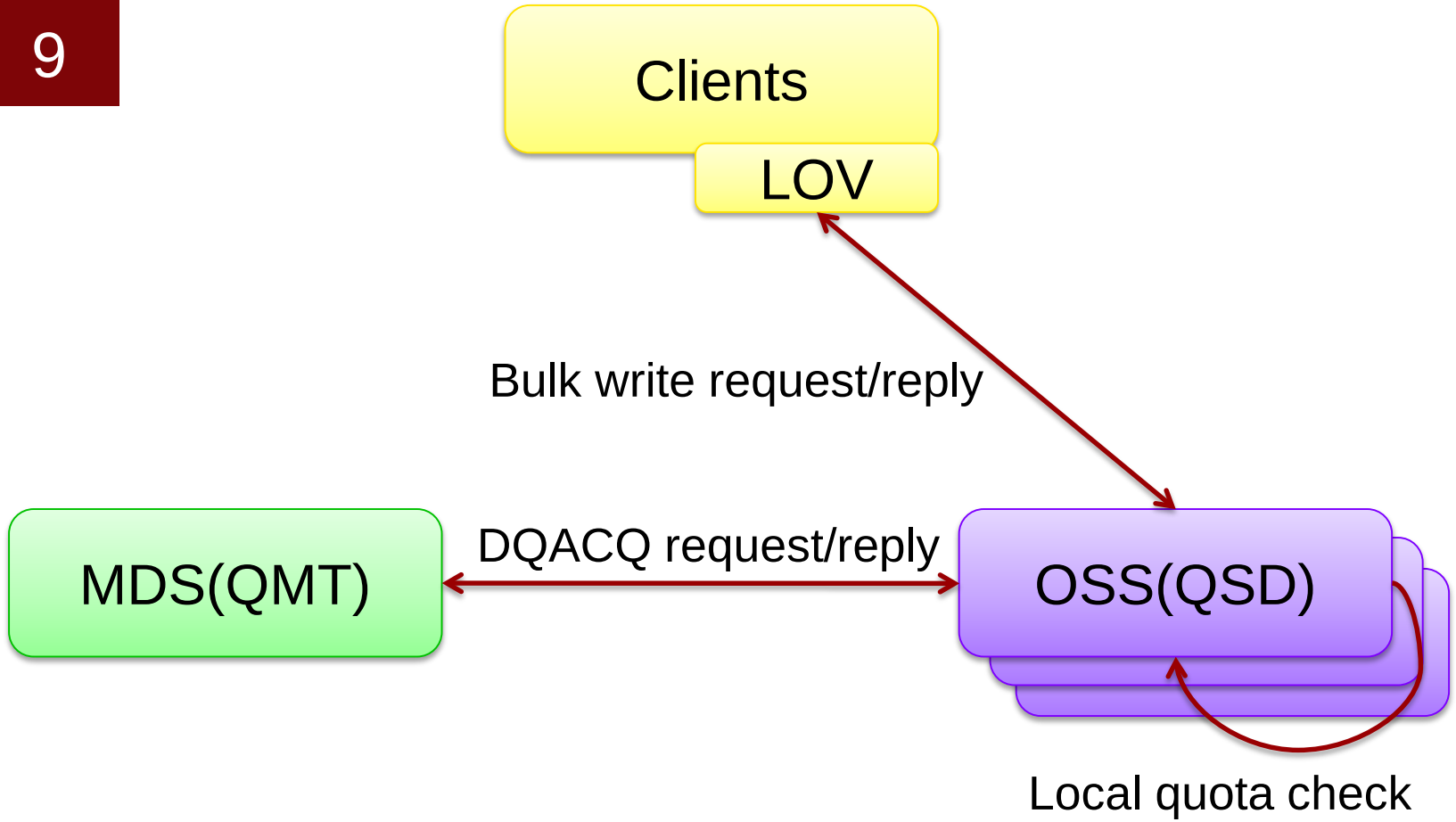
- **Per-user or per-group quota is no more sufficient to account for the use scenarios encountered by storage administrators (e.g. project based storage volume allocation)**
  - User/Group configurations usually don’t change frequently
  - Projects within a massive file system change a lot
- **Quota of small parts of a file system helps administrator to make capacity plans of entire storage's volume**
  - Space accounting and limitation enforcement based on OSTs/sub-directories/file-sets/projects enables various use cases

Semantics of Project Quota

- **Project ID**
  - Inodes that belong to the same project possess an identical identification, just like user/group ID
- **Inherit flag**
  - An inode flag which defines the behavior related to projects
- **Directory with inherit flag:**
  - All newly created sub-files inherit project IDs from the parent
  - No renaming of an inode with different project ID to the directory is allowed (EXDEV returned)
  - No hard-links from an inode with different project ID to the directory is allowed (EXDEV returned)
  - The total block/inode number of a directory will be the minimum of the quota limits and the file system capacity

Architecture of Lustre Quota

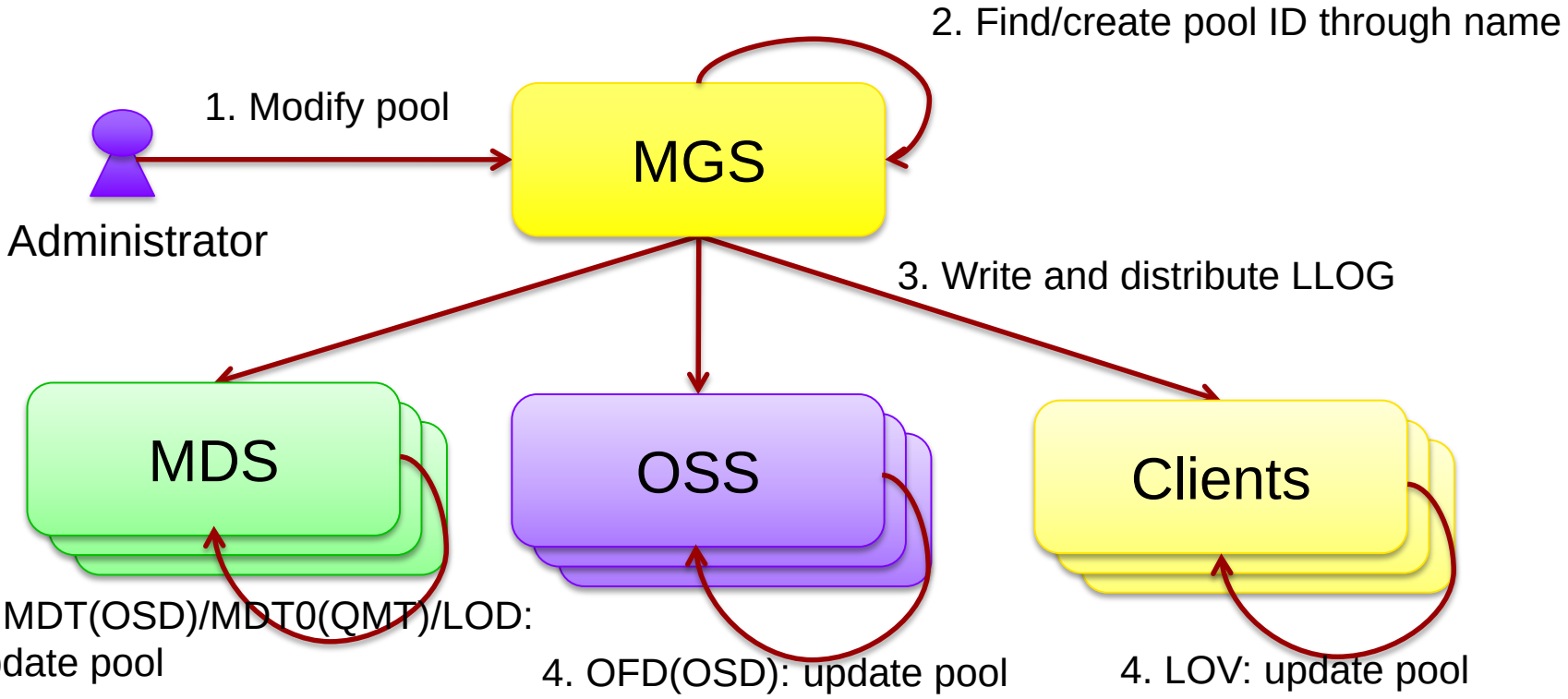
- **Quota “master”**
  - A centralized server hold the cluster wide limits
  - Guarantees that global quota limits are not exceeded and tracks quota usage on “slaves”
  - Stores the quota limits for each uid/gid/project ID
  - Accounts for how much quota space has been granted to slaves
  - Single quota master is running on MDT0 currently
- **Quota “slaves”**
  - All the OSTs and MDT(s) are quota “slaves”
  - Manage local quota usage/hardlimit acquire/release quota space from the master



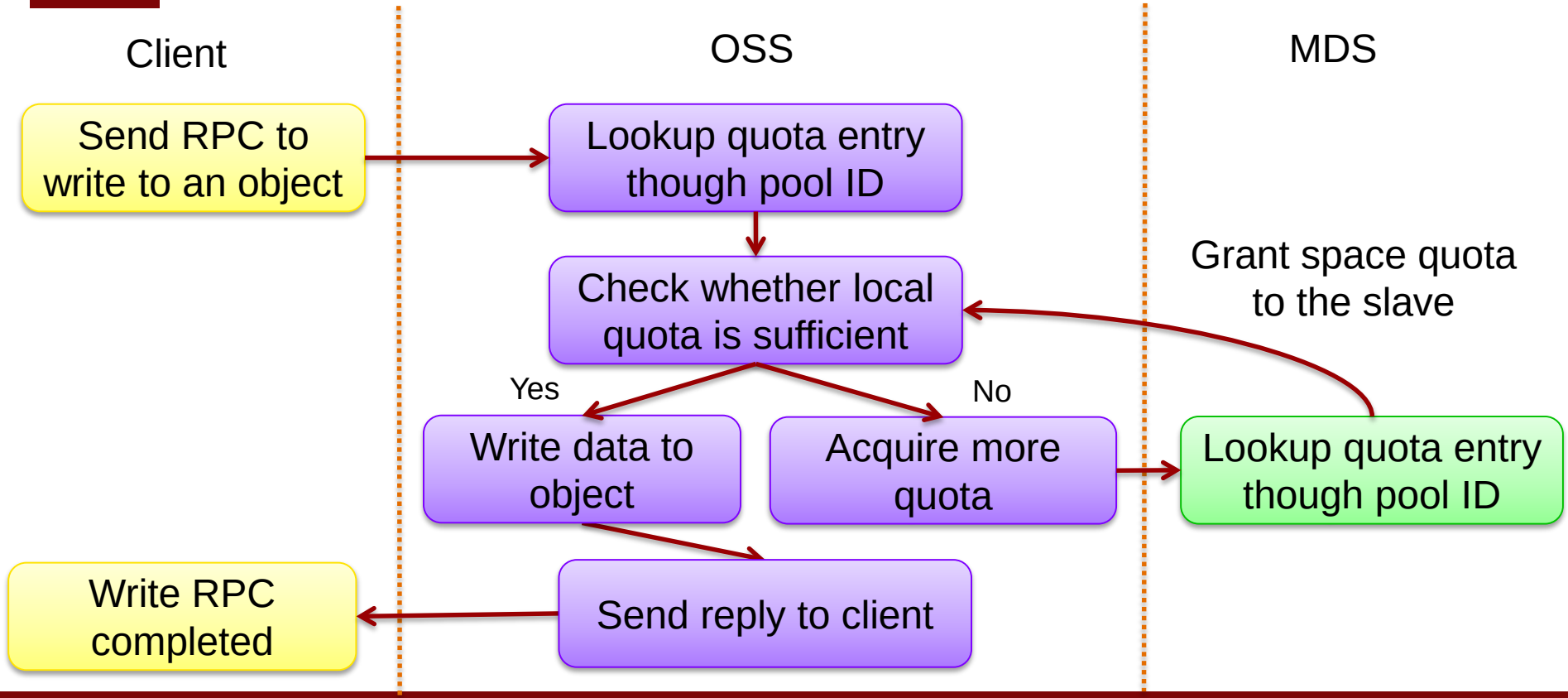
Implementation Goals

- **Integrated in current quota framework**
  - Ability to enforce both block and inode quotas
  - Support hard and soft limits
  - Support user/group/project accounting
- **Full support of pools**
  - Dynamic change of pool definition
  - Separate quotas for users/groups for each pool
- **No significant performance impact**

Design: Pool Definition in LLOG



Design: Flow of a write request



Use case #1

- **Usage accounting/restriction of sub-directories**
  - Pool/project IDs are inherited from parent directory by default
  - A quicker replacements of “du” command on big directories: “lfs quota”

```
# lfs setstripe --pool dir1 /lustre/dir1
# lfs setstripe --pool dir2 /lustre/dir2
# lfs quota -p dir1
# lfs quota -p dir2
```

Use case #2

- **Usage accounting/restriction of file sets according to attributes**

  # lfs find … /lustre | xargs lfs setstripe --pool fileset1

  # find … /lustre | xargs lfs setstripe --pool fileset2

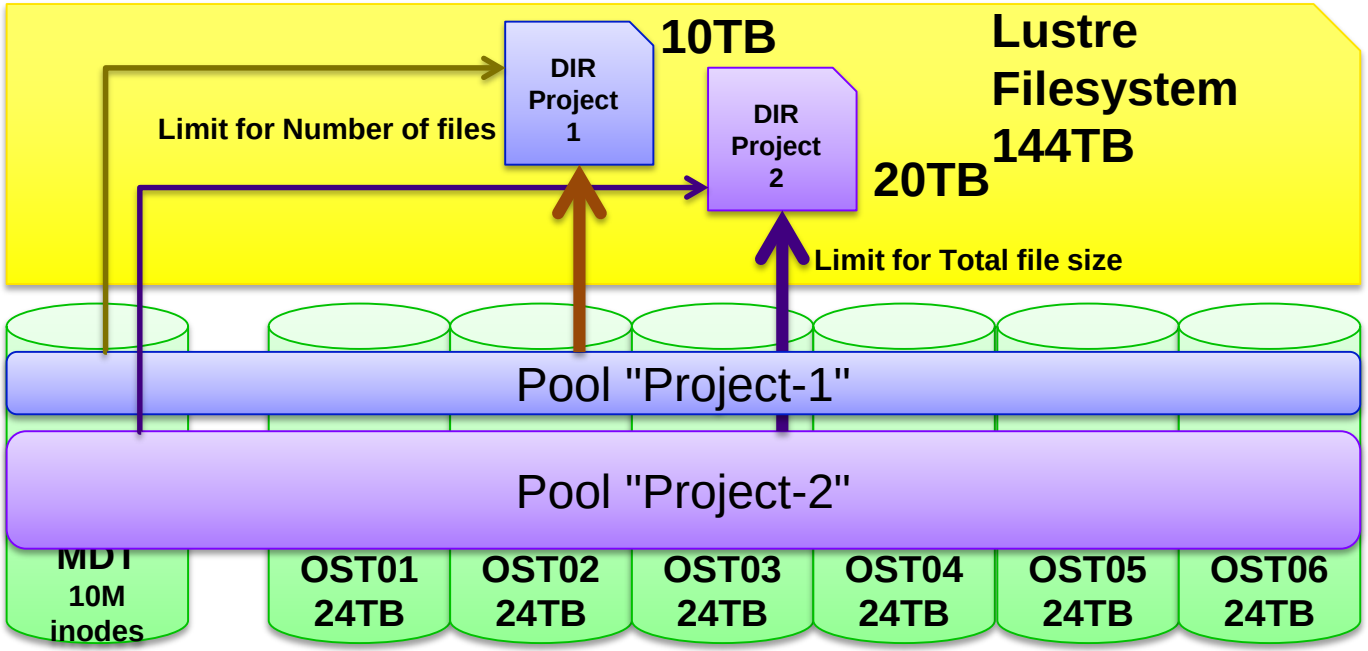
  # lfs quota -p fileset1 /lustre

  # lfs quota -p fileset2 /lustre

  - Files in the same file set can be scattered in different locations
  - The space/inode usage of file sets can be monitored in real time
  - Requirement: Need to be able to change the pool name of a file without changing its layout!

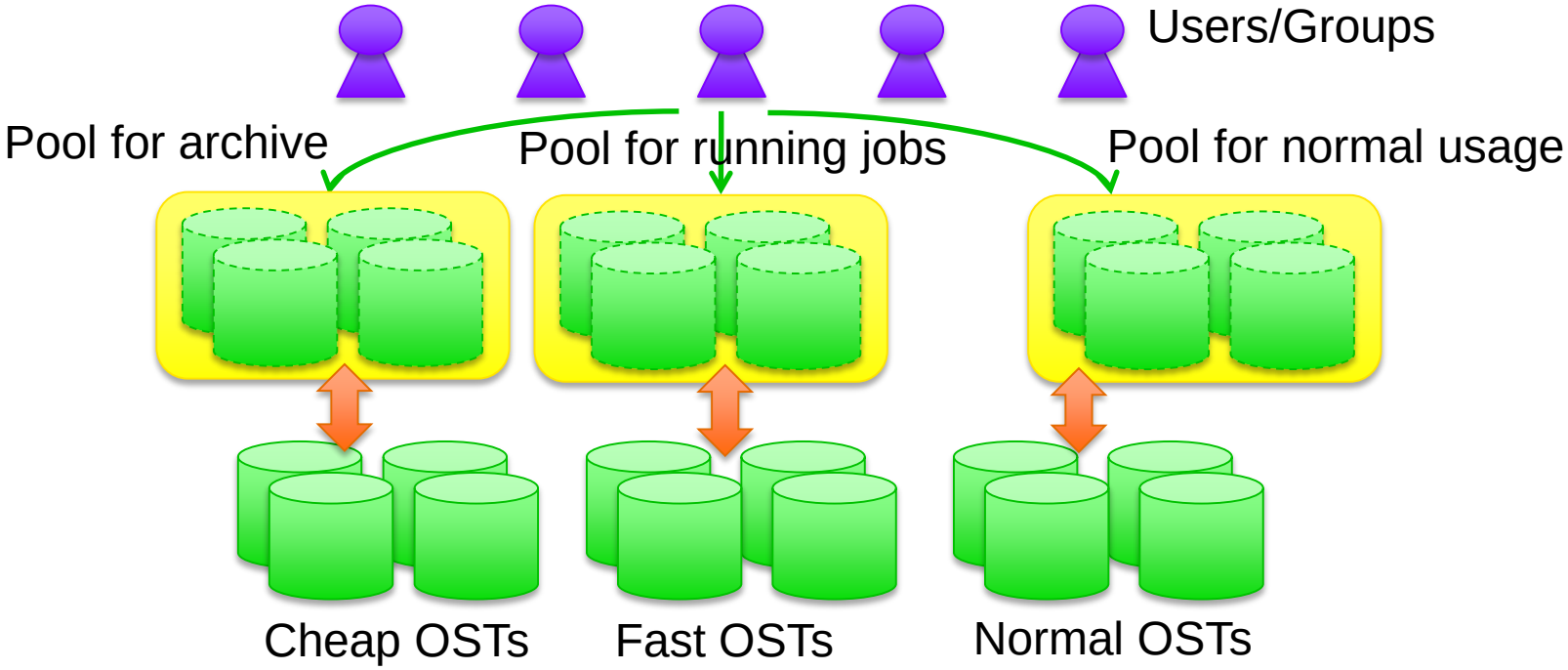
Use case #3

Total usage limitations of collaboration projects



Use case #4

Usage tracking of different OST types



Development Status

- **Lustre-2.9 will be the initial branch for project quota**
- **All critical patches that change disk format or protocol will be pushed to Lustre-2.9**
  - Backported mainline ext4 codes against ldiskfs
  - Backported mainline e2fsprogs patches against lustre-master
  - Pushed Lustre Quota cleanup patches to add new quota type
  - OST Pool enchantment patches to support “pool id”
  - All patches are tracked on LU-4017 and under active review
- **Other Lustre quota patches are ready in DDN Lustre branch and will be pushed to master after Lustre-2.9**
- **More review, tests and benchmarks are needed**

Conclusion

- We proposed a implementation of project quota in Lustre
- Lustre project quota shares the same semantics with XFS and Ext4
- The combination of project quota with OST pools enables more use cases than pure project quota
- Benchmark results show that project quotas cause very limited performance impact

Thank you!