Extensions for Project ID

Li Xi, Hongchao Zhang

ABSTRACT

Whamcloud

▶ Project ID is a mature Lustre feature widely used by Lustre users, it provides easy ways to manage subsets of files in Lustre for different purposes

▶ We are going to extend the Project ID feature with new functionality to enable new use cases, for example, Read-Only-Access or Append-Only-Access subsets of files

▶ The extensions will use existing project quota and LDLM mechanisms to avoid complexity to keep system stable

Project ID

▶ Project ID are just like UID/GID, which can be assigned to file or directory

▶ Project ID can be inherited from the parent directory (option $'-s'$ in "lfs $project$")

▶ Project ID are 32 bits for Lustre (64 bits for ZFS, 32 bits for LDiskFS)

▶ LFS command "lfs $project$" can be used to manage the Project ID in Lustre

LFS Support for Project ID

Whamcloud

▶ Using *LFS project* command to manage the project ID

- Set the project ID for file or directory

```
# lfs project –p 1000 /path/to/the/file
```

- Set the project ID for all files and sub-directories under a directory

```
# lfs project –p 1000 –r /path/to/directory
```

- Set the project ID and inherit flag for all files and sub-directories under a directory

```
# lfs project –p 1000 –r -s /path/to/directory
```

- Clear the project inherit flag

```
# lfs project –p 1000 –C –r -k /path/to/directory
```

Quota Flags

► There is a quota flags for each quota ID (UID/GID/PRJID)

► Currently, only one flags bit (default quota flag) is used in Lustre

► Quota flags can be used to control the behaviors of all the directories or files owned by the quota ID (UID/GID/PRJID)

► Together with Project ID, Project quota flags would be more useful

Immutable Project Flag

▶ When the files are backed up, it's better to set as immutable to protect data from accidental deletion, then it can be recovered at any time

▶ In Lustre, *Project ID* and *Immutable Project Flag* can be used together to achieve it easily

▶ The immutable flag will be cleared automatically after the backup files were copied out to recover

Use Case of Immutable Project Flag

▶ Back up files to be immutable automatically

Step1: create the backup directory and set an unique project ID for it

```
# mkdir /path/to/backup/directory
# lfs project -p 1000 -s /path/to/backup/directory
```

Step2: copy the files and directories into the backup directory, of which the Project ID is inherited by the newly create ones

```
# cp xxx /path/to/backup/directory
```

Step3: set the unique project ID to be immutable

```
# lfs project -p 1000 -i /mnt/lustre
```

▶ The files or directories in backup directory will be immutable automatically

Step4: copy out some files to reuse

```
# cp /path/to/backup/directory xxx
```

▶ The files out of backup directory will be cleared immutable flag automatically

Append Project Flag

▶ Some files, such as logs, are required to be open with O_APPEND mode to protect the existing data from being overwritten

▶ In Lustre, these files can be put into some designated directory, the project ID of which will be inherited by the files or directories created in it

▶ Setting the Append Flag for this project ID will enforce the files to be open with O_APPEND flag

Use Case of Append Project Flag

► Enforce the log directory to be open with O_APPEND

Step1: create the log directory and set a unique project ID for it

```
# mkdir /path/to/log/directory
# lfs project –p 1001 –s /path/to/log/directory
```

Step2: set the unique project ID to be Append-Only
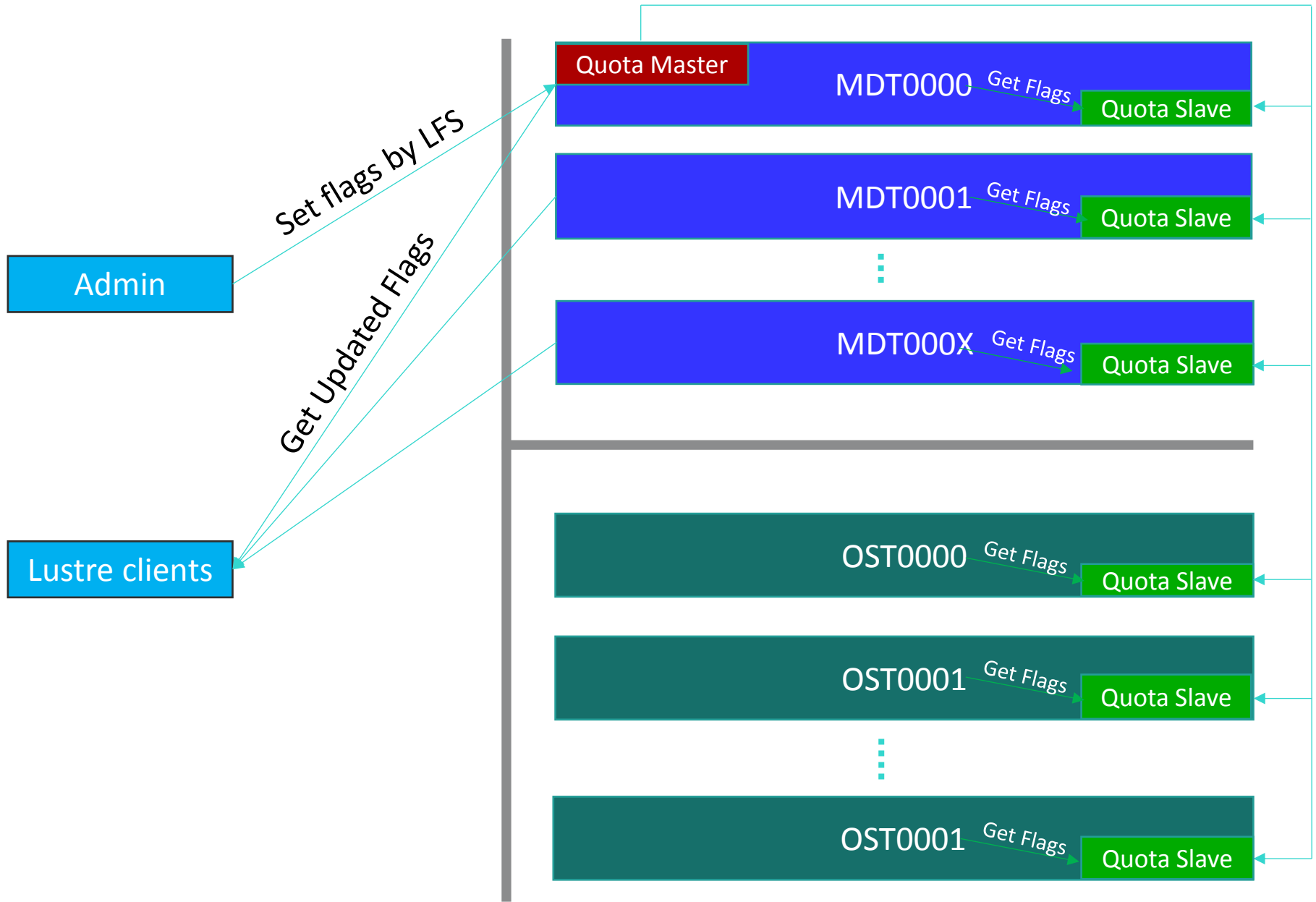
```
# lfs project –p 1000 –a /mnt/lustre
```

► The files created in the log directory will be enforced to open with O_APPEND

Audit Project Flag

► For security, the various access operations on some files are needed to be recorded for audit

► In Lustre, the administrator can use Audit Project Flag to implement it

► A separate changelog file will be generated for each project ID with Audit Project Flag

Set/Clear Project Flags

► The project flags will be managed by LFS command at client (need a Lustre mount), it will be saved on Quota Master at MDT0000

► After setting, it will be propagated to the Quota Slaves on all MDTs and OSTs by the existing quota framework (using LDLM lock glimpse callback)

► Quota Slave will save the latest setting locally, MDTs or OSTs will get the updated setting and change the flags of the inodes accordingly

Quota Master

MDT0000 Get Flags Quota Slave

MDT0001 Get Flags Quota Slave

MDT000X Get Flags Quota Slave

Admin

Set flags by LFS

Get Updated Flags

Lustre clients

OST0000 Get Flags Quota Slave

OST0001 Get Flags Quota Slave

OST0001 Get Flags Quota Slave

The updated quota flags are propagated to all MDTs or OSTs

Whamcloud

LFS Support for Project Flags

► Extend *LFS project* command to set different flags for project ID

- Set the immutable flag

  `# lfs project –p 1000 –i /mnt/lustre`

- Clear the immutable flag

  `# lfs project –p 1000 –I /mnt/lustre`

- Set the append flag

  `# lfs project –p 1000 –a /mnt/lustre`

- Clear the append flag

  `# lfs project –p 1000 –A /mnt/lustre`

- Get the current flags

  `# lfs project –p 1000 –f /mnt/lustre`

Advantages of Project Flags

► **Performance**

- Files or directories are not needed to be iterated to modify its flags
- The flags can be switched on or off quickly at any time

► **Stable**

- Using existing quota framework to avoid complexity
- Some of the flags have already been implemented in Lustre

► **Easy to use**

Next Steps

► Introduce more flags, for example, the encryption flag to indicate the client needs to supply a password to open the file

► These flags can also be extended for UID/GID, which could be used to control the access of the files owned by it

► See LU-14915 for further details

Whamcloud

Thank You!

ddn