LUG 2025
Lustre User Group

Whamcloud

Lustre Developer Day

Andreas Dilger

Lustre Principal Architect

Trash Can/Undelete for Files and Directories (2.18 WC)

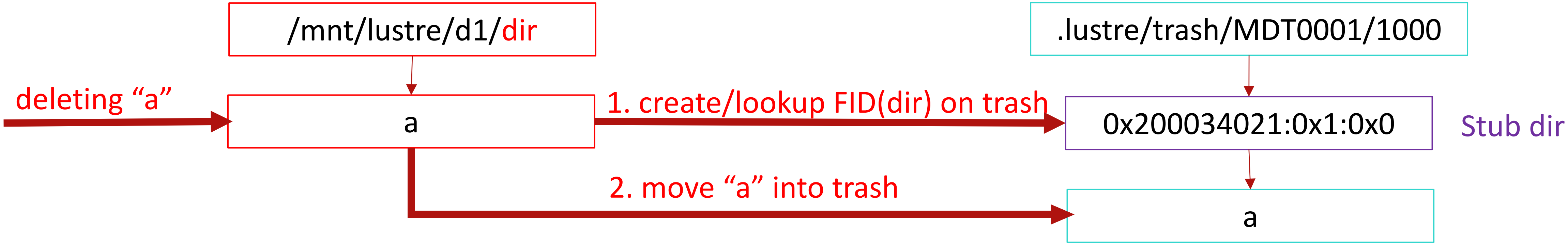
- ▶ Allow files/directories to be undeleted after `rm/rmdir`
  - Rescue users from fat-finger mistakes or malicious scripts
  - Handle "`rm -r`" properly to allow whole-tree recovery
- ▶ Deleted files in trash are flagged and treated specially
  - Removed from user/group/project quota and `df` usage
  - Files cannot be read to avoid abuse, and apps know files are deleted
- ▶ Virtual `.Trash` directory visible in every directory
  - Can use normal tools to list and recover files or directories
  - `.Trash` is hidden from normal directory listing
- ▶ Users can view and recover their own files
  - Configurable expiry time before cleanup (e.g. max age = 7d)
  - Configurable filesystem fullness threshold (e.g. 80% full)
  - More sophisticated cleanup policy in userspace (e.g. by user, project, nodemap)

https://wiki.whamcloud.com/pages/viewpage.action?pageId=351437962

Moving regular file into trash

- "Last unlink" for an inode will create (or lookup) a stub directory on the MDT that the file located
  - Stub is created in subdir named by UID of inode being deleted to isolate user's trash
  - Stub is created named by its parent's FID in trash (pFID)
- Then rename the regular file into this directory on trash
  - Add `user.del` xattr to file recording JobID of process deleting the file

"dir" on MDT1 with FID: 0x200034021:0x1:0x0; "a" is a deleting regular file under "dir";

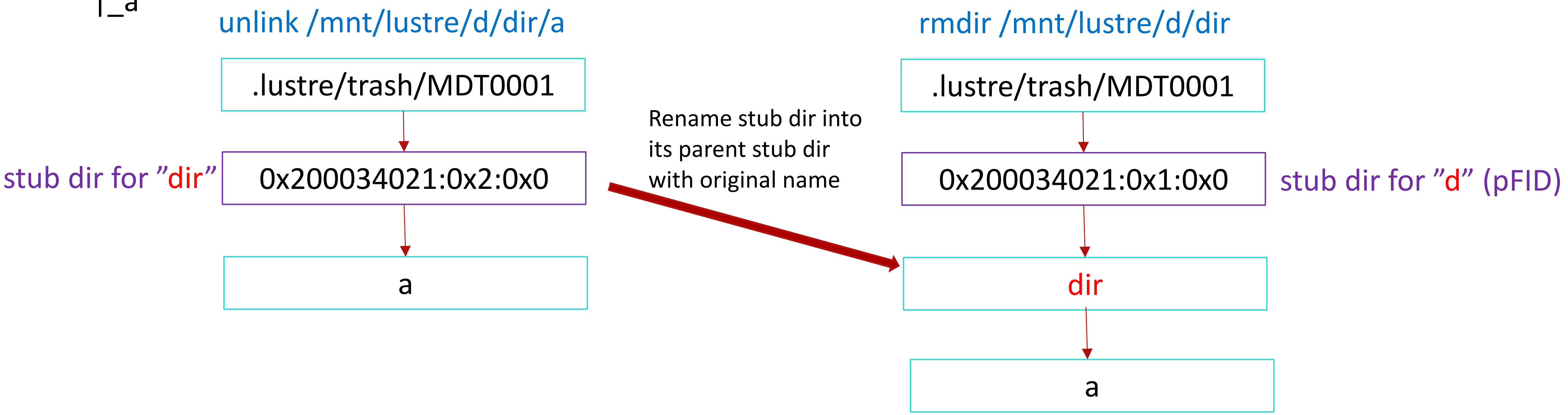


- Access trash from Lustre namespace on a client:
- # ls –R /mnt/lustre/.lustre/trash/MDT0001/1000
  .lustre/trash/MDT0001/1000/0x200034021:0x1:0x0
  .lustre/trash/MDT0001/1000/0x200034021:0x1:0x0/a

Patch: https://review.whamcloud.com/57748

Move an empty directory into trash

```
/mnt/lustre Path: /mnt/lustre/d/dir/a
|_d       FID: 0x200034021:0x1:0x0
   |_dir  FID: 0x200034021:0x2:0x0
      |_a
```



- If the sub file is a directory, we must restore the whole subtree within this sub directory under the stub dir.
- How to restore sub files in the much deeper level?
- However, it is allowed to move files in trash into another place in Lustre namespace
- The stub dir cannot be moved and will not be visible under normal access

Space Accounting of Files in Trash

- To avoid user/administrator confusion, files in Trash Can are removed from UID/GID/PROJID quotas
  - Otherwise, users have no way to reduce quota usage
- Want to allow accounting quotas under some other ID
  - One Trash ID per source ID (e.g. ID+2B) makes it easy to revert to original IDs, track per-user Trash usage
  - A single Trash ID for all files, which can be set per nodemap/tenant, easier to integrate with other tools

- Need to remove Trash Can usage from "`df`" output to avoid administrator confusion
- Need to have some mechanism to easily see trash usage
  - "`lfs df --trash`" option to show trash usage per MDT/OST?

Create/delete files with the same name repeatedly

- It must handle the special case when create/delete files with the same name under a directory repeatedly with trash enabled. i.e. under the directory "/mnt/lustre/d/d1", do operations repeatedly:
  - `touch /mnt/lustre/d/d1/tf; unlink /mnt/lustre/d/d1/tf`
  - `touch /mnt/lustre/d/d1/tf; unlink /mnt/lustre/d/d1/tf`
  - `mkdir /mnt/lustre/d/d1/tf; rmdir /mnt/lustre/d/d1/tf`

- When moving the file into trash found that the dentry index already existed in trash
  - Change the dentry name with a unique ID (timestamp) to disambiguate copies
  - Change the naming for repeated name on a same stub dir on trash:

    i.e. …/`.trash/MDT0001/`*`UID`*`/`*`pFID`*`/tf;` …/`.trash/MDT0001/`*`UID`*`/`*`pFID`*`/tf.`*`timestamp`*
  - User can select manually which version to restore, aided by timestamp to identify when it was deleted

Fault Tolerant MGS (LMR-FTM) (2.18 WC)

- Run MGS service on multiple MDS nodes for availability (LU-17819)
  - Allow clients to read config llogs from **any MGS node**, stored on MDT
  - Reduces mount time/timeouts, distributes load in large clusters
  - MGS Imperative Recovery even if “primary” MGS node restarts
- Mirror MGS config logs to remote MDTs for redundancy
  - Use RAFT Consensus algorithm to coordinate MGS cluster
  - MGS Leader election, heartbeat, consistent log updates
  - Append-only logs, matches existing MGS config llog format

https://raft.github.io/

https://en.wikipedia.org/wiki/Raft_(algorithm)

MGS Config Log Replication

- MGS Leader is elected by RAFT algorithm
  - Leader continually pinging peers to keep Leadership in control
  - Otherwise, if Leader has gone quiet start a new election and elect leader with newest logs
  - Prefer MGS Leader with local MGT device?
- MGS replicas are read-mostly, but need occasional updates
  - Updates are controlled/consistent by RAFT consensus algorithm
  - Do we need a separate election for each config llog?
  - Can validate/repair local llog files against remote replicas (checksum per record?)
- Should MGS Leader migrate to node running “`lctl set_param -P`”?
  - More overhead on first operation, local requests for subsequent updates
- Handle initial OST/MDT replication – need to avoid Leader MGS ping-pong
  - Targets should pick first MGS NID during registration?
- Handle multiple filesystems managed by single MGS
- Need to replicate MGS IR Table for recovery and NID broadcast to clients

Useful Lustre Development Links

- General development overview https://wiki.lustre.org/Development
- JIRA issue tracking system https://jira.whamcloud.com/ (LU/LUDOC projects)
- Lustre Operations Manual https://wiki.lustre.org/Lustre_Manual_Changes
- How to submit patches overview https://wiki.lustre.org/Submitting_Changes
- Gerrit patch management system https://review.whamcloud.com/ (Details)
- Commit comment style https://wiki.lustre.org/Commit_Comments
- Lustre coding style https://wiki.lustre.org/Lustre_Coding_Style_Guidelines
- Jenkins build system https://build.whamcloud.com/
- Maloo test results database https://testing.whamcloud.com/
- Lustre mailing lists https://www.lustre.org/mailing-lists/
- Lustre Slack channel using join link or use QR code on the right ->
- Autotest Test-Parameters:
  https://wiki.whamcloud.com/display/PUB/Changing+Test+Parameters+with+Gerrit+Commit+Messages
- Presentation with tips on using Autotest, Maloo, Git, and Gerrit:
  - https://wiki.lustre.org/images/8/8e/LUG2024-Lustre-Autotest-Maloo-Gerrit.pdf

Join the Lustre Slack Channel

Small Project Hackathon with Other Lustre Developers

- Start and/or finish some small Lustre project(s)
  - Several options on next page, or work on your own
- Good opportunity for new developers to meet veterans
- Knowledgeable developers available for questions
  - Quick turn-around for questions and problem solving
- Tips for effective use of Git, Gerrit, Autotest, Maloo
  - Sidebar for those of you interested

Hackathon Small Project Suggestions

Whamcloud

Low Difficulty

- LU-17648 save jobid of process deleting file
- LU-16622 mark volatile files with `I_LINKABLE`
- LU-18818 use libext2fs in `ldiskfs_write_ldd()`
- LU-17957 user immutable via atime+chmod
- LU-18891 increase default max-inherit-rr
- LU-16738 mount.lustre with many MGS NIDs
- LU-18889 add "lfs find -printf" optimization
- LU-17514 hint for number of connected clients
- LU-17000 Lustre Coverity issues

- LU-4315 lfs and lctl man pages
- LUDOC many improvements to Lustre Manual
- Other "easy" labeled tickets in Jira

Medium Difficulty

- LU-12480 add `STATX_PROJID` to Linux kernel
- LU-17515 dynamic `conns_per_peer` tuning
- LU-13527 OST FID lookup via "lfs fid2path”
- LU-13123 list client NIDs with job in `job_stats`
- LU-16671 statfs cache for project directories
- LU-18857 allow/deny MDT/OST register to MGS
- LU-15419 move quota off MDT0000
- LU-15414 mirror FLDB to all targets
- Other "medium" labeled tickets in Jira

Higher Difficulty

- LU-7880 Performance stats in OBD_STATFS
- LU-1941 FIEMAP compressed file support