Online Distributed Coherency Checking for Lustre (*lfsckNG*)

- Andreas Dilger
  Principal Engineer
  Whamcloud, Inc
Current State of *lfsck*

- Used only after serious corruption
  - OST loss/corruption, MDT corruption
  - Very slow to run checks, unusable on large systems
- Tightly integrated with *e2fsck*
  - Not suitable for other back-end filesystems
  - Not possible to do incremental checks
  - Makes e2fsprogs maintenance difficult due to db4 use
- Depends on external databases
  - Very slow to create databases
  - Offline, or outdated before ready for use
  - Databases very large sparse files, hard to transfer
Need to Move *lfsck* Forward

- Need to scale far beyond current size
  - 100B files in 1 year, 1T files in 3 years
  - Handle thousands of inodes per second efficiently

- Online coherency checking needed
  - Avoids lengthy downtime
  - No external databases needed
  - Continuous incremental checks, handle in-flight changes

- Core code isolated from backend
  - Useful for Idiskfs, ZFS, btrfs, ...
  - Use Lustre RPCs for communications
  - Share bulk RPC optimizations for lfsck, stat-ahead

- Handle distributed namespace
OSD Internal Consistency

- **Not** doing internal filesystem check
  - Partly possible, but work++ for ldiskfs
  - Online checks exist for ZFS, Btrfs already

- **Verify OSD Object Index (OI Scrub)**
  - Mapping for FID->internal inode number (2.x only)
  - May be corrupted, or after file-level backup/restore
  - Inode->FID pointer on each inode (`LMA` xattr, since 2.0)
  - Iterate in-use inodes in filesystem
  - On-demand lookup-driven check/correction also
MDT-OST Consistency

- Verify OST objects used by inode exist
  - MDT inode layout references OST objects (*LOV* xattr, all versions)
  - Verify OST object UID and GID are correct, for quota
  - Verify MDT size is correct, if needed for SOM

- Verify OST objects referenced by correct inode
  - OST objects store MDT inode back-pointer (*fid* xattr, since 1.6.0)
    - Detect if multiple inodes reference same object
    - Also used for OST recovery (*ll_recover_lost_found_objs*)
  - OST in-memory bitmap of referenced objects for orphan checks
    - Clear bit when object accessed by MDT-driven *lfsck*
    - Unreferenced objects can check MDT inode or if object is orphan
MDT-OST Consistency

Inode X
UID U
GID G

LOV EA
4 stripes

OST 0
Object J

OST 1
Object K

OST 2
Object L

OST 3
Object M

MDT

OSTs

Object J
Inode X
Stripe 0
UID U
GID G

Object K
Inode X
Stripe 1
UID U
GID I

Object L
Inode X
Stripe 2
UID U
GID G

Object M
Inode X
Stripe 3
UID U
GID G

Online Distributed Coherency Checking for Lustre

© 2011 Whamcloud, Inc.
MDT Internal Consistency

• Verify directory hierarchy
  - Inode points to parent directory (link xattr, since 2.0)
  - Can verify inode hard link count directly
  - Works for both local/remote parent directory

• Parent link list useful for other reasons:
  - FID-to-path operations (ChangeLog, error messages)
  - Update parent directory entries if migrating FID/inode
  - Rename directory loop checking
  - POSIX lookup-by-FID path permission checks
MDT Internal Consistency

Directory A

Inode X

Inode Y

Directory B

inode+xattrs

Directory entries

© 2011 Whamcloud, Inc.

Online Distributed Coherency Checking for Lustre

© 2011 Whamcloud, Inc.
Current status

• Internal back-references exist today
  - *link* xattr only since 2.0, can add during directory walk

• Prototype implementation of OI Scrub
  - Iterator for ldiskfs inodes written for OSD API
    • Virtual index that references all in-use inodes in filesystem
    • Fast linear inode table traversal with large reads
  - Iterator for DMU objects is part of standard DMU APU
  - Includes on-demand verification of FID during lookup

• Discussing bulk attribute RPC design
Thank You

- Andreas Dilger
  Principal Engineer
  Whamcloud, Inc