Lustre Userspace Server Architecture

Alex Zhuravlev (bzzz)

Andreas Dilger
Lustre Group
Contents

- Current architecture: 1.4 and 1.6
- Problems with current architecture
- Future challenges
- Requirements for new architecture
- New architecture: 2.0+
Current architecture: 1.4 and 1.6
VFS provides

- fs abstraction
- Data operations
- Metadata operations
- Cache (pagecache, icache, dcache)
- Permission checks
- Rename helpers
fsfilt provides

• Transaction API
• Direct IO
• Extended attributes
• quota
• uuids/labels
Problems with this architecture

• Not portable, depends on vfs, ldiskfs
• Lack of features
• Quality
• Amount of work unrelated to Lustre
Quality

- Complex and unstable kernel API
- VFS doesn't suit our needs: fsfilt, tricks and lots of bugs
- Complicated debugging
- Complex automatic testing system
- Hard to find people
Future Challenges

- CMD requires primitive operations
- Scalability
- Resistance to disk failures
- Features: snapshots, data cache
- Complexity of Lustre increases
Requirement for new architecture

• Portability: more OS, more disk fs

• Quality:
  > Clean, stable and documented API
  > No tricks
  > Easy to test and debug

• Scalability

• Features

• Performance shouldn't suffer much
New Architecture

- OSD replaces VFS and fsfilt:
  - MDS/OSS don't depend on VFS anymore
  - With proper OSD Lustre runs anywhere
OSD provides

- Cluster-wide object ID – FID
- Set of primitives to operate on:
  - Data
  - Indexed lookup tables
  - Regular and extended attributes
- Transactions support
- All of this with clean and good API
New architecture: DMU

- Very portable (POSIX)
- Satisfies most OSD needs
- Runs in user space
- Easier to develop and debug
Work Still Needed

• There are number things to develop
• DMU needs some work
• New architecture needs testing