Lustre – A view from the outside

Neil Brown
Kernel Engineer
neilb@suse.com

An external perspective.

- Linux Kernel developer for nearly 20 years
- Employed by SUSE for nearly 14 years
- NFS server maintainer – 2000?? to Jul 2007
- md/RAID maintainer – Feb 2002 to Jan 2016
- LWN.net contributor
- Now: interested in Lustre

Lustre: how it looks to me….

Lustre strengths

- **Modularity**
  - 25 separate kernel modules!
- **Transparency**
  - Extensive low-impact tracing
- **Testability**
  - Comprehensive test suite
  - Multiple fault-injection points

Lustre weaknesses - code

- **Duplicates of Linux functionality**
  - Tracing, Fault injection,
  - Wait event, Resizable hash tables
  - Work queues …
- **Locking and reference counting**
  - atomic_t → refcount_t → kref.
  - mutex → spinlock → RCU/lockless
- **Internal APIs**
  - Unused code
  - APIs designed for user-space (Portals)
  - “lustre” filesystem – used for both server and client

Lustre weaknesses - process

The lustre git tree has commits going back to 1999 (well before git). This is excellent.

The commits themselves are not so excellent, though recent commits are much better than earlier ones.

Two criteria:

1. One change per patch.

- Don’t mix formatting changes with code changes
- Don’t mix independent code changes

2. Explain the change clearly in the commit message

- Nothing in the patch should come as a surprise (this is subjective!)

The Linux kernel is far from perfect here either, but it is a worth goal to focus on.

Lustre: benefits from mainlining.

Practical and Social

Practical benefits from mainlining.

- **Linux internal API changes won’t break Lustre**
  - Anyone changing an API will, at least, notify us.
- **Regular automated testing**
  - Whamcloud already does this a lot
  - Various community members do more, with a complementary focus
  - Random configs, different architectures, assorted static analyses

Social benefits from mainlining

- **A voice at the table**
  - We can ask for interfaces that would be unused without Lustre
  - We can argue against changes that would hurt Lustre
  - We have more visibility with other developers
- **A culture of quality**
  - The kernel community values and encourages quality in multiple ways.
  - These complement and enhance the quality focus that Lustre already has.
  - Having someone external hold you accountable is a good motivator.

Lustre: helping world domination

Adding lustre make Linux stronger

Adding functionality with active users and maintainers is good for Linux. The experience of

- EXT4, XFS, BTRFS etc
- md/RAID, dm/LVM2, DRBD
- slab, slob, slub

shows that duplication doesn't have to be wasteful.

The market is large, healthy competition can help.

Merging communities makes both stronger.

Linux has many active and clever developers.

So does Lustre.

We can learn from each other more easily if our code is in the same place, and our membership is of the same community.

Neither side will swamp the other, but removing barriers will increase opportunities for sharing.

Together we are stronger than we are apart.

Questions?