Linux Lustre client roadmap

The good, the bad, and the ugly
The good happening upstream to backport to OpenSFS

- LU-10467 : `l_wait_event` replaced by wait queues
- LU-8130 : Migration to rhashtables
- LU-9859 : libcfs cleanup
  - cfs signal handling
  - Replace CLASSERT with BUILD_BUG_ON
  - Lustre unique things made standard for linux
- LU-4432 / LU-XXXX kthread handling cleanup
- ldlm rbtree can be replace with kernel one
The good in OpenSFS branch to push upstream

- LU-6401 / LU-6245: UAPI header completed for upstream and OpenSFS. Finished!!!
  - Potential to build tools against linux client.
- LU-9019: tick-less and 64 bit time support almost complete
  - LU-10707: routers upstream doesn’t work 😞
- LU-8703 / LU-7734: SMP rework
  - Patches for upstream need rework but are nearly ready
- LU-10785: xattr breakage upstream – patch series needs one revision
- LU-7004 / LU-9431: lctl set_param –P mostly fixed and udev events
- LU-9667: moving Inet to sysfs *
- LU-8066: sysfs + debugfs port fixes
- LU-8964: use kernel readahead, needs pdata support. Also clio cleanup
- Many IB patches needed for upstream. IB broken upstream
The bad: breakage and fall out

- LU-8066: sysfs + debugfs impacts
  - debugfs is only accessible by root
    - lctl dl uses debugfs devices file – patch exist for this
  - Pools – need solution
  - lov / osc targets – need solution
  - Stats – need solution
  - Sptlrpc + nodemap – only root?
  - Lnet stats - LU-9667 fixes this

- LU-7734: multi-rail enabled tools / lctl don’t work properly with upstream client
The badly confused

- LU-9091 : \texttt{lprocfs\_str\_with\_units\_to\_s64} – \texttt{string\_to\_size()}
- What is need to move away from Lustre debugging code.
  - Linus called it ugly. Viro is not a fan.
- The return of the \texttt{llite\_loop}?
  - See \texttt{cryptoloop.c} for non-standard loop device
  - Better yet fix upstream loop back device
- Libcfs watchdog timer
  - Only used in ptlrpc layer. Can it be replaced with something more standard
- Libcfs pdata infrastructure
  - What is missing upstream that it can’t be used directly?
The bad : Not even started

- LU-8874 : update ko2ibInd to latest RDMA changes
- LU-9680 : Improve user land to kernel space interface
  - LU-8834 : LL_IOC_FUTIMES_3 is too generic
    - Solutions – handle everything on the server side or create a generic syscall for everyone
    - Potentially ladvise ???
  - LU-9667 : Move LNet ioctls to netlink API
    - I/O forwarding friendly
  - LU-6202 : ioctl cleanup
    - Many obsolete ioctls - grab from sysfs
    - ioctl redirect is hated. Everything done with /dev/obd. Possible netlink solution
Just plan Ugly

• LU-8915: No using linux list structures as arguments
  – Lnet_selftest: really needs to be replaced. Piece of garbage
  – Nodemap cleanup. Solution could be netlink

• LU-9855: cleanup obdclass preprocessor code
  – Kill off struct obd_ops. Get ride of ugly macros.

• Sptrrpc, GSS, and checksum code integration
  – LU-8602: GSS support with newer kernel
    • Work with sunrpc gss maintainer to create common framework
      – Could merge libcfs crypto into the gss framework.
  – LU-10472: T10P checksum
  – Move cfs_crypto_hash_type and cfs_crypto_hash_algo to lustre core from libcfs
    • Use string names instead of enum cfs_crypto_hash_algo