

Upstream efforts progress

Report of milestones, May 2026

James Simmons

Storage Systems Engineer

Oak Ridge National Laboratory

ORNL is managed by UT-Battelle LLC for the US Department of Energy

Project summary

- Heavily demanded project
- large community support
 - Aeon
 - ORNL
 - HPE
 - Lustre collective
- Path to submission ?
 - One giant patch
 - Revive native lustre client with git history
 - None of the above
 - `linux/Documentation/filesystems/adding-new-filesystems`

What is needed for upstreaming

- Use modern VFS interfaces
 - Folios which is in the works
 - Explain why we don't use netfs
 - Don't use deprecated interfaces *
- Be testable
 - We have maloo + ktest (<https://tim-day-387.github.io/upstream-patch-review/#>)
 - Run at kernel CI (<http://www.kernelci.org>)
 - Integrate into xfstest
- Provide Documentation
 - linux/Documentation/filesystems/lustre
 - Submit our man pages

What are the deprecated interfaces?

- Change BUG() and BUG_ON() to WARN()
 - They might hate our LBUG handling. You can do panic_on_warn.
- Open code arithmetic for allocations
 - Use kcalloc_array(), kcalloc(), array_size(), struct_size(), size_mul(), size_add()
 - Our OBD_ALLOC macros might have to be removed. Use CONFIG_MEM_ALLOC_PROFILING
 - Also need to use new kcalloc_obj(), kcalloc_flex().
 - Use only strncpy(). No more strcpy() or strncpy()
 - Need to hash "%p". "%px" can be acceptable.
- Can't use /proc anymore
 - Only a handful left in proc. Rest in /sys/fs/lustre and /sys/kernel/debug/{lustre/lnet}
 - Also available using Netlink / YAML. lctl get_param family uses Netlink / YAML as backend

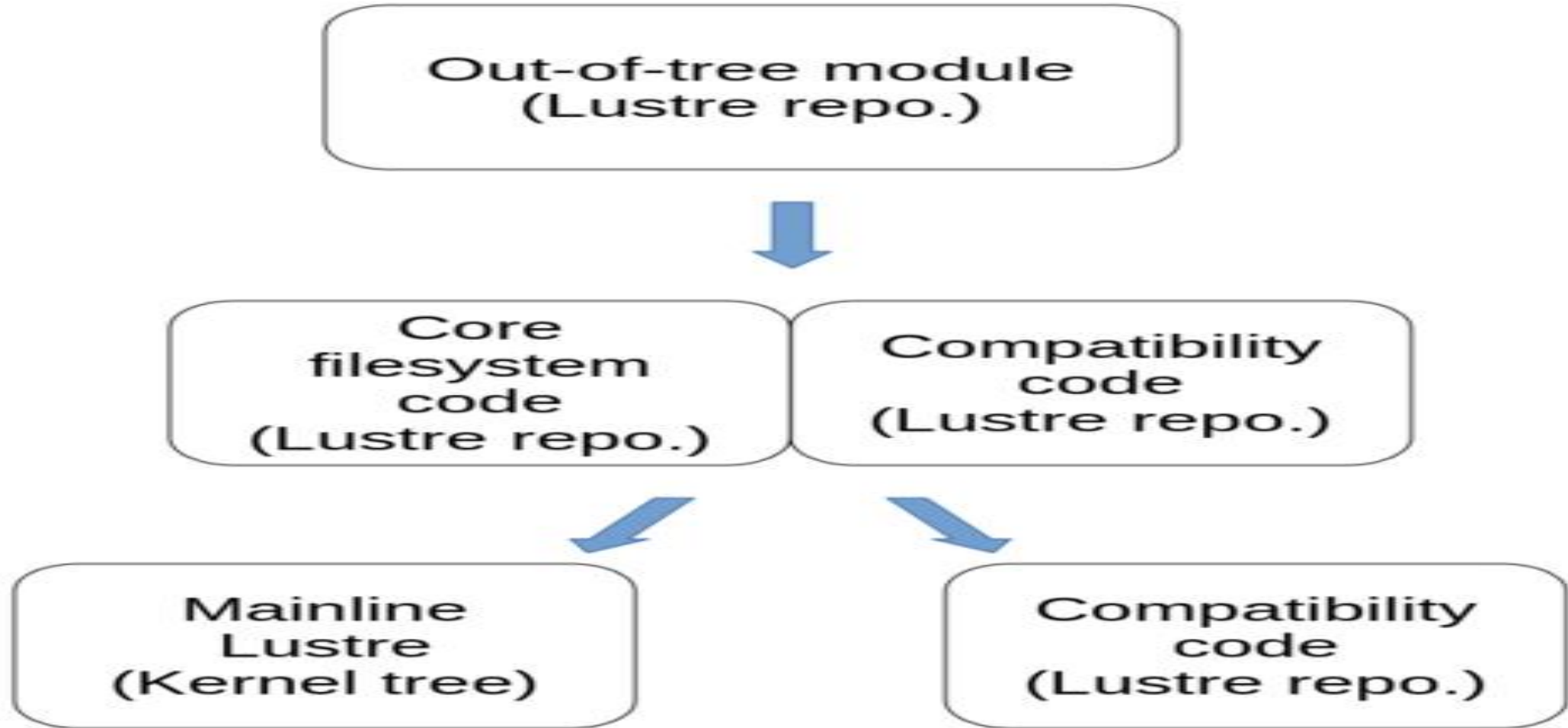
Coding work for submission

- Enforce kernel code style mostly done
- Convert to kernel-doc comments nearly done
- Enable building Lustre monolithically into kernel is done
- No longer use autotools for kernel modules
- Split kernel compat code out of libcfs. Compat abstraction for drop-in support.
- IPv6 support
 - Mostly done but more to go. Most code changes needed for complete support
- Transition to p2p dma from GPUDirect should be complete
 - Needed so we can support newer infiniband APIs
- Folio support is in the works. Please consider using RHEL10. 10x or more performance improvements.

Side products of this work

- Separate mount targets. Please use `mount -t lustre_tgt`
- Setup `cpu_pattern` with `lnet.ko` instead of `libcfs.ko`
- New debug work
 - Looking to use new upstream memory leak framework (`/proc/allocinfo`)
 - Embedded BTF in lustre modules
 - LNet eBPF tracing
- Profiling changes (stats etc)
 - YAML Netlink based instead of file open in `proc` or `debugfs`
 - Python, go, rust wrappers
 - `lljobstats` and `lljobstat` will be examples

Tale of two repos



OpenSFS source tree setup

- Use Lustre that comes with OS kernel
 - Only osd-wbcfs and ksocklnd going up at first.
 - Want to use osd-zfs, osd-lldiskfs and lnd drivers?
 - `export CBI_WITH_O2IB; export CBI_WITH_ZFS; $(src tree)/copy-builtin /usr/src/linux-source-tree`
 - Option for distros like Ubuntu with ZFS or HPE / Cray to install kfilnd driver into SuSE kernel
 - Build much lighter weight lustre kernel rpms using weak-module support
 - `./configure --disable-modules [--enable-client] [--enable-servers]`
- Use latest and greatest upstream Lustre version
 - Remember OpenSFS tree will no longer have most of the Lustre kernel code.
 - `$(src tree)/copy-native /usr/src/linux-source-tree; $(src tree)/configure ....`

Special thanks

- Native client support is a true community effort
 - Neil Brown (ex SuSE)
 - Arshad Hussain (Aeon computing)
 - Shaun Tancheff (HPE)
 - Timothy Day (Lustre Collective)
 - Whamcloud team

Conclusion

- Preparation for working with Linux tree is progressing
- Important milestones are within reach
- New features and improvements from this work that benefits the user community
- Healthy momentum to support this work
- So close to the finish line

Acknowledgments

This work was performed under the auspices of the U.S. DOE by Oak Ridge Leadership Computing Facility at ORNL under contract DE-AC05-00OR22725.