Lustre Upstreaming

Timothy Day (AWS)

James Simmons (ORNL)

Lustre Overview

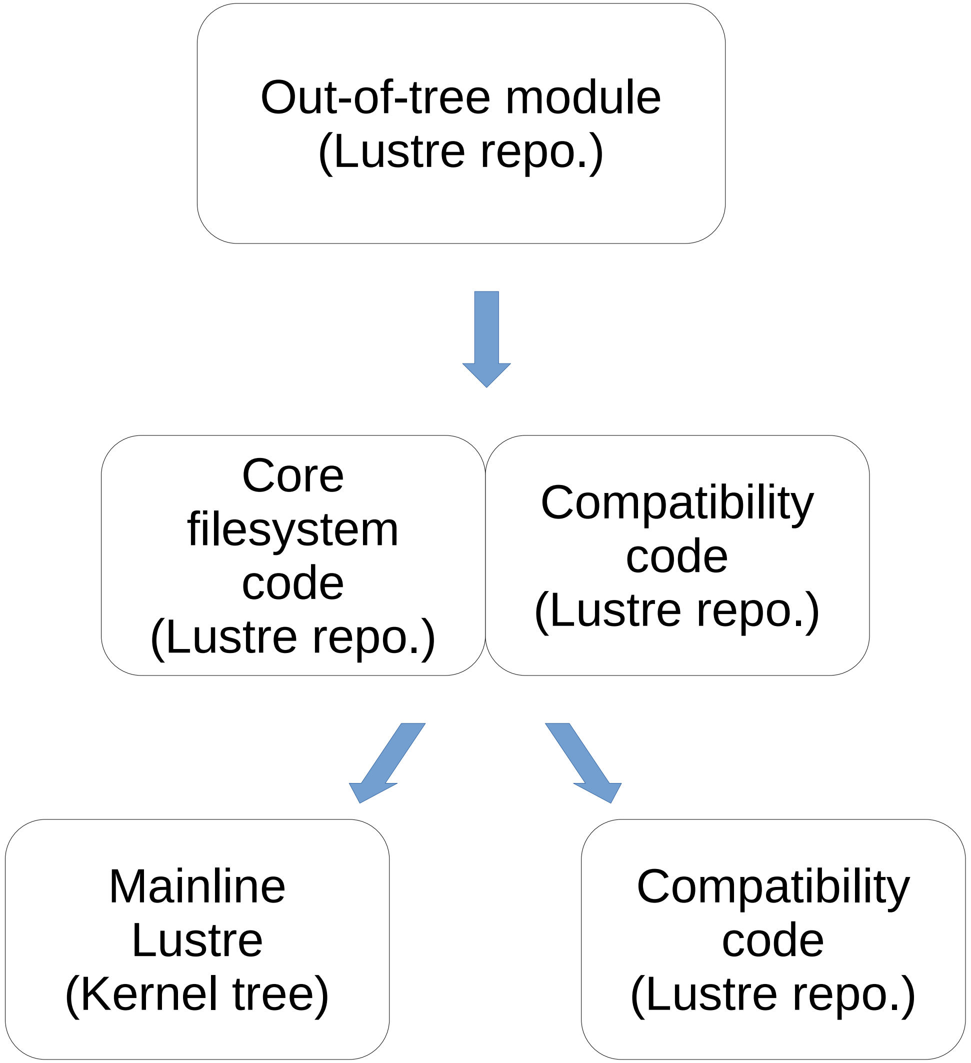
- High performance parallel filesystem
- Used most often in HPC/AI
- Single client and server implementation, both in kernel-space
- Stable wire/disk formats and good interoperability story

History

- Upstreaming is a decade long project
- Removed from staging 5+ years ago (bad fit)
- Development out-of-tree has continued at a healthy pace (100s of patches)
- Time to restart our journey towards mainline!

Development Model

- Evolve from an out-of-tree model to an upstream-first model
- Technical changes
- Process changes
- Aim to have the major changes completed in the next year


Out-of-tree module (Lustre repo.)

Core filesystem code (Lustre repo.)

Compatibility code (Lustre repo.)

Mainline Lustre (Kernel tree)

Compatibility code (Lustre repo.)

Open Questions

- How do we build confidence that Lustre upstream will work?
- Should we start with client only? Or client/server at the same time?
- What projects have successfully undergone a similar transition?

How do we build confidence?

- Staging did not turn out well…
- Lustre community needs a stronger upstream focus
- What else?

Client first? Server later? Or both?

Client Only

- Pros
  - Much closer to completion
  - Less code
- Cons
  - Harder to test (requires dedicated Lustre server)

Client and Server

- Pros
  - Easier to test (client/server co-located on same kernel)
  - Avoid work of splitting client/server
- Cons
  - Server is much larger
  - Must decide between patched ext4 or running in-memory

Who can we learn from?

What projects have successfully undergone a similar transition?

Future Work

We maintain a project page with up-to-date status, notable events/links, and a in-depth explanation of our evolving development model:

https://wiki.lustre.org/Lustre_Upstreaming_to_Linux_Kernel

Client branch:

https://github.com/jasimmons1973/lustre