Lustre + Linux

Putting the house in order

James Simmons
ORNL
The fork in the road

• State of current Lustre upstream client
  – EMC started the work spring 2013
  – Based on Lustre commit faefbfc03 (~ 2.4 version)
  – No new features added
  – Most patches address style issues
  – Removed wrappers
  – Gripes about architectural design
Trip down Intel drive: Syncing with upstream client

• Participants:
  – Historically Peng Tao from EMC
  – James Simmons from ORNL
  – John Hammond and Dmitry Eremin from Intel
  – Frank Zago from Cray

• Goals:
  – Remove technical debt
  – Meet kernel coding styles
  – Remove wrappers
  – Support newer kernels
Master renovations so far for client code

• Changes to master since Lustre upstream version got merged
  – Support up to 3.15 kernels
  – Port of proc fs handling to seq_file
  – All cfs_* wrappers are gone
  – Liblustre removal
  – Kernel coding styles enforced with new code
What about the other side of town?

• All code cleanup for clients was applied to server side code

• LU-20 : Goal of no more patching the server side
  – Patching only needed for Idiskfs support
  – LU-3406 : raid5 mmp unplug patch
  – LU-684   : Use dm flakey to test fail over
  – Drop the rest of the patches since they are upstream
  – Only need to patch latest kernel with LU-3406 for testing. Hopefully proper upstream solution will be done

• LU-6030: Idiskfs patch cleanup

• LU-6220: Push most Idiskfs patches upstream
What needs to be done to be up to code for upstream merger?

• All work tracked under LU-6215

• Needs to be in style.
  – LU-5478 : typedef removal
  – LU-6210 : Use C99 structure initializers
  – LU-6303 : unsigned/signed fixes
  – LU-5710 : Grammar and spelling fixes
  – LU-5829 : Remove unneeded exports
  – LU-6142 : Kernel indenting of code a.k.a tabathon
  – LU-6302 : kernel sparse cleanup
• Technical debt is still left

• LU-6245 : libcfs train wreck
  – User land and kernel header entanglement
  – User land and kernel source code entanglement
  – No more using libcfs.h everywhere.
  – Abstractions still left

• LU-6401 : lustre header train wreck
  – User land and kernel header entanglement
  – Use of internal kernel headers in user land

• About 6 months of work for master improvements and merging upstream
Code violations

• Upstream reviews dislike some code choices
  – Job ID handling
  – Random seed handling
  – General dislike of procfs (LU-5030)
  – Use trace point instead of current debug system
  – Others not named

• After syncing we need to address these issues
Goals going forward

- Break up the moving out of staging process
  - Move libcfs/LNet out of staging tree first.
    - Should be done by next LUG.
  - Move Lustre client out of staging
    - Address all design issues
    - Should be done by 2017 LUG
  - Merge in remain server side support
Conclusion

• Work left
  – Style issues should be ready by end of 2.8
  – The libcfs layer needs a lot of work
  – Lustre header cleanup
  – Kernel code style updates will be big update.

• Need to work as a community to push work upstream.

• We can’t forget support for latest kernels.