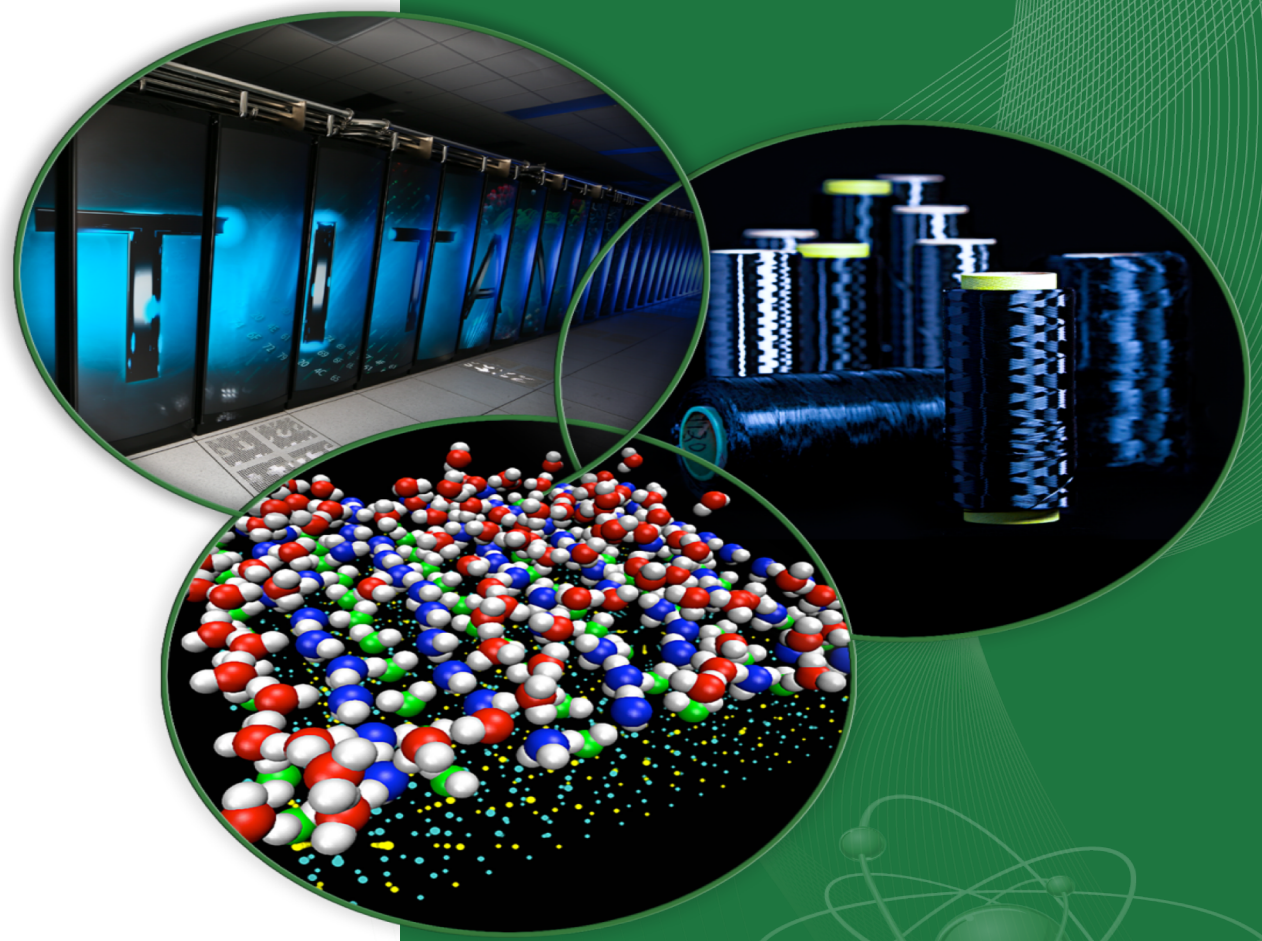
Flexible Lustre management

Making less work for Admins



How do we know Lustre condition today

- Polling proc / sysfs files
  - The knocking on the door model
  - Parse stats, rpc info, etc for performance deviations.
- Constant collection of debug logs
  - Heavy parsing for common problems.
- The death of a node
  - Have to examine kdumps and /or lustre dump

Origins of a new approach

- Requirements for Linux kernel integration.
  - No more proc usage
  - Migration to sysfs and debugfs
  - Used to configure your file system.
  - Started in lustre 2.9 and still on going.
- Two ways to configure your file system.
  - On MGS server run lctl conf_param …
    - Directly accessed proc seq_files.
  - On MSG server run lctl set_param –P
    - Originally used an upcall to lctl for configuration
    - Introduced in Lustre 2.4 but was broken until lustre 2.12 (LU-7004)
  - Configuring file system works transparently before and after sysfs migration.

Changes introduced with sysfs / debugfs migration

- sysfs has a one item per file rule.
- Complex proc files moved to debugfs
- Moving to debugfs introduced permission problems
  - Only debugging files should be their.
  - Both debugfs and procfs have scaling issues.
- Moving to sysfs introduced the ability to send uevents
  - Item of most interest from LUG 2018 Linux Lustre client talk.
  - Both lctl conf_param and lctl set_param –P use this approach
    - lctl conf_param can set sysfs attributes without uevents. See class_modify_config()
  - We get life cycle events for free
  - udev is now involved.

What do we get by using udev ?

- Under the hood
  - uevents are collect by systemd and then processed by udev rules
  - /etc/udev/rules.d/99-lustre.rules
  - SUBSYSTEM=="lustre", ACTION=="change", ENV{PARAM}=="?*", RUN+="/usr/sbin/lctl set_param '$env{PARAM}=$env{SETTING}'"
- You can create your own udev rule
  - http://reactivated.net/writing_udev_rules.html
  - /lib/udev/rules.d/* for examples
  - Add udev_log="debug" to /etc/udev.conf if you have problems
- Using systemd for long task.
  - ACTION=="change", SUBSYSTEM=="lustre", TAG+="systemd", ENV{SYSTEMD_WANTS}="my-lustre.service"
  - [Service] Type=oneshot

    ExecStart=/usr/bin/echo 'lustre tunable changed"

udev tools

- A database for udev configurations.
  - Best keep secret.
  - If you have many settings to configure just add them to /lib/udev/hwdb.d/lustre and build your binary db.
  - Read by system.d. Also can be queried.
  - man hwdb
- udevadm is your friend
  - Want to know what your attributes values are:
    - udevadm info -a -p /sys/fs/lustre/osc/lustre-OST0001-osc-ffff8d76a2646000
  - Examine uevents directly:
    - udevadm monitor -s lustre
    - For file system tunables timestamps are include (2.13)

Future uevent support for Lustre

- Only tunable are handle as of 2.12
- LU-10756 list suggested items of interest
- Lustre import life cycle
  - https://review.whamcloud.com/#/c/31407 (Incomplete)
  - Evictions
  - Recovery status
- obd device health checker
- Target mounts and state such as degraded.
- LBUGS
- Self healing file system
- Add polling to sysfs files - sysfs_notify()
- Other suggestions welcomed

Future LNet uevent support

- LNet dynamic discover landed for 2.11 - LU-9480
- LNet network health landed to 2.12 - LU-9120
- LNet sysfs support for 2.13 release - LU-9667
- Events together provide infrastructure to support uevents for:
  - LNet NID and Net notifications
  - Network timeouts.
- Possible management of LNet using udev rules.
- Other suggestions welcomed.

Writing applications for udev usage

- Udev helper library – libudev
  - Simply sysfs access in some cases
  - udev_monitor_*() to detect events
    - Xorg video device plugins
- D-Bus is aware of udev events
  - Example is USB stick insertion (udisk) and network management
  - https://www.freedesktop.org/wiki/Software/dbus/
  - Applications need libdbus
  - /usr/share/dbus-1 example of configurations
- Can use D-Bus as a message bus
  - Support of interprocess communication
  - D-Bus can communicate between different nodes.

Impact of debugfs

- Complex proc files moved to debugfs
  - Not the purpose of debugfs.
  - Both proc and debugfs files don’t scale well.
    - https://lwn.net/Articles/406975 (taskstats)
  - Only root can access it by default.
  - Good news we don’t have many debugfs files
- Replace with Netlink
  - Designed to replace ioctls – RFC 3549
  - Scales far better
  - Support multi packets so no size limitation.
  - Very flexible with API changes.
  - Container aware
  - Should work better with I/O forwarding systems.

LNet Netlink implementation

- library interface being developed for liblnetconfig : LU-9680
  - Simplify working with libnl API
  - Enables vendor specific hooks for their products
- Rewrite of LNet selftest to use Netlink
  - Kernel module is unacceptable by kernel standards
  - Current LNet self test is very fragile
  - Netlink version could be used to monitor live traffic
- LNet User Defined Selection Policy : LU-9121
  - Will land to 2.13
  - Currently ioctl based which is inflexible in the long run

Lustre Netlink implementation

- Use Netlink for device listing with ‘lctl dl’ : LU-8066
  - Non root permission problem with ‘devices’ debugfs file
  - Could be very large which is expensive with seq_files
  - Ioctl exist but is really awful.
- Use Netlink for stats : LU-11850
  - Far more scalable.
  - Will be transparent to user with lctl get_param.
    - New better interface will be create using YAML
  - Can be made configurable – only return stats of interest
    - Many times proc stat files have changes
    - Could control what stats are even collected!! Memory saver..
  - Different Netlink groups exist for stats.
    - No need to open many proc files to gather stats.

Other Lustre Netlink uses

- Fixing lustre ioctl handling : LU-6202
  - Use struct obd_ioctl that is similar to Netlink headers
  - Lustre uses ioctl redirection which is frowned on by kernel development community.
- Replace KUC with netlink : LU-7659
  - KUC is used by HSM but was designed with other uses in mind.
  - Currently it uses a pipe which is not recommend.
    - https://www.linuxjournal.com/article/8110
- Use Netlink to implement external HSM coordinator : LU-10968
- Suspect other uses in the future

Conclusion

- Greater Lustre + LNet state awareness
- Greater flexibility with Netlink
- Better performance and scalability

- Questions?