Lustre Tuning and Advanced LNET Configuration

Jesse Hanley
Rick Mohr
Jeffrey Rossiter
Sarp Oral
Michael Brim
Jason Hill
Neena Imam
Outline of Presentation

• Kernel module options
  – Tuning recommendations from Lustre manual
  – Tuning recommendations from OLCF experience

• Multi-rail LNET configurations

• Complex LNET routing configurations

• Lustre client, router, and server tuning options
Kernel Module Tuning

- The tunings applicable to a host depends on that host’s role. A client will be tuned differently than a router, which would be tuned differently than a server.
- There are some tunings that are applicable to each of these roles.
What modules and what options?

• The main modules to look at are Inet, and the transport layer (most commonly ko2iblnd or ksocklnd). Other modules we’ll look at are libcfs and ptlrpcd.

• How can we pass options to these modules?
  – First, check what parameters can be passed using `modinfo <module_name>`, e.g. `modinfo Inet`.
  – These parameters are listed within the appropriate module.conf file under `/etc/modprobe.d/`:
    ```none
    options module_name param=option param2=option2
    Example: options ko2iblnd peer_credits=63
    ```
What are good values to use?

• Often, the default values for parameters are reasonable to use.

• The following values are from the Lustre documentation and OLCF experimentation.

• Unfortunately, there may need to be a trial and error period to find appropriate parameter values for different networks.
Module: Inet

- Arguably the most important Inet parameter is the *networks* or *ip2nets* param.
  - The *networks* parameter maps a network interface to an LNET subnet
    - Format: options Inet networks=LNET(interface),LNET(interface),...
    - Example: options Inet networks=tcp0(eth1),o2ib(ib0)
  - This can be a problem to manage with large networks with many hosts. The *ip2nets* parameter allows for a single configuration file across the network
    - Format: Each node identifies the locally available networks based on the listed IP address patterns that match the node's local IP addresses.
    - Example: options ip2nets="o2ib0(ib0) 10.10.[0-1].*"
      This would put all clients on the 10.10.{0,1}.* networks on o2ib0
Module: Inet routing

- Routing is defined through a module parameter to Inet as well.
  - The *routes* parameter specifies a semi-colon separated list of router definitions.
    - routes=dest_lnet [hop] [priority] router_NID@src_lnet; \ dest_lnet [hop] [priority] router_NID@src_lnet
  - An alternative syntax consists of a colon separated list of router definitions:
    - routes=dest_lnet: [hop] [priority] router_NID@src_lnet \ [hop] [priority] router_NID@src_lnet
  - Example:
    - options Inet networks="tcp0(eth0)" routes="o2ib0 1 10.10.10.2@tcp0; o2ib0 1 10.10.10.3@tcp0"
Routing example

- Setup:
  - one TCP client,
  - one router (TCP & Infiniband connections)
  - servers (MGS, MDS, OSS) on an InfiniBand fabric.

- The LNET router has two NIDs:
  - 192.168.1.2@tcp0
  - 10.13.24.90@o2ib0

- The lustre.conf file for the client includes:
  - options lnet networks="tcp0(eth0)" routes="o2ib0 192.168.1.2@tcp0"

- On the router nodes:
  - options lnet networks="o2ib0(ib0),tcp0(eth0)" forwarding=enabled

- On the server nodes:
  - options lnet networks="o2ib0(ib0)" routes="tcp0 10.13.24.90@o2ib0"
Remaining LNET routing parameters

- auto_down
  - Default Value = 1

- avoid_asym_router_failure
  - Default value: disabled

- live_router_check_interval
  - Default value: 60

- dead_router_check_interval
  - Default value: 60

- router_ping_timeout
  - Default value: 50

- check_routers_before_use
  - Default value: off
Module: libcfs

• Module params:
  – libcfs_console_ratelimit
  – libcfs_console_max_delay
  – libcfs_console_min_delay
  – libcfs_panic_on_lbug
  – cpu_npartitions
    • Ex: options libcfs cpu_npartitions=4
  – cpu_pattern:
    • Ex: options libcfs cpu_pattern="0[0-3] 1[4-7] 2[8-11] 3[12-15]"

• Examples of cpu partitioning:

Module: ptlrpcd

- max_ptlrpcds
  - options ptlrpcd max_ptlrpcds=32

- ptlrpcd_bind_policy
  - options ptlrpcd ptlrpcd_bind_policy=3
Module: ko2ibInd

- Option “timeout”:
  - Suggested value: 100

- Option “credits”:
  - Suggested value: 2560

- Option “peer_credits”:
  - Suggested value: 63

- Option “concurrent_sends”:
  - Suggested Value: 63

- Option “fmr_pool_size”:
  - Suggested value: 1280

- Option “fmr_flush_trigger”:
  - Suggested Value: 1024

- Option “ntx”:
  - Suggested value: 5120
Module: ksockInd

- Option “sock_timeout”:  
  - Suggested value: 100
- Option “credits”:  
  - Suggested value: 2560
- Option “peer_credits”:  
  - Suggested value: 63

- Check /proc/sys/Inet/peers for indications of queued send requests
Client Tuning

- lctl set_param osc.*.checksums=0
- lctl set_param timeout=600
- lctl set_param at_min=250
- lctl set_param at_max=600
- lctl set_param ldlm.namespaces.*.lru_size=2000
- lctl set_param osc.*OST*.max.rpc_in_flight=32
- lctl set_param osc.*OST*.max_dirty_mb=64
Server Tuning

- `lctl set_param timeout=600`
- `lctl set_param ldlm_timeout=200`
- `lctl set_param at_min=250`
- `lctl set_param at_max=600`
- OSS:
  - `lctl set_param obdfilter.*.read_cache_enable=1`
  - `lctl set_param obdfilter.*.writethrough_cache_enable=1`
Summary

• Common modules
  – Inet, libcfs, ptlrpcd, ko2ibInd/ksockInd
  – What and how to tune
  – LNET routing

• Client tuning

• Server tuning
Resources

• Lustre Software Manual

• Jason Hill - “LNET Configuration”

• “LNET Router Resiliency and Tuning”

• “Manage Lustre for the Cray Linux Environment”

• Doug Oucharek – “Taming LNET”
Acknowledgements

This work was supported by the United States Department of Defense (DoD) and used resources of the DoD-HPC Program at Oak Ridge National Laboratory.