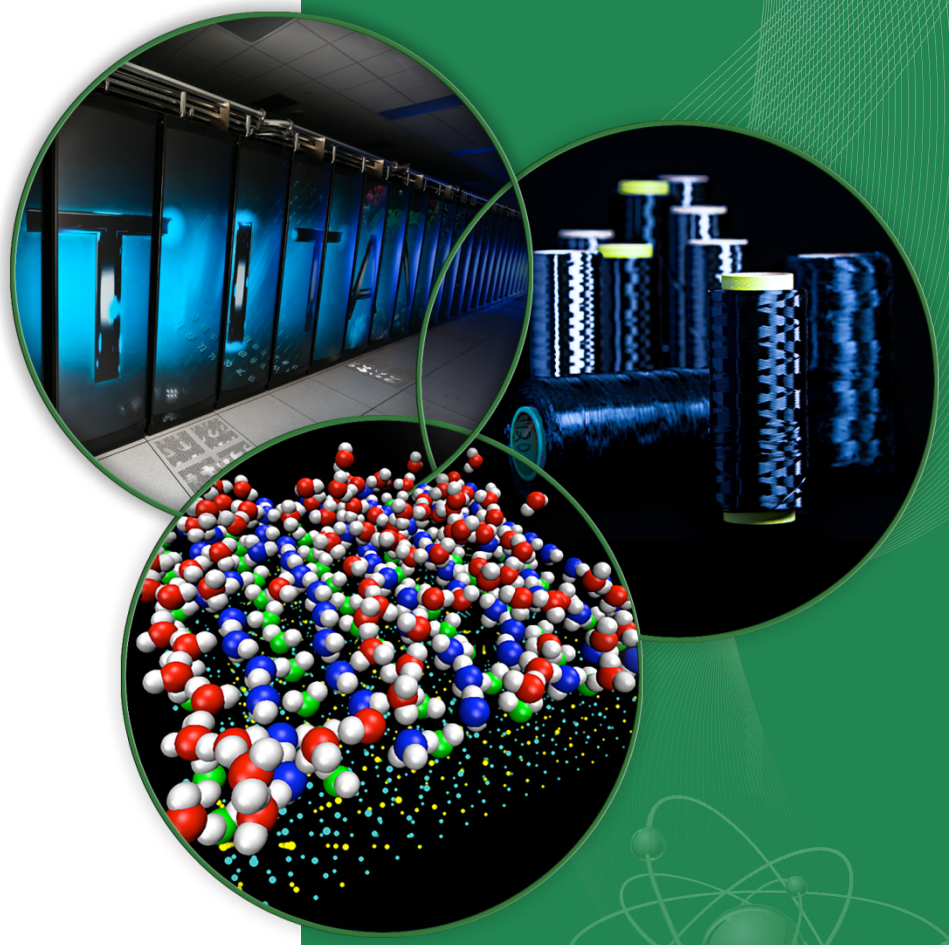
LNET Configuration

Jason Hill

Storage Team Lead

OLCF, HPC Operations

3-3-2015



Overview

- Why we need LNET
- What are parameters to module?
- What are sane defaults?
- Scenarios
  - IB/IB
  - IB/Gemini
- Questions

Why do we need LNET

- LNET is an overlay on an underlying transport mechanism
- Abstraction allows for common parameters to be set for different transports.
- Still need to configure the transport appropriately
- Allows Lustre level flow control
- Allows for peer health detection

LNET Peers – who are the people in your neighborhood?

- Peers are direct communication paths
- Server-> client
- Server-> router
- Router-> client
- Number of direct peers is important
- Affects memory utilization by several modules

What modules are we looking at?

- lnet
- ko2iblnd
- kgnilnd
- ksocklnd
- Others that I'm not an expert in

Today we focus on LNET

- The other modules are important though
- Not directly related, but have impact
- /sbin/modinfo is your friend
- Quoting can be problematic, pay attention
- Trial and error unfortunately
- Inserting comments is important (required!)
- modprobe's evolve over time, sometimes parameters need updating or removal
- First plug for configuration management

What are the parameters

- 3 types of nodes to consider
  - Lustre Client
  - Lustre Server
  - LNET Router
- All tuned differently
  - Some commonalities
- Let's take a look..

Types of parameters

- Specific to all node types
- Specific to clients only
- Specific to servers only
- Specific to routers only

Specific to all node types

- ip2nets/networks
  - Define what LNET Network/IP Range and interface to use
  - options lnet networks=tcp0(eth0),o2ib(ib0)
  - options lnet 'ip2nets="tcp0(eth0) 192.168.0.[2,4]; \
    tcp0 192.168.0.*; o2ib0 132.6.[1-3].[2-8/2]"'
- Two parameters, same meaning

What's the difference?

- LNET chooses best route based on hop count
- What if you have common modprobe.conf across servers and clients?
  - That's where the ip2nets is far more handy
  - In my opinion anyway!
- options lnet 'ip2nets="tcp0(eth0) 192.168.0.[2,4]; \ tcp0 192.168.0.*; o2ib0 132.6.[1-3].[2-8/2]"'
- Servers are at 192.168.0.[2,4], clients are on same / 16 TCP network

Specific to routers

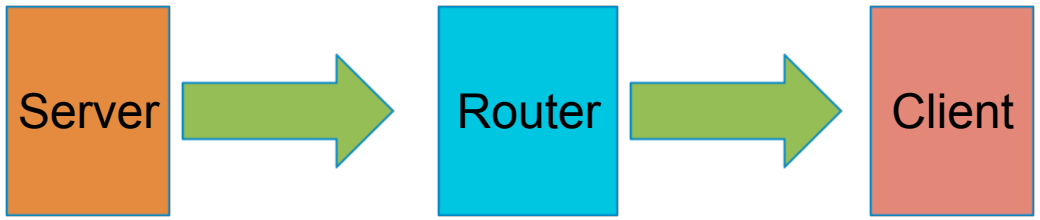
- tiny_router_buffers – size of buffer for messages of 0 size
- small_router_buffers – size of buffer for messages of 1 page in size
- large_router_buffers – size of buffer for messages >1 page in size
- peer_buffer_credits – how many messages queued per peer

Router Buffers

- Why are they important?
- Allow for messages to queue
  - Backlog on one side of the router
  - No more free peer_credits
- Need a bucket for non-data payload RPC's to not get clogged up with data RPC's
- More about backlogs in a bit…

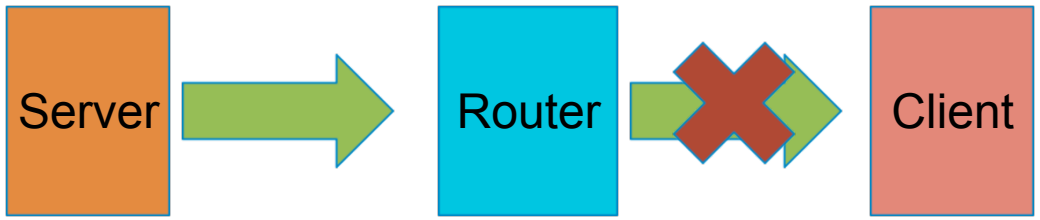
Specific to Servers/Clients

- Avoid_asym_router_failure – mark router down if it can't pass traffic "through" it



Specific to Servers/Clients

- Avoid_asym_router_failure – mark router down if it can't pass traffic "through" it



Specific to Servers/Clients

- Check_routers_before_use – does the router respond to ‘lctl ping’ before sending it traffic?
- {live,dead}_router_check_interval
  - Every X seconds send a lctl ping to determine if router is still up
  - Every Y seconds send an lctl ping to determine if router is back up

Here's where it gets specific

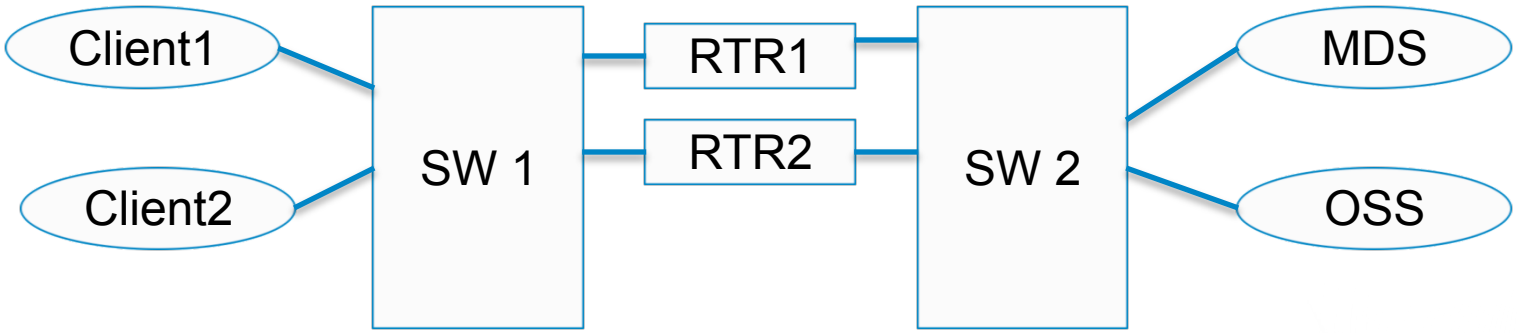
- Lustre has IB module called ko2iblnd.
- Many parameters that are important
- Depending on your routing (or lack) these are simple or complex

Ko2iblbd module parameters

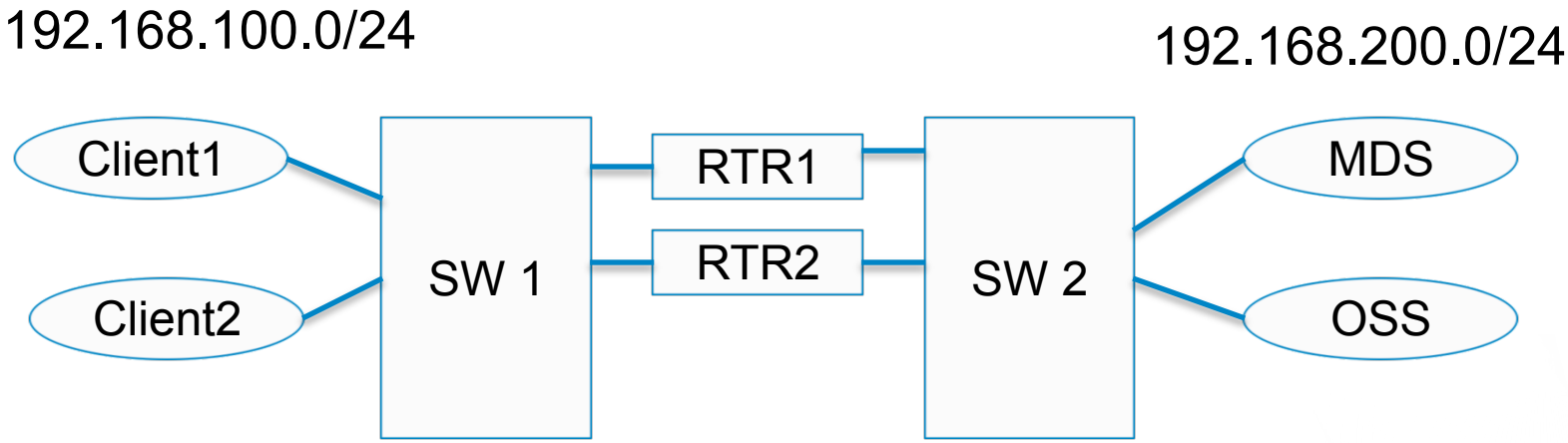
- Ntx – number of message descriptors per pool
- Credits – concurrent sends
- Peer_credits – concurrent sends per peer
- Concurrent_sends – size of work queue
- Fmr_pool_size
- Fmr_cache
- Fmr_flush_trigger
- Timeout – set to greater than timeout of underlying transmission

Simple routing example

- IB -> IB



Simple routing example



Simple routing example

- IP Address Assignments
  - Client 1: 192.168.100.10
  - Client 2: 192.168.100.11
  - Router1 (SW1): 192.168.100.1
  - Router2 (SW1): 192.168.100.2
  - Router1 (SW2): 192.168.200.1
  - Router2 (SW2): 192.168.200.2
  - MDS: 192.168.200.10
  - OSS: 192.168.200.11

Client configuration

- /etc/modprobe.d/lnet.conf

options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet networks="o2ib1(ib0)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

routes="o2ib0 192.168.200.[1-2]@o2ib1"

Router Configuration

- /etc/modprobe.d/lnet.conf

```
options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet forwarding="enabled" tiny_router_buffers=8192 \
small_router_buffers=131072 large_router_buffers=4096 \
peer_buffer_credits=64
```

Server configuration

- Assume IB connected storage for ib0 and ib1

/etc/modprobe.d/lnet.conf

```
options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet networks="o2ib0(ib2)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

routes="o2ib1 192.168.100.[1-2]@o2ib0"
```

Server configuration gotcha?

- But wait – does the MDS have 3 IB cards?
  - Likely not. It needs a different configuration

```
options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024


options lnet networks="o2ib0(ib0)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

routes="o2ib1 192.168.100.[1-2]@o2ib0"
```

- How can we get around this?
  - udev rename the device
  - Scripting

Server configuration

- Assume IB connected storage for ib0 and ib1

/etc/modprobe.d/lnet.conf

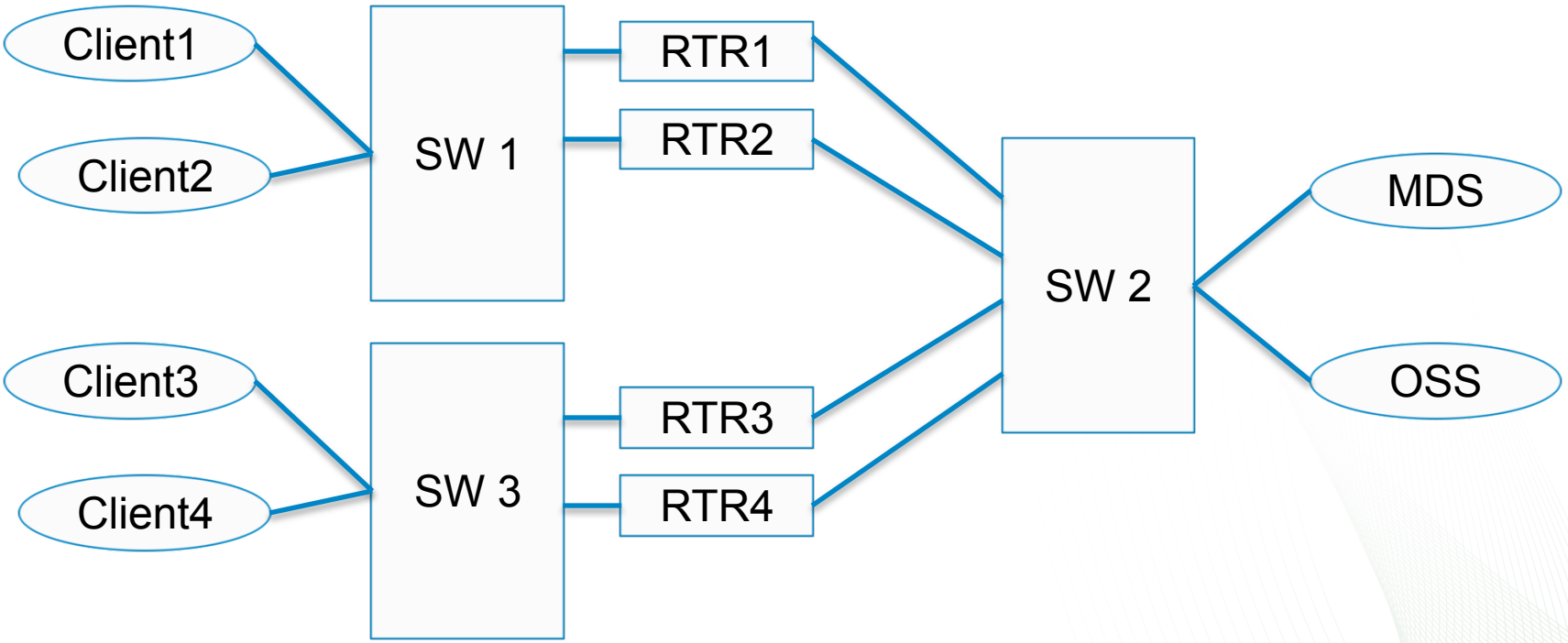
options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet networks="o2ib0(ib2)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

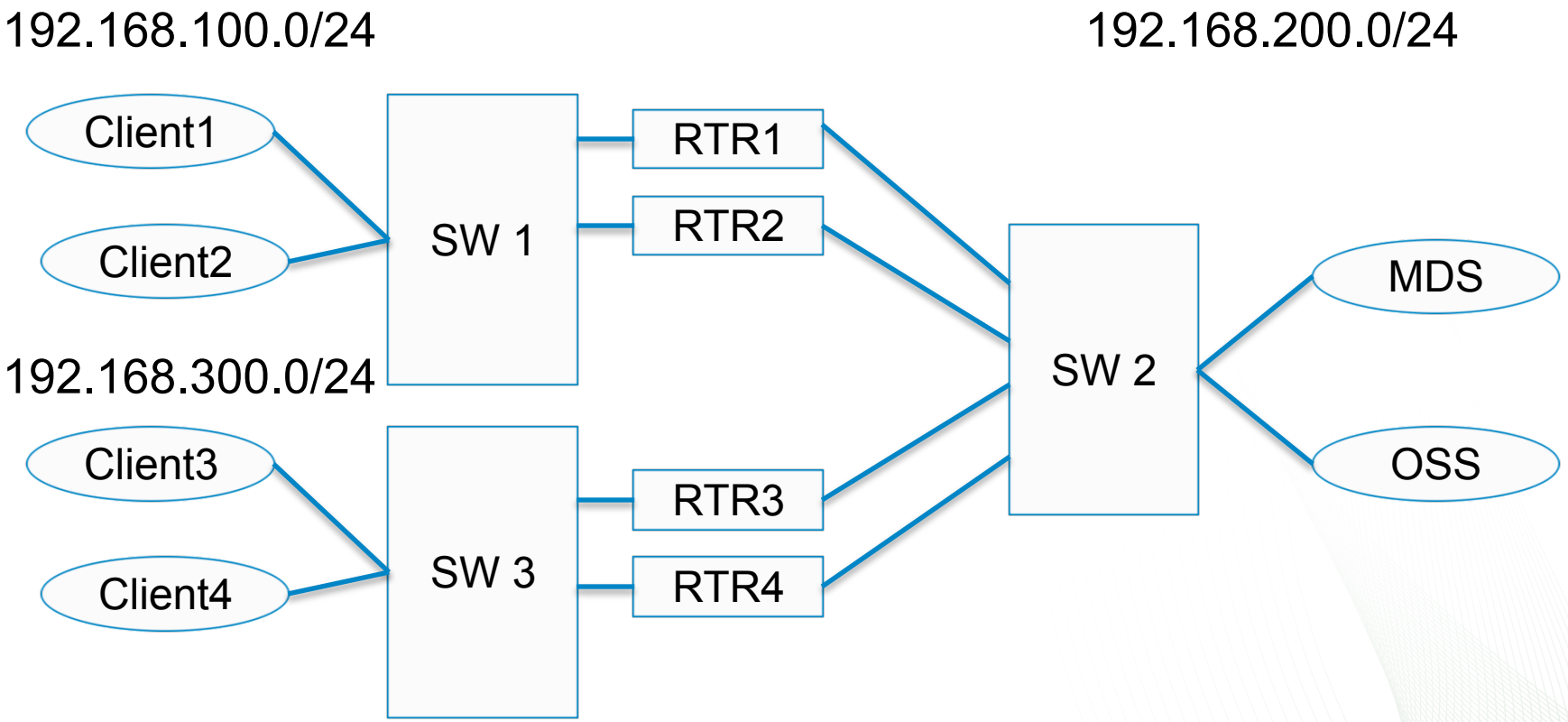
routes="o2ib1 192.168.100.[1-2]@o2ib0"

Complex Configuration

- Still looking at IB <-> IB



Complex Configuration



Complex Configuration

- IP Address Assignments
  - Client 1: 192.168.100.10
  - Client 2: 192.168.100.11
  - Client 3: 192.168.300.10
  - Client 4: 192.168.300.11
  - MDS: 192.168.200.10
  - OSS: 192.168.200.11
  - Router1 (SW1): 192.168.100.1
  - Router2 (SW1): 192.168.100.2
  - Router1 (SW2): 192.168.200.1
  - Router2 (SW2): 192.168.200.2
  - Router3 (SW3): 192.168.300.3
  - Router4 (SW3): 192.168.300.4
  - Router3 (SW2): 192.168.200.3
  - Router4 (SW2): 192.168.200.4

Client Configuration

- Client Cluster 1 (clients 1 and 2)

options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet networks="o2ib1(ib0)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

routes="o2ib0 192.168.100.[1-2]@o2ib1"

Client Configuration

- Client Cluster 2

options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet networks="o2ib2(ib0)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

routes="o2ib0 192.168.300.[3-4]@o2ib2"

Router Configuration

options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024

options lnet forwarding="enabled" tiny_router_buffers=8192 \
small_router_buffers=131072 large_router_buffers=4096 \
peer_buffer_credits=64

Server Configuration

/etc/modprobe.d/lnet.conf

```
options ko2iblnd timeout=25 peer_credits=63 concurrent_sends=63 \
fmr_pool_size=1280 pmr_pool_size=1280 fmr_flush_trigger=1024


options lnet networks="o2ib0(ib2)" check_routers_before_use=1 \
dead_router_check_interval=50 live_router_check_interval=100 \
avoid_asym_router_failure=1 \

routes="o2ib1 192.168.100.[1-2]@o2ib0 \
        o2ib2 192.168.100.[3-4]@o2ib0"
```

Some best practices

- Keep central management of IP addresses
  - DNS or wiki page or Excel spreadsheet
- Keep central management of LNET id's in use
  - Wiki, excel, Google Doc…
- Make patterns where possible (2am Stupidity)
- Monitoring LNET is a good idea

Headaches

- Routes line can get too long for module parameter
- source/arch/x86/include/asm/setup.h:#define COMMAND_LINE_SIZE 2048
- So if we're over 2048 characters including spaces the module load fails.
- 435 LNET routers on Titan alone (several other routed clusters in the OLCF)
- We need a way around this!

Default LNET config

- The default Server LNET config for Spider II is:

```
options lnet    check_routers_before_use=1 \
                dead_router_check_interval=50 \
                live_router_check_interval=100 \
                avoid_asym_router_failure=0
```

- But you said to turn on asym_router_failure earlier!!
- We are currently investigating an issue where this feature doesn't work for Titan/Spider II
- We have seen it work correctly at smaller scale and IB <-> IB environments
- LU-5758

How do we get around it?

- We exploit the defaults
- Set only what is necessary to deviate from default
- All servers are in o2ib (o2ib0)
- We then script the addition of the routes after the LNET module is loaded
- Do this both on clients and servers
- Routers are configured only through modprobe.conf

What happens if it needs to change

- Why would you need to change LNET networks, IP addressing, etc.?
  - Titan example; upgrade part of machine to GPU's leaving other part for users to compute on
  - Compute cluster outgrows it's subnet range
  - Probably many others..
- You can test in a test environment, but to implement the change in production you have to reboot and reload the module parameters

Lustre 2.7 – Dynamic LNET Config

- DLC allows you to use lnetctl to:
  - Add/remove LNET Networks
  - Add/remove LNET Routes
  - Configure/modify router buffer pools
  - Enable/Disable routing
  - Show Routing
  - Not all parameters are configurable
- DLC will allow CLI based configuration or YAML config file
  - Import and Export
- Tracking DLC work in LU-2456

Example lnetctl

- lnetctl net add –net tcp0 -if eth0 –peer-timeout 180 –peer-credits 8 –credits 1024
  - Adds tcp0 LNET Network on eth0
- lnetctl net show
- lnetctl net show -detail
- lnetctl net show -net tcp0
- lnetctl net show -net tcp0 –detail
- lnetctl route del –net tcp0 –gateway 10.1.1.2@tcp1
- Examples lifted from Amir Shehata's of Intel's LUG 2014 presentation at : http://insidehpc.com/2014/04/video-dynamic-lnet-config/

More Inetctl

- By putting field in request you can filter on that param
- Inetctl's output is YAML
  - So machine parsable
- Avoids scripting in bash for a common way to add and remove networks and routes.
  - No more rebooting to add networks
  - No more site customization

Review

- Configuring LNET can be complex
  - It can be very simple too
- Understanding underlying topology and transport is very important
- Lustre's flexibility can make this harder

Questions?

Thanks for your attention!