LNetCtl Utility Improvements

Target release	
Epic	
Document status	**DRAFT**
Document owner	Amir Shehata
Designer	
Developers	
QA	

Overview

The DLC project added the ability to dynamically configure LNet. It also added the ability to pull information from LNet and display them in YAML format. The decision was later made to make lnetctl the default utility for all LNet operations. This will entail rolling in all LNet related functionality from lctl to lnetctl.

This document will outline the following:

1. The sysfs interface design for the parameters which will be stored in sysfs and reported by lnetctl
2. The ioctl interface design for the operations which will be implemented via lnetctl

Design Principles

There has been several discussions on the mechanics of pulling information from the kernel. What has been agreed on is to use `sysfs` for simple key/value pairs. Primary examples for that are module parameters and statistics. For module parameters the hooks are already in place to add the parameters in `sysfs`. Namely under: `/sys/module/lnet/parameters`. Statistics also fall in the category of simple key/value pairs. These can also be stored under `sysfs`. The current suggested path is: `/sys/module/lnet/statistics/`. The exact hierarchy will be outlined later in the requirements.

This information can then be pulled out of `sysfs` via `lnetctl`. According to an article posted by GregKH and written by Kay Sievers, https://lwn.net/Articles/237664/, it is not suggested to use `libsysfs` for accessing `sysfs` from userspace, since it doesn't follow the rules he mentions in that article. However, it will be ideal to use an existent library, if one exists, rather than create our own `sysfs` interface. However if one doesn't exist, then the possibility of creating one for lustre should be investigated. The suggestion here is not to have each utility do it's own sysfs access, but rather consolidate this in a library which can be utilized by all Lustre/LNet utilities.

For more complex configuration operations, such as configuring network, network interfaces, peers, etc, IOCTL will still be used to invoke these operations. Sysfs is not geared to handle complex information passing to the kernel. And the IOCTL mechanism is already in place to handle more complex kernel operations.

Requirements

`sysfs` Structure

ID	Status	Class	Description
sysfs-01		MUST HAVE	Create a statistics directory under /sys/modules/lnet `/sys/modules/lnet/statistics`

sysfs-02		MUST HAVE	Create a directory for each Luster Network type
			ex:
			`/sys/modules/lnet/statistics/lolnd/`
			`/sys/modules/lnet/statistics/o2ib/`
			`/sys/modules/lnet/statistics/tcp/`
			`/sys/modules/lnet/statistics/gni/`

sysfs-03		MUST_HAVE	Create a directory for each separate lustre network under the network type ex: `/sys/modules/lnet/statistics/o2ib/1`

sysfs-04		MUST HAVE	Create a directory for each configured Network interface under the Lustre Network
			ex:
			`/sys/modules/lnet/statistics/o2ib/[NUM]/ib[NUM]/`
			`/sys/modules/lnet/statistics/tcp/[NUM]/eth[NUM]/`

sysfs-05		MUST HAVE	Place the Network Interface statistics as attribute files under the Network interface directory `ex: send_count, recv_count, ...`
sysfs-06		MUST HAVE	Create a peer directory under statistics. `/sys/modules/lnet/statistics/peers/`
sysfs-07		MUST HAVE	Create a separate entry per peer named with the peer's primary_nid ex: `/sys/modules/lnet/statistics/peers/192.168.23.3@o2ib/`
sysfs-08		MUST HAVE	Create a separate entry per peer network interface (peer_ni) named with the NID of the peer_ni ex: `/sys/modules/lnet/statistics/peers/192.168.23.3@o2ib/192.168.23.4@o2ib2`
sysfs-09		MUST HAVE	Add peer statistics, each in its own attribute file, under the peer_ni directory. ex: `state, max_ni_tx_credits, etc`
sysfs-10		MUST HAVE	Create a directory for statistics under the ko2iblnd module `/sys/modules/ko2iblnd/statistics`
sysfs-11		MUST HAVE	Create a directory for peers under the statistics directory `/sys/module/ko2iblnd/statistics/peers/`
sysfs-12		MUST HAVE	Create a directory for each peer_ni under the /sys/module/ko2iblnd/statistics/peers/. Under each peer_ni directory create a directory for each local_ni which has connections established with the peer_ni. This is needed because there could be multiple LND peers for a single LNet peer. One local_ni can talk to multiple peer_nis. ex: `/sys/module/ko2iblnd/statistics/peers/peer_NID/local_NID/`
sysfs-13		MUST HAVE	Create a link from the LNet peer_ni directory to the specific LND peer-ni directory, which contains all instances of LND peer-nis associated with that LNet peer. ex: `/sys/modules/lnet/statistics/peers/192.168.23.3@o2ib/192.168.23.4@o2ib2/LND_peers/`

sysfs-14		MUST HAVE	Add LND peer_ni statistics each in a separate attribute file under the LND peer_ni directory as defined above. The statistics to be added are as follows: • number of active connections • number of messages waiting for a connection to be established • number passive connects • number active connects • number of times connection races were triggered • the last time the peer was alive • the number of users for this peer (refcount)
sysfs-15		MUST HAVE	Add a directory under each LND peer NI director that will contain all the connections established to that peer. ex: `/sys/modules/ko2iblnd/statistics/peers/peer_ni/local_ni/conns`
sysfs-16		MUST HAVE	Add a directory per connection under the connections directory. The name shall be "<num>" ex: `/sys/modules/ko2iblnd/statistics/peers/peer_ni/local_ni/conns/1/`
sysfs-17		MUST HAVE	Add all the connections stats as attribute files under the relevant <num> directory The statistics to be added are as follows: • the number of users for this connection (`ibc_refcount`) • the state of the connection (`ibc_state`) • number of incomplete sends (`ibc_nsends_posted`) • number of incomplete no-ops (`ibc_noops_posted`) • number of available credits (`ibc_credits`) • number of credits to return (`ibc_outstanding_credits`) • number of reserved credits (`ibc_reserved_credits`) • last communication error on that connection (`ibc_comms_error`) • Note if the connection is immediately closed on error then this value will probably never be seen. • number of queued messages of type IMMEDIATE (`ibc_tx_queue`) • number of queued messages not requiring credits of type `PUT_NAK, PUT_ACK, PUT_DONE, GET_DONE.` (`ibc_tx_queue_nocred`) • number of queued messages of type `PUT_REQ, GET_REQ.` (`ibc_tx_queue_rsrvd`) • number of transmits waiting for completion (`ibc_active_txs`)
sysfs-18		MUST HAVE	provide a reset attribute file under `/sys/modules/lnet/statistics/`. When anything is written to it, then all statistics are reset.
sysfs-19		MUST HAVE	provide a reset attribute file under `/sys/modules/ko2ilbnd/statistics/`. When anything is written to it then all statistics are reset.
sysfs-20		MUST HAVE	provide a reset attribute file under `/sys/modules/socklnd/statistics/`. When anything is written to it then all statistics are reset.

`lnetctl` Requirements

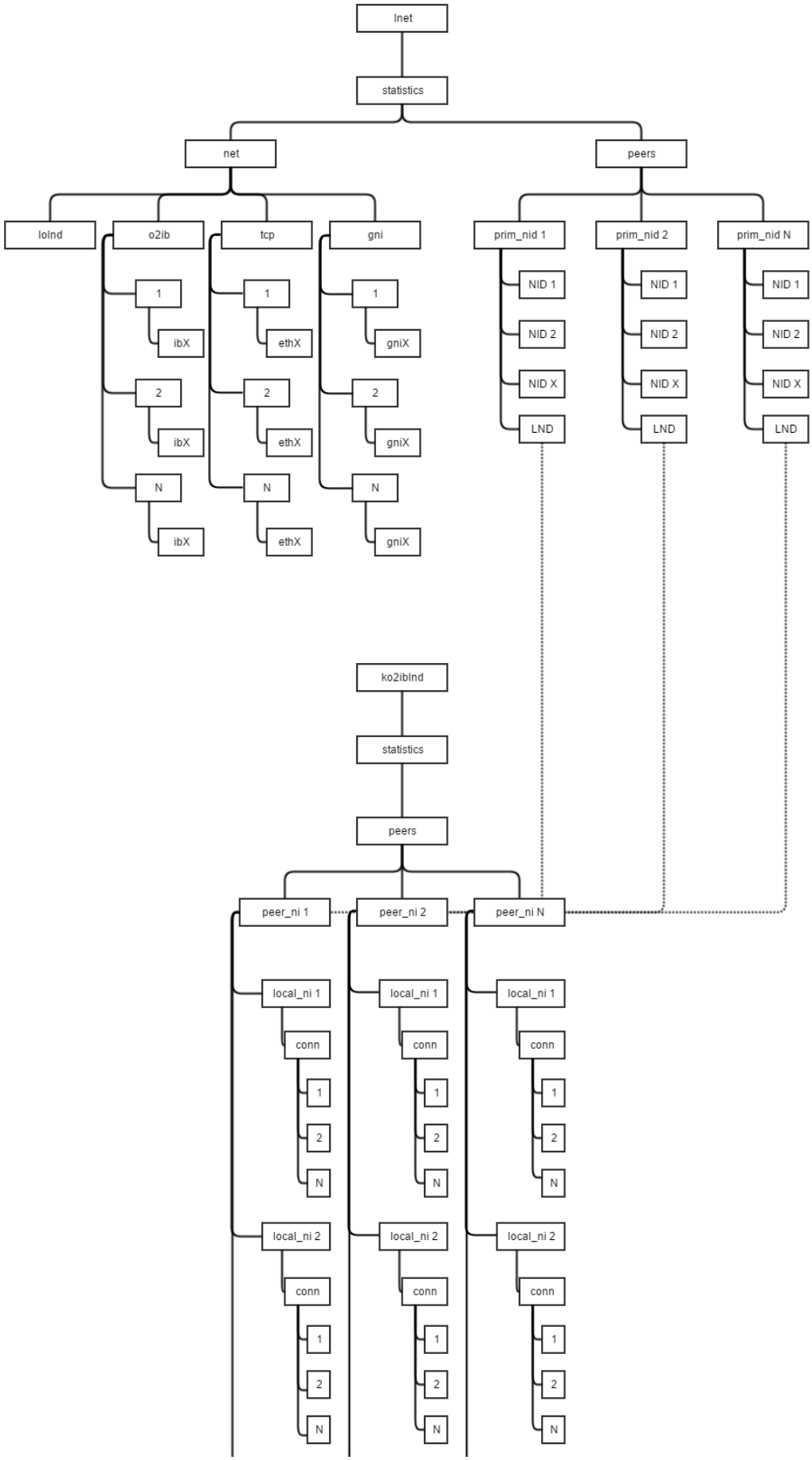
ID	Status	Class	Description
ctl-01		MUST HAVE	lnetctl shall traverse the /sys/lnet/statistics and /sys/<LND>/statistics directory and gather the stats and displays them in YAML format

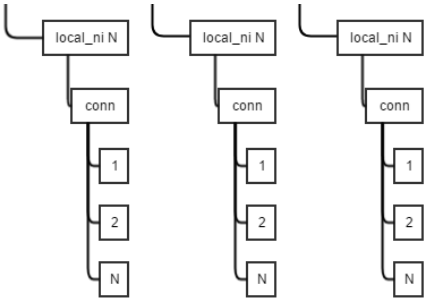
ctl-02		MUST HAVE	lnetctl shall display the peer stats using the following hierarchy:
			`peer:`
			`- primary nid: 192.168.1.1@o2ib`
			`Multi-Rail: True`
			`peer ni:`
			`- nid: 192.168.1.1@o2ib`
			`<list of stats>`
			`LND peer ni:`
			`- ni: <local ni>`
			`<list of stats>`
			`connections:`
			`- connection: <num>`
			`<list of stats>`

ctl-03		MUST HAVE	lnetctl shall show the LND peer only when a the --lnd_peer option is provided. the LND peer information will be displayed as an addition as represented the hierarchy described above ex: `lnetctl peer show --lnd_peer`
ctl-04		MUST HAVE	lnetctl shall allow peers to be filtered on their primary NID ex: `lnetctl peer show --prim_nid <nid>`
ctl-05		MUST HAVE	lnetctl shall allow peers to be filtered on their networks. The semantics are "Show all peers on <network>. ex: `lnetctl peer show --net <net>`
ctl-06		MUST HAVE	lnetctl shall use the function `libcfs_str2nid()` to resolve a string to a NID. This function is able to resolve an IP address or a hostname into a NID, for socklnd and o2iblnd networks. It can also handle gni networks
ctl-07		MUST HAVE	lnetctl shall show the LND connections per peer only when the --connections option is provided. The LND connection information will be displayed per peer as represented in the hierarcy above. By providing the --connection option the --lnd_peer option is activated automatically as it is necessary to display the connection information per lnd peer. ex: `lnetctl peer show --prim_nid <nid> --connections`
ctl-08		MUST HAVE	LNet shall extend the del peer command to shutdown all active connection to a peer.
ctl-09		NICE-TO-HAVE	LNet shall allow the termination of a single connection to a peer in the case when multiple connections are opened to the same peer. `lnetctl peer del connection --nid <nid> --id <connection id as displayed in the show>`
ctl-10		MUST HAVE	lnetctl shall provide a command to reset all statistics `lnetctl statistics reset`

sysfs structure

Overview


lnet
statistics
net
peers
lolnd
o2ib
tcp
gni
prim_nid 1
prim_nid 2
prim_nid N
1
ibX
2
ibX
N
ibX
1
ethX
2
ethX
N
ethX
1
gniX
2
gniX
N
gniX
NID 1
NID 2
NID X
LND
NID 1
NID 2
NID X
LND
NID 1
NID 2
NID X
LND
ko2iblnd
statistics
peers
peer_ni 1
peer_ni 2
peer_ni N
local_ni 1
conn
1
2
N
local_ni 2
conn
1
2
N
local_ni 1
conn
1
2
N
local_ni 2
conn
1
2
N
local_ni 1
conn
1
2
N
local_ni 2
conn
1
2
N



Example

For the sake of demonstrating the sysfs structure, lets assume the following system:

Node A with the following NIDs:

- 192.168.1.2@o2ib
- 192.168.1.3@o2ib
- 192.168.1.4@o2ib

Peer B with the following NIDs

- 192.168.2.2@o2ib
- 192.168.2.3@o2ib
- 172.168.2.2@o2ib1

Currently the following connections are established

- 2 connections between 192.168.1.2@o2ib and 192.168.2.2@o2ib
- 2 connections between 192.168.1.2@o2ib and 192.168.2.3@o2ib
- 1 connection between 192.168.1.3@o2ib and 192.168.2.2@o2ib
- 1 connection between 192.168.1.4@o2ib and 192.168.2.3@o2ib

The sysfs directory structure would be as follows:

```
/sys/modules/lnet/statistics/o2ib/
/sys/modules/lnet/statistics/o2ib/ib0/
/sys/modules/lnet/statistics/o2ib/ib0/send_count
/sys/modules/lnet/statistics/o2ib/ib0/recv_count
/sys/modules/lnet/statistics/o2ib/ib0/drop_count
/sys/modules/lnet/statistics/o2ib/ib0/<other stats>
/sys/modules/lnet/statistics/o2ib/ib1/
/sys/modules/lnet/statistics/o2ib/ib1/send_count
/sys/modules/lnet/statistics/o2ib/ib1/recv_count
/sys/modules/lnet/statistics/o2ib/ib1/drop_count
/sys/modules/lnet/statistics/o2ib/ib1/<other stats>
/sys/modules/lnet/statistics/o2ib/ib2/
/sys/modules/lnet/statistics/o2ib/ib2/send_count
/sys/modules/lnet/statistics/o2ib/ib2/recv_count
/sys/modules/lnet/statistics/o2ib/ib2/drop_count
/sys/modules/lnet/statistics/o2ib/ib2/<other stats>

/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.2@o2ib/
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.2@o2ib/state
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.2@o2ib/max_n
i_tx_credits
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.2@o2ib/avail
```

```
able_tx_credits
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.2@o2ib/<other
stats>
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.3@o2ib/
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.3@o2ib/state
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.3@o2ib/max_n
i_tx_credits
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.3@o2ib/avail
able_tx_credits
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/192.168.2.3@o2ib/<other
stats>
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/172.168.2.2@o2ib/
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/172.168.2.2@o2ib/state
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/172.168.2.2@o2ib/max_n
i_tx_credits
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/172.168.2.2@o2ib/avail
able_tx_credits
/sys/modules/lnet/statistics/peers/192.168.2.2@o2ib/172.168.2.2@o2ib/<other
stats>

/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.2@o2ib/
/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.2@o2ib/tx
_queue_noop
/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.2@o2ib/tx
_queue_cr
/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.2@o2ib/<o
ther statistics>

/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.3@o2ib/
/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.3@o2ib/tx
_queue_noop
/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.3@o2ib/tx
_queue_cr
/sys/module/ko2iblnd/statistics/peers/192.168.2.2@o2ib/192.168.1.3@o2ib/<o
ther statistics>

/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.2@o2ib/
/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.2@o2ib/tx
_queue_noop
/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.2@o2ib/tx
_queue_cr
/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.2@o2ib/<o
ther statistics>

/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.4@o2ib/
/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.4@o2ib/tx
_queue_noop
/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.4@o2ib/tx
_queue_cr
```

```
/sys/module/ko2iblnd/statistics/peers/192.168.2.3@o2ib/192.168.1.4@o2ib/<o
ther statistics>
```

Phases

All statistics are currently pulled from the kernel via the ioctl interface. As indicated above for simple key/value pairs the sysfs interface is preferred. However, the work will be divided in multiple phases or patches in order to concentrate more on adding new functionality rather than rewriting existing ones. These phases are equivalent to gerrit patches.

Phase 1

Add the necessary infrastructure to create sysfs directories and files. This might just boil down to using the provided APIs, but we'd probably want to create _sysfs.c file that will contain all the necessary code to handle sysfs for lnet.

Phase 2

Create the lnd peer sysfs structure and all associated stats.

Add the hooks necessary for sysfs to be updated when a read is performed on the exported attributes.

Phase 3

Create the user space control to pull up and display the lnd peer sysfs

Phase 4

Create the lnd peer connections sysfs structure and all associated stats.

Add the hooks necessary for sysfs to be updated when a read is performed on the exported attributes

Phase 5

Create the user space control to pull up and display the lnd peer connection sysfs stats

Phase 6

Create the LNet peer sysfs structure and assocated stats.

Phase 7

Update the user space control to traverse the sysfs directory tree rather than use ioctl.

Phase 8

Create the LNet network sysfs structure and associated stats.

Phase 9

Update the user space control to traverse the sysfs directory tree rather than use ioctl