OPENFABRICS ALLIANCE

13th ANNUAL WORKSHOP 2017

LNET MULTI-RAIL RESILIENCY

Amir Shehata, Lustre Network Engineer

Intel Corp

March 29th, 2017

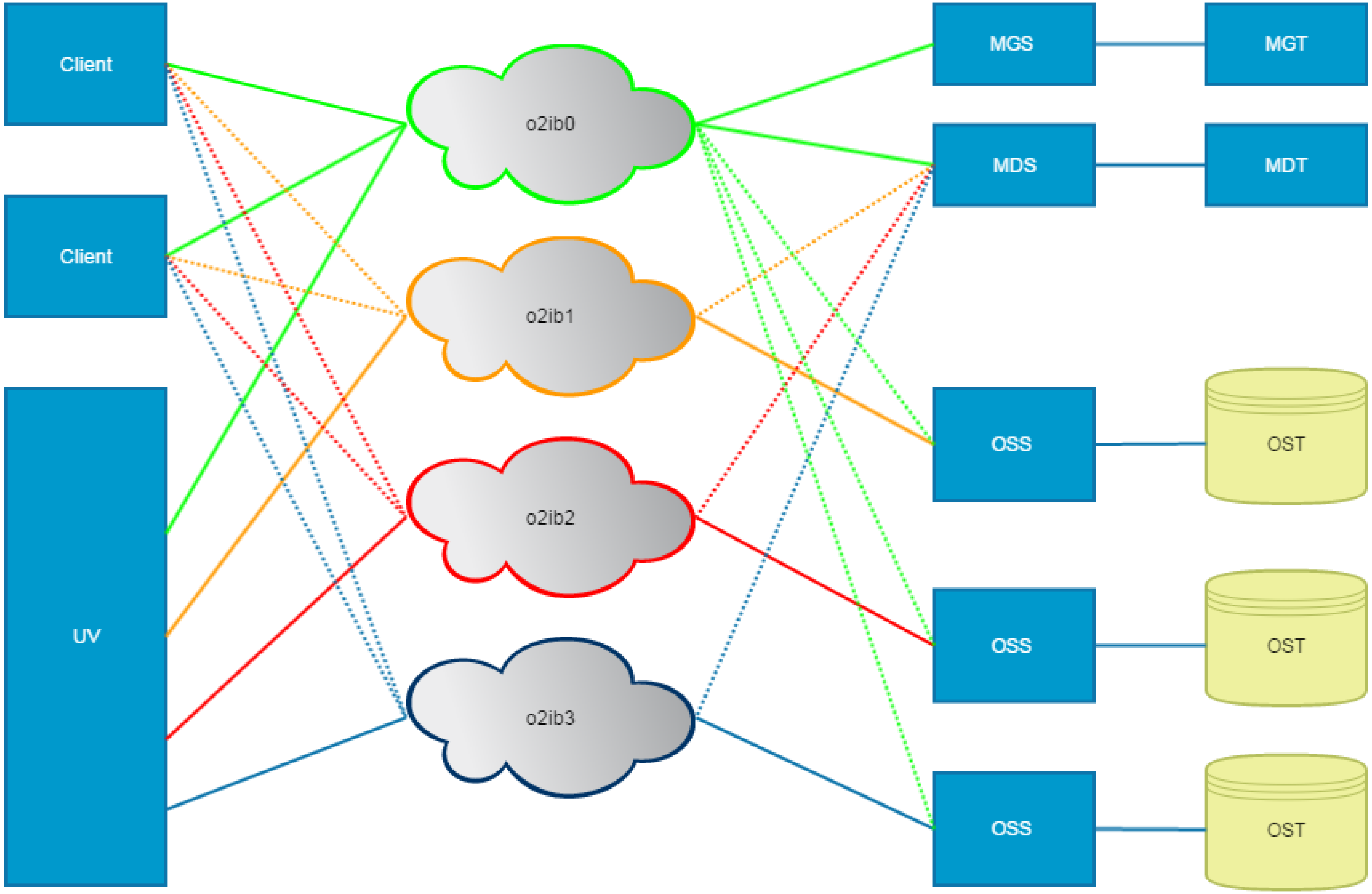
OUTLINE

- **Multi-Rail Recap**
  - Base Multi-Rail
  - Dynamic Discovery
- **Multi-Rail performance on SGI System**
- **LNet Resiliency**
  - LNet Messages
  - Message failures
  - Failure Handling
- **Interface Health Tracking**
- **Reporting**
- **Summary**

LUSTRE NETWORK MULTI-RAIL (MR)

- **Prior to MR**
  - Only one Network Interface (NI) per LNet network.
  - Multiple NIs == Multiple LNet networks.
  - Resulted in a complex configuration.

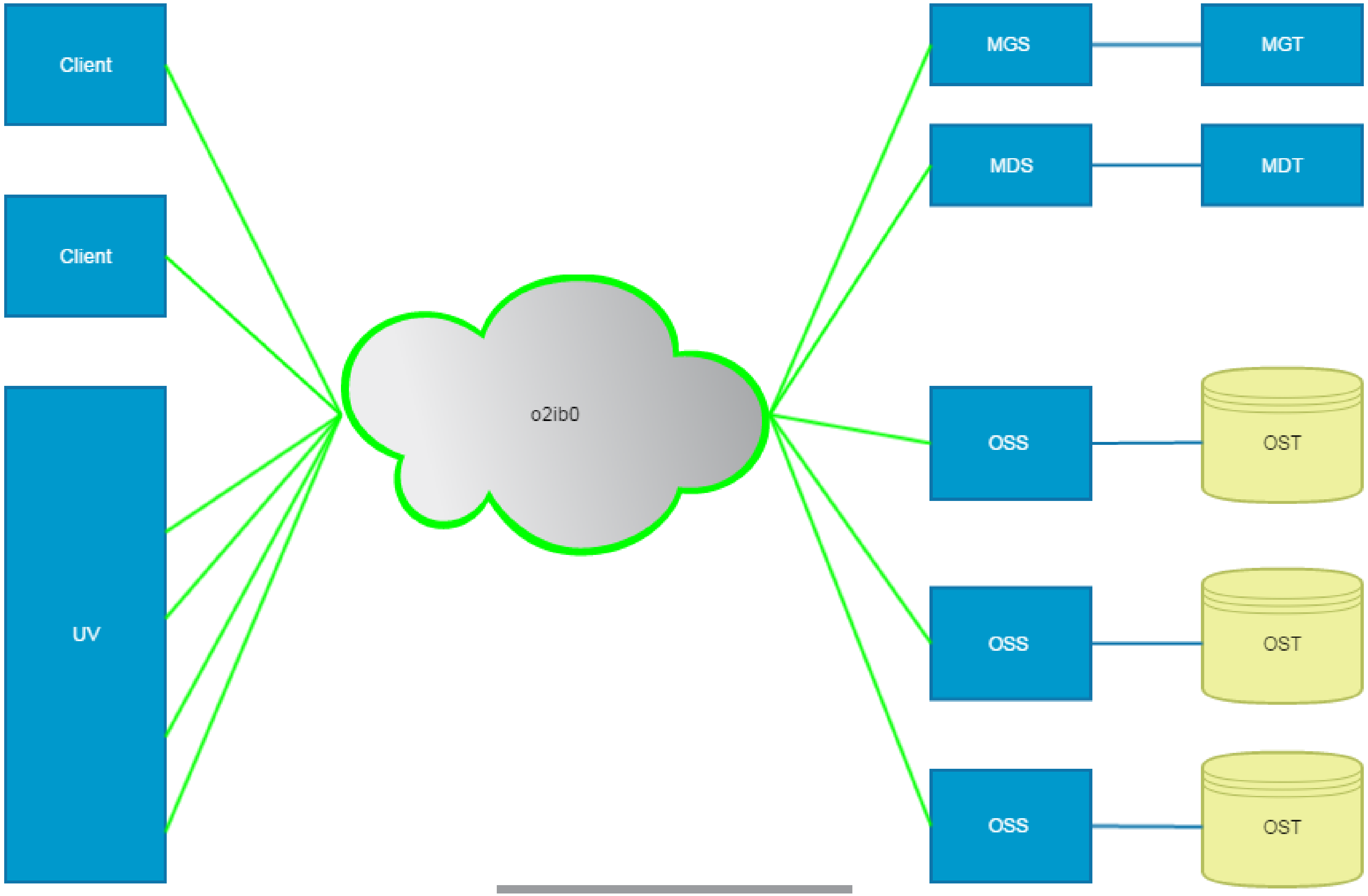
WITHOUT MULTI-RAIL



DRIVER

- **Add support for big Lustre Nodes**
  - SGI UV 300: 32 Socket NUMA system
  - SGI UV 3000: 256 Socket NUMA system
- **Large systems need lots of bandwidth.**
- **NUMA systems benefit when memory buffers and interfaces are close in the system's topology.**
- **Support different HW (ex OPA, MLX, ETH).**
- **Simplify configuration.**

WITH MULTI-RAIL



BENEFITS OF MULTI-RAIL

- **Multi-Rail implemented at the LNet layer allows:**
  - Multiple interfaces connected to one network.
  - Multiple interfaces connected to different networks.
  - Interfaces are used simultaneously.
- **LNet level implementation allows use of interfaces from different vendors. For example:**
  - OPA and MLX are incompatible.
  - OPA on o2ib0, MLX on o2ib1.
  - MR can use both simultaneously to communicate with peers on the same networks.

BASIC CONFIGURATION

- **Two MR configuration steps**
  - Configure the local networks and the interfaces on each network
    - EX: o2ib0 with ib0, ib1 and ib2.
  - Configure peers
    - In this step each node is configured with all the Multi-Rail peers it needs to know about.
    - Peers are configured by identifying on the node the peer's primary NID and all of its other NIDs.

EXAMPLE CONFIGURATION

Peer B's configuration on Peer A

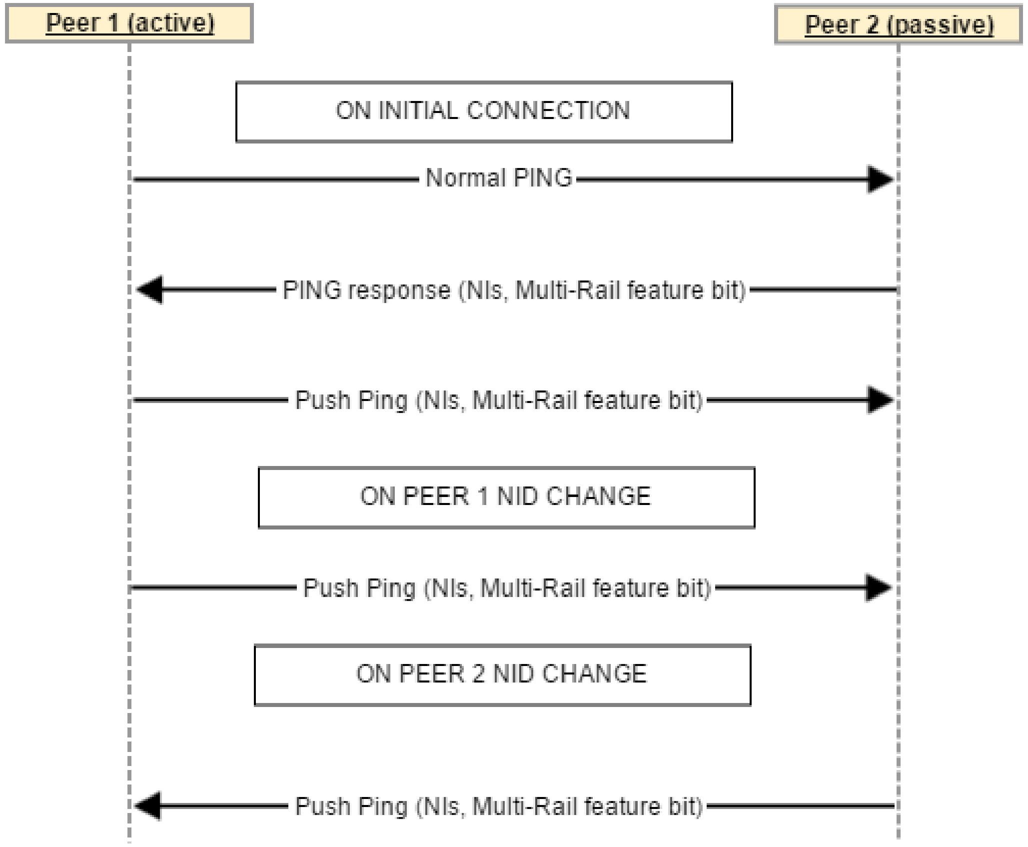
```
peer:
  - primary nid: 192.168.122.30@tcp
    Multi-Rail: True
    peer ni:
      - nid: 192.168.122.30@tcp
        state: NA
      - nid: 192.168.122.31@tcp
        state: NA
      - nid: 192.168.122.32@tcp
        state: NA
```

Peer A's configuration on Peer B

```
peer:
  - primary nid: 192.168.122.10@tcp
    Multi-Rail: True
    peer ni:
      - nid: 192.168.122.10@tcp
        state: NA
      - nid: 192.168.122.11@tcp
        state: NA
      - nid: 192.168.122.12@tcp
        state: NA
```

DYNAMIC DISCOVERY

- **Manual configuration can be error prone.**
- **Dynamic Discovery reduces configuration burden.**



LNET MR SUMMARY

- **MR allows multiple interfaces, increasing bandwidth.**
- **Good for servers that need more bandwidth to serve more clients.**
- **Ability to add more interfaces to the server.**
- **Useful for large clients with many interfaces, like the UV.**

MULTI-RAIL PERFORMANCE – THE SYSTEM

- **SGU UV 300: 32 socket of Xeon Processors**
- **16 TB of memory**
- **8 Omni-Path network interfaces**
- **8 C2112-GP2-EX Object Storage Systems (OSS)**
- **4 P3700 NVMe devices LDISKFS Object Storage Target (OST) per OSS**

MULTI-RAIL PERFORMANCE

- **Theoretical maximum performance of the system:**
  - P3700 Sequential Write: 34560 MB/s
  - Sequential Read: 86400 MB/s
- **Multi-Rail performance:**
  - Sequential Write: 31990.18 MB/s
  - Sequential Read: 68593.35 MB/s

LNET RESILIENCY

- **But what about resiliency?**
- **Lustre error handling is expensive.**
- **It can involve evicting clients (depending on RPC lost).**
- **What can LNet do to address network failures?**
- **Other interfaces can be tried for the same peer, before giving up on a message.**

LNET MESSAGES

- **LNet has four main message types:**
  - PUT
  - ACK
  - GET
  - REPLY
- **ACK is an optional response to a PUT**
- **REPLY is the response to a GET**

LNET MESSAGE TRANSACTIONS

- **That gives us the following combinations to handle:**
  - `PUT`
  - `PUT + ACK`
  - `GET + REPLY`
- **An RPC can be one or more combination of the above.**

GOAL

- **In order for LNet to be resilient it must try all local interfaces and all remote interfaces before it declares a message failed.**
- **But it must not go on resending messages indefinitely**
  - There has to be an upper time limit after which LNet declares a message failed.
- **Resiliency logic needs to be implemented at the LNet level to be able to fail across different network types if needed: OPA, MLX, ETH.**

STRATEGY

- **An LNet network can have directly connected nodes .**
- **Or LNet routers can introduce extra hops between the source and the destination.**
- **LNet resiliency is concerned with ensuring that an LNet message is delivered to the next hop**
  - EX if the Source and the Destination are separated by multiple hops, each hop will be responsible for ensuring that a message is received and potentially acknowledged by the immediate next-hop.

PUT

- **A PUT message is used to indicate that data is about to be RDMAed from the node to the peer.**
- **This is the simplest LNet message.**
- **Since there is no response to this message if it is dropped on the remote end, there is nothing that LNet can do.**
- **Upper layers which request PUT on its own need to implement their own timeouts.**

PUT+ACK

- **The sender of the PUT can explicitly request an ACK from the receiver.**
- **The ACK message is generated by the LNet layer once it receives the PUT.**
  - It does not wait for the upper layer to process the PUT.
- **Since LNet on the sender node expects an ACK if it is not received it can deliver a timeout failure event to the upper layer.**

GET+REPLY

- **GET semantics means that the sender node is requesting data from the peer.**
- **A GET always expects a REPLY**
  - LNet does not wait fro upper layer to process the GET.
- **Since LNet knows that a REPLY is expected if it is not received it can deliver a timeout event to the upper layer.**

LNET RESILIENCY SUMMARY

- **LNet Resiliency is concerned with ensuring that a message is delivered to the next-hop.**
- **In `PUT+ACK` and `GET+REPLY` cases LNet can maintain a timeout.**

FAILURES TO HANDLE

There are three classes of failure that LNet needs to handle:

1. ***Local interface failure*: There is some issue with the local interface that prevents it from sending or receiving messages.**
2. ***Remote interface failure*: There is some issue with the remote interface that prevents it from sending or receiving messages.**
3. ***Path Failure*: The local interface is working properly but messages never reach the peer interface.**

LOCAL INTERFACE FAILURE

- **The LND reports failures it gets from the hardware to LNet.**
  - **`EX: IB_EVENT_DEVICE_FATAL, IB_EVENT_PORT_ERR`**
- **LNet refrains from using that interface for any traffic.**
- **Retransmit on other available interfaces.**
- **It will keep attempting to retransmit the message until a configurable peer timeout is reached.**
- **On peer-timeout a failure is propagated to the higher levels.**

REMOTE INTERFACE FAILURE

- **There are cases when a remote interface can be considered failed:**
  - Address is wrong error
  - No route to host
  - Connection can not be established
  - Connection was rejected due to incompatible parameters
  - ?? (Needs further investigation)
- **In this case the local interface is working, but is unable to establish a connection with the remote interface.**
- **LND reports this error to the LNet layer.**
- **LNet attempts to resend the message to a different remote interface.**
- **If none are available, then fail message.**

PATH FAILURE

- **The LNDs currently implement a protocol which ensures that a message is received by the remote LND within a transmit timeout.**
- **If an LND message is not acknowledged the transmit timeout on that message expires.**
- **Where the timeout expires tells us where the failure resides, either due to a problem with the local interface or somewhere on the network.**

PATH FAILURE BREAKDOWN

- **LND Timeouts occur due to the following reasons.**
- **The message is on the sender's queue but is not posted within the transmit timeout**
- **The message is posted but the transmit never completes**
- **The message is posted, the transmit is completed, but the remote never acknowledges**
- **If it's a local or remote interface issue, then it's dealt with as previously indicated.**
- **Otherwise an entirely new pathway between the peers is attempted.**

INTERFACE HEALTH TRACKING

- **A relative health value is kept per local and remote interface.**
- **On soft errors, such as timeouts, the interface health value is decremented.**
- **That interface is selected at a lower priority for future messages.**
- **The health value recovers over time.**
- **The result is consistently unhealthy interfaces are preferred less.**

REPORTING

- **Currently `lnetctl` is a utility used to configure the different LNet parameters.**
- **`lnetctl` will extract LNet Resiliency information and show it on demand.**
- **This enhances the ability to debug a cluster when needed.**
- **This information can be displayed on a GUI. EX: Intel Manager for Lustre (IML).**

SUMMARY

- **Multi-Rail increases a node’s bandwidth.**
- **LNet Resiliency builds on top of the Multi-Rail infrastructure to allow resending of LNet messages over different interfaces on failures.**
- **The LNet level implementation allows messages to be sent over different networks and HW.**

13th ANNUAL WORKSHOP 2017

THANK YOU

Amir Shehata, Lustre Network Engineer

Intel Corp