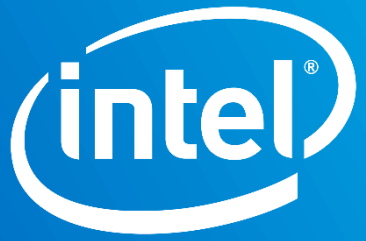


Multi-Rail LNet for Lustre*

Olaf Weber
Senior Software Engineer
SGI Storage Software

Introduction

Multi-Rail LNet

Multi-Rail is a long-standing wish list item known under a variety of names:

- Multi-Rail
- Interface Bonding
- Channel Bonding

The various names do imply some technical differences

This implementation is a collaboration between SGI and Intel

- First discussed at the LAD'15 Lustre Developer Summit.
- Presented at LUG'16.

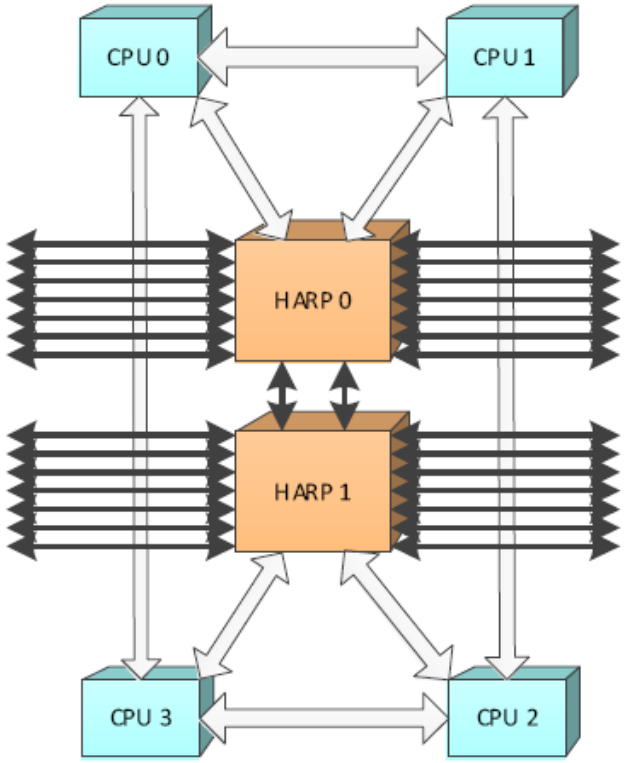
Our goal is to land multi-rail support in Lustre* 2.10

What Is Multi-Rail?

Multi-Rail allows nodes to communicate across multiple interfaces:

- Using multiple interfaces connected to one network
- Using multiple interfaces connected to several networks
- These interfaces are used simultaneously

Big Clients



We want to support big Lustre* nodes

- SGI UV 300: 32-socket NUMA system
- SGI UV 3000: 256-socket NUMA system

These systems contain hardware equivalent to an entire cluster of smaller systems.

For example, 16 dual-socket nodes have 16 connections to the cluster's network.

A single 32-socket system should have more than one.

Server Bandwidth

In big clusters, bandwidth to the server nodes becomes a bottleneck

Adding faster interfaces implies replacing much or all of the network

- Adding faster interfaces only to the servers does not work

Adding more interfaces to the servers increases the bandwidth

- Using those interfaces requires a redesign of the LNet networks
- Without Multi-Rail each interface connects to a separate LNet network
- Clients must be distributed across these networks

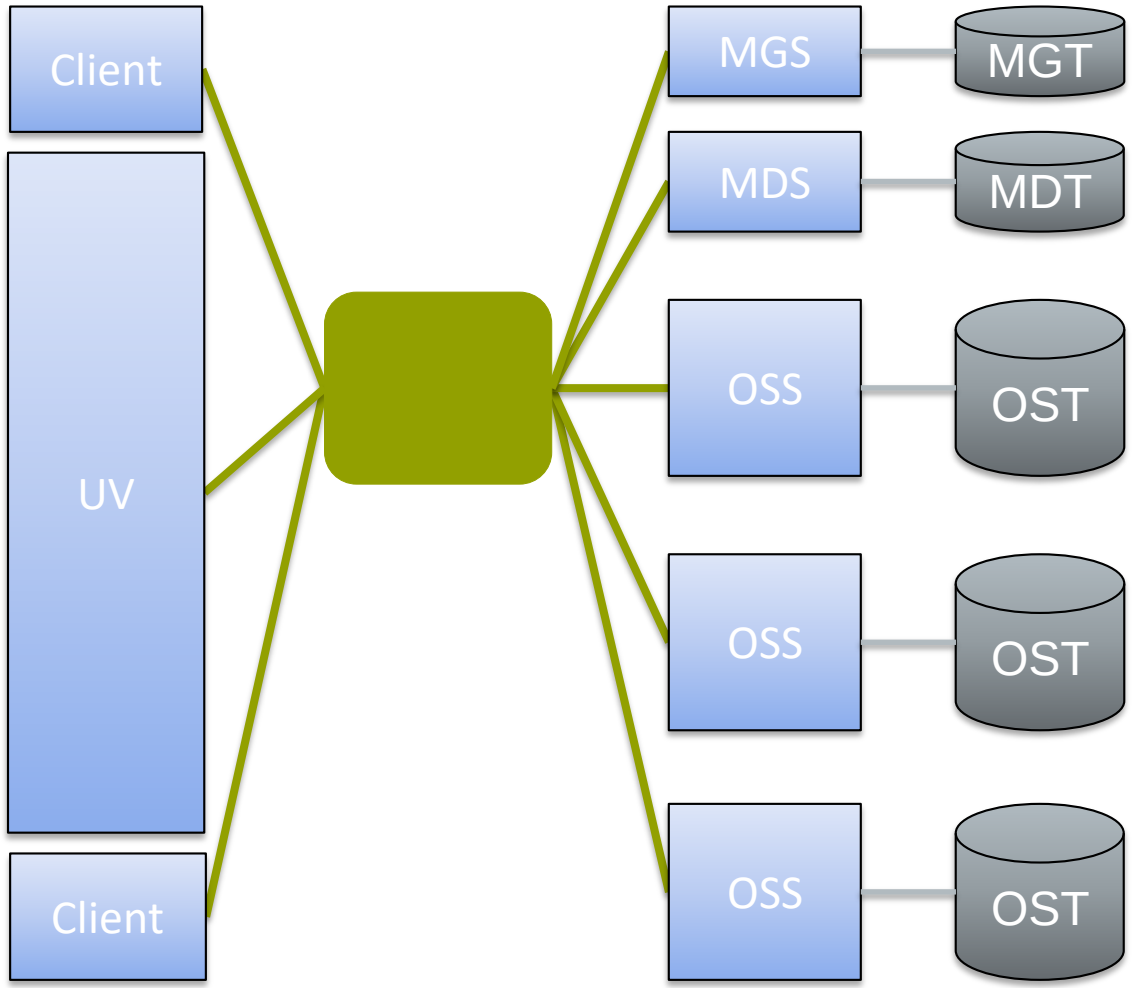
It should be simpler than this

The Multi-Rail Project

- Add basic multi-rail capability
  - Multiplexing across interfaces, as opposed to striping across them
  - Multiple data streams are therefore needed to profit
- Extend peer discovery to simplify configuration
  - Discover peer interfaces
  - Discover peer multi-rail capability
- Configuration can be changed at runtime
  - This includes adding or removing interfaces
  - *lnetctl* is used for configuration
- Compatible with non-Multi-Rail nodes
- Add resiliency by using alternate paths in case of failures

Deploying Multi-Rail LNet

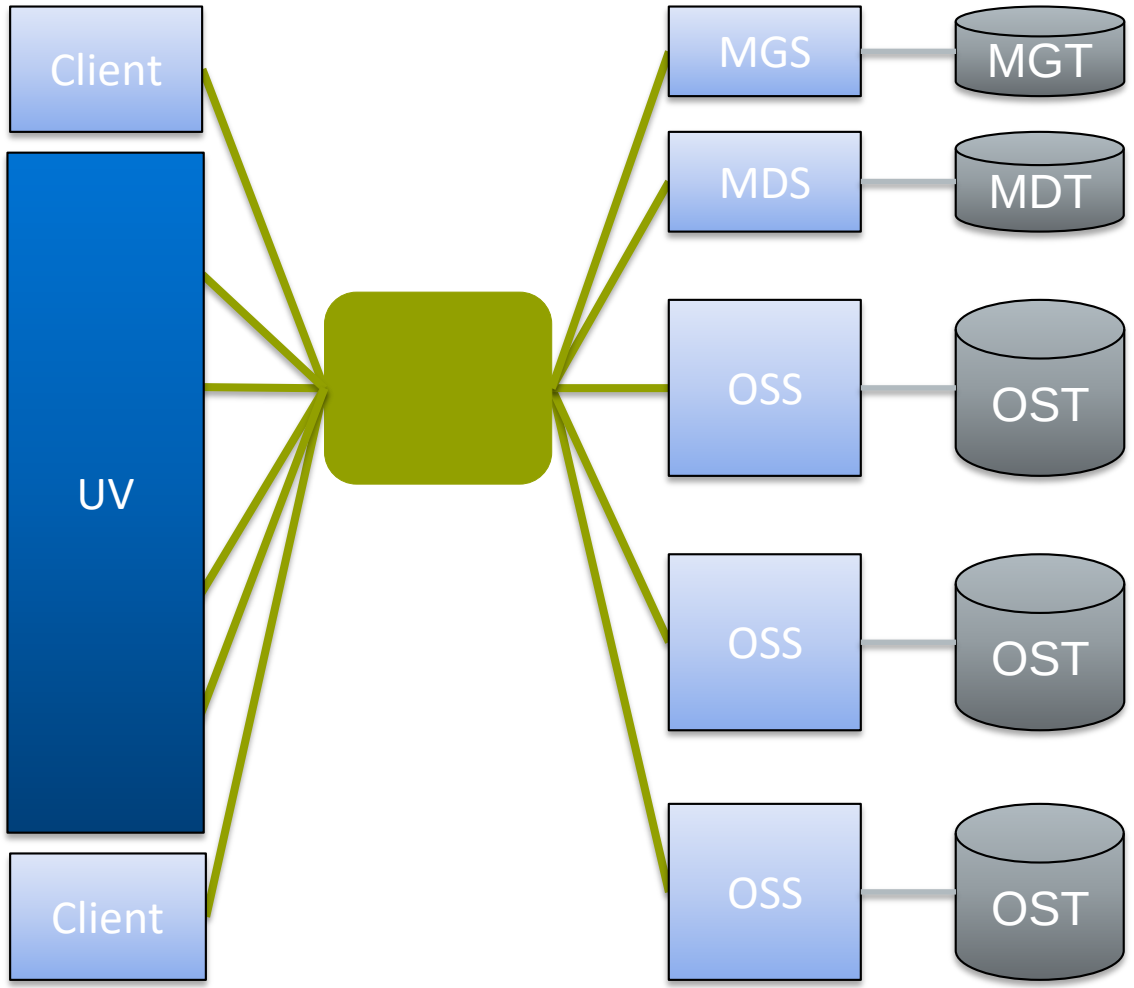
Example Cluster



A big UV node has just been added to a cluster.

It has the same bandwidth available to it as the smaller client nodes.

Multi-Rail Big Client



The UV node has been given multiple interfaces.

The UV node is the only node that *must* run a Multi-Rail aware version of Lustre*.

Configuring the Big Client

/etc/sysconfig/lnet.conf

- Interface list

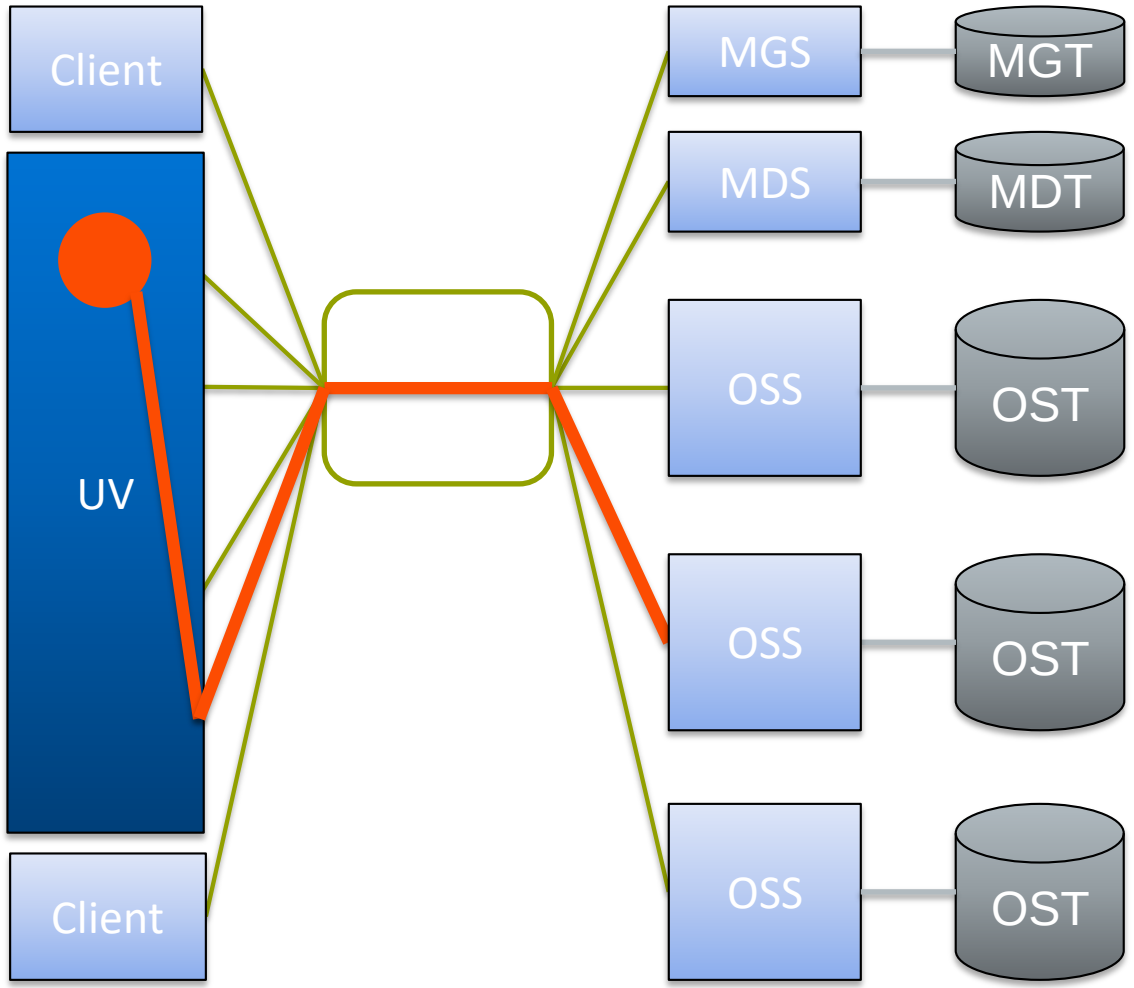
```
ip2nets:
 - net-spec: o2ib0
   interfaces:
      0: ib0[0]
      1: ib1[4]
      2: ib2[8]
      3: ib3[12]
```

Extra configuration on the big node:

- Big nodes benefit from setting up CPTs:
  - Use **`cpu_patterns=N`** to map the structure of the machine to CPTs.
- Bind the interfaces to their CPTs.
  - The example gives 4 interfaces evenly spaced on a 16-node system.
- We're looking at ways to simplify this by improving the default settings.

The special configuration is limited to the big node.

Multi-Rail Big Client: LNet traffic flow

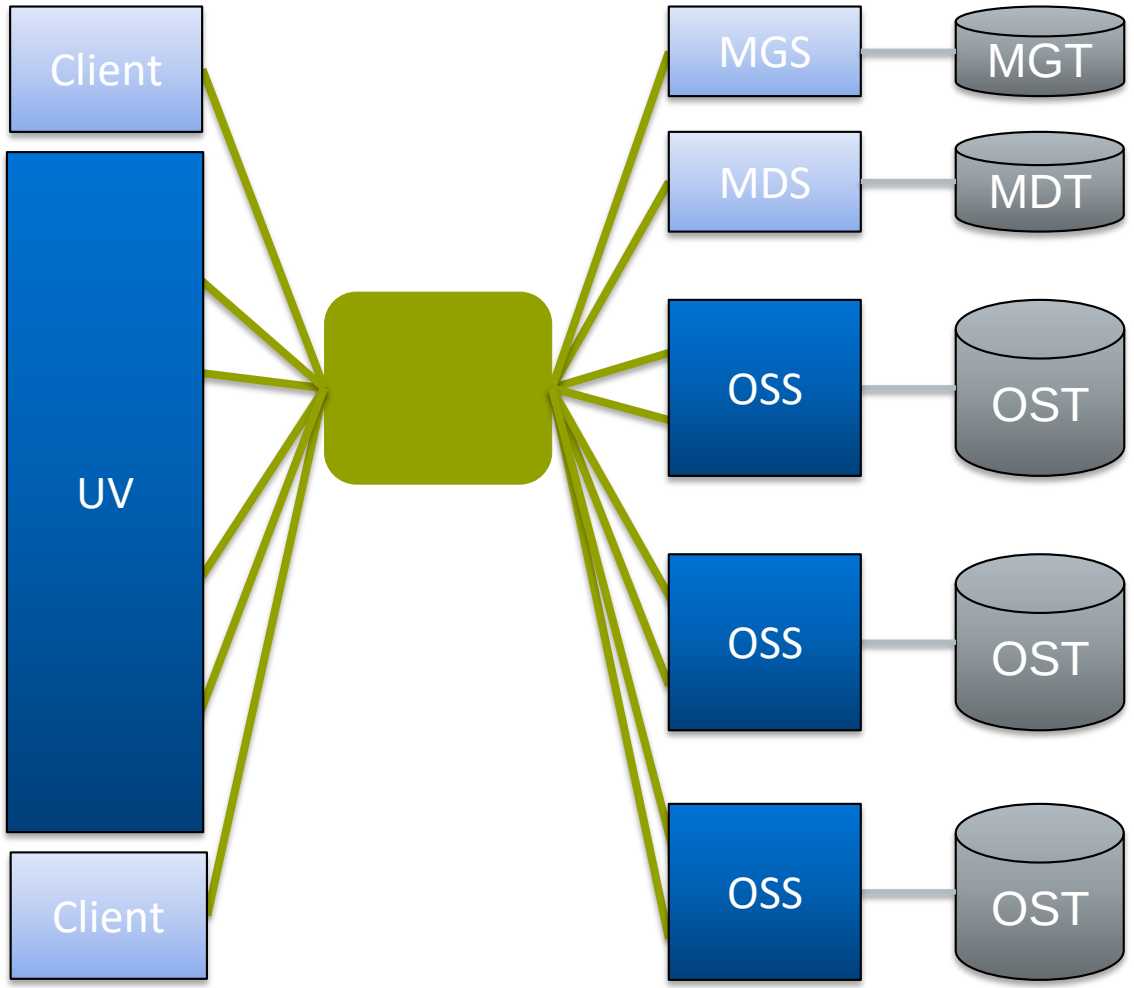


A Multi-Rail node selects a single interface to talk to a non Multi-Rail peer node.

Different interfaces are used to talk to different non Multi-Rail peers.

In a big node, a process may have to use a remote interface. This effectively adds another network hop.

Multi-Rail OSS Nodes



The OSS nodes run Multi-Rail aware Lustre*.

Extra interfaces have been added to give them additional bandwidth.

Bear in mind that the OSTs of an OSS may not be able to fill that bandwidth.

Configuring the OSS nodes

/etc/sysconfig/lnet.conf

- Interface list

```
ip2nets:
 - net-spec: o2ib0
   interfaces:
      0: ib0
      1: ib1
```

- Pattern rule

```
ip2nets:
 - net-spec: o2ib0
   ip-range:
      0: 10.0.0.*
```

Extra configuration on the OSS nodes:

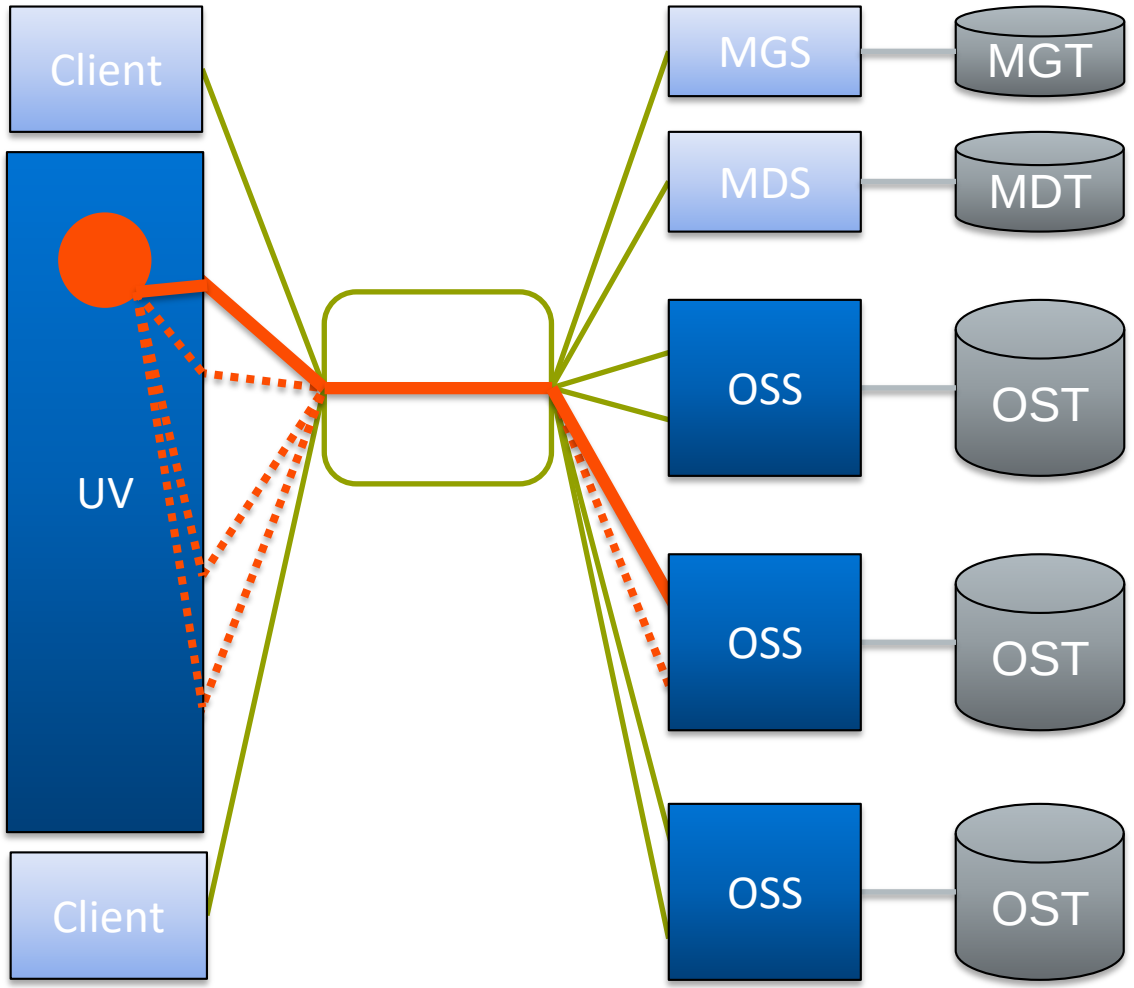
- An OSS node likely does not need CPTs defined
- As with the big client, we can list all interfaces.

A pattern rule is simpler:

- Adds all interfaces on the 10.0.0.* subnet.

Peer discovery takes care of finding which NIDs belong to a peer node.

Multi-Rail OSS Nodes: LNet traffic flow

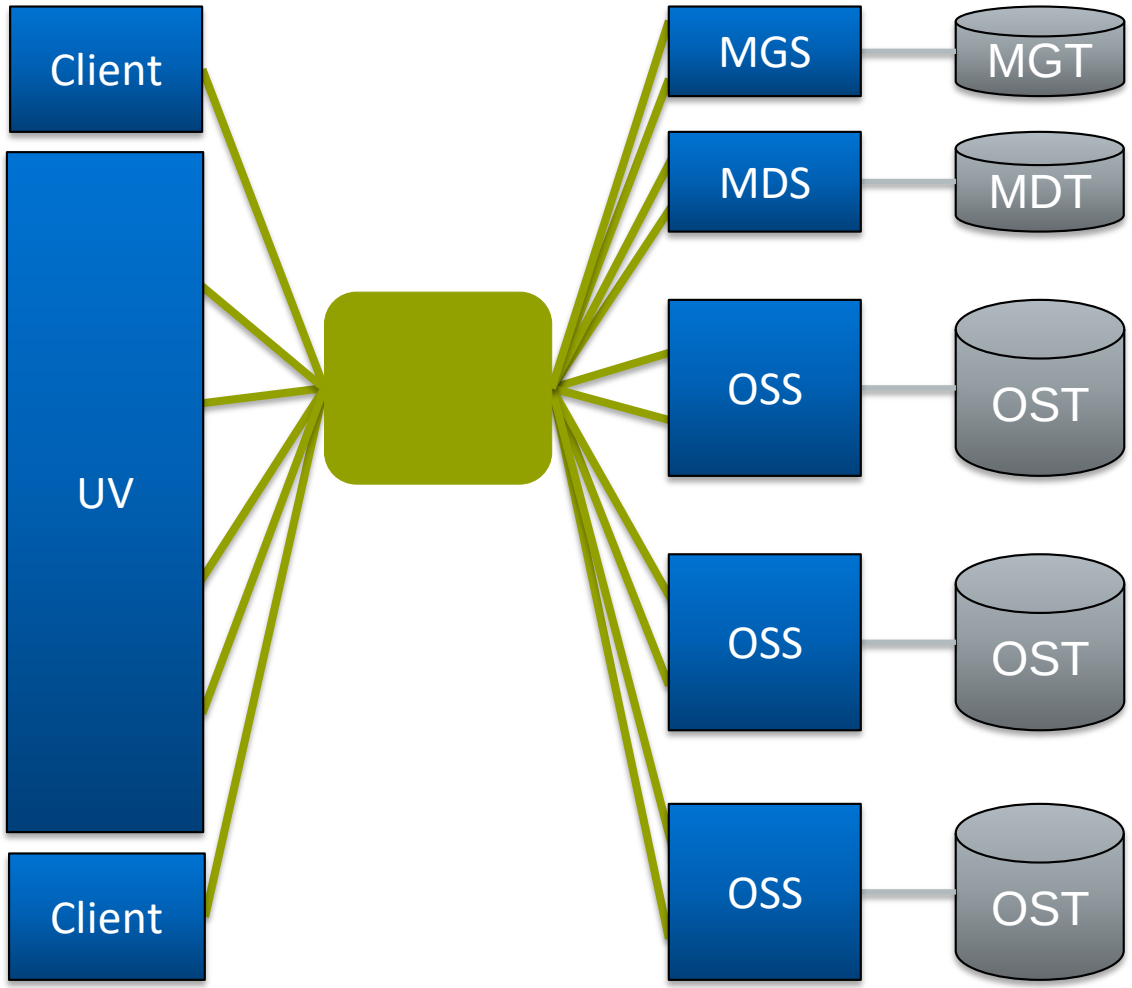


When both nodes run Multi-Rail aware Lustre*, all interfaces on both nodes can be used, provided that there is a route between them.

The big client node is free to choose the interface pair: any of the red paths can be used.

In this example, the closest interface was used.

Multi-Rail Cluster

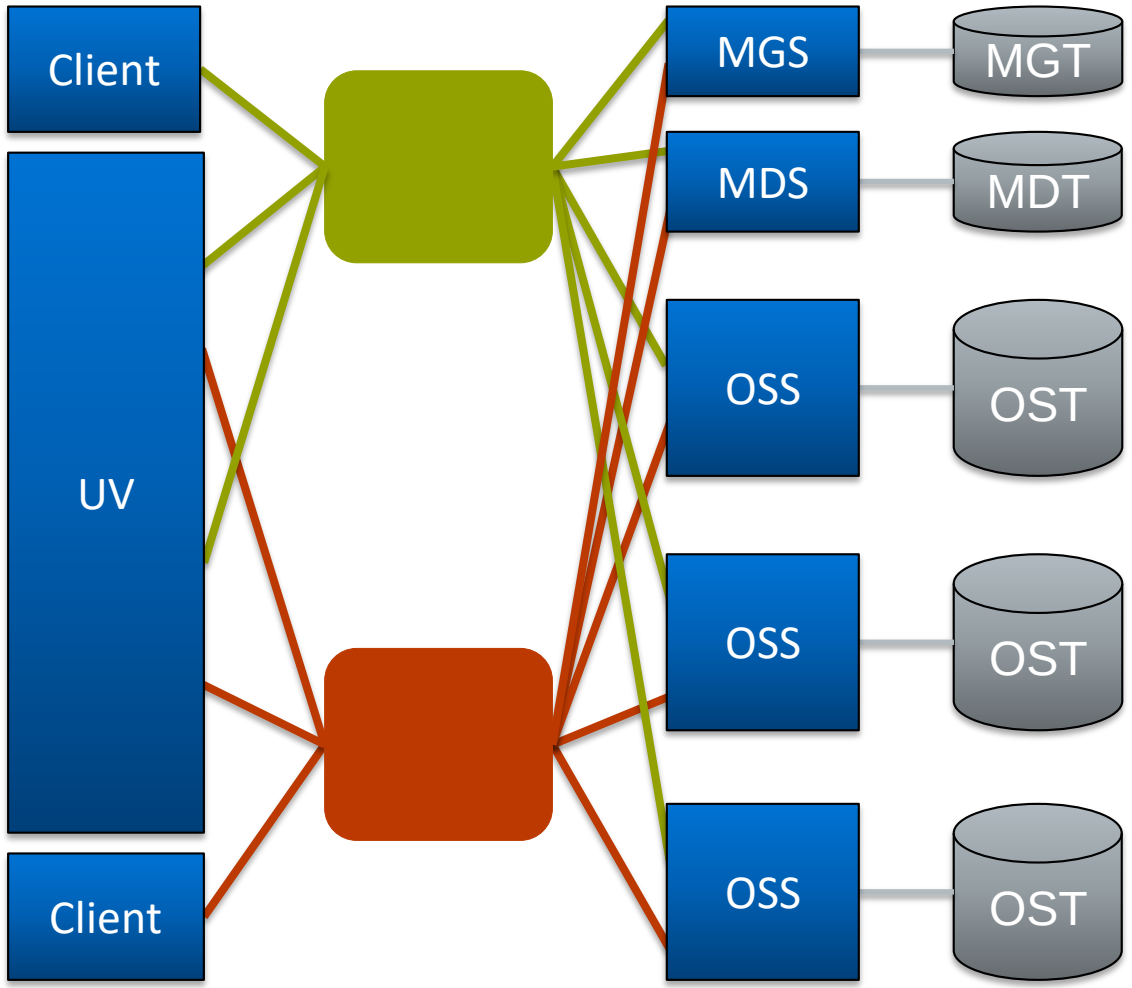


All nodes run Multi-Rail aware Lustre*.

The server nodes have also been given multiple interfaces.

Configuration of the other nodes is similar to how the OSS nodes are configured.

Multi-Rail Cluster: Multiple LNets



When there are multiple networks, all interfaces on all networks are used.

This is a case where specifying the NIDs of the MGS/MDS/OSS can be useful.

- The MGT/MDT/OST was built with the IP address on the **green** network.
- The **red** network is added.
- A client that only connects to the **red** network needs these peers configured.
- A client that connects to the green network can use peer discovery.

Configuring Peer Nodes

/etc/sysconfig/lnet.conf

- Peer configuration

```
peer:
  - primary nid: 10.0.0.1@o2ib0
    peer ni:
      - nid: 10.0.1.1@ob2ib1
  - primary nid: 10.0.0.2@ob2ib0
    peer ni:
      - nid: 10.0.1.2@ob2ib1
[…]
  - primary nid: 10.0.0.100@o2ib0
    peer ni:
      - nid: 10.0.0.101@o2ib0
      - nid: 10.0.1.100@o2ib1
      - nid: 10.0.1.101@o2ib1
```

Explicit peer configuration example:

- The colors refer to the networks in the previous slide.

The Primary NID identifies a peer.

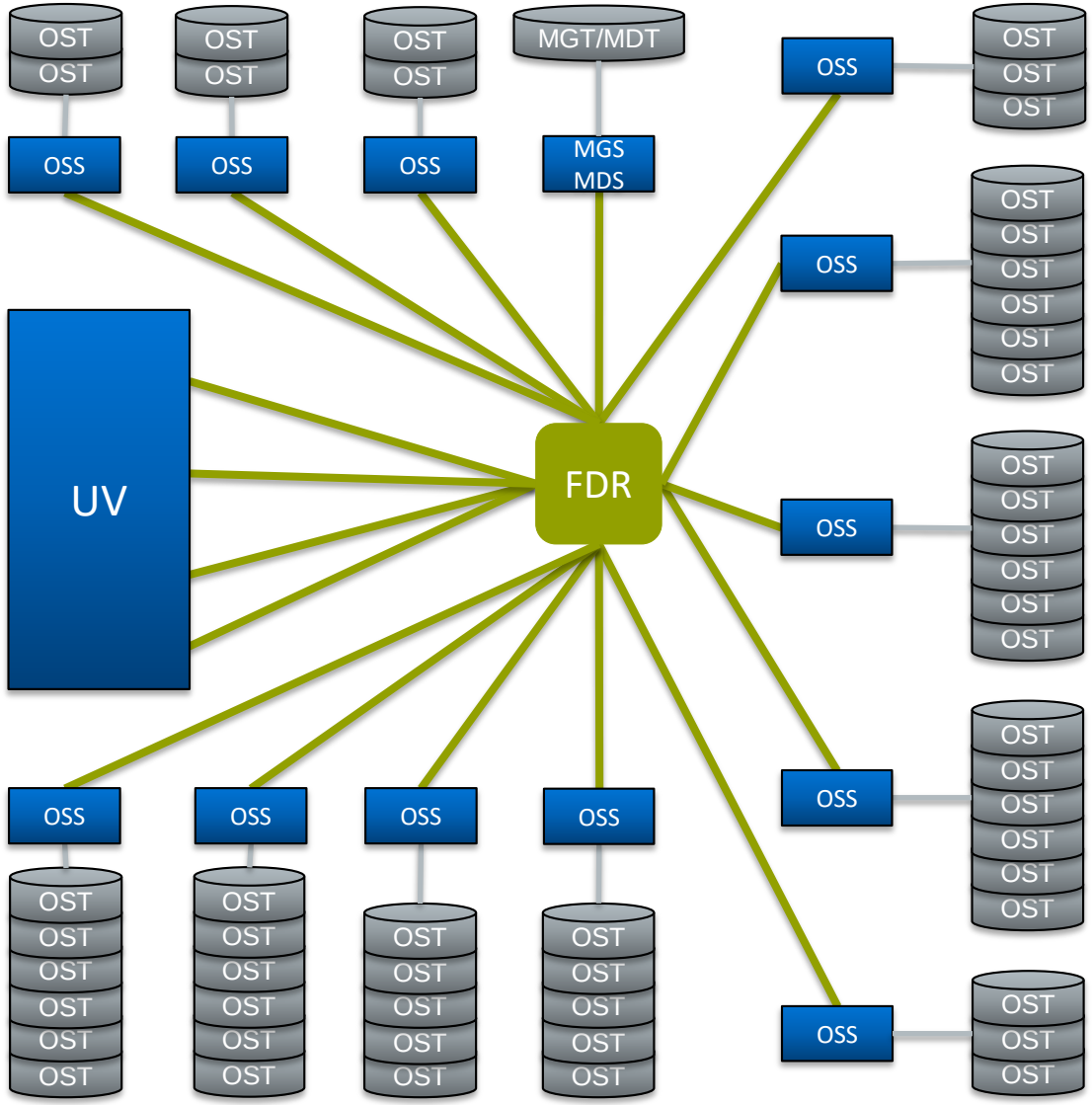
- LNet maps the NIDs to the Primary NID.
- The Primary NID shows up in log files.
- LNet logs both Primary and actual NID.

If peer discovery is also enabled, it will complain if it find a discrepancy.

- But the explicit configuration is used.

Performance

SGI's Multi-Rail Test Cluster



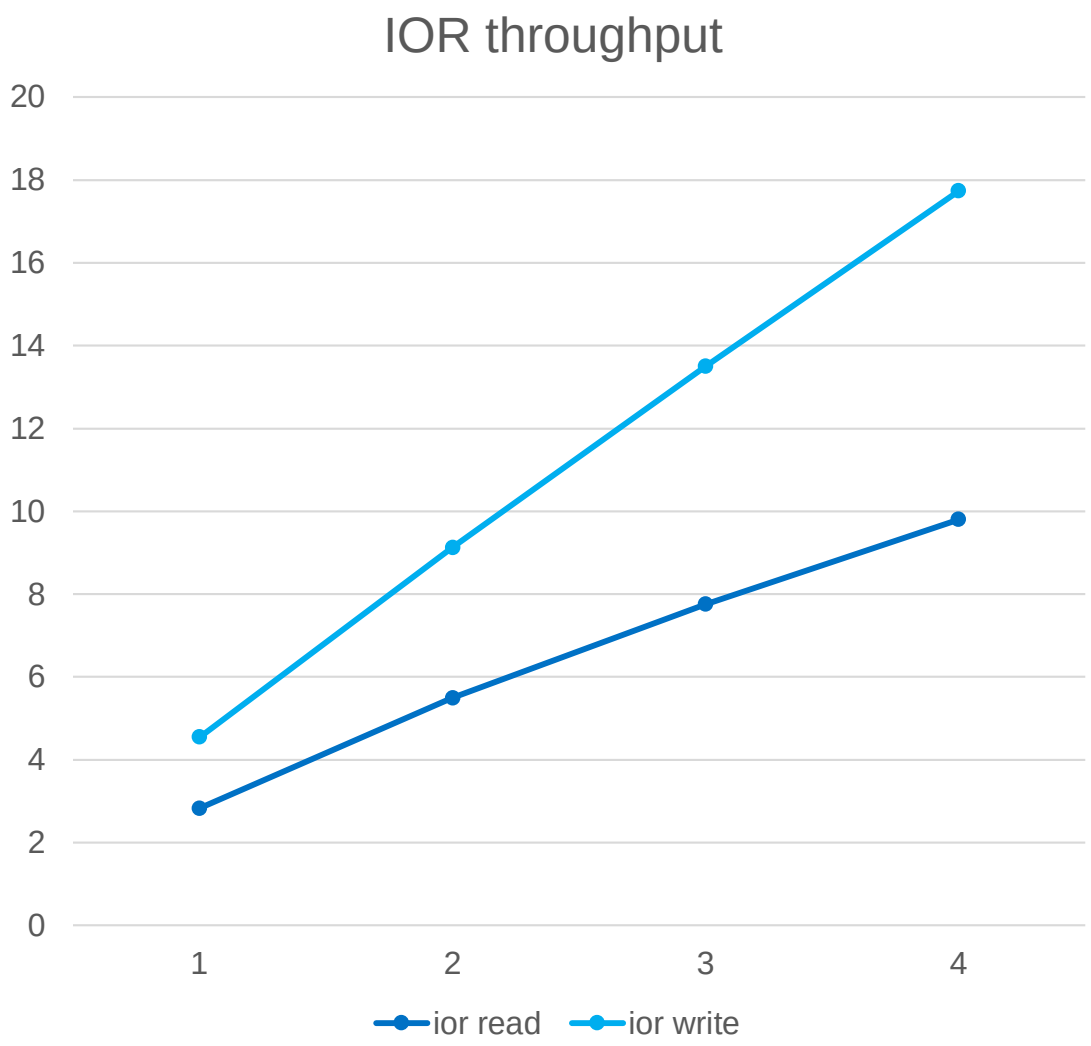
SGI's test cluster is built around a 16 socket/160 core SGI UV 2000.

12 OSS with 52 OST built from a gaggle of different hardware.

The fabric is FDR infiniband.

Using older hardware enables us to dedicate a system of this size to development.

Increasing Performance by Adding Interfaces



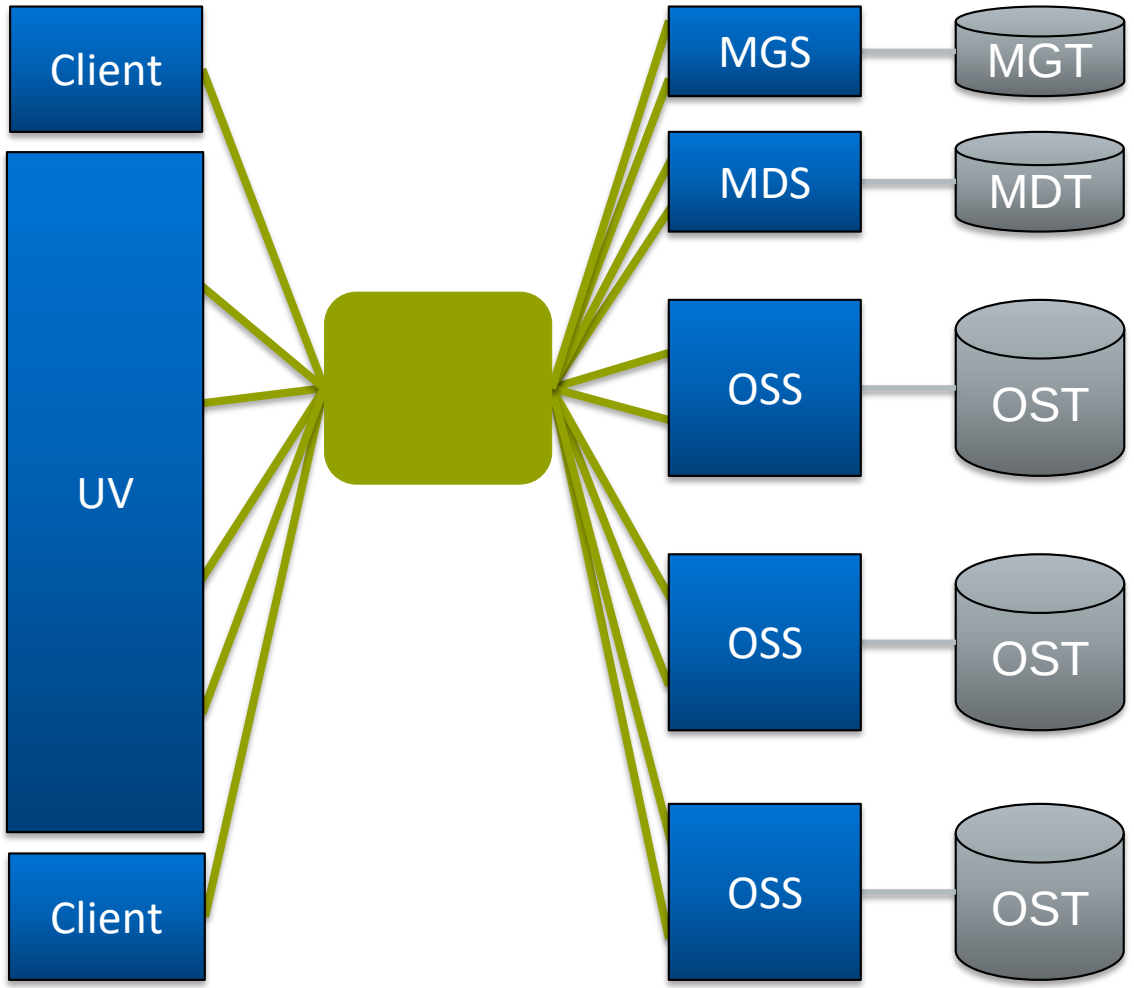
Older hardware does not lend itself to the latest in performance.

The plot gives I/O throughput in GiB/s against number of IB interfaces.

Performance increases almost linear with number of interfaces.

Upcoming Features

Multi-Rail for Resiliency

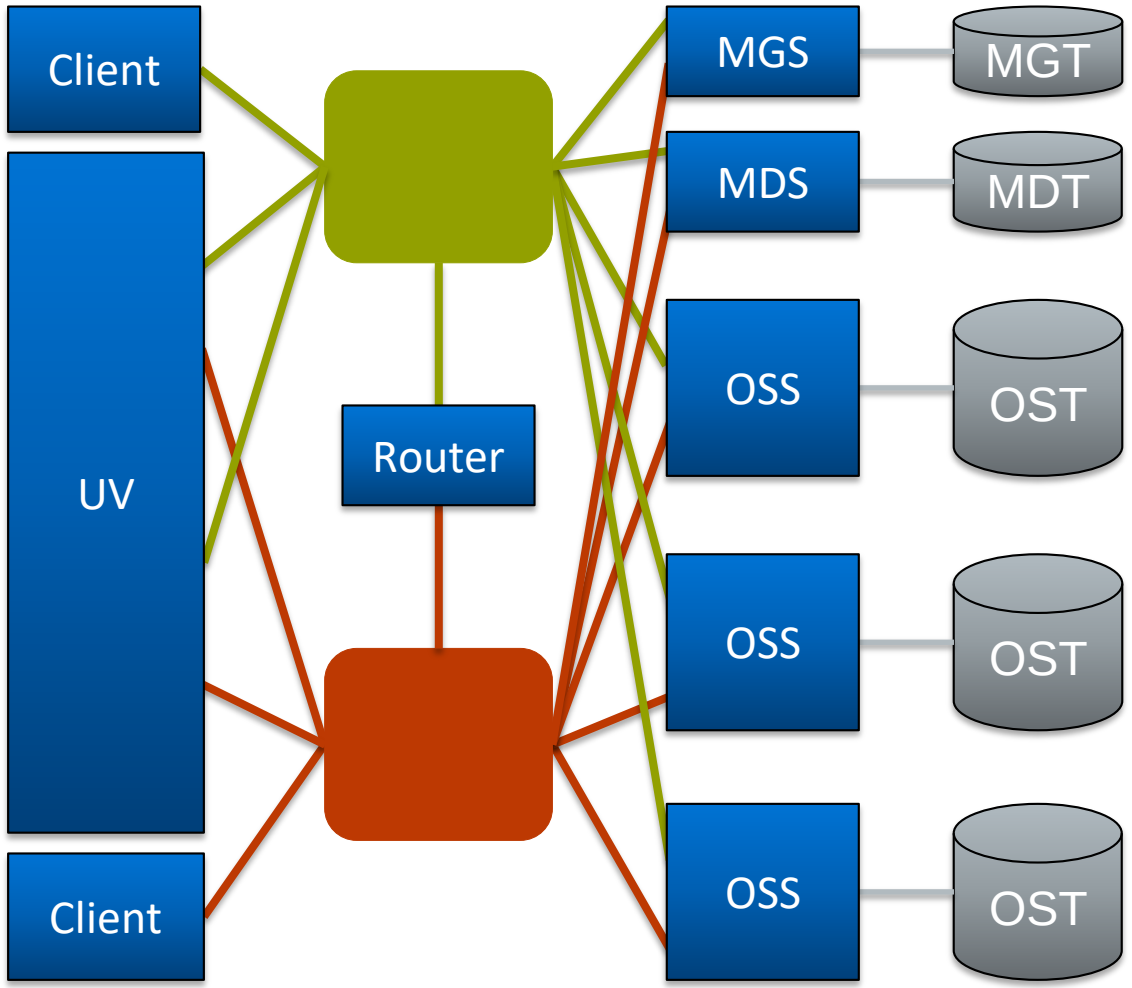


Nodes are connected with multiple interfaces.

If an interface / connection fails, the other interfaces are used to carry the load.

On failure, a message will be resent from and to a different interface, if one is available.

Connecting Multiple LNets for Resiliency

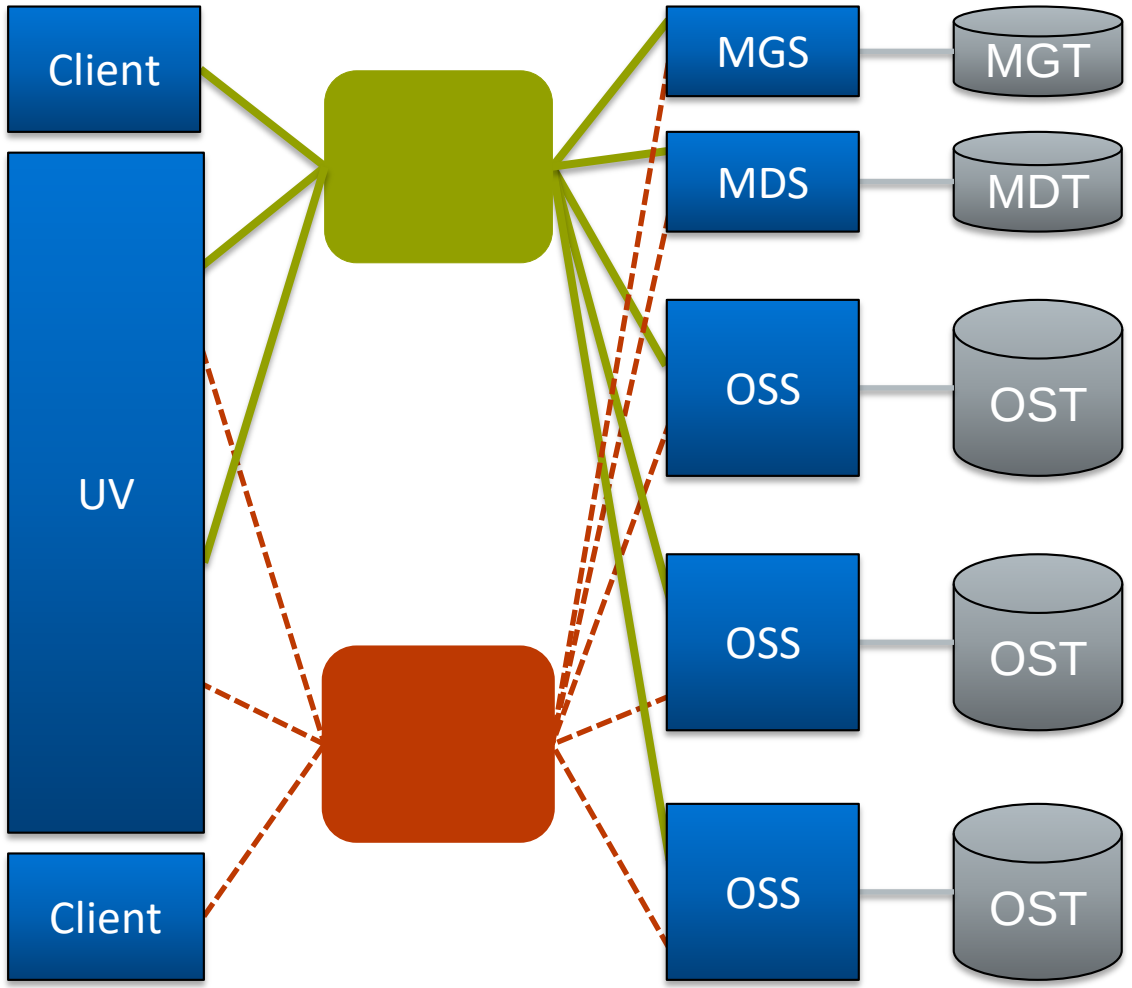


When there are multiple networks, they can be connected with routers.

The resiliency carries over to this configuration as well.

The router ensures that a single-connected client can still talk to the servers if a server interface goes down.

Prioritizing LNet Networks



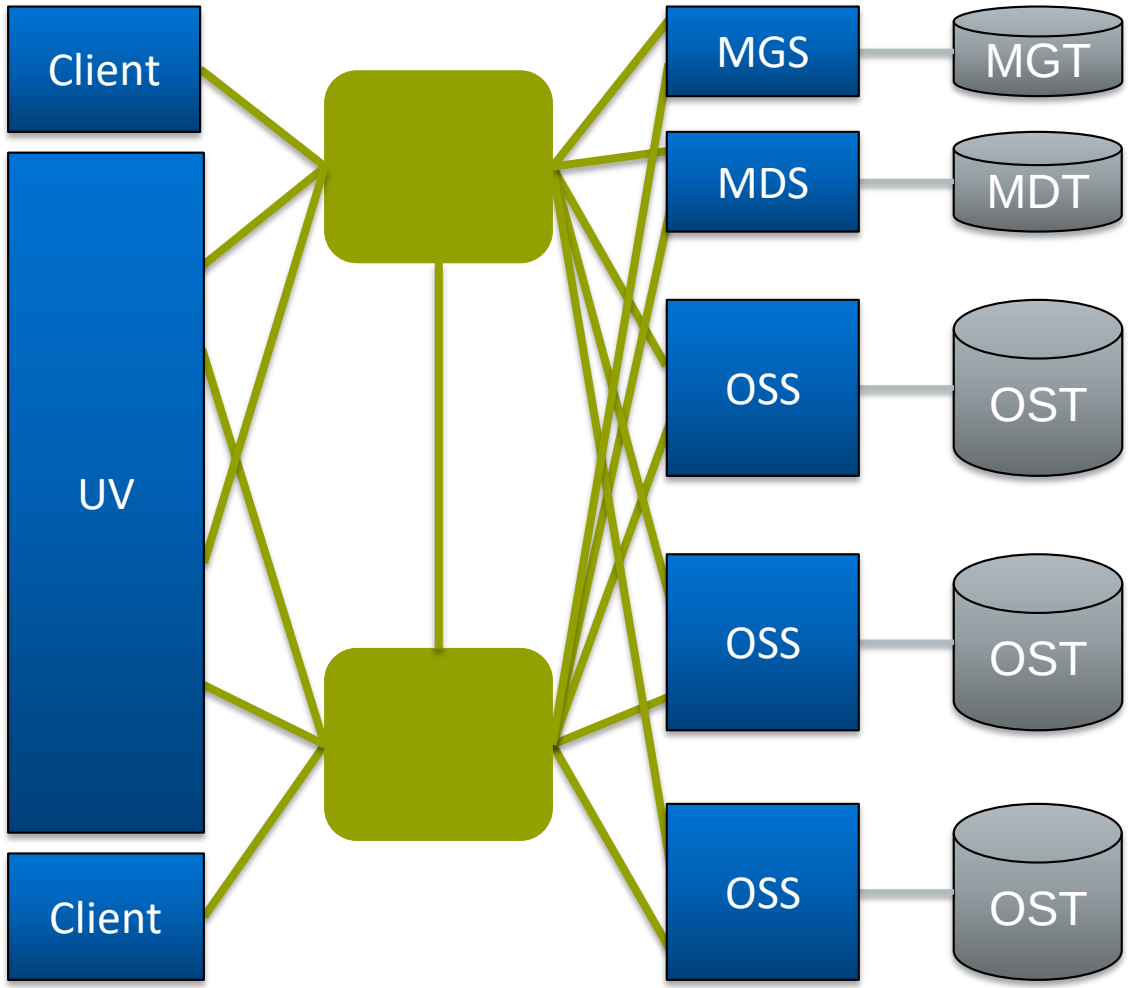
An active/standby network setup.

- A slow standby network.
- Or smaller capacity.
- Or load-balanced across networks.

The nodes that can use the active network will do so.

Nodes that can only use the standby network will fall back to using it.

One LNet with Bottleneck



Instead of a router a link connects two fabrics into one LNet network.

A link like this is a bottleneck.

Advanced routing makes it possible to avoid the bottleneck during normal operation.

The link will be used when a server loses one of its connections to the network.

Summary

Status of the Multi-Rail Project

Code development is done on the multi-rail branch of the master repository.

- Adding basic multi-rail capability
  - Committed to multi-rail branch
- Extend peer discovery to simplify configuration
  - In review on multi-rail branch
- Added resiliency
  - Under development
- Advanced routing rules
  - Under development

Estimated project completion time: end of this year.

Project Wiki Page

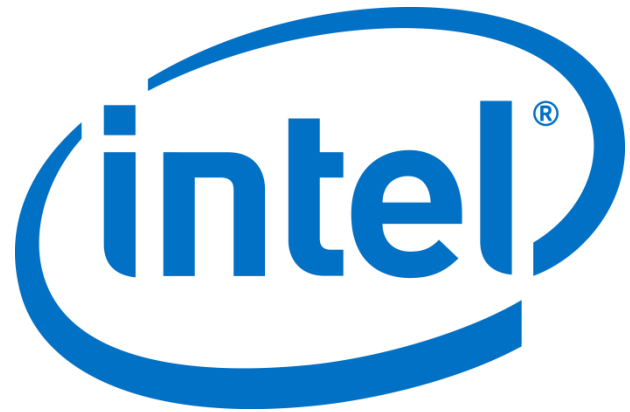
The public wiki page for the project is

htttp://wiki.lustre.org/Multi-Rail_LNet

Here you can find:

- Design documents
- Previous presentations
- Other documentation


sgi


intel