OPENFABRICS ALLIANCE

Lustre* Use of OFED and the Future

Doug Oucharek, Intel Corp.

#OFADevWorkshop

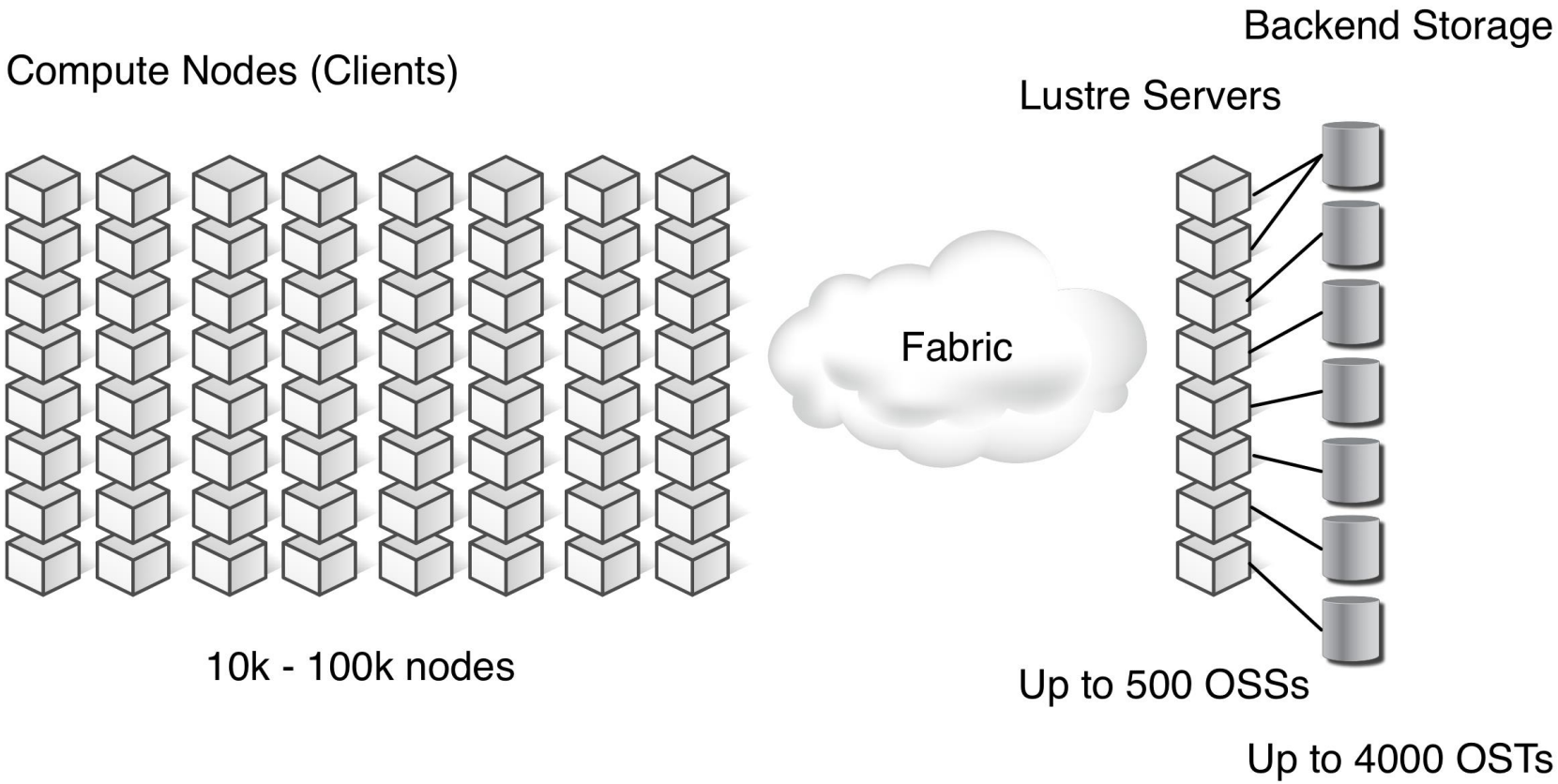
ANNUAL INTERNATIONAL OPENFABRICS SOFTWARE DEVELOPERS' WORKSHOP

* Some names and brands may be claimed as the property of others.

Overview

- Review of Lustre and future requirements on Fabric
- Summary of Requirements given at end
- A New Approach
  - Usually we see bottom-up style presentations
  - Trying top-down

50,000 ft View: The Product



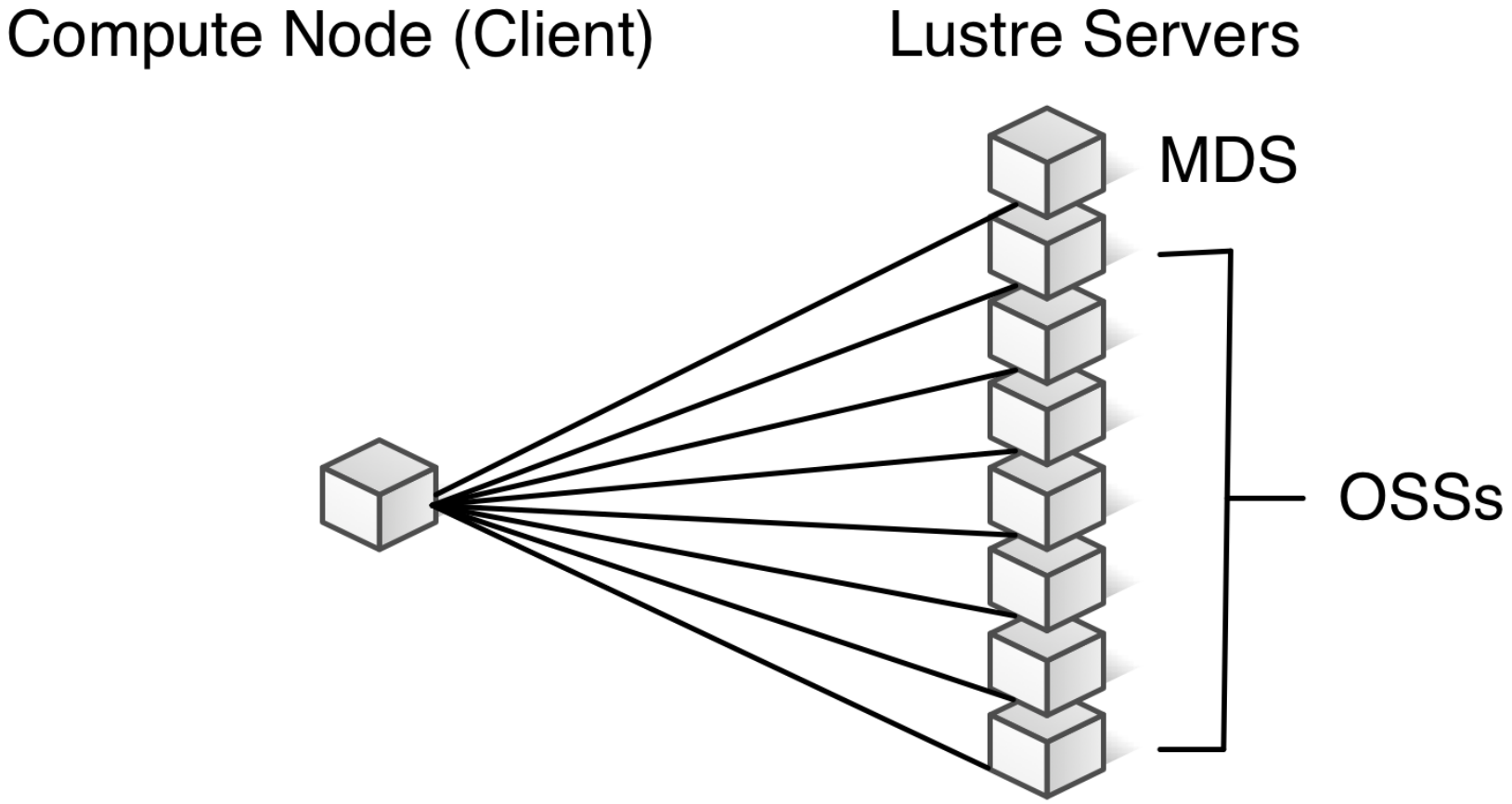
By the Numbers

	2012	2020
Nodes	10-100K	100K-1M
Threads/node	~10	~1000
Total concurrency	100K-1M	100M-1B
Object create	**100K/s**	**100M/s**
Memory	1-4PB	30-60PB
FS Size	10-100PB	600-3000PB
MTTI	1-5 Days	6 Hours
Memory Dump	< 2000s	< 300s
Peak I/O BW	**1-2TB/s**	**100-200TB/s**
Sustained I/O BW	**10-200GB/s**	**20TB/s**

Lustre Markets

- HPC
  - Core fan out -> Need equal access to Network Interface
- Enterprise
  - Security -> Encryption Support (IPSec-like)
  - Longer Distances (WAN) -> FMR or equivalent
- Cloud
  - VM Support by Fabric

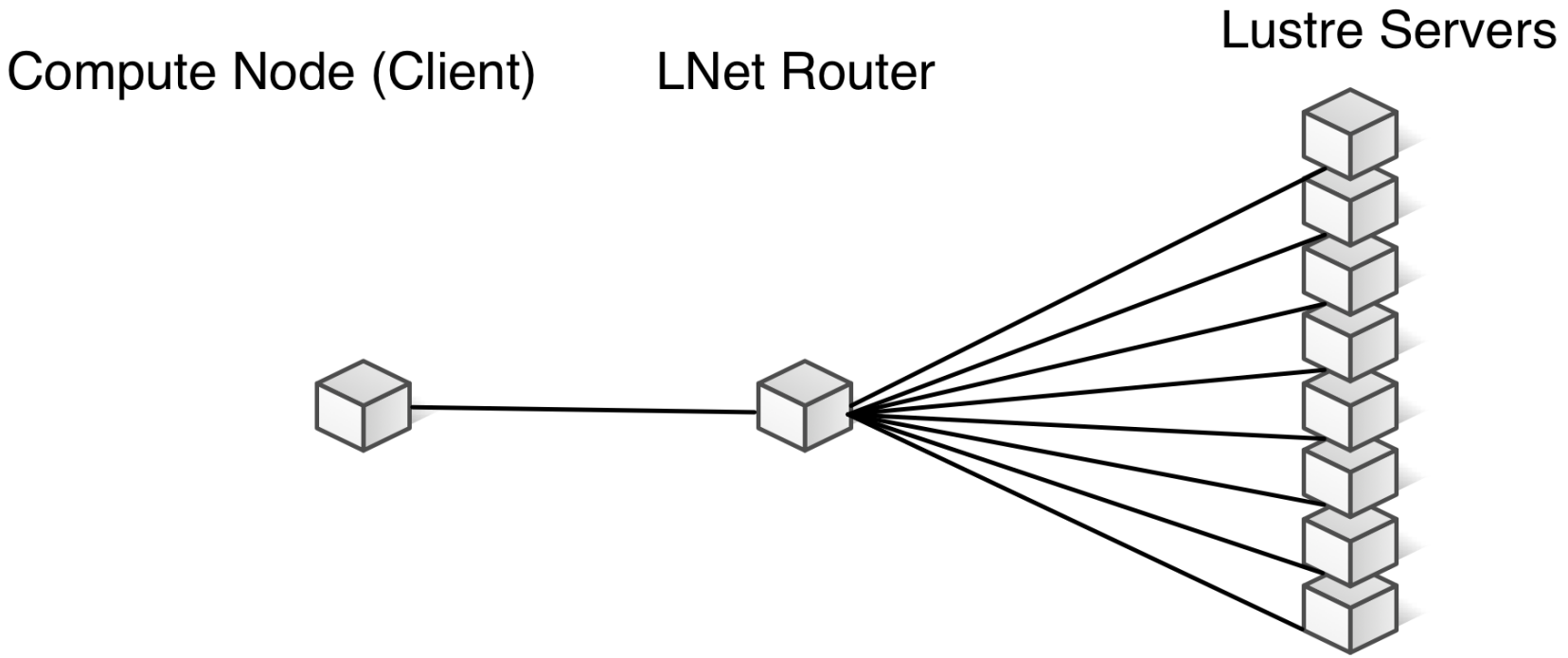
40,000 ft View: Client Server



Usage Pattern

- Striping
  - RAID0 Behaviour
- Peer Connections
  - Created on demand: RC QPs
  - Multiple Server Messages in Parallel
  - Monitor Connections via LNet Ping
- DNE : Multiple Meta-Data Servers
- Self Imposed Flow Control
  - Credits (for network) and Peer Credits (for each connection)
  - SLA with Fabric would be Great
  - Latency is OK, message loss is not

30,000 ft View: Getting Connected



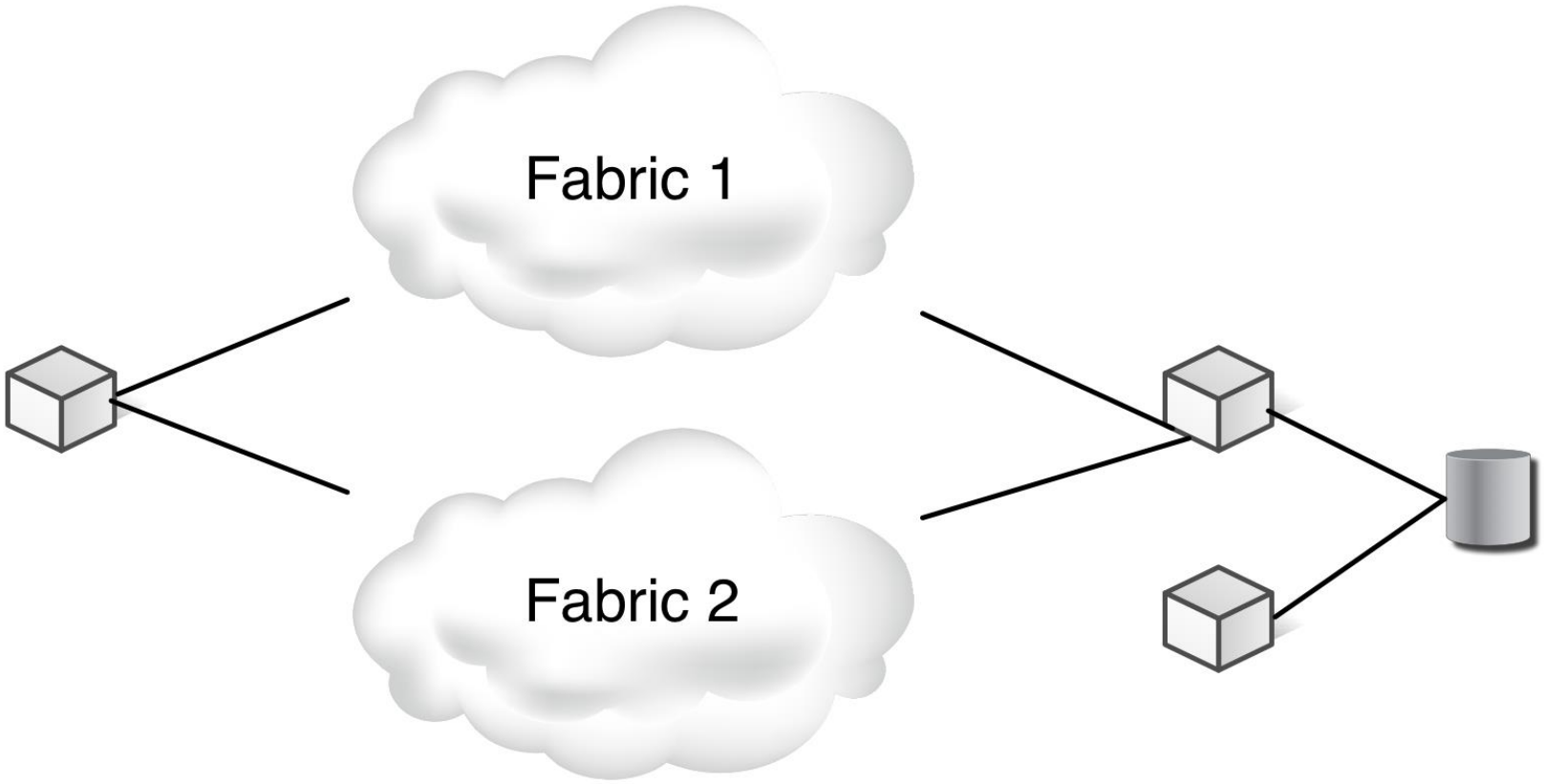
LNet Routing

- Why Route? (From ORNL paper: Network Contention and Congestion Control:Lustre Fine-Grained Routing – Matt Ezell)
  - Control **bandwidth** by varying the number of routers
  - Control **I/O paths** by selecting which routers to use
  - Control **route computation** by partitioning fabrics
  - If you are using multiple network types, you have to route traffic
- Required: Single OFED stack Optimized for all interfaces in router
  - Preferably upstream in kernel.org

Other Services

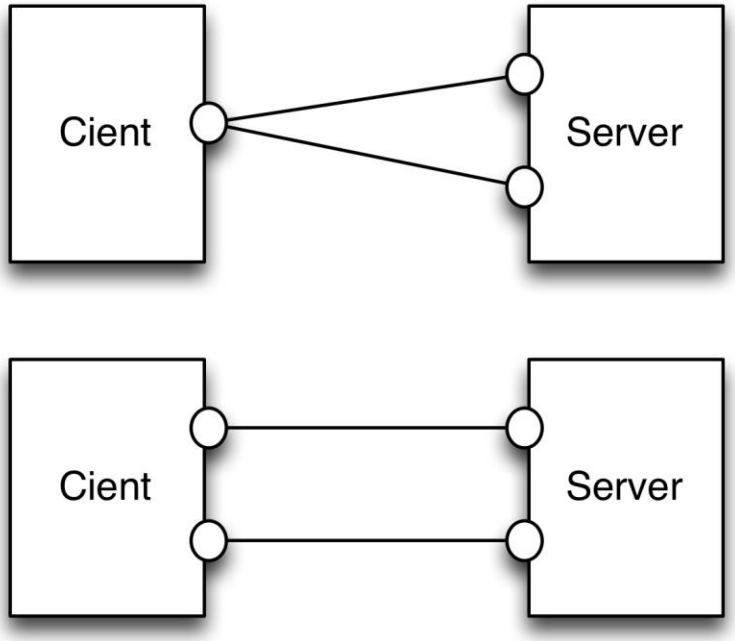
- Server Discovery
  - Use IPv4 (do not use IPoIB)
  - Can not support IPv6 any time soon (see 2012 LUG presentation: LNET Support for IPv6 is Long Overdue – Isaac Huang)
- Mostly Static Configuration
  - Painful for administration
  - Could really use DHCP for Cloud
  - Problem with Lustre/LNet and not fabric

20,000 ft View: Fault Handling



Channel Bonding

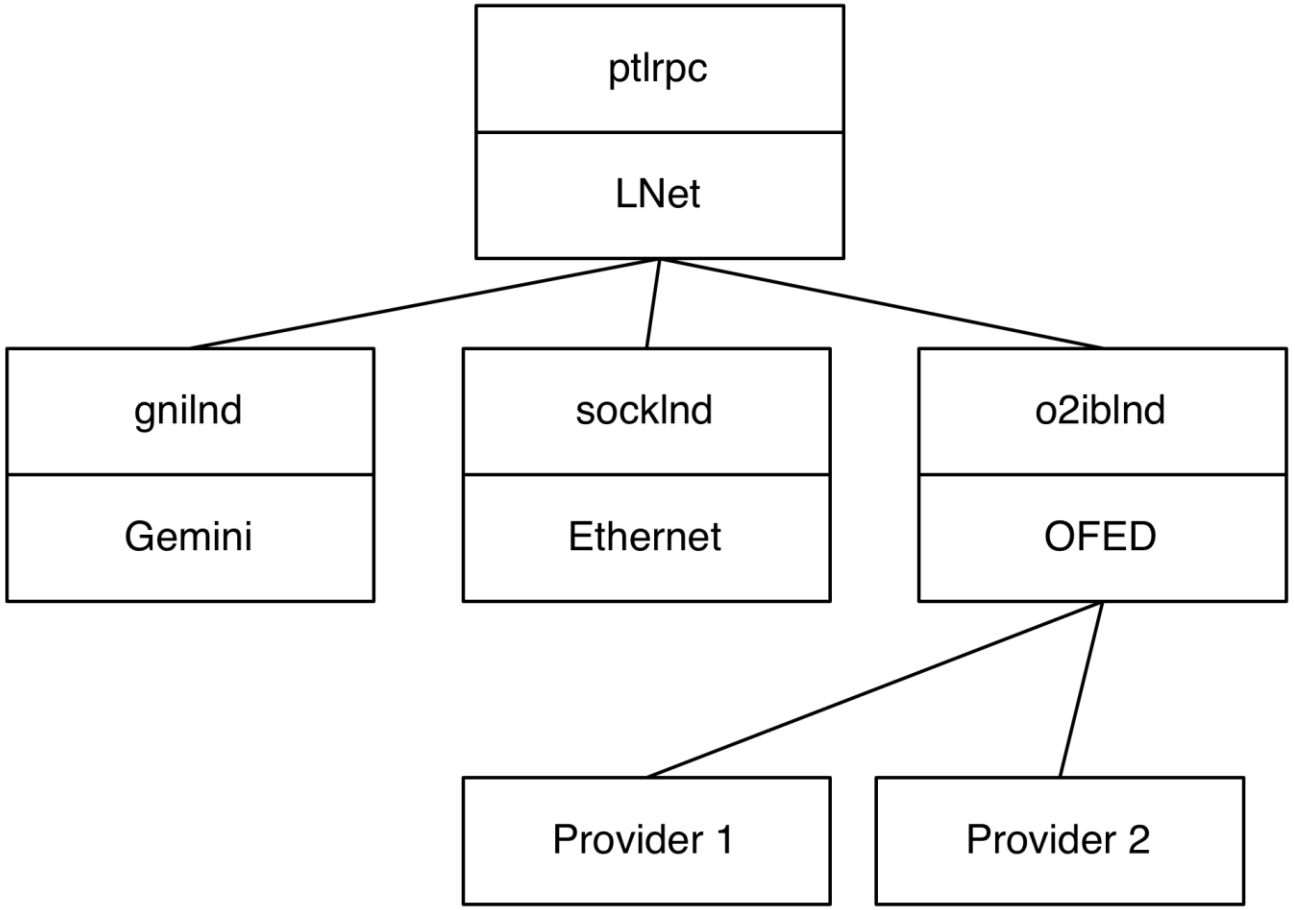
- Coming later this year
- Provides both:
  - Bandwidth
  - Redundancy
- Can Mix Fabrics supported by OFED
- Need: Good feedback when to abandon a pathway



Fault Handling

- Could use help with fault detection
  - Fault threshold passed in fabric
- Pings not good for scalable fault detection
  - See 2012 LUG paper: Lustre Ping Evictor Scaling in LNET Fine Grained Routing Configurations - Cray
  - Found a 4% - 11% reduction in throughput
  - Example: 25,000 clients, 360 OSSs, 4 OSTs per OSS:
  - 36M pings every 75s
- Need proper Health Network
  - High priority messaging in Fabric
- Packet arrival order: Does not matter to Lustre

10,000 ft View: Network Stack



OFED Usage

- Only PUTs Supported in o2iblnd
  - Therefore: only RDMA writes
  - Done to avoid limitations of RDMA reads
  - Need a way to manage RDMA read limitations
- Use Reliable Connections
  - Uses a lot of resources
  - Would be better if we could use Reliable Datagrams
- Choices:
  - <4k = Use Immediate
  - >4k = Use RDMA write

Runway: APIs used by o2iblnd

- ib_post_recv()
- ib_post_send()
- ib_poll_cq()
- ib_req_notify_cq()
- ib_create_cq()
- ib_destroy_cq()
- ib_mtu_enum_to_int()
- ib_destroy_fmr_pool()
- ib_create_fmr_pool()
- ib_fmr_pool_unmap()
- ib_fmr_pool_map_phys()
- ib_dereg_mr()
- ib_reg_phys_mr()
- ib_query_device()
- ib_dealloc_pd()
- ib_get_dma_mr()
- ib_alloc_pd()
- rdma_set_reuseaddr()
- rdma_resolve_addr()
- rdma_destroy_id()
- rdma_accept()
- rdma_connect()
- rdma_disconnect()
- rdma_create_qp()
- rdma_destroy_qp()
- rdma_bind_addr
- rdma_listen

Requirement Summary

1. Equal core access to Network Interface
2. Encryption support
3. Latency support -> FMR or equivalent
4. VM Support
5. SLA with Fabric
6. Single OFED stack Optimized for all supported interfaces. Preferably upstream (kernel.org)
7. Fault/Error Counters from Fabric
8. High Priority Messaging (Health Network)

Requirement Summary Cont

9. Control over RDMA read limitations from app
10. Support Reliable Datagrams

Thank You

OPENFABRICS ALLIANCE

ANNUAL INTERNATIONAL OPENFABRICS SOFTWARE DEVELOPERS' WORKSHOP