Cray's Implementation of LNET Fine Grained Routing CUG 2013

Mark Swan, Cray Inc.

Nic Henke, Xyratex International Inc.

Introduction

- **LNET Primer**
  - Background
  - Flat LNET
  - LNET Fine Grained Routing
  - Bandwidth matching
- **The Complexity Issue**
- **Complexity Reduction**
  - Cray LNET Configuration and Validation Tool (CLCVT)
  - Other live validation
- **Future Work**
  - My LNET routers are where?
  - My LNET IB network looks like what?
  - There is something other than Cray Sonexion?

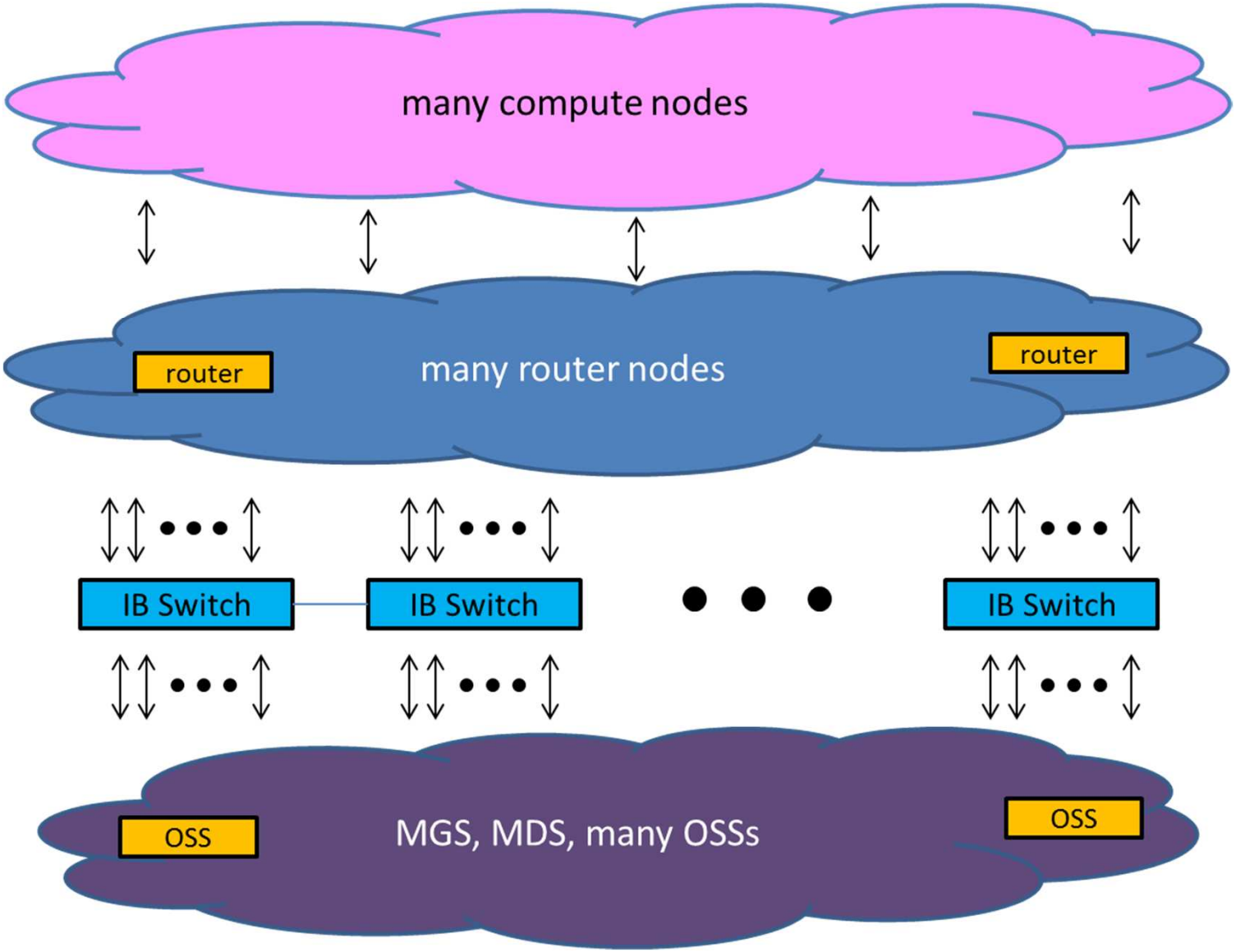
LNET Primer
Background

- LNET is the Lustre NETworking layer
- LNET routers are the bridge between Cray's high speed network (SeaStar, Gemini, Aries) and external Lustre servers (Infiniband).
- LNETs define what pool of LNET routers can be used to communicate with a given MGS, MDS, or OSS.

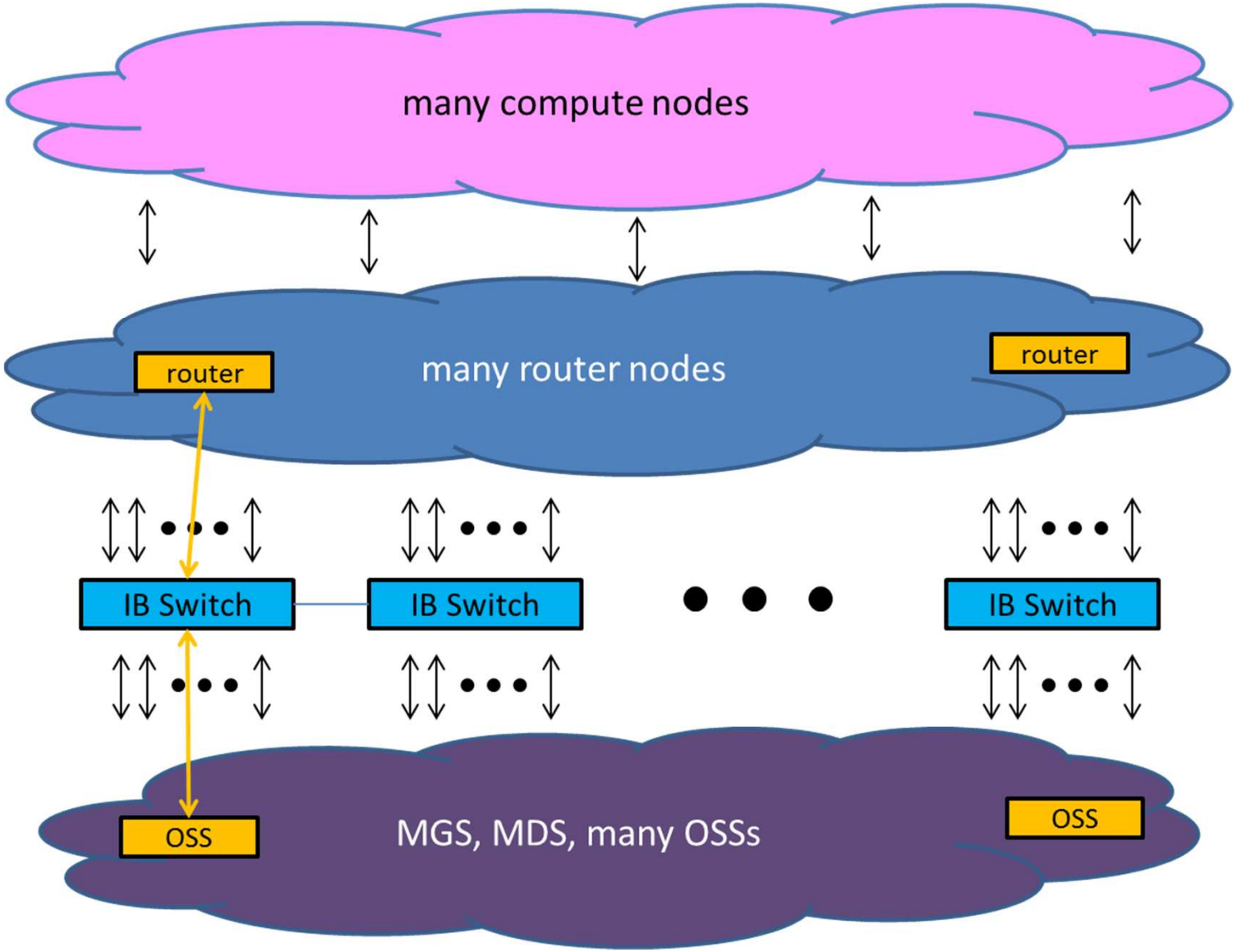
LNET Primer
Flat LNET

- All external Lustre servers (MGS, MDS, OSSs) are assigned to the same LNET.
- Flat LNET is used by esLogin and “whitebox” clients.
- All LNET routers are used round-robin to access any of the Lustre servers.
- No preferred pathways.
- Works just as good as Fine Grained Routing at very small scale.
- Some paths are high performance (when the LNET routers are plugged into the same Infiniband leaf as the server).
- Some paths are horrible (multiple hops across Inter-Switch-Links (ISLs) between Infiniband switches).
- For large numbers of routers and servers, you will never achieve maximal performance.

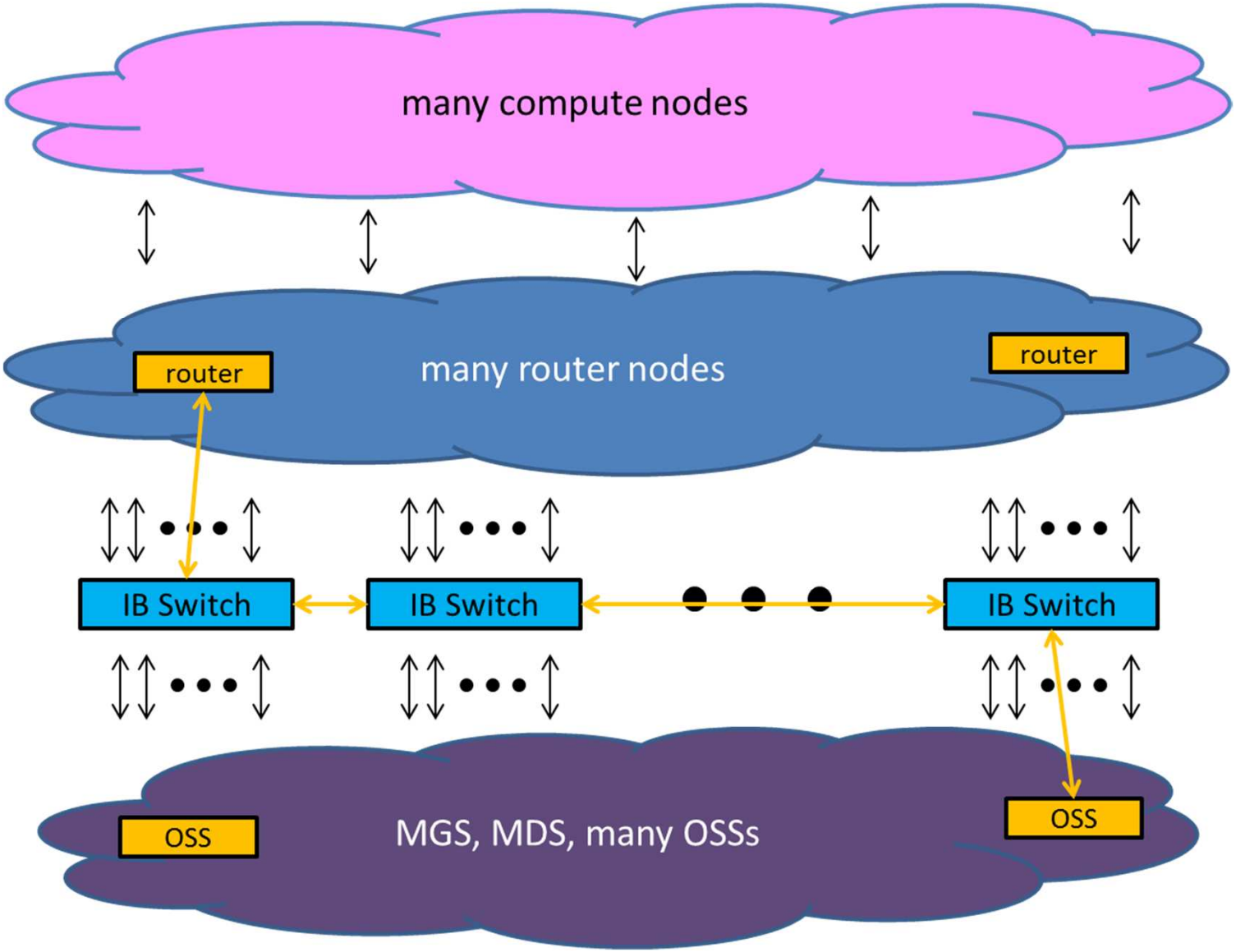
LNET Primer
Flat LNET (cont'd)



LNET Primer
Flat LNET (cont'd)



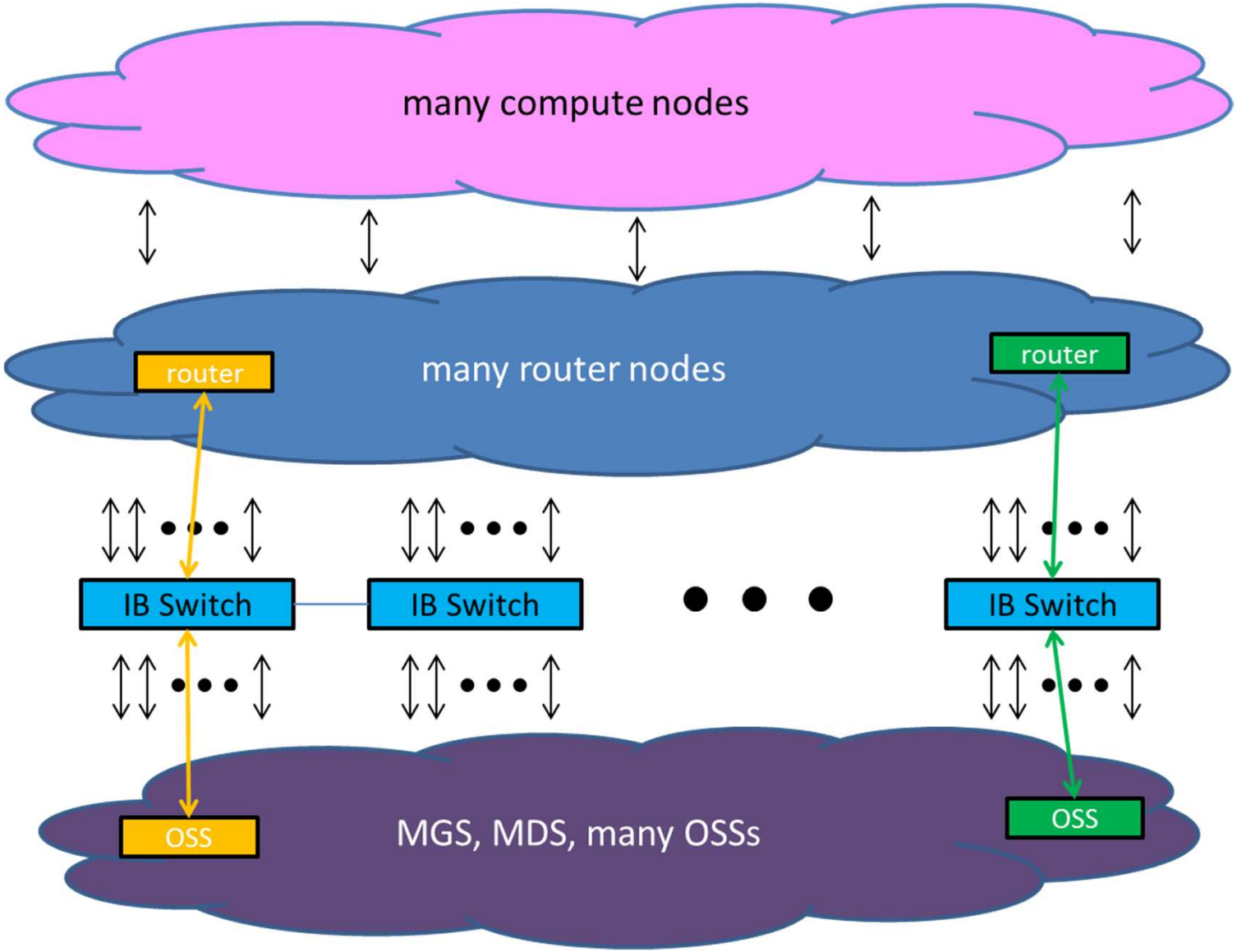
LNET Primer
Flat LNET (cont'd)



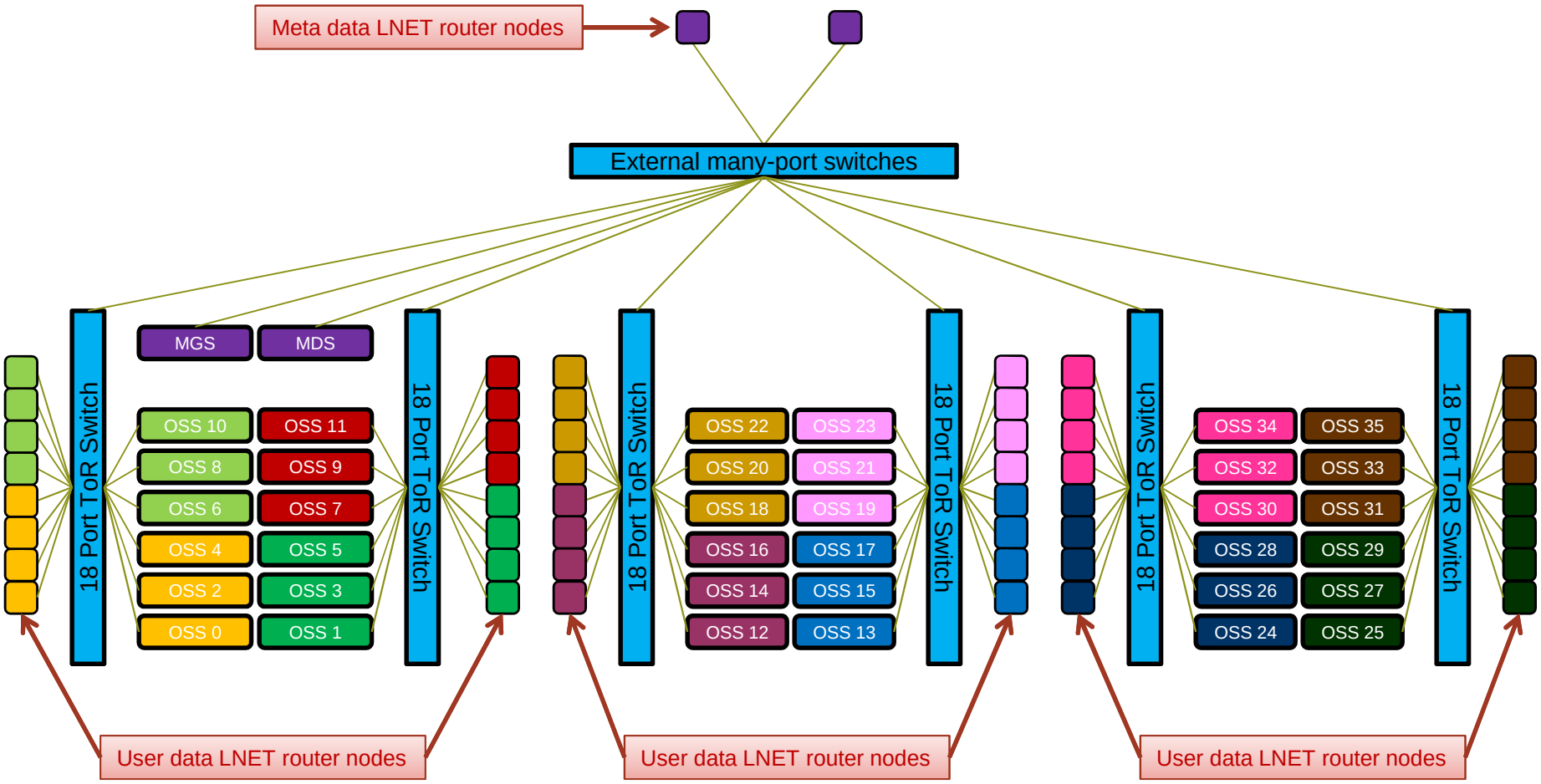
LNET Primer
LNET Fine Grained Routing (FGR)

- More narrowly defines the highest performance pathway from a client (compute node) to a server (MGS, MDS, OSS).
- Maximizes bandwidth between a set of LNET routers to a set of OSSs so I/O is driven at highest rates.
- Avoid Inter-Switch-Links (ISLs).
- Scales perfectly (as far as we can tell).
- Not a Sonexion-only solution.

LNET Primer
LNET FGR (cont'd)



LNET Primer
LNET FGR (cont'd)



LNET Primer
Bandwidth matching

	1	2	3	4	5	6
Sonexion-1600 OSS	3.0	6.0	9.0	12.0	15.0	18.0
Cray XE LNET Router	2.6	5.2	7.8	10.4	13.0	15.6
Cray XC30 LNET Router, single HCA	5.5	11.0	16.5	22.0	27.5	33.0
Cray XC30 LNET Router, dual HCA	4.2	8.4	12.6	16.8	21.0	25.2

* Rates are in GigaBytes per second (GB/s)

The Complexity Issue

- **Scale**
  - 10s to 100s of hosts, at least 2x that for cable endpoints
  - For each host we need multiple pieces of information
- **Numbers Are Confusing**
  - 172.16.4.3, 172.16.4.4, 172.16.4.5, 172.16.4.6
  - 172.16.4.3, 172.15.4.4, 172.16.5.4, 172.16.4.5
- **Formats**
  - ibnetdiscover
  - [4] "H-0002c9030052bc2c"[1](2c9030052bc2d) # "yellow04 HCA-1" lid 71 4xQDR
- **When Things Go Wrong**
  - Actual vs Intended vs Recorded Intentions

Complexity Reduction
Cray LNET Configuration and Validation Tool (CLCVT)

- Simple and descriptive input file format
- Understands Cray Sonexion IB switch configuration
- Handles multiple Cray clients connected to multiple Sonexion file systems
- Generates secondary routes for LNET groups
- Generates appropriate “lnet.conf” file contents for each system
- Generates a cable table
- Performs live validation of IB connectivity
- Performs live validation of LNET group membership
- Performs live validation of LNET destinations (i.e., cable check)

Complexity Reduction
CLCVT (cont'd)

```
[info]
clustername = snx99999n
SSU_count = 2
clients = mark

[mark]
lnet_network_wildcard = gni0:10.128.*.*

o2ib1000: c0-0c2s2n0, c0-0c2s2n1, c0-0c2s2n2, c0-0c2s2n3
o2ib1002: c0-0c2s2n0, c0-0c2s2n1
o2ib1003: c0-0c2s2n2, c0-0c2s2n3

[snx99999n]
lnet_network_wildcard = o2ib0:10.10.100.*

o2ib1000: snx99999n002, snx99999n003 ; MGS and MDS
o2ib1002: snx99999n004, snx99999n006 ; OSSs 2/4
o2ib1003: snx99999n005, snx99999n007 ; OSSs 3/5
```

Future Work

- Application I/O could potentially benefit from knowing where [in the file system] their data exists and which LNET routers service those OSTs. The application placement tool would need to understand this information and place the application (or pieces of it) near the appropriate LNET router nodes.
- CLCVT could provide better live validation if it understood the IB topology.
- CLCVT could do live validation on the server side too.
- CLCVT is mainly used for FGR implementations for Cray Sonexion and could benefit from understanding other Lustre implementations such as Cray esFS.

Thank you