Linux Clusters Institute: HPC Networking

Kyle Hutson, System Administrator, Kansas State University

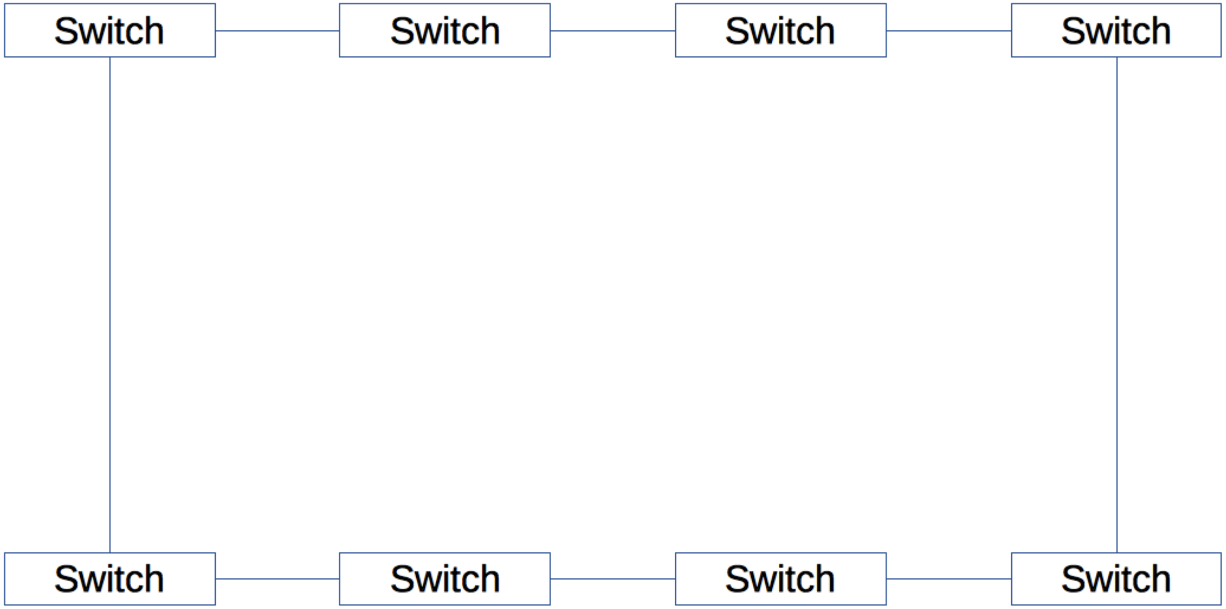
Network Topology

- Design Goals:
  - Maximum bandwidth between any two points
  - Minimum latency between any two points
  - Minimize hops – most of the latency is in switching/processing
  - Keep the cables short
    - ... Or use less of them
  - Costs are per {switch, port, cable}
  - Want collective operations to be fast (for most HPC workloads)
  - Redundancy, filesystems, etc may impose additional requirements

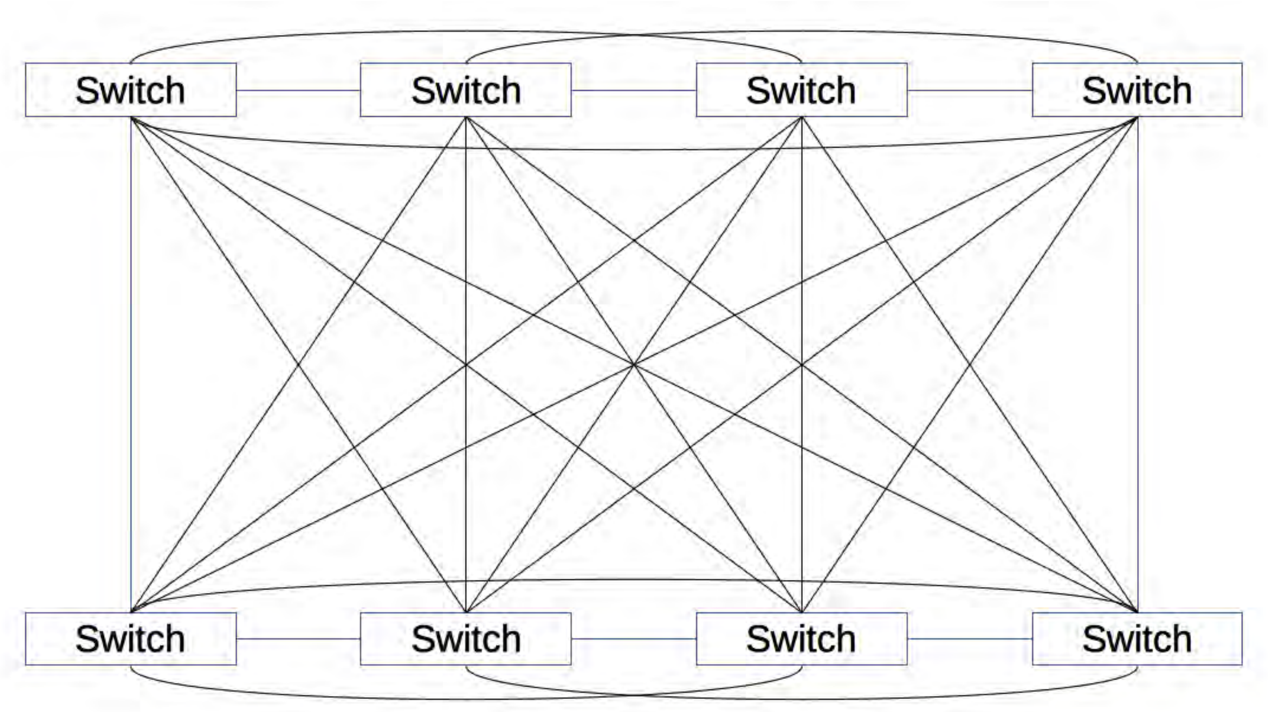
Network Topology

- Let’s start the discussion by physical and logical topologies
- I’ll give some examples and you tell me the advantages and disadvantages of each
    - (Hint: If there were no disadvantages to any of these, we would only have one slide which showed what everybody does)

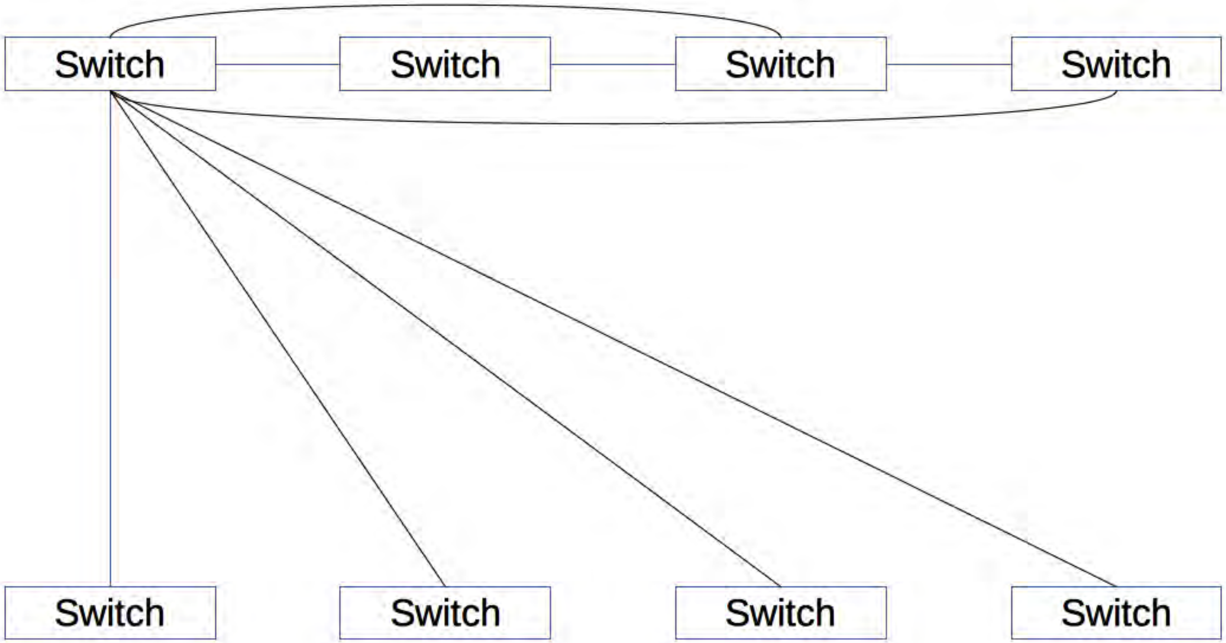
Network Topology



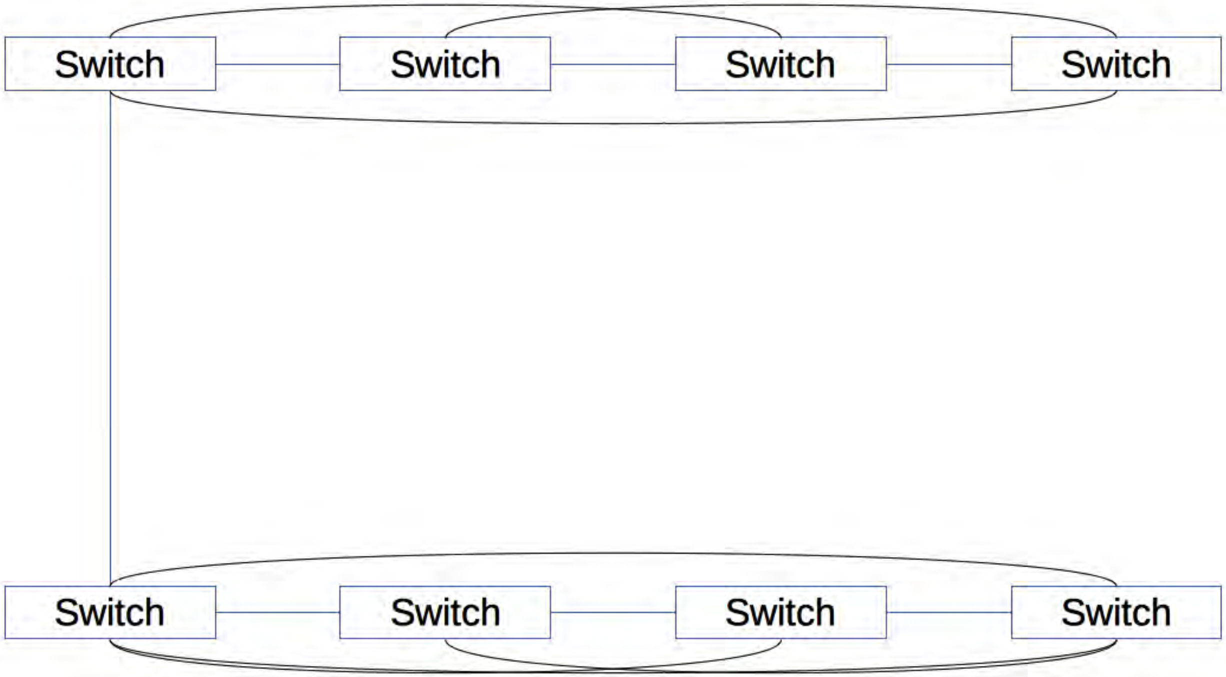
Network Topology



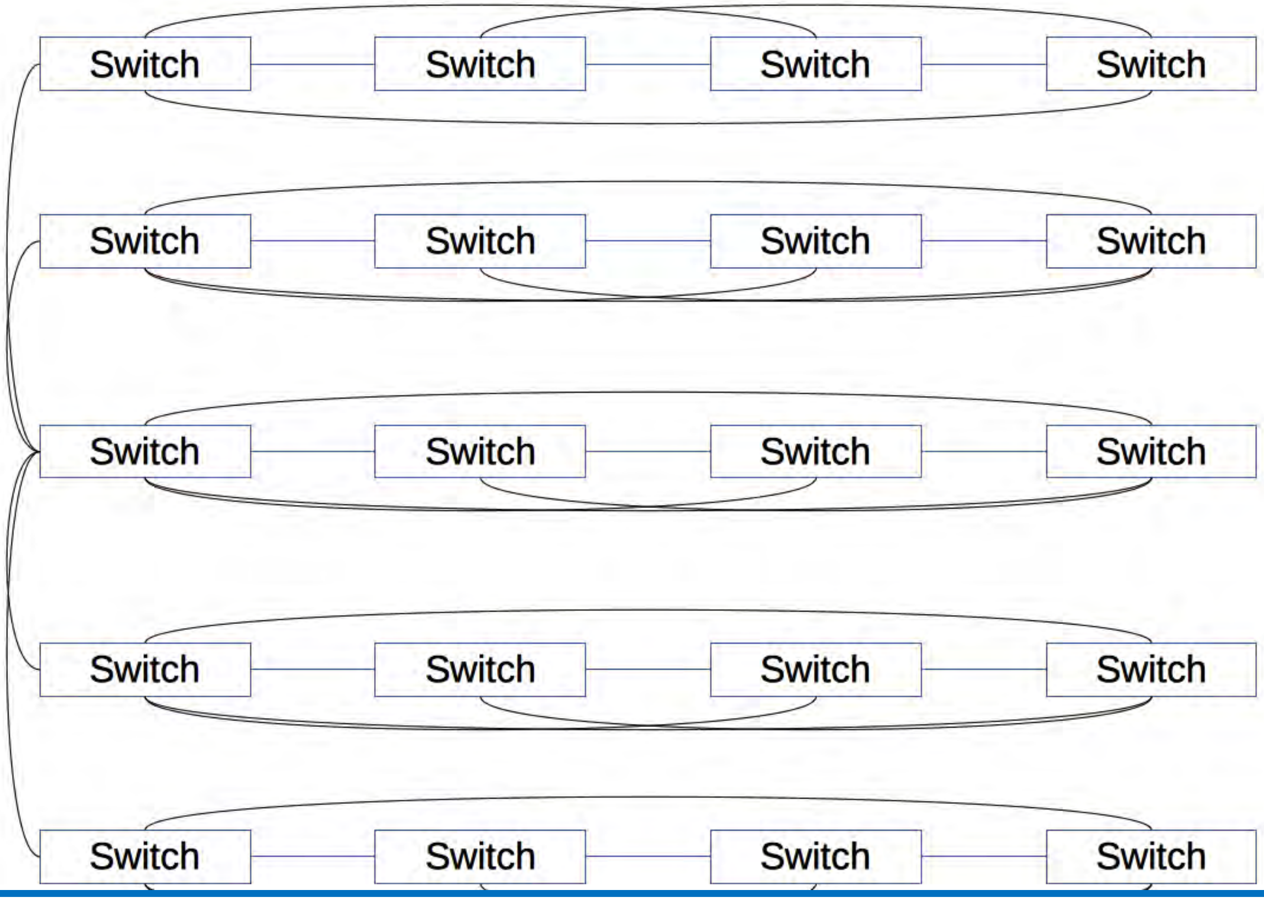
Network Topology



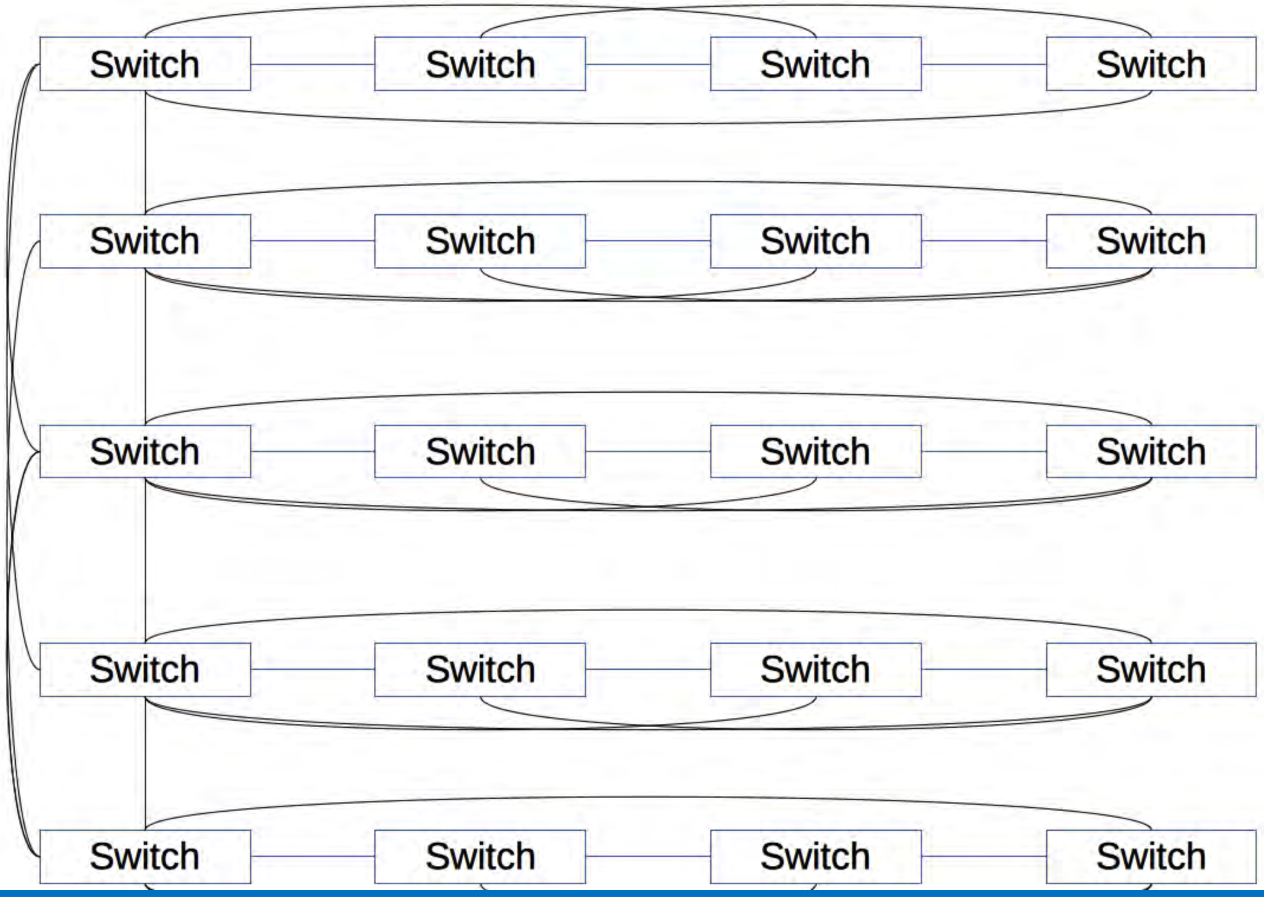
Network Topology



Network Topology



Network Topology



Ethernet

- Where do we use it?
- Overview
  - Switched – packets go only where they need to
  - MAC address table – scaling
- What will a good managed switch give you?
  - Interface for managing
  - VLANs
  - MAC address tables
  - Switch port functions (speed, MTU, uplink)
  - Statistics
  - Stack functions
- What will a bad managed switch give you?
  - Headaches
  - Interoperability issues

Ethernet

- Advantages:
  - Cost
  - Ubiquitous and well-understood – Most problems are Google-able
  - Can be quite fast (100Gbps)
  - Many vendors
  - Versatile
    - Auto-speed-sensing and changing (at least on BaseT hardware)
    - VLANs
    - Spanning Tree
    - LACP (aka “port bonding”)

Ethernet

- Disadvantages
  - Poor latency
  - Lossy (unless you spend $$$$ on switches and NICs)
  - Occasional interoperability issues
  - Lots of little things that can break
  - Because things can be “mostly working” can be difficult to pinpoint

Ethernet

- Typically higher latency and more complexity than IB
- Plug and play, in theory
- Higher speed variants tend to be a bit less reliable than you'd expect
- Familiar software TCP/IP is a first class citizen
- RoCE provides VERBS on certain vendor's hardware
- Not that common in the real world
- Mellanox has a VERBS emulation layer and LibFabrics can operate over TCP if you don't have RoCE

High Speed Interconnect

- Significant percentage of the cost of a cluster (1/3 or more)
- High bandwidth, low latency
- Exact requirements depend on cluster workload
- Typically, want some hardware offload capability
- RDMA
- Needs to present a software interface compatible with your desired middleware (MPI?)
- Reliability desirable (may carry filesystem and other important traffic)
- Typically short range (single room, or at least single building)
- Proprietary and open solutions available

Hardware Offload - RDMA

- “Remote Direct Memory Access” – The CPU designates a memory region, which the network cards then transfer
- CPU can (and should!) continue doing computation during the transfer
- Exposed to user software through various special purpose APIs
- Most commonly: Verbs
- LibFabric is an up-and-coming alternative
- One API for a variety of hardware (at least in theory)
- Some Vendors have their own – PSM, Portals, etc
- Software often wrapped with a higher level middleware layer, almost always MPI for HPC use cases
- The Open Fabric Alliance conference/mailing lists are a good source of information about RDMA

Hardware Offload – RDMA – High level workflow

- Create a TCP/IP connection to the nodes of interest and exchange their high-speed interconnect addresses (IB and similar generally don’t have a DNS like thing)
- Pin a memory region for transfer on both nodes (so the kernel won’t move the data before the network card is done)
- Open a context to the network card and create a queue pair
- Exchange metadata about the transfer over TCP
- Put a work request in the queue specifying the region to transfer
- Wait for a completion message (or better, do something else with the CPU).
- Cleanup

RDMA Configuration Pitfall: Pinned Memory

- Memory must be “pinned” for RDMA transfers so that the NIC knows where to find (or put) the data being transferred is in physical memory and can be sure that the OS won’t move it
- Having a lot of pinned memory can interfere with normal OS operations like paging and the handling of out-of-memory events, so the amount a user can pin is usually restricted
- If a user can’t pin their entire buffer, the transfer will fail (unless they are using very smart middleware)
- This is implemented on Linux with “ulimit”
-
  ```
  printf “%s\n%s” “soft memlock unlimited” “hard memlock unlimited” >> /etc/security/limits.conf
  ```

Interconnect Hardware - Options

- InfiniBand
- Ethernet (the high-end variety – 40/50/100Gbit)
- OmniPath
- Cray Proprietary (Aries at the moment)
- SGI Proprietary (NUMALINK)
- Historically:
  - IBM Proprietary – BlueGene
  - Myrinet

InfiniBand

- Popular since the early 00's
- "Lossless"
- Standard, but only a few current manufactures
  - Nearly everyone (with IB) in HPC uses Mellanox gear (or someone else's stuff made with Mellanox ASICs)
  - Oracle and Obsidian also have some IB hardware
  - QLogic's IB was bought by Intel in 2012
- Speeds are rated by generation and number of lanes
  - Current generation is EDR: 25 gigabits/lane
  - Current implementations almost always use 4x – effectively 100 gigabits
  - FDR (14gb/lane) and QDR (8gb/lane) can sometimes still be found in the wild
- Modern implementations use the QSFP+ connector (fiber or short copper runs)

InfiniBand

- Spanning-tree-like capabilities are built-in (via the SM)
- Generally the switch side “Just Works”™
- Auto-speed changes
- Client-side gets trickier, especially for performance tuning
  - OFED (Open Fabrics Enterprise Distribution)
  - Tweaking kernel parameters
- Do yourself a favor – don’t mix vendors. It theoretically works but…
- I suggest also keeping firmware versions consistent
- You can mix speeds (with caveats)

InfiniBand

- Centrally Managed (everyone uses OpenSM)
- Verbs RDMA is the native API
- IP (or even emulated Ethernet with recent kernels) available with significantly reduced performance
- Statically routed via a per-port linear forwarding table in nearly every case
- Subnet routing barely supported (hardware that can do this is very new)
- 16bit assigned addresses
- 64bit hardware addresses
- 128bit addresses for inter-subnet routing (again, barely any real hardware)

InfiniBand - Terms

- LID: Local assigned address (like an IP address)
- GUID: Hardware address (like an ethernet MAC address)
- SM: Subnet manager: Centralized software that determines the routing table and various other network properties
- Queue Pair: A pair of queues used for RDMA transactions (one send queue, one receive queue). Could be a software construct, but they usually require some hardware resources on the NIC and are therefore finite in number
- HCA: “Host Channel Adaptor” : A network card (NIC)
- Other high-speed interconnects sometimes call these “HFI”: Host Fabric Interfaces or HBA: Host Bus Adaptors

OmniPath

- Shiny, new competitor to InfiniBand from Intel
  - Result of Intel’s purchase of QLogic tech. and Cray IP
  - Hardware incompatible with IB
- Software compatible with IB
- PCIe cards don’t have as much hardware offloading as IB
- Intel chips with OPA on-board Real Soon Now™
- Potential to be less expensive
- In active development

IP – Quick Review

- IPv4 addresses are four bytes, typically dotted, for example: 192.168.1.1
- A subnet mask is a bitmask used to mark which machines can communicate directly
- For example, if my subnet mask is 255.255.255.0, then any machine with an address 192.168.1.* can talk with any other machine with a similar address
- 255.255.0.0 would mean 192.168.*.* can talk
- Subnets can also be described in CIDR notation, which is a IP address prefix and a number of bits
- 192.168.1.1/32 would be a single host
- 192.168.1/24 would be 255 hosts and have a netmask of 255.255.255.0
- If you need to talk to a host outside your subnet, you send to a “gateway” which is on both subnets and it will forward your packets

Proprietary Interconnects

- Can be good or bad – Single vendor is usually a good idea
- Consider whether your workload works on the vendor’s MPI
- Does your PFS support the vendor interconnect?
- Expandability?
- Can you interoperate with other technologies? Do you want to?
- Can the vendor support the product for your desired system life?
- Smaller support options
  - Lots of InfiniBand installs – you can Google
  - …but some vendors have very good user groups.

Proprietary Interconnects – NUMALink/UV

- Maybe a cluster/traditional network isn’t right for you
- Multi-socket machines have a “network” on the motherboard interconnecting the CPUs
- NUMALink lets you extend this to a few racks of machines while maintaining a single system image
- Run threaded applications (or even single processes, really slowly) with (up to) the resources of several racks of hardware.
- Expensive, but really convenient for non-distributed memory workloads
- NUMALink is made by SGI/HPE, but others have tried similar things (Oracle still has some 16 socket SPARC systems for example).

Management Networks

- Two types:
  - ssh/software updates/config management/etc. (user accessible)
  - Admin only
- Uses:
  - Monitoring/sensors
  - Depending on compute node type (stateless) maybe be used during boot
  - Typically used for Lights-Out Hardware interface
  - IPMI
  - Might want to put IPMI/hardware on separate VLAN(s) for security
  - Needs to be cheap and reliable but not necessarily fast
  - Sometimes used for compute node internet access, especially if your main interconnect doesn't support IP well (more on that later)
  - Usually gigabit Ethernet

Stateless Nodes

- One way to save cost on compute nodes is to not include disks and instead boot over the network
- Requires RAM disk
- PXE Protocol typically used
- Pass a file to download from a tftp server in a DHCP response
- Firmware downloads that file (a Linux kernel, for example) and boots it
- Semi-supported over InfiniBand
- Typically done over the management network
- Might be a reason to make your management network reasonably fast
  - xCAT
  - Cobbler
- Multicast is sometimes used, which can require switch support
  - SystemImager/SGI SMC

SDN (Software Defined Networking)

- Will make your network fully buzzword-compliant
- Typically Ethernet only (currently)
- Switches are “slaves” to a software “master”
- Can give very complicated rules
- Mostly experimental, but showing great promise
- Examples:
  - Bypassing firewalls for certain traffic
  - Configuring all switches so they know computers by MAC

Internet / WAN

- Usually Ethernet
- Usually have a security boundary between the WAN and the rest of the cluster
- Dedicated login/head nodes
- External interface is typically ssh
- May have other site specific firewall considerations
- Compute nodes are typically behind some kind of NAT, but could be publicly addressable (if you have address space to burn).
- Consider redundancy and whether your applications can handle multipath routing
- Lots of vendors/site specific – hard to give general advice

Diagnostics - Hardware

- Optics seated?
  - Most lock in some way
- Fibers crossed?
  - Try flipping one side. LC connectors may be keyed incorrectly
- Cables pinched/bent?
  - Check the bend radius specification
- Slow performance? Try reseating your NIC and/or SFP
- Ensure ports are enabled on both sides
- ifconfig [interface] up
- InfiniBand: ensure the SM is running, otherwise you may not even get a link light

Diagnostics

- ethtool [interface]
- ibstat
- ibv_devinfo

```
# ibstat
CA 'mlx5_0'
                CA type: MT4115
                Number of ports: 1
                Firmware version: 12.14.2036
                 Hardware version: 0
                Node GUID: 0x7cfe9003008ddacc
                System image GUID: 0x7cfe9003008ddacc
                Port 1:
                                 State: Active
                                 Physical state: LinkUp
                                Rate: 100
                                Base lid: 5
                                LMC: 0
                                SM lid: 29
                                Capability mask: 0x2651e848
                                Port GUID: 0x7cfe9003008ddacc
                                Link layer: InfiniBand
```

Diagnostics - InfiniBand

- Ibdiagnet
  - General diagnostics
- Ibnetdiscover
  - What can we talk to?
- Ibtopodiff
  - Are the cables in the right places?
- Ibportstate
  - Is the port enabled?
- perfquery
  - Each port has counters for various events. The vendor can tell you which are problematic in what quantity

Diagnostics - RDMA

- ib_write_bw #server
- Ib_write_bw -b --run_infinitely ip_address_of_server #client

Diagnostics – Packet Capture

- Extremely helpful for figuring out what's happening to your packets
- Ethernet: tcpdump
- IB (Mellanox Only): ibdump
- Visualization: wireshark

Questions?