LCI workshop
on high performance clustered computing

# Linux Clusters Institute: HPC Networking

Ben Matthews, Software Engineer, NCAR

NCAR
NATIONAL CENTER FOR ATMOSPHERIC RESEARCH

# Types of Networks

- Highspeed Interconnect
- External (Internet access)
- Management/Boot

# 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
- 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

- 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

# InfiniBand

- Buy from one vendor and run consistent firmware versions
  - Hardware generations can mix in theory, but here be dragons!
- Low latency and good performance
- Good software support
- Simple – tends to just work
- Some scaling limitations
  - 16bit addresses
  - Typically 36port switching unit (more on this later)

# OmniPath

- Shiny new thing from Intel, based on QLogic's InfiniBand and Cray's interconnect IP
- Sort of like InfiniBand, but not compliant with the standard
- Lots of new management software
- 48 port switching unit
- Low latency, but not as much hardware offloading as Mellanox (yet?)
- Cheaper?
- Proprietary software interface, but can do IP or VERBS with significant performance loss
- Currently first generation, expect changes to come soon.

# Ethernet

- Well understood, comes in lots of different flavors from many vendors
- 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
  - Buy from a single vendor and/or test at small scale
- 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
- Lossy (unless you have fancy flow-control supporting NICs and switches) - ECN

# 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

# Ethernet – Quick Review

- In a **switched** Ethernet network, each switch port tracks the hardware addresses of devices connected to each port. When you send a packet, it is forwarded to the port with the destination device connected
- The address table for each switch port is finite, so at large scale more sophisticated routing is required
- At the Ethernet level, networks can be subdivided by using “VLAN”s, in which case each packet is tagged with a VLAN number
  - Most **managed** switches can enforce that ports only receive packets for their assigned VLAN (or they can be configured to pass all packets with the tags)

# Ethernet – Quick Review

- Ethernet doesn't handle redundant paths particularly well
  - Redundant links might be disabled by the spanning tree protocol (given smart switches)
  - Or packets might just get forwarded around in circles ;-(
- Various protocols exist for aggregating the performance of multiple ports (or using them in failover configurations). This is often called "bonding" or link aggregation and requires configuration on both sides of the link
  - LACP is a common protocol for managing bonds
  - Relies on hashing of the packet headers to distribute packets
    - 4x 10Gb links != 1x 40Gb link
- More recently, "Software defined networking" has started to become popular, which allows for fancier topologies.

# 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

- For system administration
  - pdsh/ssh/software updates/config management/etc
  - Monitoring/sensors
- Depending on the high-speed interconnect, may be needed for job launch and user access
- 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
- 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

# Internet/WAN

- Usually Ethernet, Obsidian and Mellanox (MetroX) have products for long haul IB for interconnecting sites
- Usually have a security boundary between the WAN and the rest of the cluster
  - Dedicated login/head nodes
  - External interface is typically ssh (one-time-password recommended!)
  - 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

# Topology

- I have more than ~48 nodes, but I can't buy a switch that big. What do I do?
- I bought this many hundred port switch, but the performance is bad, why?
- Wow networks are expensive!

- Do your applications really need every node to be able to talk to every other node simultaneously (one definition for a "non-blocking" network)?
- Can we oversubscribe uplinks a bit and save money?
  - How should we do so?

# 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

# Topology: Crossbar

- Everything can talk to everything else directly
- Basically a big matrix of wires
- Doesn't scale at all, but is often the basic element used to build up other types of networks
- 36 ports for most current InfiniBand, 48 port available for Ethernet and OmniPath
- Even single switches might have a "blocking factor" and are therefore not really crossbars
  - Read the specs when purchasing network gear ☺

# Topology: Types of Switches

- "Leaf" or "Top of Rack" ("TOR")
  - Switching unit that connects directly to compute nodes
  - Typically one or more in each rack to simply cabling
  - Typically 24-48 ports
    - Mellanox IB is 36 ports
    - OPA is 48 ports
- Director
  - Large switches which aggregate connections from the leaf switches
  - Typically not a flat crossbar
  - Hundreds to a thousand ports per chassis is typical
  - **Expensive**

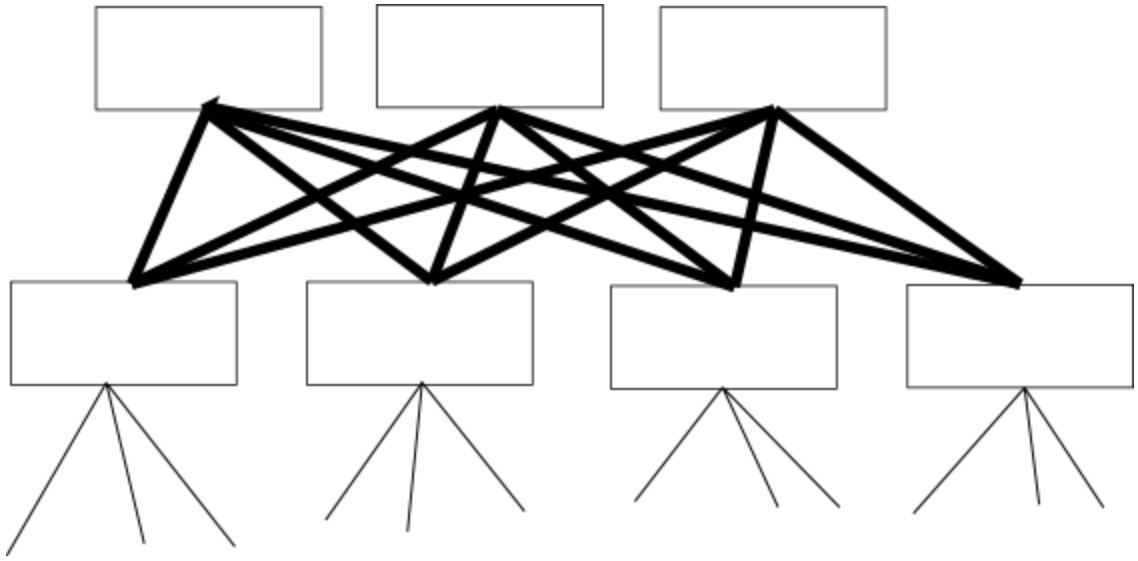
# Topology: Island

- Simple way to scale up a network
- Buy a bunch of crossbar switches and connect them
- Cheap and simple. Good performance within each “island”
- Not so good performance system wide

# Topology: Fat Tree

- Build a tree of switches
- Half the ports on each switch go to other switches higher in the tree
  - To save cost, you could use less than half the ports as uplinks and end up with an oversubscribed or "trimmed" tree at the cost of performance
- Half the ports go to other switches lower in the tree or to compute nodes
- Still non-blocking, and scalable
- Cost grows rapidly with scale

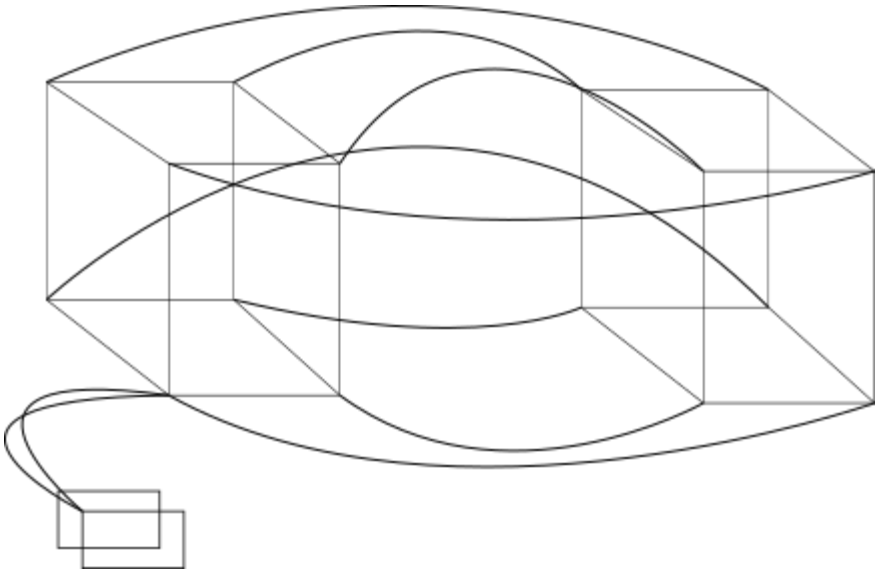
# Topology: Fat Tree



# Topology: Hypercube and Torus

- N-dimensional mesh of squares or rings (respectively)
- A bit less costly to scale, but typically more hops between the farthest compute nodes
- Performance gets worse as the paths get longer
  - If your program mostly needs near neighbor communication these are good options
  - … if your scheduler is reasonably smart
  - Not so good for more random communication patterns
  - Need smart (topology aware) collectives in your middleware

# Topology: Hypercube



# Software Considerations: Thundering Herd

- Static addressing
- Prefer static host files over DNS
  - Multiple caching DNS close to the compute nodes
- Prefer static passwd/group files over LDAP/AD or scale up your LDAP and have a good path to it
- Consider hardware redundancy for any subnet managers/routers/DHCP/etc
- Consider how to manage hardware discovery
  - How does a replacement system get an address assignment?
  - Most cluster management software has some solution for this
  - Two nodes with the same address = bad
- You can get away with a lot when you have only 10s of nodes, but these considerations get more important with scale
  - At massive scale, even things like ARP and multicast start to break down

# Hardware Considerations: Managed Switches

- InfiniBand is easy to manage in-band. Managed switches might not be worth it
  - Management interface mostly gives better hardware monitoring and a nice way to manage ports
- Much more important for Ethernet – VLAN support, diagnostics, etc  get progressively more essential with scale
- Unfortunately managed switches all use different obnoxious interfaces and are additional stateful things to configure
  - Think about how you're going to restore config when a switch dies
- Software defined (Ethernet) networking might solve this in the near future

# 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?