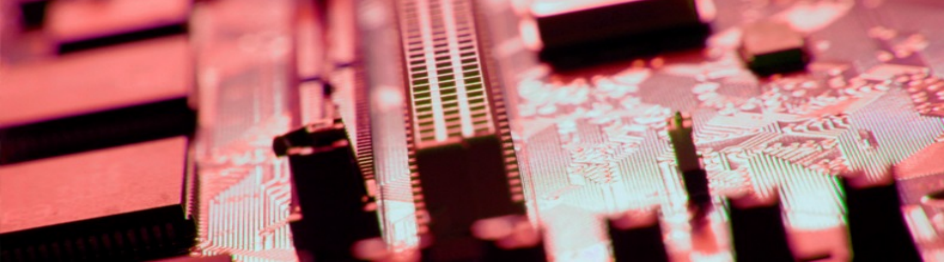


LCI workshop
on high performance clustered computing

Linux Clusters Institute: Intermediate Networking

Vernard Martin
HPC Systems Administrator
Oak Ridge National Laboratory

OAK RIDGE National Laboratory

About me

- Worked in HPC since 1992
- Started at GA Tech in 1990 as a Grad Student
  - Center for Experimental Research in Computer Science (CERCS)
  - BBN Butterfly, Thinking Machines
  - Linux clusters before clusters were clusters (i.e. pre-Beowulf)
- 1994 – IBM SP-2. Focused on debugging applications
- 1995 – Linux Beowulf (Quad Pentium Pro units)
- 2001 – Vertex Pharmaceuticals first cluster – Vampire
- 2005 – Biostatistics cluster for Emory School of Public Health
- 2010 – Bioinformatics cluster for Center for Comprehensive Informatics
- 2013 – Scientific cluster for CDC Infection Disease group
- 2015 – HPC operations for Titan at ORNL (#4 on Top500)

Some assumptions….

- Familiarity with:
  - TCP/IP, DMZs
  - Switches, Routers, Firewalls
  - Networking testing/diagnostic tools: ip/ipconfig, netstat, tcpdump,wireshark/ntop
  - Network attribute trade offs: Latency vs bandwidth, reliability, scalability, etc.
  - Public vs. Private (RFC 1918) Addressing
  - Subnetting/Gateways
  - DNS
  - IPv4 vs IPv6
  - Logical vs Physical topology

Agenda/Topics

- Networking at Scale
- InfiniBand
  - Basics
  - Hardware
  - Subnet Managers
  - Topologies
  - Omni-path
- InfiniBand Practicum

Scaling

Issues of Scale

- 32 nodes isn't really a problem (where node = mainboard)
- 2500 nodes: how do you interconnect? How do you debug that much I/O?

Scaling in clusters is **usually** about I/O limits

In the software, it's about balancing communication costs with computation costs.

Hardware and software both have bounds on their costs.

I/O Limitations

- Communication infrastructure has built in limitations
  - CPU-to-CPU communication aka Northbridge
    - Most common technique addressed.
    - Most benefit in multi-socket setups. Less for multi-core.
    - NUMA is most common. Software libraries can compensate some
  - CPU-to-GPU communication aka GPU interconnect (i.e. NVlink)
    - Only useful for GPU-heavy computations
    - Fairly new technology and still evolving
  - Node-to-Node communication
    - Ethernet (10Gb/40Gb/100Gb) – limited by Binary Exponential back-off
    - InfiniBand – Network Trees: Max switch size ==> Max cluster size

Networking considerations

- How does this affect storage?
  - Cost per port on switch
  - Cost per port on the node
  - Cabling!
  - Latency != Bandwidth
- Common host channel adapters (HCAs)?
  - HCA is an IB-Host Channel Adapter
    - Sometimes called an HBA interchangeably
  - InfiniBand
    - Mellanox
    - Intel
    - Oracle has proprietary stuff
  - Fibrechannel
  - Intel OmniPath

Vendor Management

- Clusters manage very differently than Enterprise servers
  - Enterprise is all about the "7 9s".
  - It’s all about availability of the overall service, not the size of the service
    - “Can I schedule jobs” vs “only jobs of size < 1024 nodes can run”?
  - Graceful degradation is often acceptable.
    - E.g.. Failures in RAID arrays. Things slow down but do not stop.
  - Clusters, simply by their scale, are always going to have losses.
    - In some cases, you can't have graceful degradation.
    - Alternatives: checkpointing, rescheduling and backfill etc.

Vendor Management (cont.)

- How to manage integrators that don't want to integrate?
  - Nature of the beast: Vendor want to sell you EVERYTHING.
    - Their stuff.
    - Their partners stuff.
  - Shunt issues to other areas
    - "It's the motherboard BIOS, not our HCAs"
    - "Only supported on this **exact** configuration"
  - Play Softball
    - Be very polite but also very insistent.
    - Be very detailed. Deluge them with all the info they need.
    - Show instances where it works in almost identical environment
  - Play Hardball
    - Call them at every issue (i.e. get your money's worth out of support)
    - Replace equipment under warranty for small errors (usually expect some)
    - Last resort: Legal team goes over the support contract and handles it.

Vendor Management (cont.)

Your personal relationship with your vendor sales rep is CRITICAL.

Sales reps move around as well. That relationship can migrate as well.

Understand your limits on what you can tolerate in production

Understand your options in both short and long term.

Ethernet and InfiniBand

- Tradeoffs between Ethernet and InfiniBand
  - Cost/Speed/Bandwidth
  - Speed/Bandwidth

    Ethernet is traditionally much cheaper (Both host port and switch port)
    - InfiniBand has much better latency
    - Ethernet usually doesn't require code modification (i.e. no special libs)
    - Ethernet handles "jumbo packets" better
  - Coding
    - Have to write code using API that understands IB protocol (i.e. rdma)
    - Writing code is still not easy. Rewriting in MPI is **hard**
    - Re-writing apps is not always viable or possible (i.e. commercial codes)

How will you access storage?

- Node to node is different than node to storage
- Filesystem for storage might dictate network as well
  - Lustre
    - The status quo. Open source lags very slightly behind commercial.
    - More hardware to run it.
    - Scales better
    - Fsck/recoveries more expensive
  - GPFS
    - Rock solid. IBM product is supported well. Open source, not so much.
    - More stable but *tends* to have less scalability with the same hardware
  - NFS
    - Tried and true. Fully known.
    - Scales abysmally.
    - pNFS is better but not in same class as Lustre and GPFS.

Ethernet

- Speeds
  - 40GbE, 25/40GbE, 100Gbe, 400Gbe?
  - Considerations for use in switches, wiring
    - Speed (Can the cable handle it?
      - Cat6e is RJ45 that does 10Gb up to 100meters
      - 40Gb: 4 pair twisted pair can do 30 to 150 **m** vs single-mode fiber can go 2 **km**
    - Density (Everything is networked = Need more Networking ports)
    - Power Over Ethernet (VoIP phones, WLAN Aps, IoT devices)
    - Software Defined Networking (VXLNA and NVGRE support)
  - MTU 9000 (i.e. Jumbo Frames)
    - Have to have same MTU in full path
    - Behaves horrible when you don't.
    - Works best with using VLANS to guarantee non-mixed traffic
  - Dealing with system configurations (i.e. TCP tuning)
    - Host based tuning (tcp buffer sizes, fair queueing, cpu governor settings, etc.
  - Topology considerations
    - Your switch choice can cripple a network design
    - Top of Rack switches are very different than other cores switches

InfiniBand

- QDR, FDR, EDR
  - Bandwidth
    - SDR 10Gb, DDR 20GB, QDR 40Gb, FDR 56Gb, EDR 100GB, HDR 200Gb
  - Discuss the rate at which IB hardware is released.
  - Discuss the *cost* of new IB hardware when it's released.
    - http://clusterdesign.org/cgi-bin/network/network
  - Discuss the cost of 4x links vs. 8x links vs 12x links
  - Discuss the theoretical vs. Observed throughput for the aggregations.
- Topology considerations
- Dynamic routing with predictable patterns

Communication Service Types

- IBA provides several different types of communication services between endnodes:
  - Reliable Connection (RC): a connection is established between endnodes, and messages are reliably sent between them.
    [Acknowledged, connection oriented]
  - Reliable Datagram (RD): [Acknowledged, Multiplexed]
  - (Unreliable) Datagram (UD): a single packet message can be sent to an endnode without first establishing a connection; transmission is not guaranteed.
    [Unacknowledged, Connectionless]

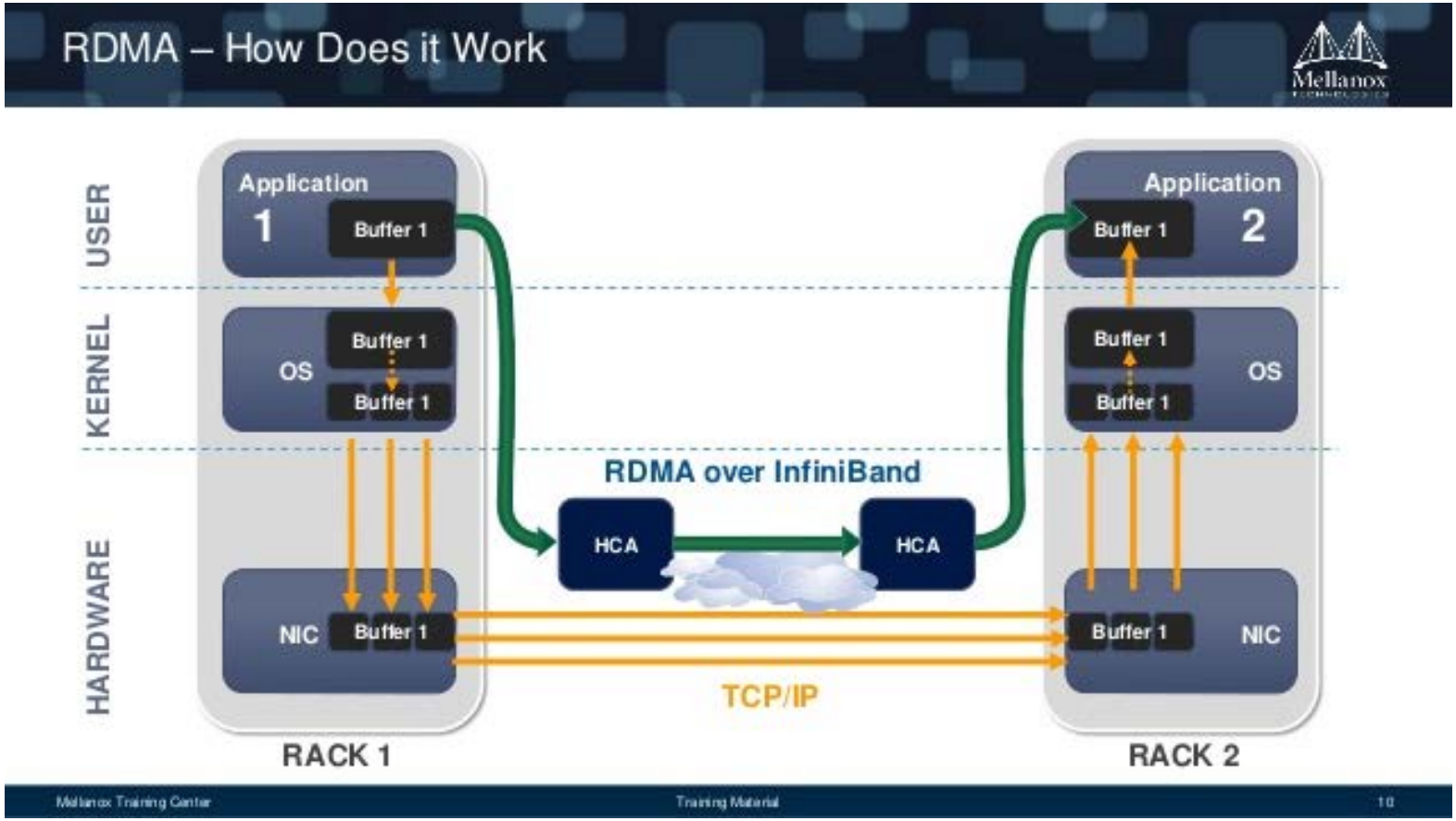
Communication Service Types (cont.)

- Unreliable Connection (UC): a connection is established between endnodes, and messages are sent, but transmission is not guaranteed. This is optional. • Reliable Datagram (RD): a single packet message can be reliably sent to any endnode without a one-to-one connection. This is optional. [Unacknowledged, Connection oriented]
- Raw Datagram (optional) (Raw): single-packet unreliable datagram service with all but local transport header information stripped off; this allows packets using non-IBA transport layers to traverse an IBA network, e.g., for use by routers and network interfaces to transfer packets to other media with minimal modification. [Unacknowledged, Connectionless]

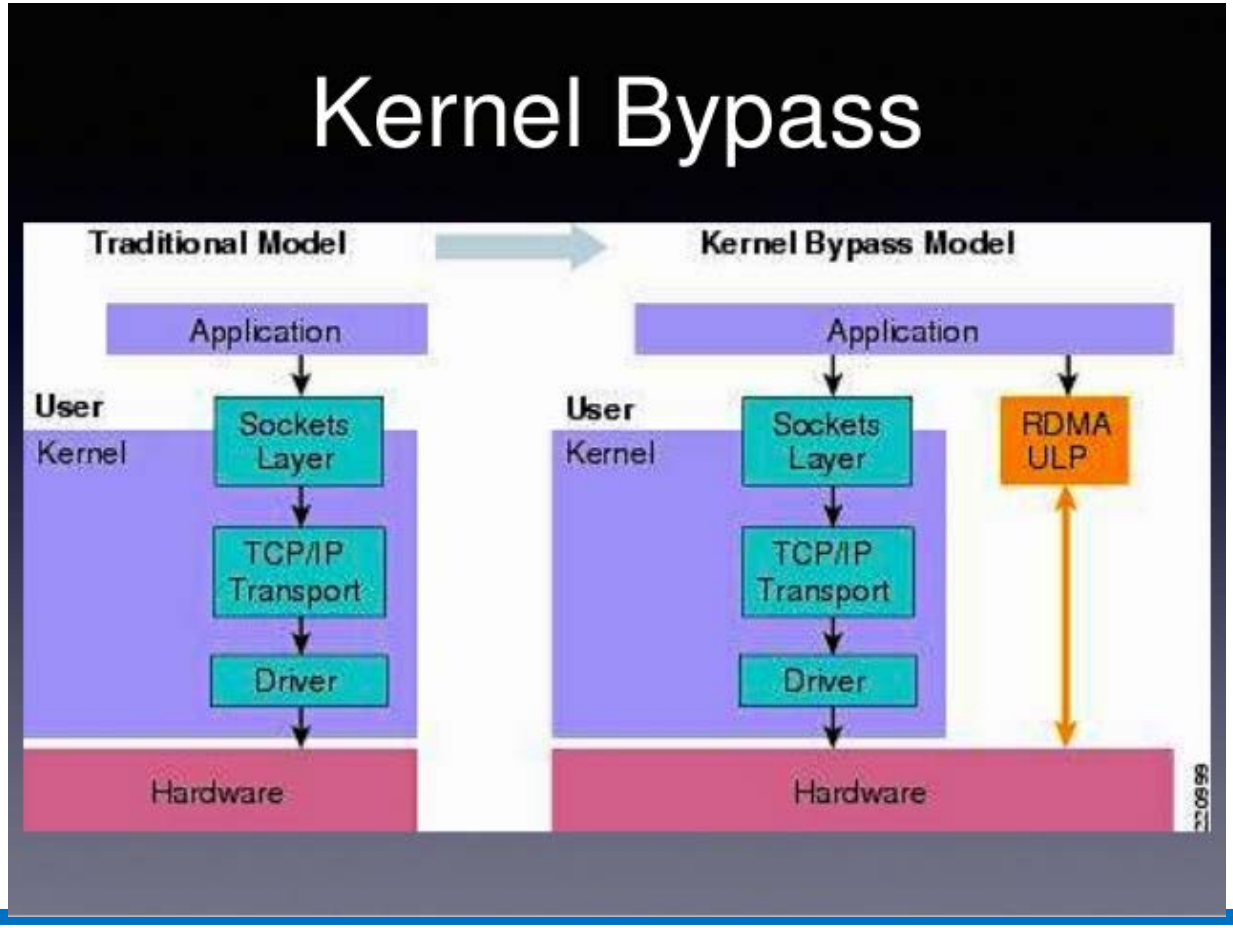
What is InfiniBand?

- Industry-standard defined by the InfiniBand Trade Association
- Interconnect technology connecting CPUS and I/O
- Switch Fabric Architecture
- **Lossless** fabric: Maximum Bit Error Rate specified in IB spec
- Super high performance
  - High bandwidth (starting at 10Gb/s and commonly 100Gb/s)
  - Low latency – fast application response across nodes
    - **< 1 microsecond end to end**
- Low CPU utilization with Remote Direct Memory Access (RDMA)
- **Unlike Ethernet, traffic communication bypasses the OS and the CPU**

RDMA Pathway



RDMA Pathway (cont.)



Host Channel Adapters/Switches/Routers

- HCA
  - Device that terminates an IB link and executes transport-level functions and supports the verbs interface
- Switch
  - A device that moves packets from one link to another of the same IB subnet
- Router
  - A device that transports packets between different IBA subnets
- Bridge/Gateway
  - InfiniBand to Ethernet

HCA

- Equivalent to a NIC (Ethernet)
  - GUID – Global Unique ID
- Converts PCI to InfiniBand
- CPU offload of transport operations
- End-to-end QoS and congestion control
- HCA bandwidth option:
  - Single Data Rate 2.5GB/S * 4 = 10
  - Double Data Rate 5 GB/S * 4 = 20
  - Quadruple Data Rate 10GB/S * 4 = 40
  - Fourteen Data Rate 14 Gb/s * 4 = 56
  - Enhanced Data Rate 25 Gb/s * 4 = 100

Global Unique Identifier (GUID) – Physical Address

- Any InfiniBand node requires GUID&LID addresses
- GUID (Global Unique Identifier)- 64 bits address,
  - “Like a Ethernet MAC address”
  - Assigned by IB vendor
  - Persistent through reboots
  - All ports belong to the same “basic “ switch will share the switch GUID
- LID (Local Identifier) – 16 bit L2 address
  - Assigned by the subnet manager when port is active
  - Not persistent through reboots (algorithm assigned)
  - used for packets switching within an IB fabric
- IB Switch “Multiple” Address GUIDS
  - Node = Is meant to identify the HCA as a entity
  - Port = Identifies the port as a port
  - System = Allows to combine multiple GUIDS creating one entity

IB Fabric Basic Building Block

- A single IB switch is the basic building block of the network
- Create larger networks via combinations (i.e. topologies)

Can segment the network – Partitions

- Define different partitions for different:
  - Customers
  - Applications
  - Security Domains
  - QoS guarantees
- Each partition has an Identifier named PKEY

IB Basic Management Concepts

- Node: Any managed entity – End Node, Switch, Router
- Manager: Active entity; sources commands and queries
  - The Subnet Manager (SM)
- Agent: passive (mostly) entity that will reside on every node, responds to Subnet Manager queries
- Management Datagram (MAD):
  - Standard message format for manager-agent communication
  - Carried in an unreliable datagram (UD)

Objectives of Subnet Management

- Initialization and configuration of the subnet elements
- Establishing best traffic paths between source to destination through the subnet
- Fault isolation
- Continue these activities during topology changes
- Prevent unauthorized Subnet Managers

Subnet Manager Rules & Roles

- Every subnet must have at least one
  - Manages all elements in the IB fabric
  - Discover subnet topology
  - Assign LIDs to devices
  - Calculate and program switch chip forwarding tables (LFT pathing)
  - Monitor changes in subnet
- Implemented anywhere in the fabric
  - Node, Switch, Specialized device
- No more than one **active** SM allowed
  - 1 Active (Master) and remaining are Standby (HA)

Fabric Discovery (A)

1. The SM wakes up and starts the Fabric Discovery process
2. The SM starts " **conversation** " with every node , over the InfiniBand link it is connected to. In this stage the **discovery stage**, the SM collects:
    1. Switch Information followed by port information
    2. Host information
3. Any switch which is already discovered will be used as a gate for the SM , for further discovery of switch links and their neighbors.
4. The SM gathers info by sending and receiving Subnet Management Packets
    1. Two types of routing info
        1. Directed routing table based on Nodes GUIDS and port number
        2. LID routing: based on LiDS which have been assigned to each node by the SM

LID assignment

- After the SM finished gathering any needed subnet information, it assigns a base LID and LMC to each one of the attached end ports
    - The LID is assigned to at the port rather than device level
    - Switch external ports do not get/need LIDs
- The DLID is used as the main address for InfiniBand packet switching
- Each **Switch port** can be identified by the **combination** of LID & port number
- After the SM finished gathering all Fabric information , including direct route tables, it assigns a LID to each one of the NODES
- Populates the tables with relevant route options to each one of the nodes

Tracking Fabric Status – SM Sweeps

- Light sweep :
  - Routine sweep of the Subnet Manager
  - By default runs every 30 second
  - Requires all switches to switch and port info
- Light Sweep traces :
  - Ports status change
  - New SM speaks on the subnet
  - Subnet Manager changes priority
- Any change traced by the light sweep will cause a heavy sweep
  - i.e. changes of status of a switch will cause an on line IB TRAP that will be sent to the Subnet Manager and cause heavy Sweep
  - Heavy sweep : causes all SM fabric discover to be performed from scratch

Fat Trees

In a switch fabric – a network topology that uses switches – the main goal is to connect a large number of endpoints (usually servers) by using switches that only have a **limited** number of ports.

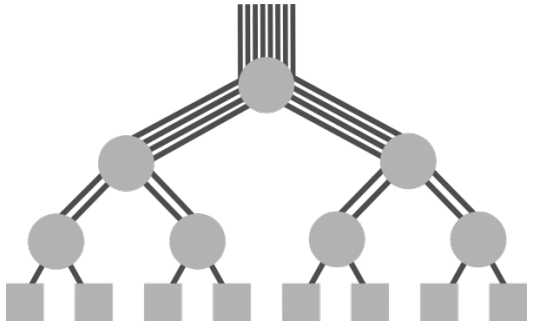
By cleverly connecting switching elements and forming a topology, a network can interconnect an impressive amount of endpoints.

Fat-Trees are a tree topology with the distinctive feature that for any switch, the number of links going *down* to its children is equal to the number of links going up to its parent in the upper level.

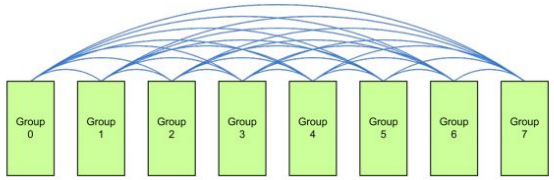
Therefore, the links get “fatter” towards the top of the tree and the switch at the root of the tree has the most links compared to any other switch below it.

Fat-Tree

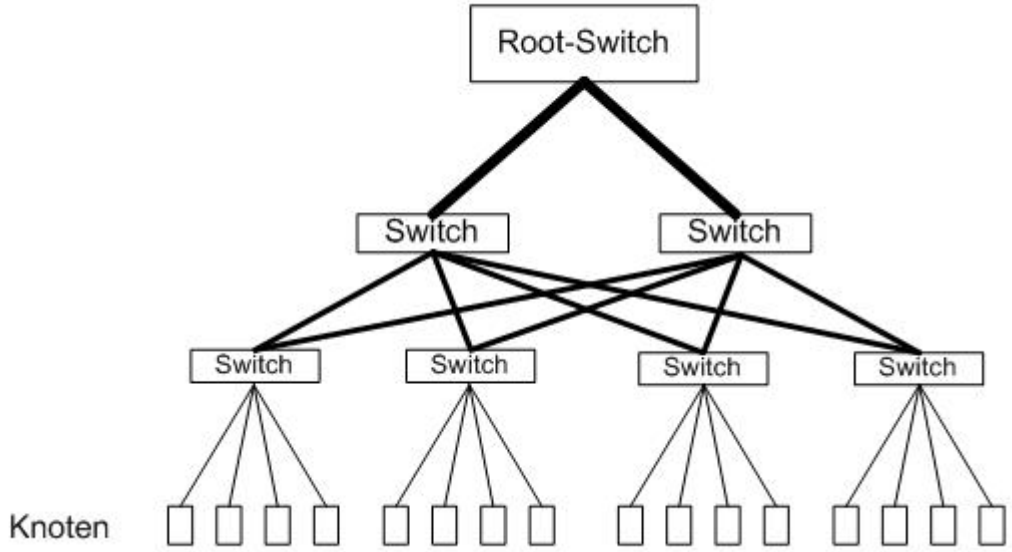
- Most common network configuration is Fat-Tree
  - Nodes are at the edge
  - Switches are hierarchically
  - Uplink bandwidth between switches is higher
  - Bisection bandwidth could be less than within switch due to switch limitation or cost
  - Placing workload within a switch provides full bandwidth for workload
  - Placing workload as close as possible will reduce latency decreasing the amount of hops required between nodes



- Dragonfly networks are similar to fat-tree in grouping of nodes on edge
  - Groups of switches interconnected connected
  - Generally uses the same interface as Fat-Tree with a slightly different algorithm



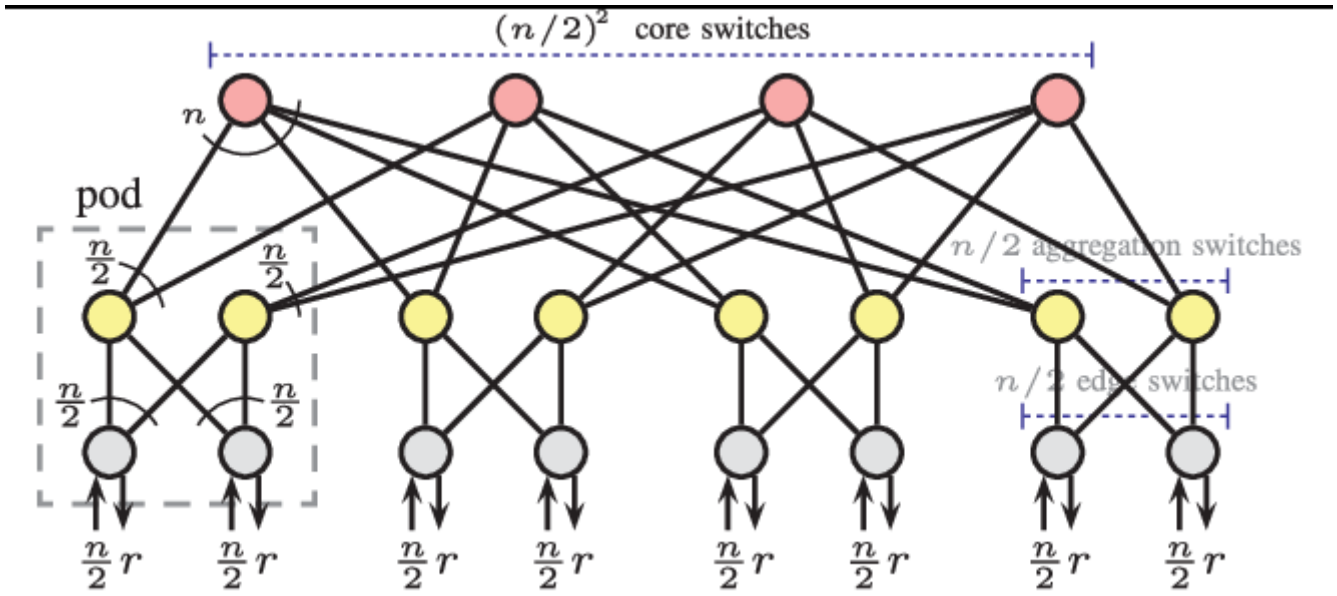
Two level fat tree



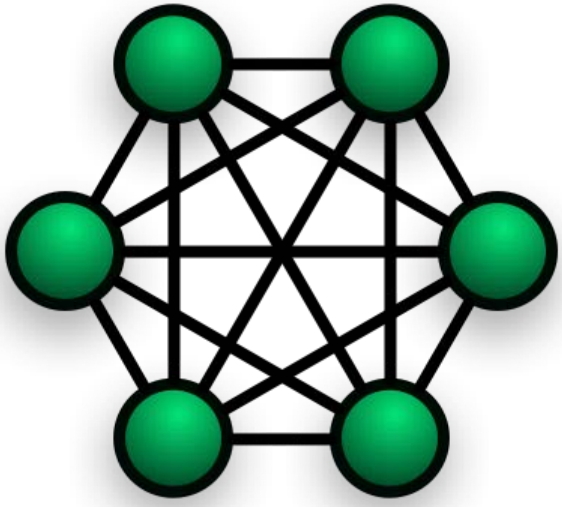
Communication

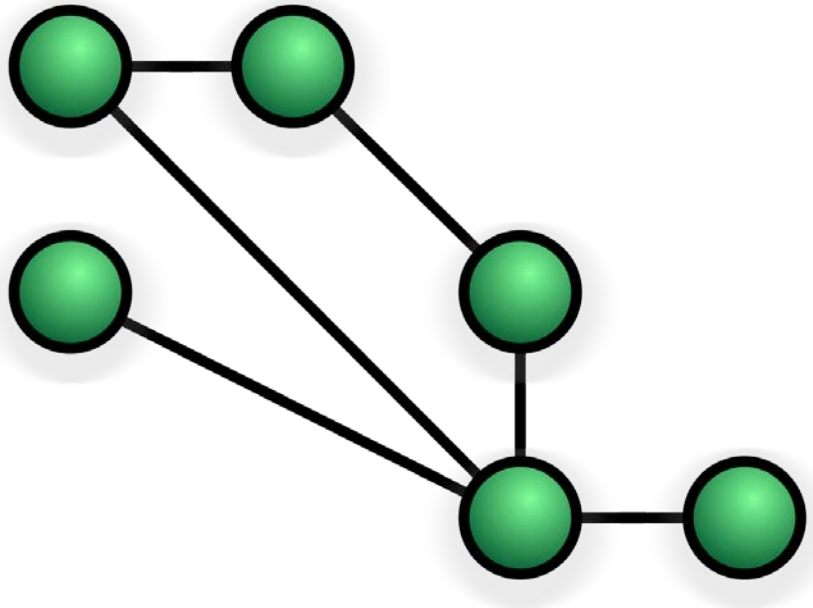
- If two servers connected to the *same* edge switch wish to communicate, they can do so via that particular edge switch, without referring to the core level. If, on the contrary, the servers are connected to *different* edge switches, the packets will travel up to any of the core switches and then down to the target edge switch.
- The subnet manager of an InfiniBand network calculates the best routes between each pair of nodes on the network and provides these routes to the nodes when they want to communicate.

Maximum size of a Fat-Tree

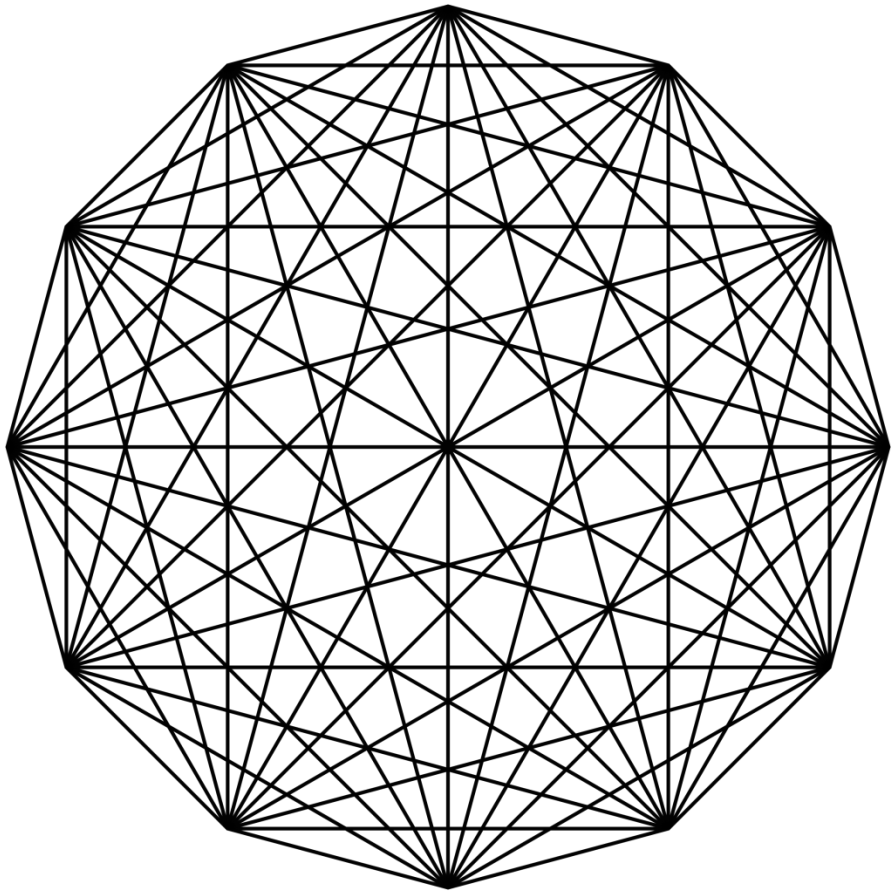


Mesh (full and partial)

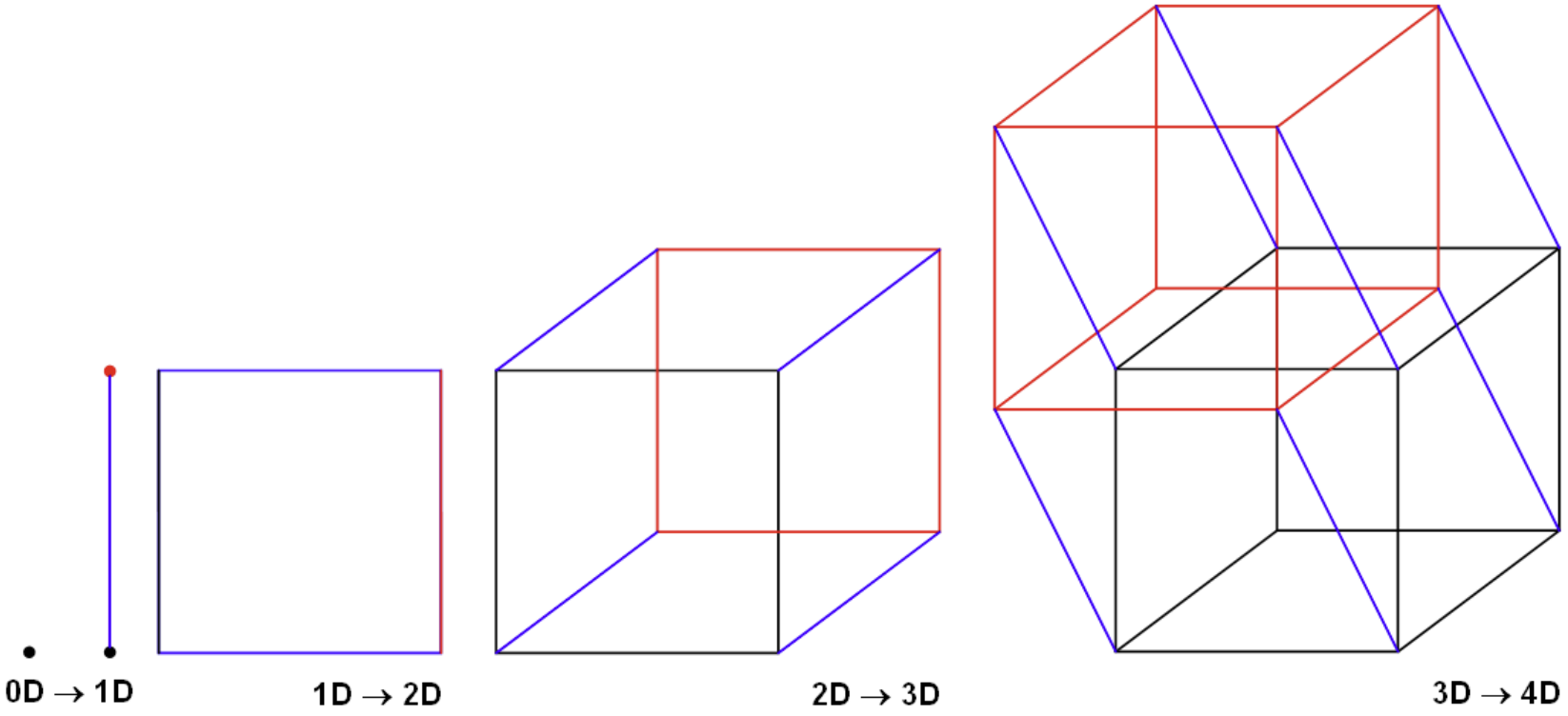




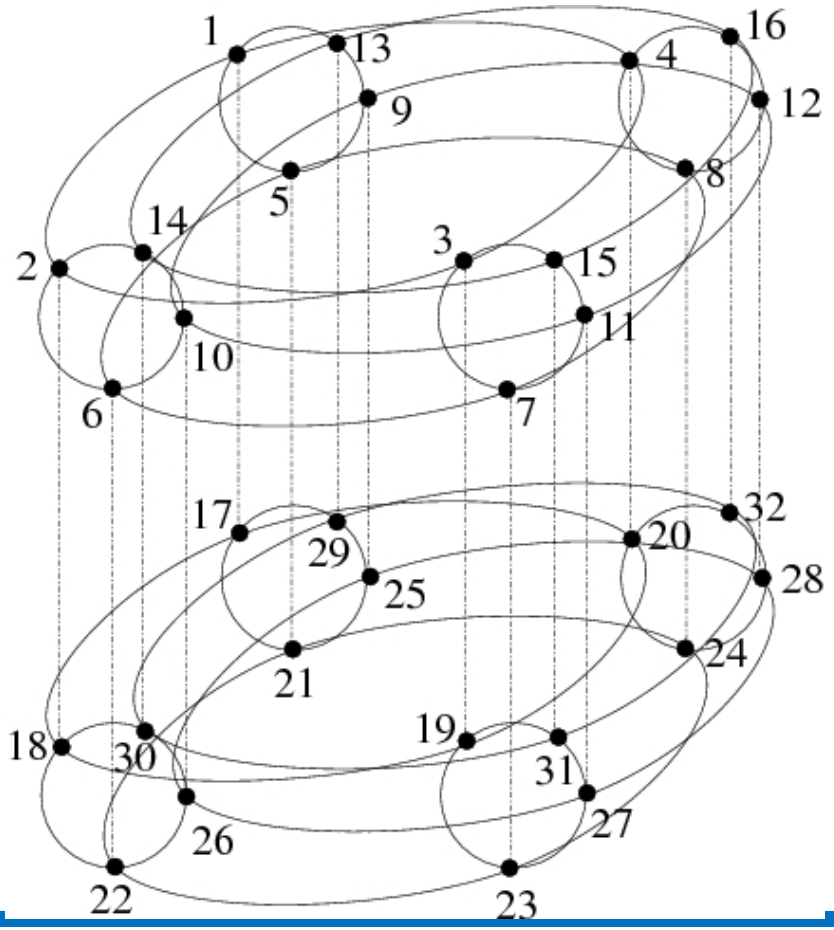
All-to-All



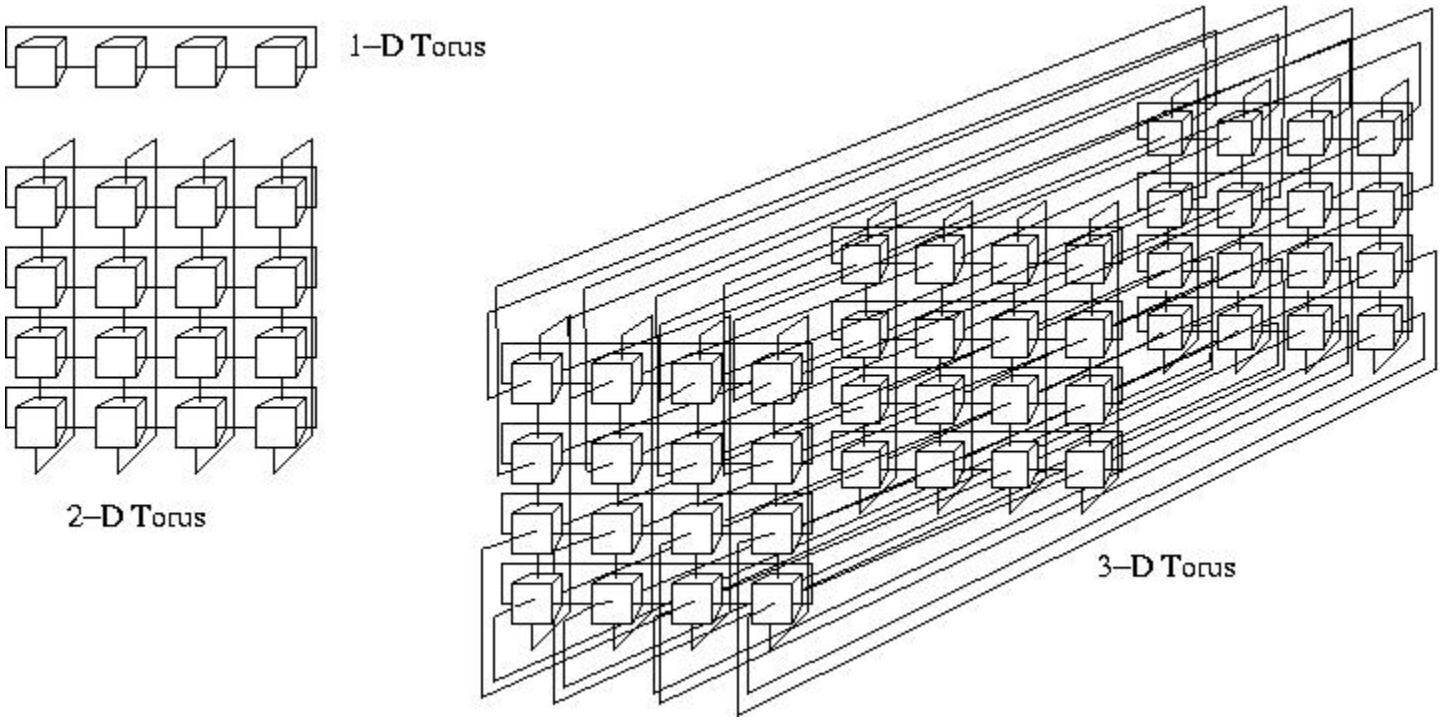
HyperCube (n=1,2,3,4)



Hypercube N=5 (dots are nodes)

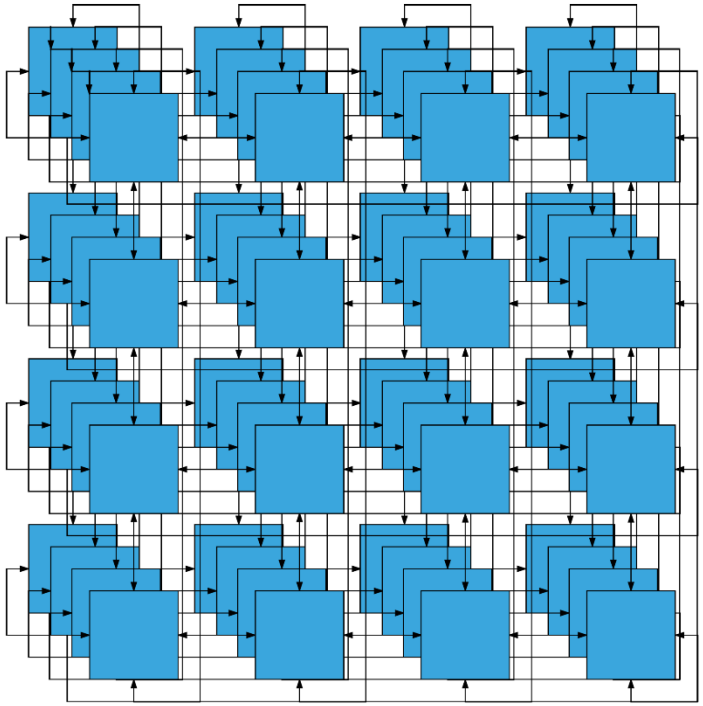


Torus

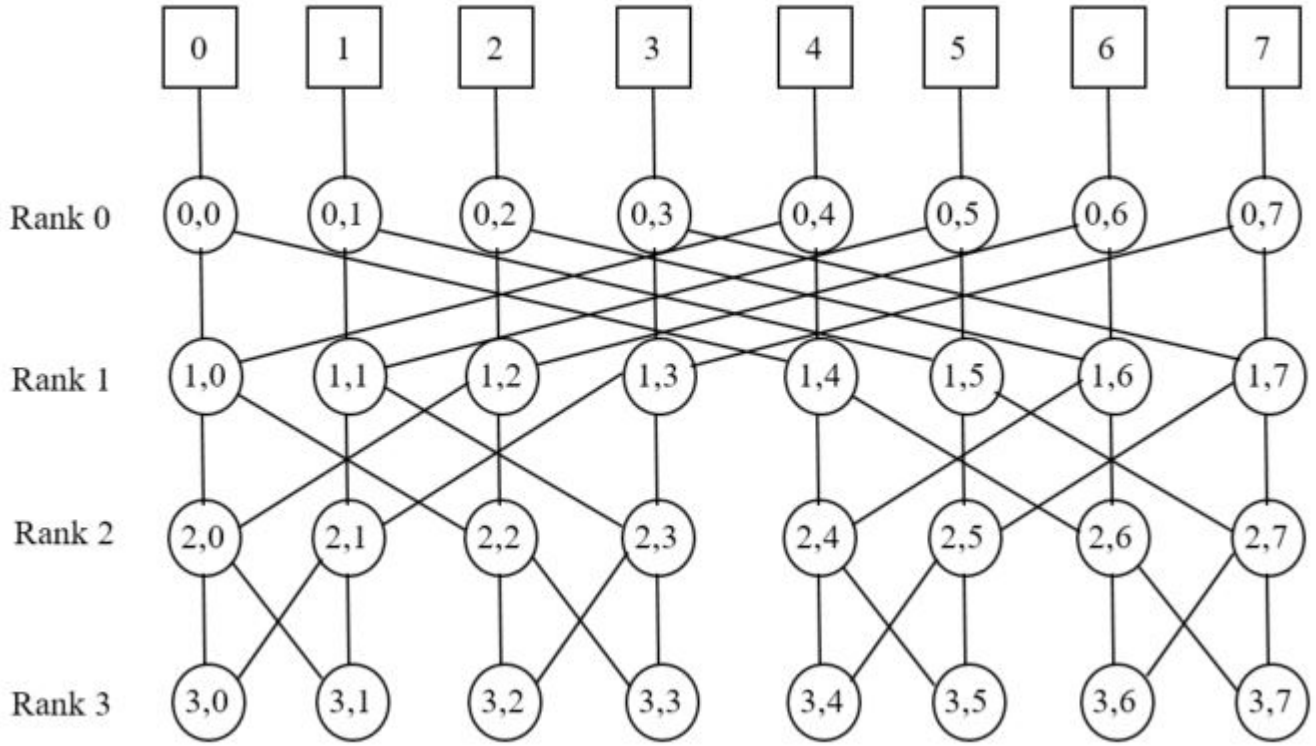


3-D Torus Interconnect Topology Aware

- A 3-D Torus is a mesh network with no switches
- All sides are connected to each other
- Network Congestion can be a problem if jobs are placed where communication is flowing through each other
- Locality is important for performance
- Not all links have to be the same speed
- Blue Waters at NCSA uses this topology



Butterfly


0
1
2
3
4
5
6
7
Rank 0
0,0
0,1
0,2
0,3
0,4
0,5
0,6
0,7
Rank 1
1,0
1,1
1,2
1,3
1,4
1,5
1,6
1,7
Rank 2
2,0
2,1
2,2
2,3
2,4
2,5
2,6
2,7
Rank 3
3,0
3,1
3,2
3,3
3,4
3,5
3,6
3,7

Physical Layer

- Industry standard Media types
  - Copper: 7 Meter QDR , 3 METER FDR
  - Fiber: 100/300m QDR & FDR
- 64/66 encoding on FDR links
  - Encoding makes it possible to send digital high speed signals to a longer distance enhances performance & bandwidth effectiveness
  - X actual data bits are sent on the line by Y signal bits
  - 64/66 * 56 = 54.6Gbps
- 8/10 bit encoding (DDR and QDR)
  - X/Y line efficiency (example 80% * 40 = 32Gbps)

Passive Copper cables

- FDR 3 meter
- FDR10 5 meter
- QDR 7 meter
- 40GbE 7 meter
- 10Gbe 7 meter

Passive Fiber cables

- FDR 300 meter
- FDR10 100 meter
- QDR 300 meter
- 40GbE 100 meter

How to handle IB Switching Hardware?

- Hardware
  - Host Channel Adapter (HCA) i.e. card
  - IB Switch
- Software (drivers and subnet manager)
  - Very Kernel dependent
  - Have to recompile for new versions
  - Open Fabrics Enterprise Distribution (OFED) vs Mellanox (MLNX)
  - Subnet Manager
    - Software
    - Embedded into the switch
  - Drivers
    - Loadable kernel mods based on HCA (i.e. mlx4_ib for newer mellanox)
    - Monolithic kernels (uncommon)

Resources

- ES Net / FasterNet
  - Tuning guides are priceless
- InfiniBand resources
  - Need dedicated people that:
    - Like kernel drivers
    - Like debugging
    - Attention to detail is **not** optional
  - Vendors usually don't/can't help except for the base level
  - Make friends (i.e. other sites, conferences, colleagues)
- InfiniBand Trade Association https://www.infinibandta.org
- OpenIB Alliance https://www.openib.org

Intel Omni-Path

- Intel's attempt at making end-to-end PCIe network solution
  - Developed to solve Xeon Phi networking constraints
  - NOT InfiniBand
    - 100Gbps per port, 25GB/s bandwidth (bi-directional)
    - Switch port-to-port latency: 100-110ns
  - Less proven. only been around since 2015. First delivery 2016.
    - Less != Non. VERY large installations exist that work quite well.
  - Ground up redesign. Targeting the Exascale market.
    - Designed to withstand higher bit error rate
    - Tradeoff: more expensive fabric both in HCA and Cabling
  - Verbs interface proposed so that IB apps can still run
    - 2016 it was published with software libs available.
  - Open specification.

Omni-Path Fabric Architecture

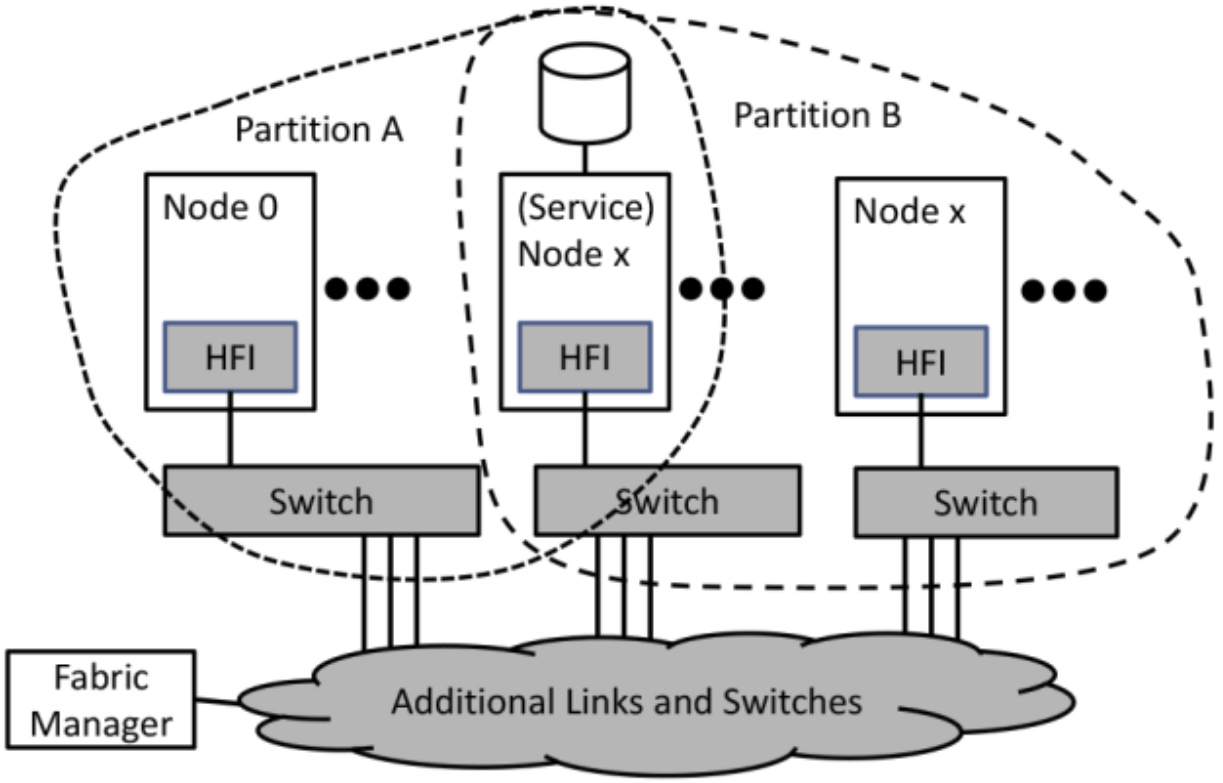


Figure 1. Intel Omni-Path Architecture Elements

Intel Omni-Path Show & Tell

- 6x OPA Host Fabric Adapter 100
  - PCIe Low Profile
  - Multi-core scaling
  - 12 Send DMA engines
  - Large MTU support (10K)
  - 8 virtual lanes for QoS
  - ASIC designed to scale up 160M messages/second and 300M bidirectional messages/second

Intel Omni-Path Show & Tell

- 6x OPA Passive Copper cable - 3 meter

Intel Omni-Path Show & Tell

- 1x OPA Edge Switch - 2 PSU, 1U 24 port
- 100Gbps Switch

InfiniBand and MPI: The Dance of the Fairies

- MVAPICH2 – Network Based Computing Laboratory @ Ohio state
  - IB, Omni-Path, Ethernet/iWarp, RoCE
  - Variants for Nvidia GPUS, hypervisors, containers, Intel KNC
- OpenMPI
  - OpenMPI Team: 30+ contributes including 15+ major players
  - IB, Ethernet
- Intel MPI
  - Supports IB via the abstraction layer DAPL
  - Ethernet
  - OmniPath

Benchmarks

- Top500 list
  - #6 Oakforest-PACS
    - PRIMERGY CX1640 M1, Intel Xeon Phi 7250 68C 1.4GHz, 556,104 cores
  - #12 Stampede 2- Texas Advanced Computing Center/ U of Texas
    - PowerEdge C6320P, Intel Xeon Phi 7250 68C 1.4Ghz, 285,600 cores
  - #13 MareNostrum – Barcelona Supercomputing Center/Spain
    - Lenovo SD530, Xeon Platinum 8160 24C 2.1Ghz, 148,176 cores
  - #14 MarconiIntelXeonPhi – CINECA, Italy
    - Intel Xeon Phi 7250 68C 1.4Ghz
  - #45 LLNL CTS-1 Quartz – LLNL, US
    - Tundra Extreme Scale, Xeon E5-2695v4 18C, 2.1Ghz
  - #46, #61, #65,#66, #77, #86, #97, #98

Infiniband and MPI

- Threads and Mutexs
  - MPI 3.1 standard allows for thread-safe calls
  - However, only latest versions of all packages support his.
  - Bad coding can cause havoc
  - Workflows created from diverse components don't play well together
  - e.g. An inherently serialized component may force the IB to use less optimized methods (i.e. screw up the caching or shared-mem)
- Incompatible MPIs
  - Not all apps work with all versions of MPI
  - Even though they support the same standard!
  - "module load" will not save your workflow