High-Performance Computing Networks at BYU

Lloyd Brown

October 11, 2011

What makes a supercomputer, super?

HPC or High-Performance Computing, is characterized by workloads and hardware requirements

- Significantly larger compute capability than an average system
- Used to solve problems that are too large to easily be solved on a single, traditional system
- May utilize specialty hardware and software
- No specific threshold for capacity

Nature of HPC Computing

In HPC, speedup comes from one of two sources:

- Using faster resources (eg. faster clock speeds)
- Using more resources (eg. using more processors) or *Parallelism*

Physics generally limits us on the faster resources, so we spend more time on parallelism.

Parallelism and Communication Needs

- When utilizing multiple resources (eg. multiple processors), the program must:
  - Split up the workload
  - Provide necessary coordination among resources
- The algorithm and data determine the nature of communication needs
- In general, for HPC problems, communication is key.

What kinds of HPC systems are out there?

There are two major categories of HPC systems:

- Systems which utilize specialty hardware, including:
  - Processors
    - Vector Processors (eg. Cray)
    - Specialty Serial Processors (eg. Itanium, Power5, etc.)
  - Accelerators
    - GPU
    - Intel MIC
    - FPGA
    - Cell
  - Specialty/Proprietary Interconnects
    - Infiniband
    - NUMALink
- Commodity Hardware:
  - Stock processors (eg. x86, x86_64)
  - Stock interconnects (Ethernet)

What is Infiniband? And why do I care?

Infiniband is the most common high-performance interconnect used in HPC. It:

- is switched-fabric architecture (more like Fibre Channel than like Ethernet)
- utilizes multiple speeds, lanes, and links
- provides:
  - extremely high bandwidth
  - extremely low latency (one-way $< 10\ \mu$s, compared to approx. 32 μs for 1GbE)
- Speedup comes mostly from:
  - Short protocol stack (very little above layer 2)
  - Low-latency switching (very little decision making in the switch)
  - Remote Direct Memory Access (RDMA)

Terms

HCA Host Channel Adapter - The interface device that connects a host to the network

GUID Globally-unique Identifier; hardware address on each HCA or switch; like a MAC address

LID Logical Identifier (address) assigned by the subnet manager to the HCA; kinda like an IP, but resides in the upper part of layer 2

SM Subnet Manager, a hardware or software device that assigns LIDs to GUIDs, and pre-loads the switch forwarding tables

Lanes/Links/Speeds

Infiniband utilizes multiple lanes per physical link. Each link has a certain speed based on the standard:

	SDR	*DDR*	*QDR*	*FDR*
1x	2.5 Gb/s	5 Gb/s	10 Gb/s	14 Gb/s
4x	10 Gb/s	20 Gb/s	40 Gb/s	56 Gb/s
12x	30 Gb/s	60 Gb/s	120 Gb/s	168 Gb/s

Encoding Overhead

Infiniband uses bit-line encodings to guarantee bit transitions for clock synchronization:

- SDR, DDR, QDR - 8b/10b encoding (8 data bytes encoded in 10 bytes total; 20% overhead)
- FDR and beyond - 64b/66b encoding (64 data bytes encoded in 66 bytes total; 3% overhead)

	SDR	*DDR*	*QDR*	*FDR*
1x	2.5 Gb/s raw 2 Gb/s net	5 Gb/s raw 4 Gb/s net	10 Gb/s raw 8 Gb/s net	14 Gb/s raw 13.6 Gb/s net
4x	10 Gb/s raw 8 Gb/s net	20 Gb/s raw 16 Gb/s net	40 Gb/s raw 32 Gb/s net	56 Gb/s raw 54.3 Gb/s net
12x	30 Gb/s raw 24 Gb/s net	60 Gb/s raw 48 Gb/s net	120 Gb/s raw 96 Gb/s net	168 Gb/s raw 162.9 Gb/s net

How Infiniband is Managed

Infiniband is designed as a trusted network. The network is managed by a *subnet manager* which does the following:

- Periodically sweep the network, looking for topology changes, checking for errors, etc.
- Build a cohesive model of the network topology
- Load the switch forwarding tables with the LID/Port mapping

Infiniband Topologies

Infinband puts very little restriction on the physical topology of the network. Basically, since the Subnet Manager loads all the forwarding tables, as long as you can build an appropriate graph parsing algorithm, and implement it in a subnet manager, you can use a topology. This allows some much more interesting topologies than the common Ethernet and TCP/IP networks usually use.[1]

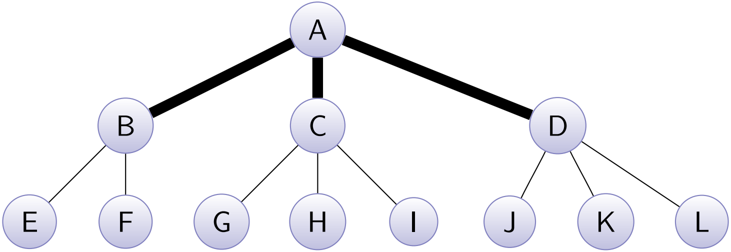
[1] Technically you can use any topology with Ethernet and TCP/IP as well, but it takes a huge amount of work, with lots of VLANs and stub routers, etc., to work around the Spanning-tree protocols. I don't recommend trying it.

Possible Topologies

- Tree/Fat-Tree
- Fully-connected Mesh
- Rectangular Mesh
- Toroidal Mesh
- Hypercube
- Folded-Clos Network

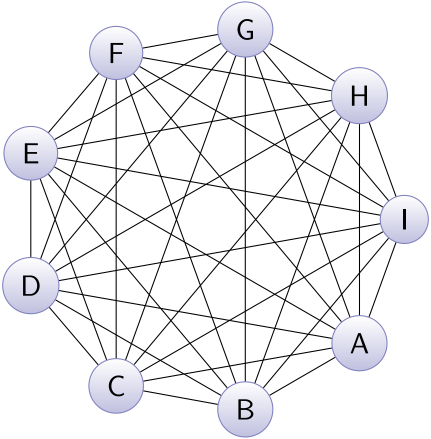
Fat Tree Example

A *Fat Tree* is basically a tree with increased bandwidth (faster links or more links) between upper tiers relative to lower tiers



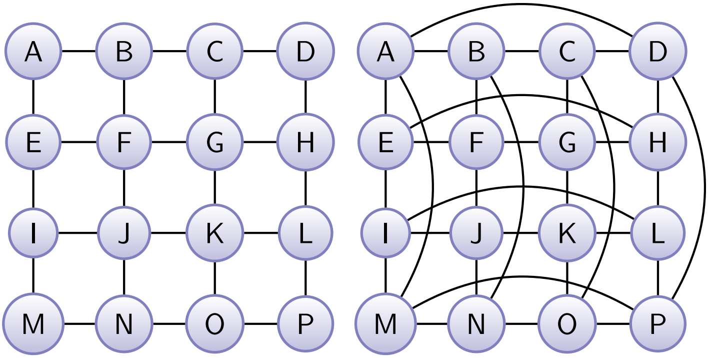
Fully-connected Mesh Example

Shortest hop-count (1 hop) from any point to any other point; takes a huge amount of cables.



Rectangular/Toroidal Mesh Example

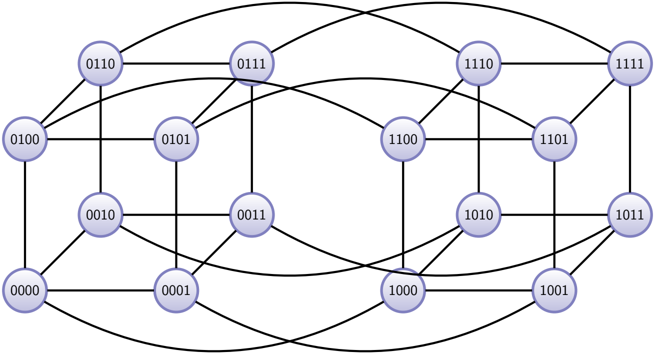
Excellent for large topologies (no spine switches to buy); higher hop count than other options, depending on size



(a) Rectangular (b) Toroidal

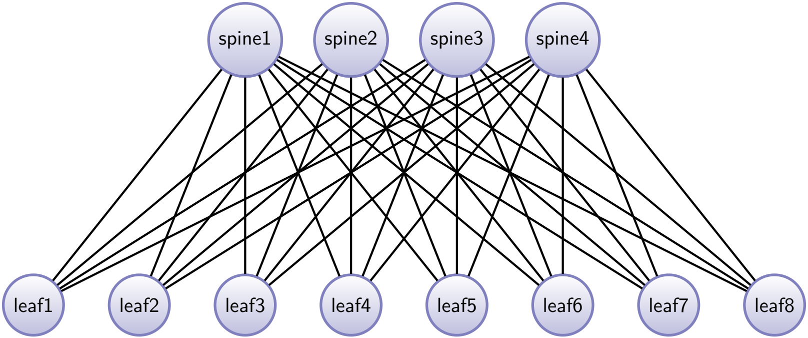
Hypercube example (4-d)

Used rarely; for *d* dimensions, no more than *d* hops from any other point in the topology.

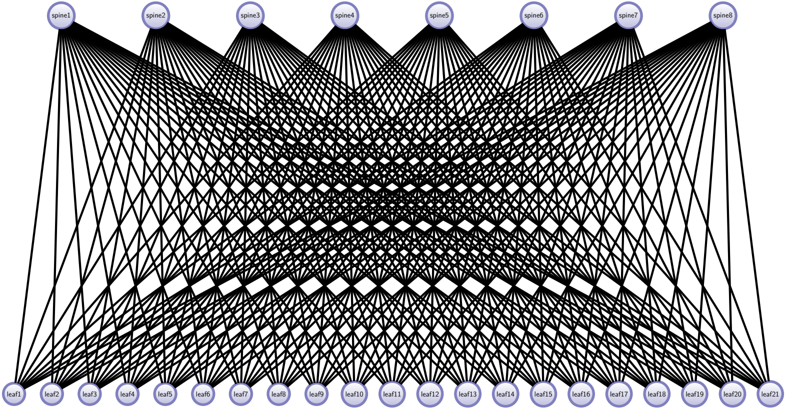


Folded Clos Network Example

Most common approach for small or medium-scale Infiniband fabrics; each leaf has 1 link to each spine



BYU Supercomputing's Clos Network



Upper Layer Stack

The protocol stack includes several optional components to enable application communication:

- **SRP** SCSI RDMA Protocol - Block Storage Protocol; competing with iSER
- **iSER** iSCSI extensions for RDMA - Block Storage Protocol; competing with SRP
- **IPoIB** IP over Infiniband - not the most efficient, but works
- **Verbs** Native IB API for general application use
- **SDP** Sockets Direct Protocol - basically sockets protocol for IB

Other (usu. Proprietary) Extensions

Other extensions exist, usually implemented in a proprietary fashion, including the following:

FCoIB
: Fibre-Channel traffic over IB

ETHoIB
: Ethernet over IB

FlexBoot
: PXE-like network booting

Message Passing

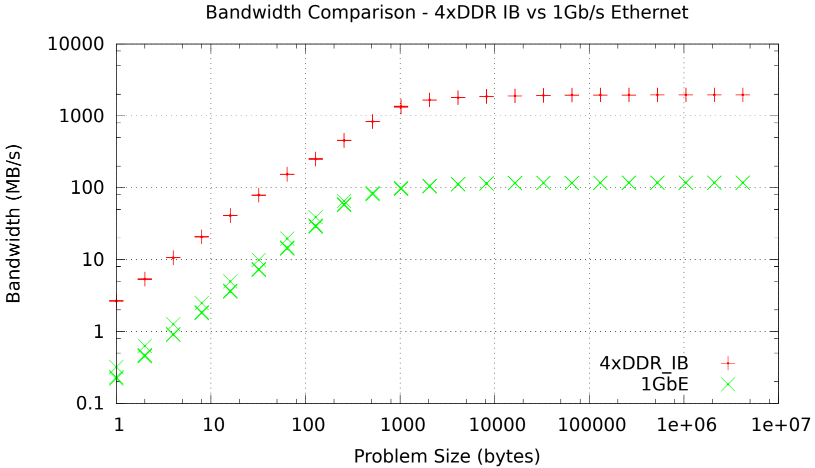
Any application can utilize IB, if it is written or ported to do so. In HPC, most applications use a message-passing library like MPI, which in turn uses the Verbs API to do its work.
Several dozen MPI implementations exist, but the most common that can utilize Infiniband are:

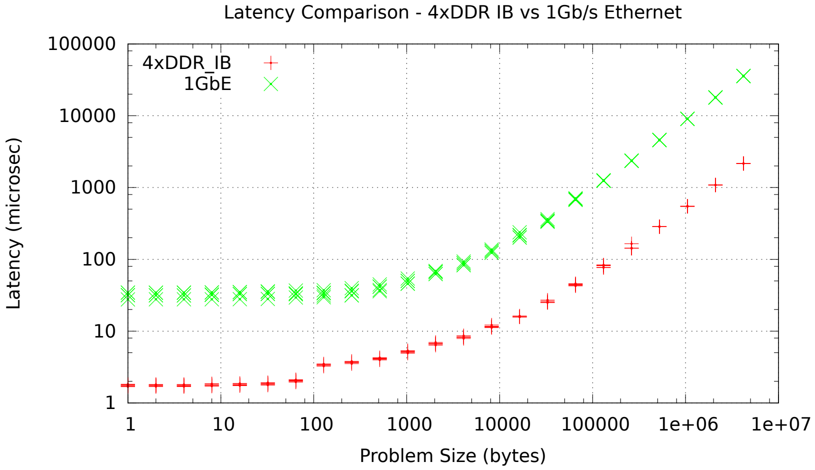
- OpenMPI
- MVAPICH
- Intel MPI
- HP/Platform MPI

Costs

In general:

- Gigabit Ethernet comes on-board for most hosts, so it has very little cost
- 10-Gigabit Ethernet is coming on-board for some hosts
- Per-port cost for 4xQDR Infiniband (40 Gb/s) is usually less than 10-Gigabit Ethernet, but this changes
- Infiniband HCAs can be repurposed (via firmware change) to be 10-Gigabit Ethernet NICs


Bandwidth Comparison - 4xDDR IB vs 1Gb/s Ethernet
Bandwidth (MB/s)
10000
1000
100
10
1
0.1
1
10
100
1000
10000
100000
1e+06
1e+07
Problem Size (bytes)
4xDDR_IB
1GbE


Latency Comparison - 4xDDR IB vs 1Gb/s Ethernet
4xDDR_IB
1GbE
Latency (microsec)
100000
10000
1000
100
10
1
1
10
100
1000
10000
100000
1e+06
1e+07
Problem Size (bytes)

Questions?

Any questions?