High-Performance Computing Networks at BYU

Lloyd Brown

November 1, 2012

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
- Therefore for HPC problems, communication is key.
  - For inter-process communication
  - For communicating with data storage

What kind of communication are we talking about?

- Programs that utilize multiple processors to split up work, need to communicate between threads or processes, to coordinate efforts, report on results, etc.
- Communication between threads/processes on the same host ("*Intra*-node" communication) is extremely fast (usually via shared memory)
- If the processes are on different hosts, we have to go out to some communication fabric ("*Inter*-node" communication)
  - There's a lot of research in speeding up *intra*-node communication, but that's more of a Computer Science or Electrical Engineering problem. We'll sepend our time today on *inter*-node communication

Technologies for *inter*-node communication

Examples of technologies for *inter*-node communication include:

- Ethernet
- Fibre Channel
- Infiniband
- RS-232

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)

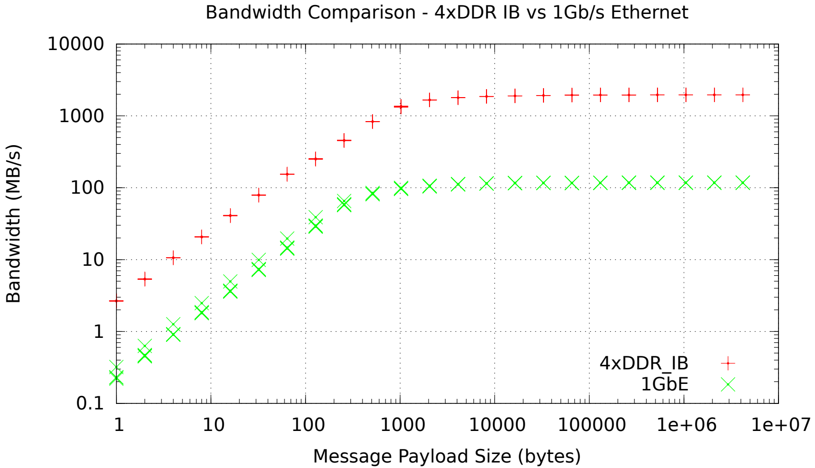
	SDR	*DDR*	*QDR*	*FDR*
1x	2.5 Gb/s raw 2 Gb/s net	5 Gb/s raw 4 Gb/s net	10 Gb/s raw 8 Gb/s net	14 Gb/s raw 13.6 Gb/s net
4x	10 Gb/s raw 8 Gb/s net	20 Gb/s raw 16 Gb/s net	40 Gb/s raw 32 Gb/s net	56 Gb/s raw 54.3 Gb/s net
12x	30 Gb/s raw 24 Gb/s net	60 Gb/s raw 48 Gb/s net	120 Gb/s raw 96 Gb/s net	168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

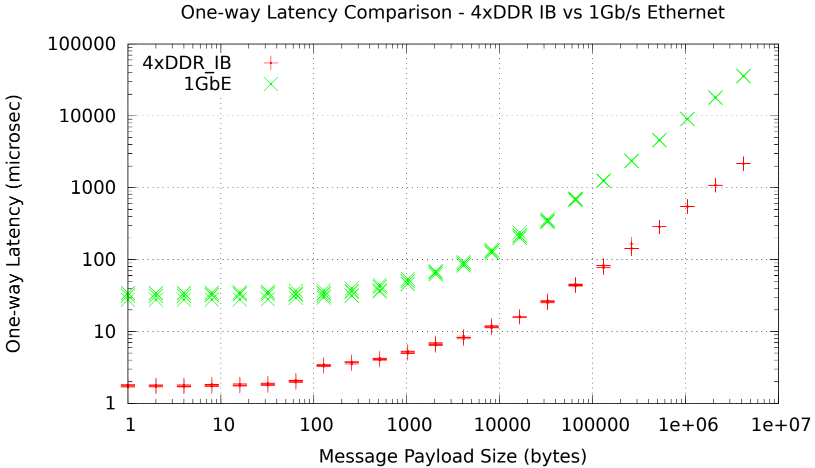
Performance at BYU's FSL

The graphs shown in the next couple of slides represent the bandwidth and latency performance of 4xDDR Infiniband vs 1Gb/s Ethernet at the Fulton Supercomputing Lab.

- All tests were performed host-to-host with one intervening switch (eg. host-switch-host)
- All tests utilize increasing message sizes, to demonstrate where one effect ends and the other starts
- Tests were performed using the "osu_bw" and "osu_latency" binaries from the OSU Micro-Benchmarks for MPI (a.k.a. "OMB")[1]

[1] http://mvapich.cse.ohio-state.edu/benchmarks/


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
Message Payload Size (bytes)
4xDDR_IB
1GbE


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

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.

- The Subnet Manager loads all the forwarding tables into the switches
  - as long as you can build an appropriate graph parsing algorithm, and implement it in a subnet manager, you can use a topology
  - allows some much more interesting topologies than those commonly Ethernet and TCP/IP networks usually use.[2]

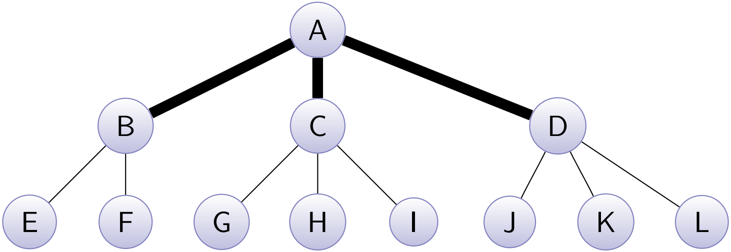
[2] Technically you can use any topology with Ethernet as well. It just takes a huge amount of very-messy work, for very little benefit. I don't recommend trying it.

Possible Topologies

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

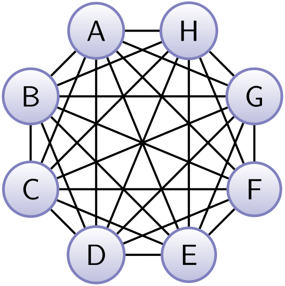
Fat Tree Example

A *Fat Tree* is basically a tree with increased bandwidth (faster links or more links) between upper tiers relative to lower tiers; Ethernet has no problems with this one, so it's not terribly exciting



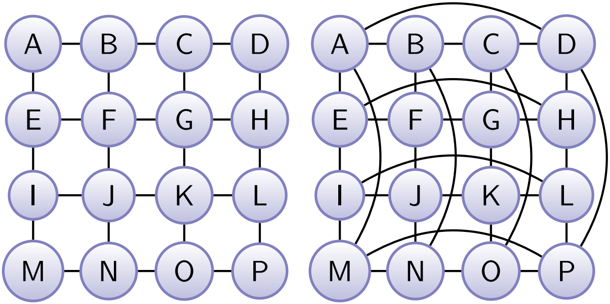
Fully-connected Mesh Example

- Pro: Shortest hop-count (1 hop) from any point to any other point
- Con: takes a huge amount of cables, and the cable count increases very, very quickly.



Rectangular Mesh / Torus Example

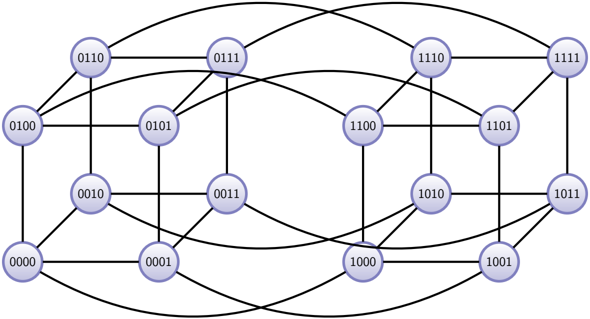
- Pro: Excellent for large topologies (no spine switches to buy)
- Con: Higher hop count than other options, depending on size and shape



(a) Rectangular (b) Torus

Hypercube[3] example (4-dimensional)

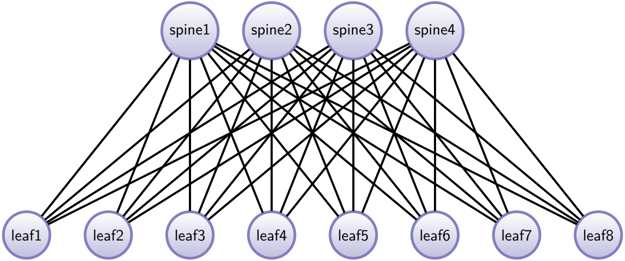
- Pro: for *d* dimensions, no more than *d* hops from any other point in the topology
- Con: cables/ports at each endpoint increase linearly with the dimension



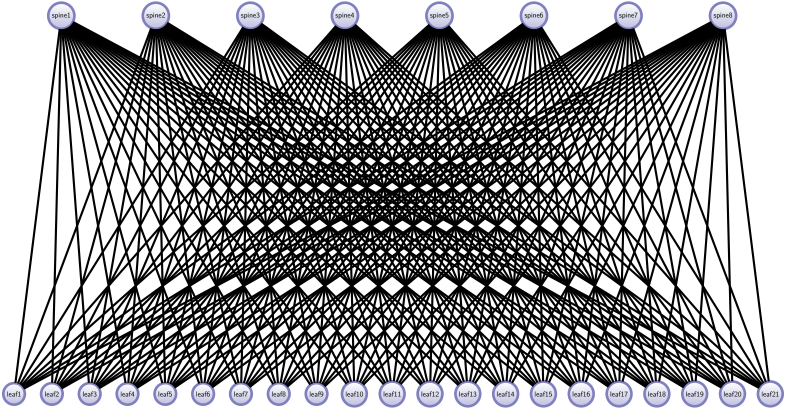
[3] Note that this is really just a special case of a Torus.

Folded Clos Network Example

- Pros:
  - Most common approach for small or medium-scale Infiniband fabrics
  - Well understood (how larger IB switches are designed internally)
  - Redundant; 1 link from each leaf to each spine
- Con: Scalability limited by the port count of spine & leaf switches



BYU Supercomputing's Clos Network



What are some important characteristics for evaluating topologies?

Questions?

Any questions?