High-Performance Computing Networks at BYU

Lloyd Brown

March 17, 2010

1. Outline
2. What is HPC?
3. Types of HPC
4. Infiniband
   - Terminology
   - Physical Layer Characteristics
   - Encoding
   - Subnet Management
   - Topologies
5. Building your own HPC Cluster
   - Hardware
   - OS Software
   - Infrastructure Software
   - Computational Software
6. Summary/Conclusions

What makes a supercomputer, super?

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

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
        - GPUs
        - FPGAs
    - Specialty/Proprietary Interconnects
        - Infiniband
        - NUMALink
        - Quadrics
        - Myrinet
- 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 (commonly 40Gb/s)
  - extremely low latency (one-way < 10 μs, compared to approx. 32 μs for 1GbE)

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*
1x	2.5 Gb/s	5 Gb/s	10 Gb/s
4x	10 Gb/s	20 Gb/s	40 Gb/s
12x	30 Gb/s	60 Gb/s	120 Gb/s

Does this look familiar?

- Is there any other common computer technology that looks like this from a physical layer? Using multiple lanes, with per-lane speed doubling each successive iteration of the standard?
- How about *PCI-Express*?

Encoding Overhead

The Infiniband standard uses an 8b/10b encoding, meaning that the net speed is 80% of the raw speed:

	SDR	*DDR*	*QDR*
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
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
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

How Infiniband is Managed

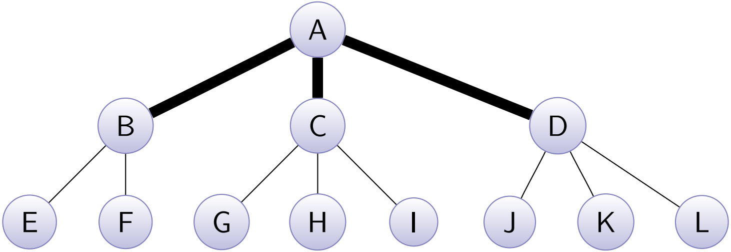
Infiniband is designed as a trusted network. The network is managed by a *subnet manager* which does the following:

- 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

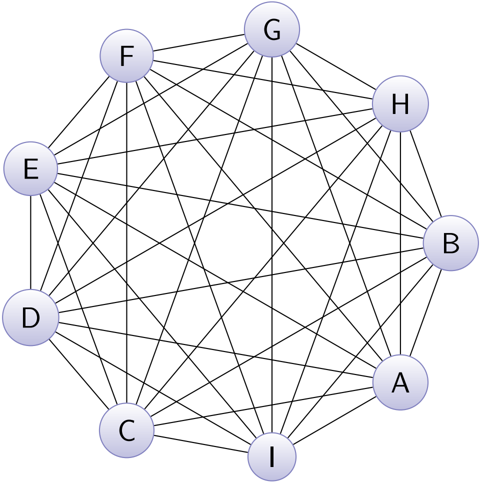
Possible Topologies

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

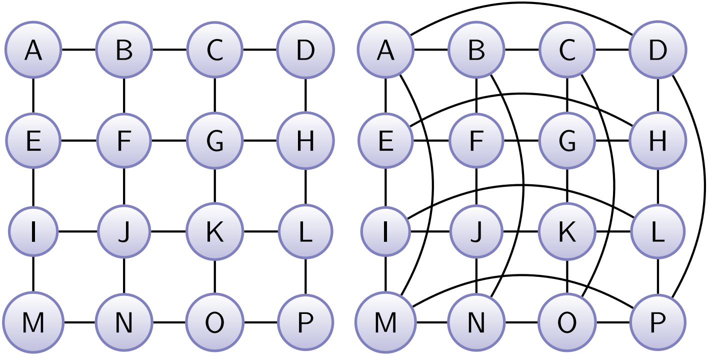
Fat Tree Example



Fully-connected Mesh Example



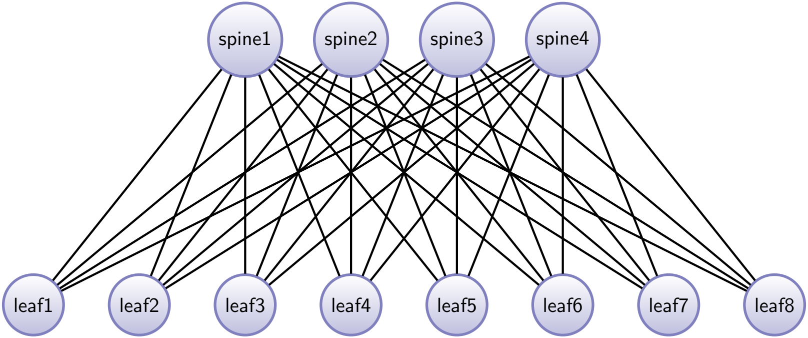
Rectangular/Toroidal Mesh Example



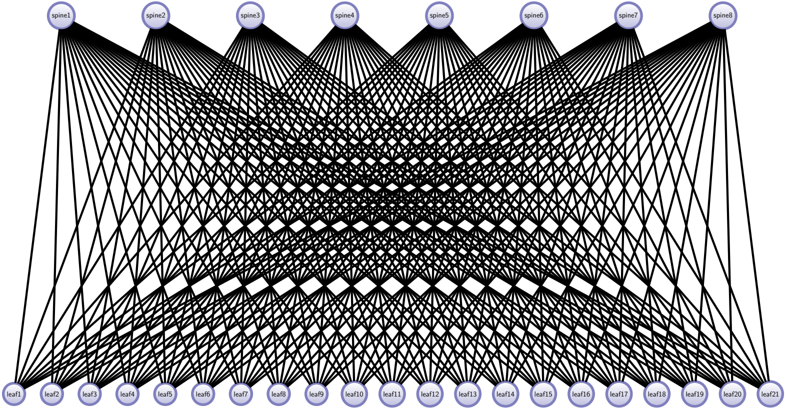
(a) Rectangular

(b) Toroidal

Clos Network Example



BYU Supercomputing's Clos Network



Components needed to build an HPC Cluster

To build your own HPC cluster, consider the following components:

- Hardware
- Operating System Software
- Infrastructure Software
- Computational Software

Hardware Considerations

When considering system hardware, be aware of the following considerations:

- If a process is using resources on multiple nodes, it's significantly easier if the nodes' hardware is homogeneous.
- You need to know the task's or software's requirements, and build the system appropriately in the following areas:
    - Processor features and speed
    - RAM
    - Network Performance (bandwidth and latency)
    - Storage requirements (total capacity, throughput, and IOPS)

Compute Node Operating System

- Each computational node needs to have a functioning operating system. *Linux* is the most common, usually installed either through a *golden image* approach (usually vendor-provided), or scalable, scripted installer, eg. *NPACI Rocks*.
- You will need to make sure your computational software is supported on the system. For example, many more commercial software packages run on *RedHat Enterprise Linux* than on *Ubuntu*.

Organizing the effort

- If you're the only person using the system, you can just run your tasks directly. If, however, you need to allow multiple users to have access, etc., you will probably need a queuing mechanism, eg. Moab/Torque, PBSPro, Slurm, SGE, LoadLeveler, LSF
- You will need to monitor the system for hardware and software failures. Think something like *ganglia*.

Actually doing work

In order for the system to be useful, you need software to do some calculations. Some things to consider here:

- How will I get the software to utilize all the resources (eg. processors) available? Do I need to use some form of communication framework like MPI to coordinate efforts, or will I just launch independent tasks
- Is there any form of tuning that I can do to make the software more efficient? For example, if it's compiled software, am I taking advantage of compilation optimizations, eg. SSE, or specialty BLAS implementations like Intel MKL or GotoBlas?

What does this all mean?

- In general, clusters of commodity hardware are the cheapest approaches to HPC, but it will vary depending on situation.
- It is possible to set up a small HPC cluster without much hardware cost, or any real software cost. Just don't expect anything über-cool like Infiniband.
- You absolutely must understand your software, and its requirements
- Not everything works like Ethernet and TCP/IP. Network technologies like Fibre Channel and Infiniband throw away a number of the basic assumptions of Ethernet.

Questions?