Engineering Cheap Supercomputers

CS100 Lecture
3:30PM CB106; Tuesday, October 21, 2003

Hank Dietz
Professor and James F. Hardymon Chair in Networking
Electrical and Computer Engineering Department
University of Kentucky
Lexington, KY 40506-0046
hankd@engr.uky.edu
http://aggregate.org/hankd/
What Is A Supercomputer?

• One of the most expensive computers?
• A very fast computer?
• Really two key characteristics:
  1. Computer that solves big problems quickly…
     stuff that wouldn’t fit on a PC
     stuff that would take too long to run on a PC
  2. Performance of the computer can scale…
     more money buys a faster machine
• A well-engineered supercomputer can be cheap
The Key Is Parallel Processing!

- Simultaneous execution of portions of a computation
- Process $N$ "pieces" simultaneously, can get up to factor of $N$ speedup
- Modular hardware designs
  - Relatively easy to scale
  - Higher availability (if not reliability)
The Evolution Of Supercomputers

• Most fit survives, even if it’s ugly
• Rodents outlast dinosaurs...

and bugs will outlast us all!
When Does Supercomputing Make Sense?

• When you need results **now**!
  • Top500 speed-up *1.4X every 6 months*!
    Just waiting might work....
  • Optimizing your existing code helps a lot;
    do that first!

• When your application takes enough time per run to justify the effort and expense

• Our new technologies don’t change the basics... they primarily improve price/performance
What Is A Cluster Supercomputer?

- Not a "traditional" supercomputer...?
- Is *The GRID* a cluster?
- Is a *Farm* a cluster?
- A *Beowulf*?
- A supercomputer made from **interchangeable parts**
  - What are interchangeable supercomputer parts?
  - Some PC parts you don’t need (and shouldn’t want)
- Often, Linux PC "nodes"
Parts... Vs. In A Traditional Supercomputer

- Processors: primarily *AMD Athlon* and *Intel Pentium 4* (within 2X of best available @ very low cost)
- Motherboards
- Memory
- Disks
- Video
- Network
- Physical Packaging
- But parts are tuned for PC use, compatibility... not for use in a cluster supercomputer
AMD Athlon
Types Of Hardware Parallelism

Want hardware able to use all the parallelism in our software

<table>
<thead>
<tr>
<th>Hardware Type</th>
<th>Description</th>
<th>Parallelization Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pipelining</td>
<td>overlapped execution</td>
<td>automatic HW</td>
</tr>
<tr>
<td>Superscalar/VLIW/EPIC</td>
<td>1 instruction stream feeds multiple pipelines</td>
<td>automatic HW/SW</td>
</tr>
<tr>
<td>SWAR</td>
<td>SIMD (vector) Within A Register</td>
<td>SW vectorization</td>
</tr>
<tr>
<td>SMP</td>
<td>Symmetric MultiProcessor using shared memory</td>
<td>SW threads, etc.</td>
</tr>
<tr>
<td>Cluster</td>
<td>Homogeneous, dedicated, tightly-coupled PCs</td>
<td>SW messaging, etc.</td>
</tr>
<tr>
<td>&quot;Beowulf&quot;</td>
<td>Cluster using mostly commodity parts</td>
<td>SW messaging, etc.</td>
</tr>
<tr>
<td>Farm</td>
<td>Mostly homogeneous, dedicated PCs</td>
<td>SW messaging, etc.</td>
</tr>
<tr>
<td>Grid</td>
<td>Heterogenous, intermittently available PCs</td>
<td>SW messaging, etc.</td>
</tr>
</tbody>
</table>
Engineer To Meet Application Needs

• Know your application(s)
• Tune your application(s)
• Know your budget (money, power, cooling, space)
• Hardware configuration options
• Software configuration options
Engineering A Cluster

- This is a *systems* problem
- Optimize integrated effects of:
  - Hardware configuration
  - Computer architecture
  - Compiler optimization/parallelization technology
  - Operating system
  - Application program
- The payoff for good engineering can be huge!
  (The penalty for bad engineering is huge.)
One Aspect: The Interconnection Network

- Parallel supercomputer "nodes" need to talk
- Bandwidth
  - Bits transmitted per second through a path
  - **Bisection bandwidth** is more important...
- Latency
  - Time to send something from here to there
  - Harder to improve than bandwidth...
Latency Determines Parallel Grain Size

\[ t_{\text{grain}} \]

P0

P1

Engineering Cheap Supercomputers
Network Design

- Assumptions
  - Links are bidirectional
  - Fixed maximum number of network interfaces per node
  - Point-to-point message communications
- Topologies
- Hardware
- Software
No Network
Direct Connections (Fully Connected)
Toroidal Hyper-Meshes

- A generic term for the most common network topologies that use nodes as through-routing elements
- Includes rings, meshes, and hypercubes
- Can have any dimensionality: 1D, 2D, 3D, 4D, etc.
- Optionally have toroidal "wrap around" links
- Through-routing tends to imply high latency, and some processor overhead
Toroidal 1D Mesh, AKA Ring
Physical Layout of a Ring

Engineering Cheap Supercomputers
Non-Toroidal 2D Mesh

Engineering Cheap Supercomputers
Non-Toroidal 3D Mesh or 3-Cube

Engineering Cheap Supercomputers
Switch Networks: What’s a Switch?

- **Hub**: a shared bus in a box
- **Full Duplex Repeater (FDR)**: two buffered busses
- **Switch**: interconnected switching elements
  - **KxK switch chips interconnected (usually in a ring)**
  - **Bandwidth** depends on interconnect; want *wire speed* and *non-blocking*
  - **Latency** a little higher than a hub or FDR
- **Router**: an expensive switch with smart routing abilities
  - **Bandwidth** can be helped by *trunking*
  - **Latency** often higher than dumb switch (due to routing processor)
The Ideal Switched Network

- No pair of nodes are separated by more than the latency of a single switch
- *Bisection Bandwidth* of each switch contributes to the total bisection bandwidth
Simple Switch (8-port)

Engineering Cheap Supercomputers
Channel Bonding

- The original "Beowulf" technology
- Use multiple, parallel, ethernet NICs
  - Dynamically load share between paths;
    Often, 2-5 100Mb/s NICs can match one Gb/s
  - Latency is not seriously degraded
- Beowulf implementation clones MAC addresses
  - Confuses switches if two NICs reachable
  - We are building a version that doesn’t
- Bonding performance limited by:
  NIC overhead, PCI bus, interrupt traffic
Channel Bonding (2-way, 8-port switches)
Switch Fabrics

- Number of nodes exceeds ports per switch... use multiple switches for connectivity
- Big switches tend to use internal ring-of-rings (often with mediocre bandwidth & latency)
- Most supercomputers, especially large clusters, use hierarchical switch fabrics
  - Most common are trees and fat trees
  - Want bisection bandwidth preserved at all tree levels
Tree (4-port switches)

Engineering Cheap Supercomputers
A Better Tree (4-port switches)
Fat Tree (4-port switches)
Our Flat Neighborhood Network (FNN) Insight

• The ideal switch network uses one wide switch, but all PCs don’t have to share the same switch

• In a FNN:
  • Every pair of PCs shares at least one switch (each switch is a network "neighborhood")
  • Multiple NICs/PC connect to multiple neighborhoods
  • Topology is flat (no switch is connected to another switch)
"Universal" FNN (4-port switches)

Engineering Cheap Supercomputers
Flat Neighborhood vs. Fat Tree

- Typically, FNN uses no more switches than Fat Tree, but uses dumb switches and more NICs
- Flat vs. Fat Latency
  - 8 PCs, 4 port: 1.0 vs. \((1.0+3.0\times6)/7 = 2.7\) switch delays
  - 64 PCs, 32 port: 1.0 vs. 2.5 switch delays
- Flat vs. Fat Pairwise Bandwidth
  - 8 PCs, 4 port: 1.29 vs. 1.0 NIC bandwidth units
  - 64 PCs, 32 port: 1.48 vs. 1.0 NIC bandwidth units
- Incremental improvement using FNNs:
  E.g., a 4th NIC/PC yields 1.86 NIC bandwidth units
KLAT2, Gort, & Klaatu

--

Engineering Cheap Supercomputers
Behind The KLAT2 Supercomputer

Engineering Cheap Supercomputers
KLAT2’s Asymmetric, Computer-Evolved, FNN

KLAT2’s flat neighborhood network
Above: physical wiring
Right: neighborhood pattern

Engineering Cheap Supercomputers
KLAT2’s FNN Changed The World...

- Single-switch latency, high (bisection) bandwidth
- KLAT2’s Firsts:
  - 1st network designed (evolved) by computer
  - 1st network deliberately asymmetric
  - 1st computer to break $1000/GFLOPS
- 160+ news stories & articles about KLAT2
- Various awards:
  2000 Gordon Bell (for CFD Price/Performance)
  2000 HPC Games, Most Innovative Architecture
  2001 Computerworld Smithsonian, among 6 ITs most advancing science
But... "Limited" Scalability for FNNs?

![Graph showing minimum ports per switch using no more than 5 NICs/PC]
Summer 2002: Sparse FNNs

- FNN ensures single-switch latency for all node pairs; SFNN only for specified node pairs
- SFNNs are a qualitative improvement in scalability:
  - The number of node pairs communicating grows slowly; each additional node adds $O(1)$ to $O(\log N)$, not $O(N)$
  - Theoretically scales to over 10,000 nodes... using standard parts!
- Looks great in simulations....
- Does it work in practice?
June 2003: Kentucky Asymmetric Zero (KASY0)

- Improved & generalized FNN interface software
- New cluster configuration software (Warewulf)
- SFNN for KASY0 vs. FNN for KLAT2
  - 128+4 nodes vs. 64+2 nodes
  - Bandwidth 36Gb/s vs. 25Gb/s; Latency identical
  - 3 NICs/24-port switches vs. 4 NICs/32-port;
    Hardware cost $4,200 vs. $8,100
- SFNNs are really the first step toward fully general "engineered networks" meeting specific latency and bandwidth requirements for particular communications... there will be a KASY1 ;-)
Engineering Cheap Supercomputers
Immediate Results

• KASY0’s Firsts:
  • 1st Terascale supercomputer in Kentucky
    (1.062TFLOPS vs. 0.672TFLOPS SDX HP Superdome)
  • 1st with physical layout optimized by computer
  • 1st supercomputer to break $100/GFLOPS;
    total cost of $39,454.31, over 482GFLOPS on Linpack
  • Over 50 students helped build KASY0 (vs. 30 for KLAT2)
  • 40+ news stories & articles, two more invited, etc.
• KASY0 set new record for POVRay render speed
  (at any price!)
What The Heck Is POV-Ray?
Supercomputers R Us:
We Make Supercomputing Cheap!

- Aggregate Function Communication
  *hardware helping processors agree on things*
- SWAR (SIMD Within A Register)
  *MMX, 3DNow!, SSE, etc., for scientific computing*
- Flat Neighborhood Networks (FNNs)
  *asymmetric optimized design that outperforms all others*
- Helping scientists/engineers use our technologies
  *ME, CE, Bio, etc. at UK; other universities from Louisville to Keele*
You Can Help Build A Supercomputer!

9AM-5PM, Wed., July 16, 2003, in 672 Anderson Tower...

FREE PIZZA*
* Some supercomputer assembly required.

Keep building KASYS, the TeLICPS supercomputer "not all-GAAS"
An advanced supercomputer experience week
For more details see: http://aggregate.org/KASY3/

Help build the future
Tuesday, April 11, 2000
Room 672A... yes, all day!

Engineering Cheap Supercomputers
The network has evolved.