Supercomputing in Plain English

An Overview of High Performance Computing

Henry Neeman, Director
OU Supercomputing Center for Education & Research
University of Oklahoma
Wednesday August 29 2007

This is an experiment!

It’s the nature of these kinds of videoconferences that **failures are guaranteed to happen!**

NO PROMISES!

So, please bear with us. Hopefully everything will work out well enough.

Access Grid/VRVS

If you're connecting via the Access Grid or VRVS, the venue should have been sent to you by e-mail, hopefully.

NOTE: So far, we haven't had a very successful test of AG or VRVS.

iLinc

We only have about 40-45 simultaneous iLinc connections available.

Therefore, each institution has at most one iLinc person designated.

If you're the iLinc person for your institution, you've already gotten e-mail about it, so please follow the directions.

If you aren't, you can't become it, because we're completely out of iLinc connections.

Quicktime Broadcast

If you don't have iLinc, you can connect via Quicktime:

rtsp://129.15.254.141/neeman_02.sdp

We strongly recommend using Quicktime player, since we've seen it work.

When you run it, traverse the menus

File -> Open URL

Then paste in the rstp URL the Movie URL space, and click OK.

Phone Bridge

If all else fails, you can call into our phone bridge:

1-866-285-7778, access code 6483137#

Please mute yourself and use the phone to listen.

Don’t worry, I’ll call out slide numbers as we go.

To ask questions, please use Google Talk.

Google Talk

To ask questions, please use our Google Talk group chat session (text only).

You need to have (or create) a gmail.com account to use Google Talk.

Once you've logged in to your gmail.com account, go to:

http://www.google.com/talk/

and then contact the user named:

oscer.sipe

Alternatively, you can send your questions by e-mail to oscer.sipe@gmail.com.

This is an experiment!

REMINDER:

It's the nature of these kinds of videoconferences that **failures are guaranteed to happen!**

NO PROMISES!

So, please bear with us. Hopefully everything will work out well enough.

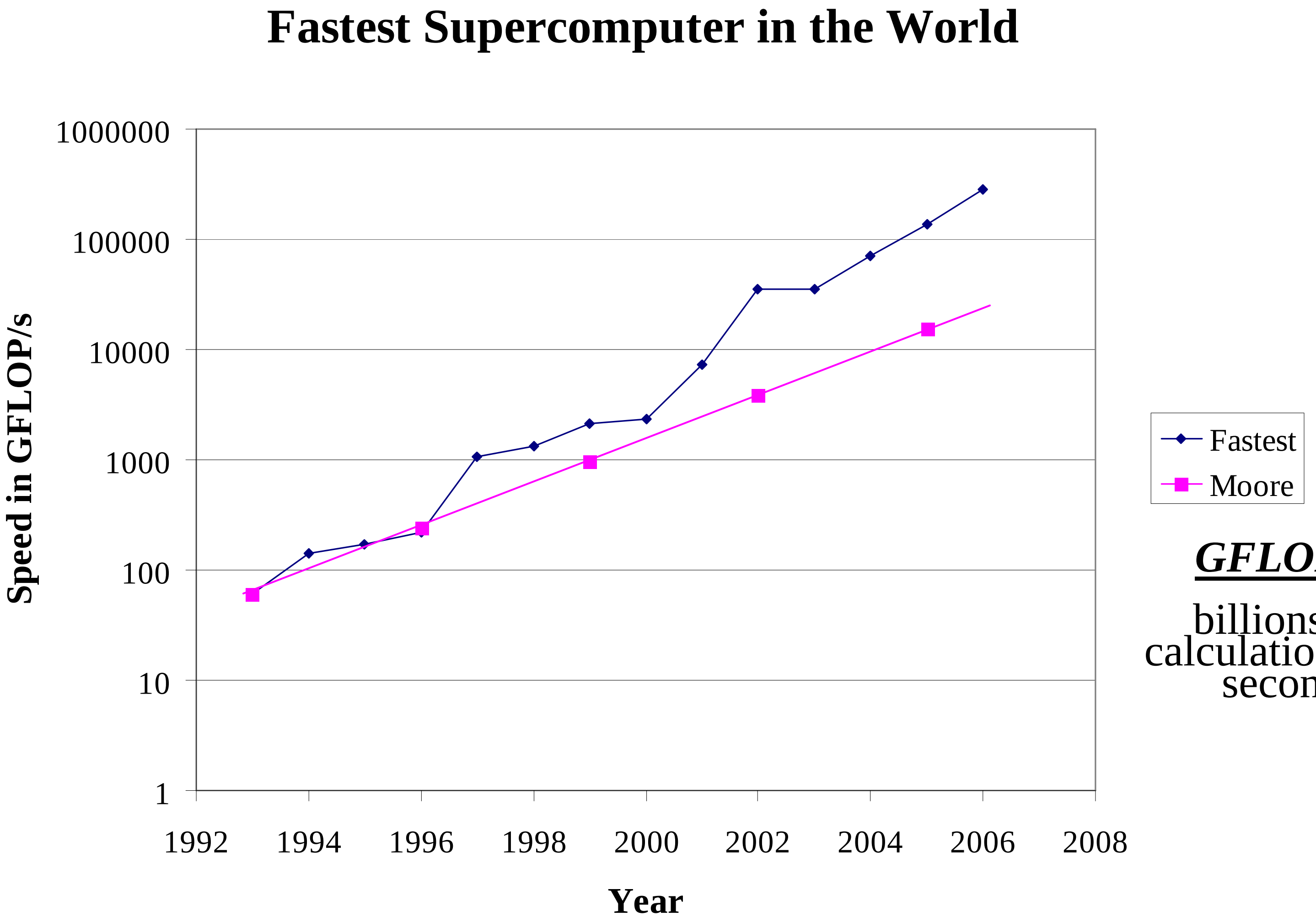
People

Things

What is Supercomputing?

- ***Supercomputing*** is the **biggest, fastest computing right this minute**.
- Likewise, a ***supercomputer*** is one of the biggest, fastest computers right this minute.
- So, the definition of supercomputing is **constantly changing**.
- **Rule of Thumb**: A supercomputer is typically at least 100 times as powerful as a PC.
- **Jargon**: Supercomputing is also known as ***High Performance Computing*** **(HPC)** or ***High End Computing*** **(HEC)** or ***Cyberinfrastructure*** **(CI).**

Fastest Supercomputer vs. Moore



GFLOPs:

billions of calculations per second

What is Supercomputing About?

Size

Speed

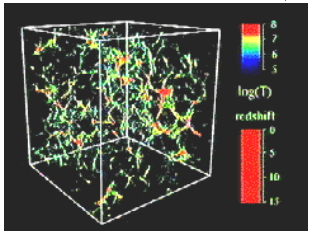
What is Supercomputing About?

- **Size**: Many problems that are interesting to scientists and engineers **can't fit on a PC** – usually because they need more than a few GB of RAM, or more than a few 100 GB of disk.
- **Speed**: Many problems that are interesting to scientists and engineers would take a very very long time to run on a PC: months or even years. But a problem that would take **a month on a PC** might take only **a few hours on a supercomputer**.

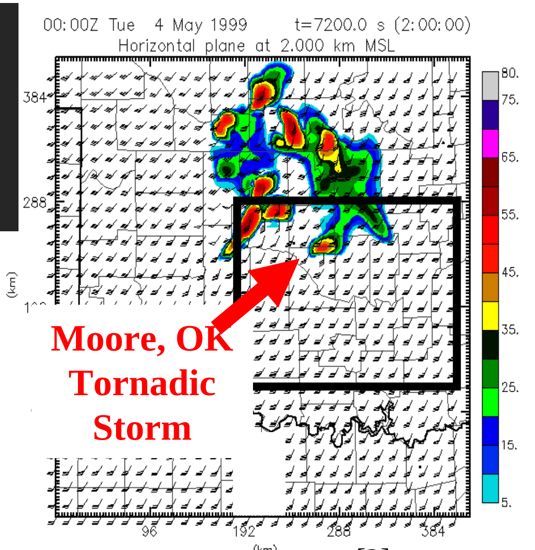
What Is It Used For?

- ***Simulation*** of physical phenomena, such as
  - Weather forecasting
  - Galaxy formation
  - Oil reservoir management

[1]



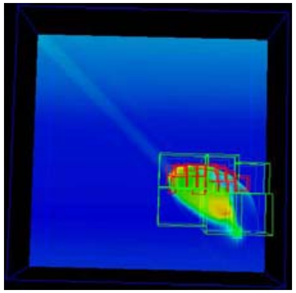
- ***Data mining***: finding **needles** of information in a **haystack** of data, such as
  - Gene sequencing
  - Signal processing
  - Detecting storms that could produce tornados



May 3 1999[2]

- ***Visualization***: turning a vast sea of **data** into **pictures** that a scientist can understand

[3]



What is OSCER?

- Multidisciplinary center
- Division of OU Information Technology
- Provides:
  - Supercomputing **education**
  - Supercomputing **expertise**
  - Supercomputing **resources**: hardware, storage, software
- For:
  - Undergrad students
  - Grad students
  - Staff
  - Faculty
  - Their collaborators (including **off campus**)

Who is OSCER? Academic Depts

- Aerospace & Mechanical Engr
- Biochemistry & Molecular Biology
- Biological Survey
- Botany & Microbiology
- Chemical, Biological & Materials Engr
- Chemistry & Biochemistry
- Civil Engr & Environmental Science
- Computer Science
- Economics
- Electrical & Computer Engr
- Finance
- Health & Sport Sciences
- History of Science
- Industrial Engr
- Geography
- Geology & Geophysics
- Library & Information Studies
- Mathematics
- Meteorology
- Petroleum & Geological Engr
- Physics & Astronomy
- Radiological Sciences
- Surgery
- Zoology

More than 150 faculty & staff in **24 depts** in Colleges of Arts & Sciences, Atmospheric & Geographic Sciences, Business, Earth & Energy, Engineering, and Medicine – with **more to come**!

Who is OSCER? Organizations

- Advanced Center for Genome Technology
- Center for Analysis & Prediction of Storms
- Center for Aircraft & Systems/Support Infrastructure
- Cooperative Institute for Mesoscale Meteorological Studies
- Center for Engineering Optimization
- Fears Structural Engineering Laboratory
- Geosciences Computing Network
- Great Plains Network
- Human Technology Interaction Center
- Institute of Exploration & Development Geosciences
- Instructional Development Program
- Interaction, Discovery, Exploration, Adaptation Laboratory
- Langston University Mathematics Dept
- Microarray Core Facility
- National Severe Storms Laboratory
- NOAA Storm Prediction Center
- OU Office of Information Technology
- OU Office of the VP for Research
- Oklahoma Center for High Energy Physics
- Oklahoma Climatological Survey
- Oklahoma EPSCoR
- Oklahoma Medical Research Foundation
- Oklahoma School of Science & Math
- Robotics, Evolution, Adaptation, and Learning Laboratory
- St. Gregory’s University Physics Dept
- Sarkeys Energy Center
- Sasaki Applied Meteorology Research Institute
- Symbiotic Computing Laboratory

Biggest Consumers

- **<u>Center for Analysis & Prediction of Storms</u>**: daily real time weather forecasting
- **<u>Oklahoma Center for High Energy Physics</u>**: simulation and data analysis of banging tiny particles together at unbelievably high speeds
- **<u>Advanced Center for Genome Technology</u>**: bioinformatics (e.g., Human Genome Project)

Who Are the Users?

Over 380 users so far, including:

- approximately 100 OU faculty;
- approximately 100 OU staff;
- **over 150 students;**
- over 80 off campus users;
- … more being added every month.

Comparison: The National Center for Supercomputing Applications (NCSA), after **20 years of history** and **hundreds of millions in expenditures**, has about **2150 users;*** the TeraGrid is 4000 users.†

* Unique usernames on cu.ncsa.uiuc.edu and tungsten.ncsa.uiuc.edu
† Unique usernames on maverick.tacc.utexas.edu

Why OSCER?

- Computational Science & Engineering has become **sophisticated enough** to take its place alongside experimentation and theory.
- **Most students** – and most faculty and staff – **don’t learn much CSE**, because it’s seen as needing too much computing background, and needs HPC, which is seen as very hard to learn.
- **HPC can be hard to learn**: few materials for novices; most documents written for experts as reference guides.
- **We need a new approach**: HPC and CSE for computing novices – **OSCER’s mandate!**

Why Bother Teaching Novices?

- Application scientists & engineers typically know their applications very well, much better than a collaborating computer scientist ever would.
- Commercial software lags far behind the research community.
- Many potential CSE users don't need full time CSE and HPC staff, just some help.
- One HPC expert can help dozens of research groups.
- Today's novices are tomorrow's top researchers, especially because today's top researchers will eventually retire.

What Does OSCER Do? Teaching

Science and engineering faculty from all over America learn supercomputing at OU by playing with a jigsaw puzzle (NCSI @ OU 2004).

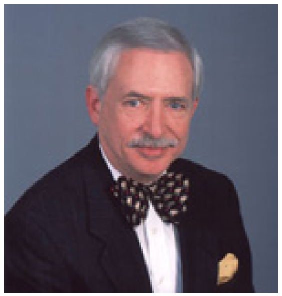
What Does OSCER Do? Rounds

OU undergrads, grad students, staff and faculty learn how to use supercomputing in their specific research.

Okla. Supercomputing Symposium

Wed Oct 3 2007 @ OU

Over 180 registrations already!

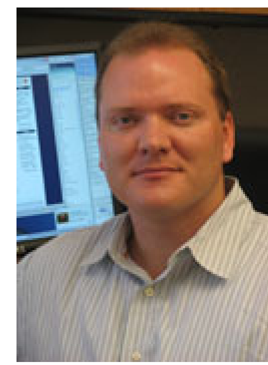


2003 Keynote:
Peter Freeman
NSF
Computer &
Information
Science &
Engineering
Assistant Director

2004 Keynote:
Sangtae Kim
NSF Shared
Cyberinfrastructure
Division Director

2005 Keynote:
Walt Brooks
NASA Advanced
Supercomputing
Division Director

2006 Keynote:
Dan Atkins
Head of NSF's
Office of
Cyberinfrastructure



2007 Keynote:
Jay Boisseau
Director
Texas Advanced
Computing Center
Univ Texas Austin

FREE!

http://symposium2007.oscer.ou.edu/

2007 OSCER Hardware

- **TOTAL:** **14,300 GFLOPs***, 2038 CPU cores, 2766 GB RAM
- **Dell Pentium4 Xeon 64-bit Linux Cluster**
  - 1024 Pentium4 Xeon CPUs, 2176 GB RAM, 6553 GFLOPs
- Aspen Systems Itanium2 cluster
  - 64 Itanium2 CPUs, 128 GB RAM, 256 GFLOPs
- **Condor Pool:** 730 student lab PCs, 7583 GFLOPs
- **National Lambda Rail** (10 Gbps network)
- **NEW! Tape library** (100 TB) – online soon

* GFLOPs: billions of calculations per second

Pentium4 Xeon Cluster

1,024 Pentium4 Xeon CPUs

2,176 GB RAM

23,000 GB disk

Infiniband & Gigabit Ethernet

OS: Red Hat Linux Enterp 4

Peak speed: 6,553 GFLOPs*

*GFLOPs: billions of calculations per second

topdawg.oscer.ou.edu

Pentium4 Xeon Cluster

DEBUTED AT #54 WORLDWIDE,
#9 AMONG US UNIVERSITIES,
#4 EXCLUDING BIG 3 NSF CENTERS

CURRENTLY #289 WORLDWIDE,
#29 AMONG US UNIVERSITIES,
#20 EXCLUDING BIG 4 NSF CENTERS

www.top500.org

DELL™

topdawg.oscer.ou.edu

Hardware: Itanium2 Cluster

64 Itanium2 1.0 GHz CPUs
128 GB RAM
5,774 GB disk
OS: Red Hat Linux Enterprise 4
Peak speed: 256 GFLOPs*
*GFLOPs: billions of calculations per second

schooner.oscer.ou.edu

What is a Cluster?

"… [W]hat a ship is … It's not just a keel and hull and a deck and sails. That's what a ship needs. But what a ship is ... is freedom."

– Captain Jack Sparrow

"Pirates of the Caribbean"

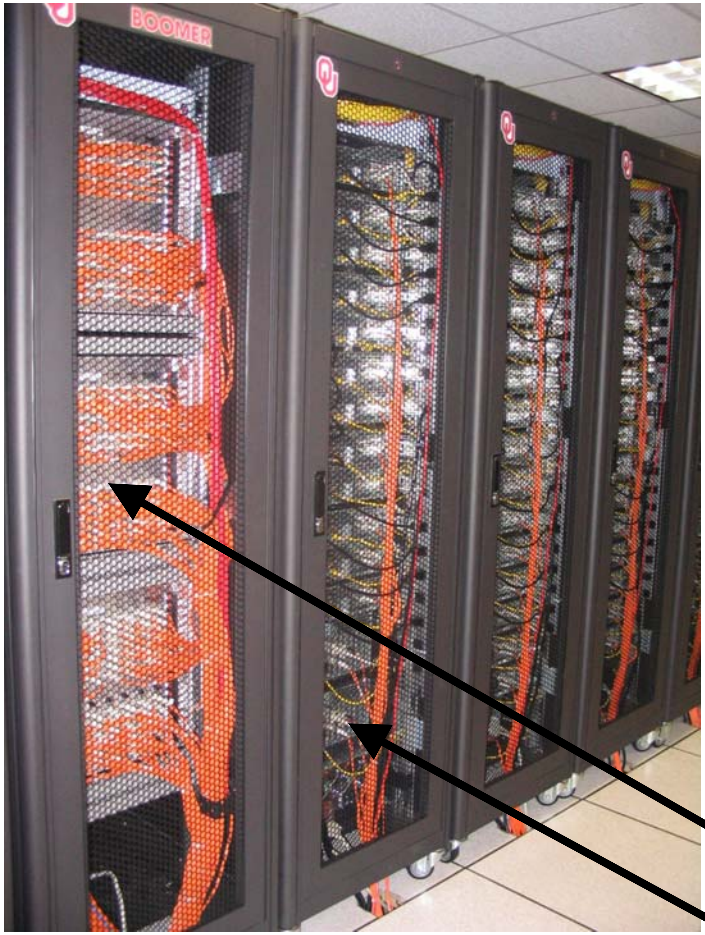
What a Cluster is ….

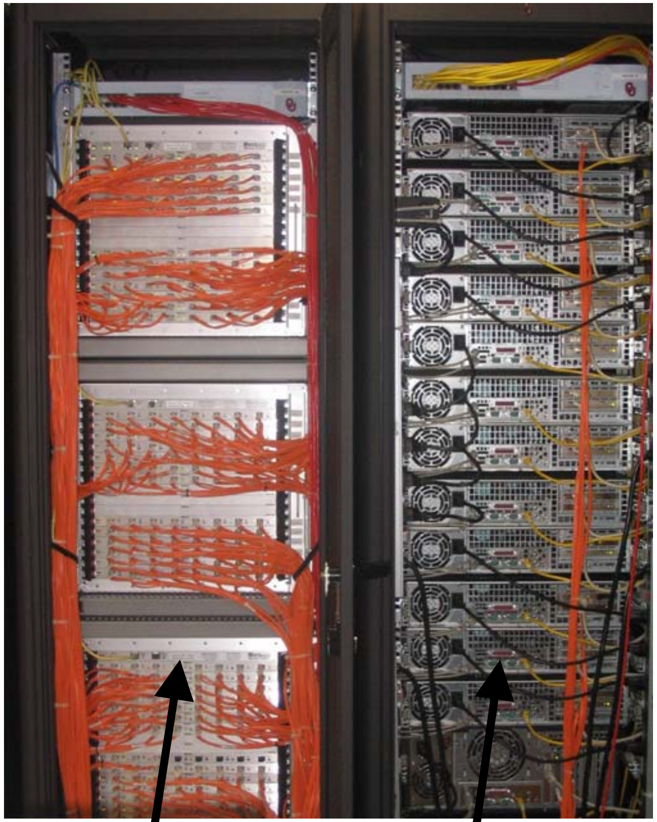
A cluster **needs** of a collection of small computers, called ***nodes***, hooked together by an ***interconnection network*** (or ***interconnect*** for short).

It also **needs** software that allows the nodes to communicate over the interconnect.

But what a cluster **is** … is all of these components working together as if they're one big computer ... a **super** computer.

An Actual Cluster



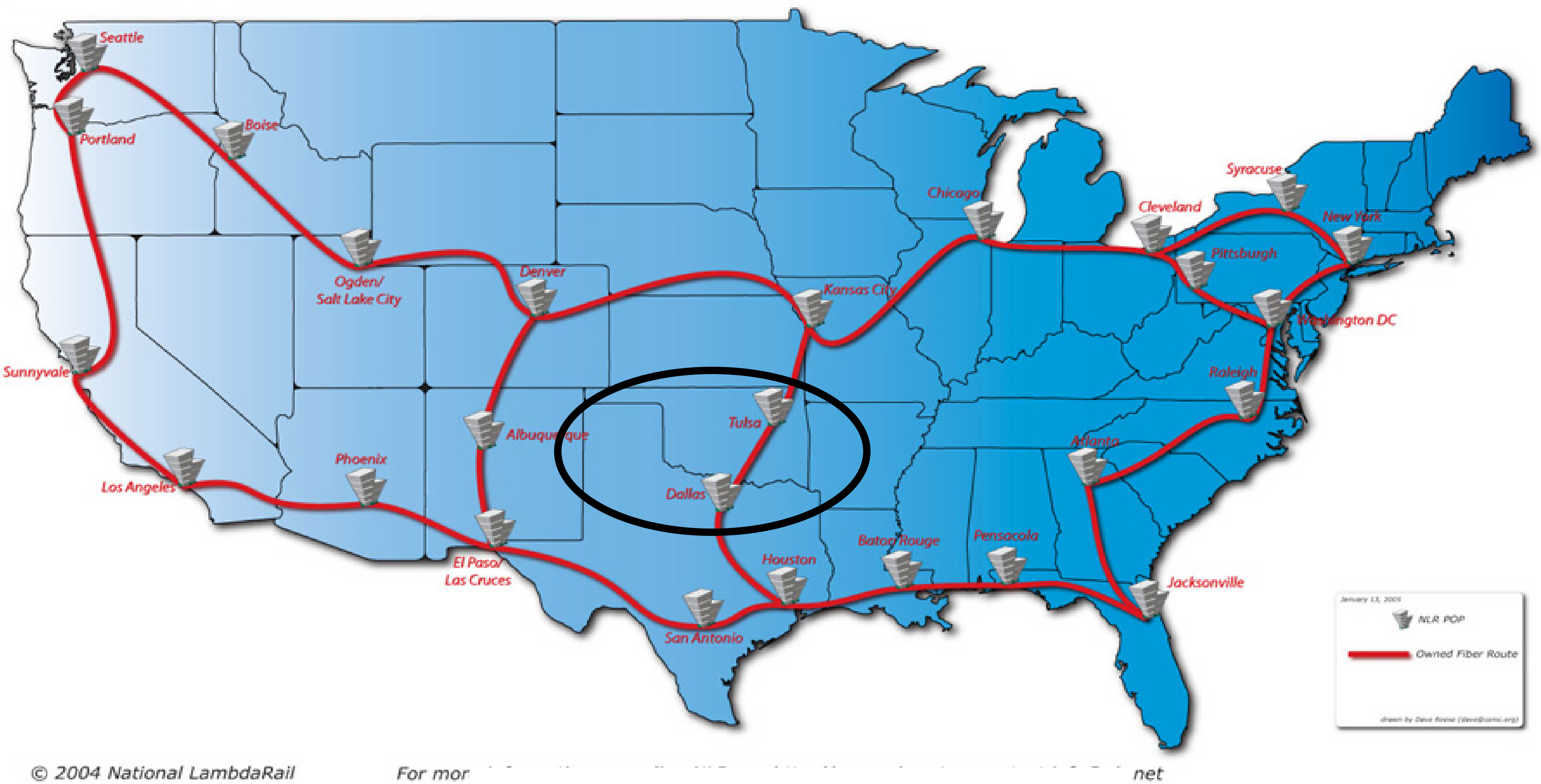


Interconnect

Nodes

NEW! National Lambda Rail

The ***National Lambda Rail*** *(NLR)* is the next generation of high performance networking.



Condor Pool

- **Condor** is a software package that allows number crunching jobs to run on idle desktop PCs.
- OU IT is deploying a large Condor pool (730 desktop PCs, currently 265 operational) during 2007.
- When fully deployed, it'll provide a huge amount of additional computing power – 4 times as much as was available in all of OSCER in 2005.
- And, the cost is very very low.
- Also, we've been seeing empirically that Condor gets about 89% of each PC's time.

NSF CI-TEAM Project

In 2006, OSCER received a grant from the National Science Foundation's Cyberinfrastructure Training, Education, Advancement, and Mentoring for Our 21st Century Workforce (CI-TEAM) program.

Objectives:

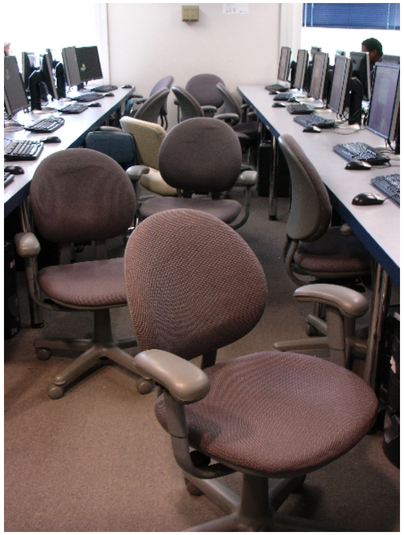
- Teach Cyberinfrastructure to EVERYBODY!
- Provide Condor resources to the national community
- Teach users to use Condor and sysadmins to deploy and administer it
- Teach bioinformatics students to use BLAST over Condor

OU NSF CI-TEAM Project

Cyberinfrastructure Education for Bioinformatics and Beyond

Objectives:

- **teach** students and faculty to use **FREE** Condor middleware, which steals computing time on idle desktop PCs;
- **teach** system administrators to deploy and maintain Condor on PCs;
- **teach** bioinformatics students to use BLAST on Condor;
- **provide** Condor Cyberinfrastructure to the national community (**FREE**).



OU will provide:

- **Condor pool** of 750 desktop PCs (already part of the **Open Science Grid**);
- ***Supercomputing in Plain English*** workshops via videoconferencing;
- Cyberinfrastructure ***rounds*** (consulting) via videoconferencing;
- **drop-in CDs** for installing full-featured Condor on a Windows PC (Cyberinfrastructure for **FREE**);
- **sysadmin consulting** for installing and maintaining Condor on desktop PCs.

OU's team includes: High School, Minority Serving, 2-year, 4-year, masters-granting; 11 of the 15 institutions are in 4 EPSCoR states (AR, KS, NE, OK).

OU NSF CI-TEAM Project

Participants at OU
(29 faculty/staff in 16 depts)

- Information Technology
  - OSCER: **Neeman (PI)**
- College of Arts & Sciences
  - Botany & Microbiology: Conway, Wren
  - Chemistry & Biochemistry: **Roe (Co-PI)**, Wheeler
  - Mathematics: White
  - Physics & Astronomy: Kao, **Severini (Co-PI)**, Skubic, Strauss
  - Zoology: Ray
- College of Earth & Energy
  - Sarkeys Energy Center: Chesnokov
- College of Engineering
  - Aerospace & Mechanical Engr: Striz
  - Chemical, Biological & Materials Engr: Papavassiliou
  - Civil Engr & Environmental Science: Vieux
  - Computer Science: Dhall, Fagg, Hougen, Lakshmivarahan, McGovern, Radhakrishnan
  - Electrical & Computer Engr: Cruz, Todd, Yeary, Yu
  - Industrial Engr: Trafalis
- OU Health Sciences Center, Oklahoma City
  - Biochemistry & Molecular Biology: Zlotnick
  - Radiological Sciences: **Wu (Co-PI)**
  - Surgery: Gusev

Participants at other institutions
(19 faculty/staff at 14 institutions)

- California State U Pomona (**masters-granting**, **minority serving**): Lee
- Contra Costa College (**2-year**, **minority serving**): Murphy
- Earlham College (**4-year**): Peck
- Emporia State U (**masters-granting**, **EPSCoR**): Pheatt, Ballester
- Kansas State U (**EPSCoR**): Andresen, Monaco
- Langston U (**masters-granting**, **minority serving, EPSCoR**): Snow
- Oklahoma Baptist U (**4-year**, **EPSCoR**): Chen, Jett, Jordan
- Oklahoma School of Science & Mathematics (**high school, EPSCoR**): Samadzadeh
- St. Gregory's U (**4-year**, **EPSCoR**): Meyer
- U Arkansas (**EPSCoR**): Apon
- U Central Oklahoma (**masters-granting**, **EPSCoR**): Lemley, Wilson
- U Kansas (**EPSCoR**): Bishop
- U Nebraska-Lincoln (**EPSCoR**): Swanson
- U Northern Iowa (**masters-granting**): Gray

Supercomputing

Supercomputing Issues

- The tyranny of the ***storage hierarchy***
- ***Parallelism***: doing many things at the same time
  - ***Instruction-level parallelism***: doing multiple operations at the same time within a single processor (e.g., add, multiply, load and store simultaneously)
  - ***Multiprocessing***: multiple CPUs working on different parts of a problem at the same time
    - ***Shared Memory*** Multithreading
    - ***Distributed*** Multiprocessing
- **High performance compilers**
- **Scientific Libraries**
- **Visualization**

A Quick Primer on Hardware

Henry’s Laptop

Dell Latitude D620[4]

- Pentium 4 Core Duo T2400
  1.83 GHz w/2 MB L2 Cache
- 2 GB (2048 MB)
  667 MHz DDR2 SDRAM
- 100 GB 7200 RPM SATA Hard Drive
- DVD±RW/CD-RW Drive (8x)
- 1 Gbps Ethernet Adapter
- 56 Kbps Phone Modem

Typical Computer Hardware

- Central Processing Unit
- Primary storage
- Secondary storage
- Input devices
- Output devices

Central Processing Unit

Also called ***CPU*** or ***processor***: the "brain"

Parts:

- ***Control Unit***: figures out what to do next -- e.g., whether to load data from memory, or to add two values together, or to store data into memory, or to decide which of two possible actions to perform (***branching***)
- ***Arithmetic/Logic Unit***: performs calculations – e.g., adding, multiplying, checking whether two values are equal
- ***Registers***: where data reside that are **being used right now**

Primary Storage

- ***Main Memory***
  - Also called ***RAM*** ("Random Access Memory")
  - Where data reside when they're **being used by a program that's currently running**
- ***Cache***
  - Small area of much faster memory
  - Where data reside when they're **about to be used** and/or **have been used recently**
- Primary storage is ***volatile***: values in primary storage disappear when the power is turned off.

Secondary Storage

- Where data and programs reside that are going to be used **<u>in the future</u>**
- Secondary storage is *<u>non-volatile</u>*: values **<u>don't</u>** disappear when power is turned off.
- Examples: hard disk, CD, DVD, magnetic tape, Zip, Jaz
- Many are ***<u>portable</u>***: can pop out the CD/DVD/tape/Zip/floppy and take it with you

Input/Output

- Input devices – e.g., keyboard, mouse, touchpad, joystick, scanner
- Output devices – e.g., monitor, printer, speakers

The Tyranny of the Storage Hierarchy

The Storage Hierarchy

[5]

Fast, expensive, few

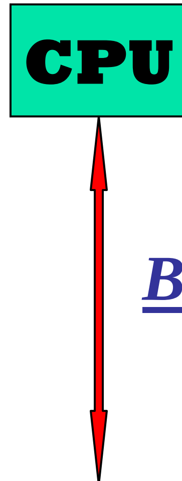
Slow, cheap, a lot

- Registers
- Cache memory
- Main memory (RAM)
- Hard disk
- Removable media (e.g., DVD)
- Internet

[6]

RAM is Slow

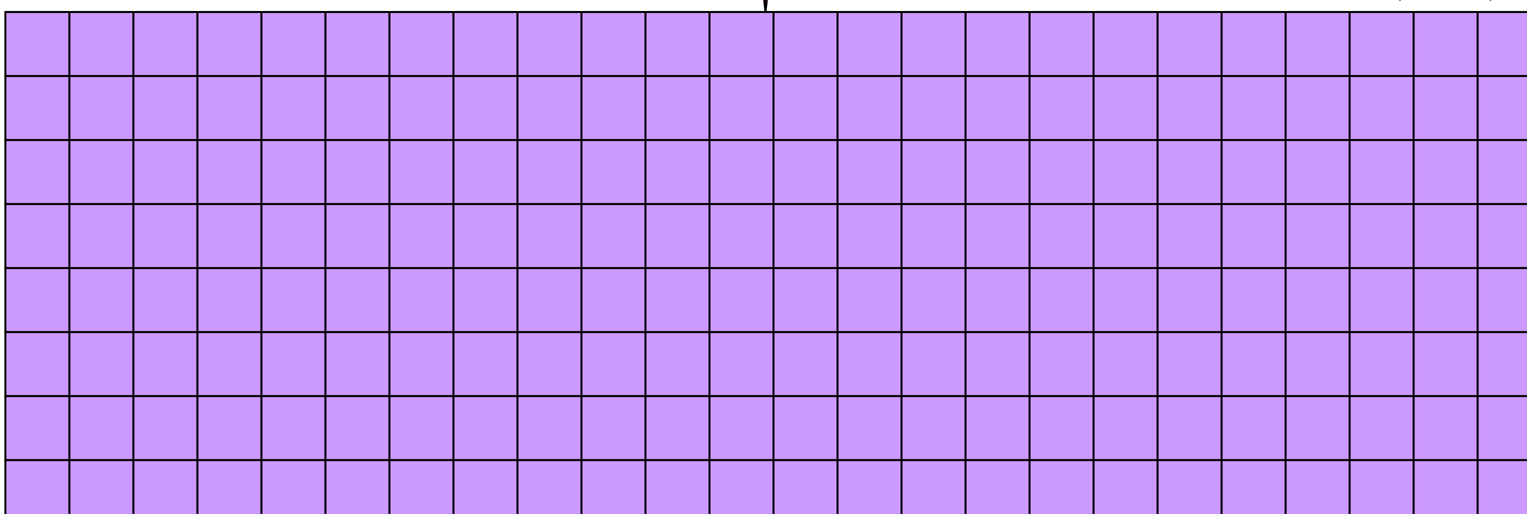
The speed of data transfer between Main Memory and the CPU is much slower than the speed of calculating, so the CPU spends most of its time waiting for data to come in or go out.



351 GB/sec[7]

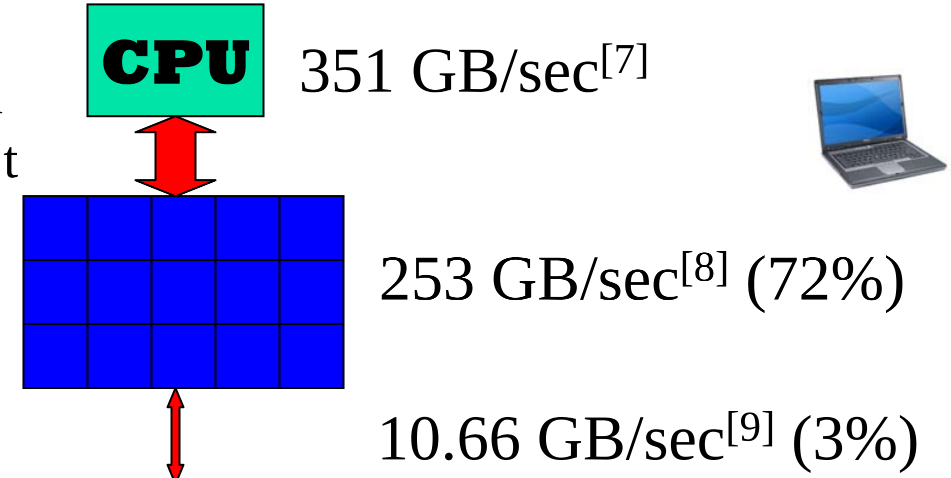
Bottleneck

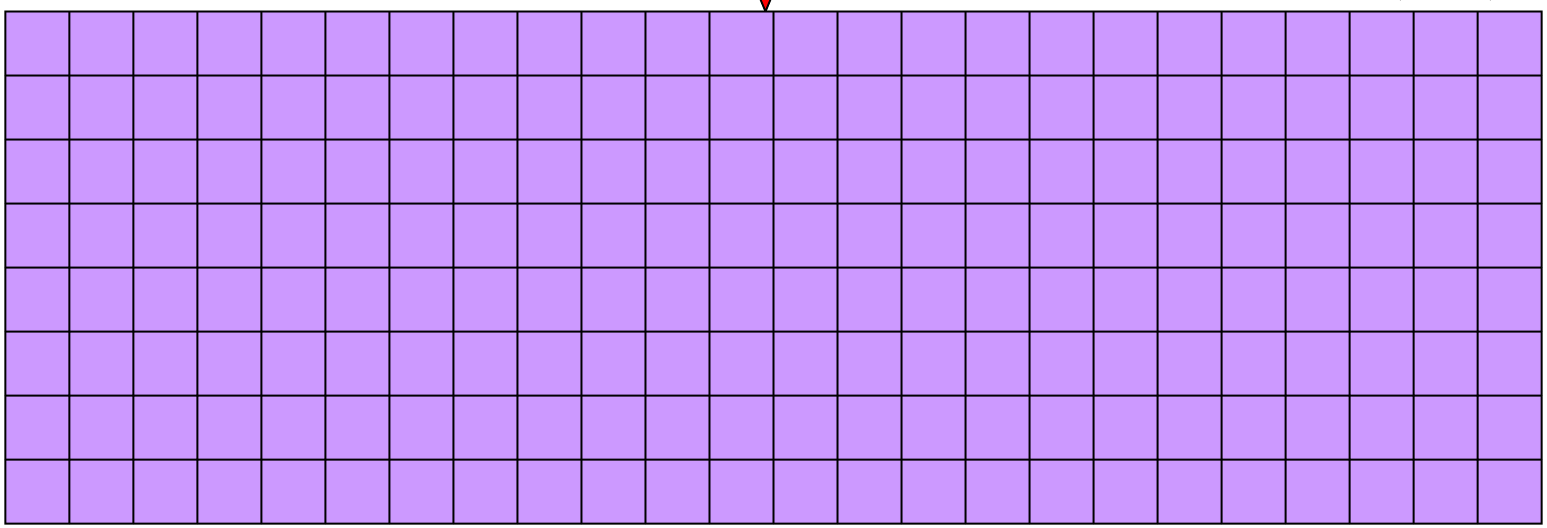
10.66 GB/sec[9] (3%)



Why Have Cache?

Cache is nearly the same speed as the CPU, so the CPU doesn't have to wait nearly as long for stuff that's already in cache: it can do more operations per second!





Henry’s Laptop, Again

Dell Latitude D620[4]

- Pentium 4 Core Duo T2400
  1.83 GHz w/2 MB L2 Cache
- 2 GB (2048 MB)
  667 MHz DDR2 SDRAM
- 100 GB 7200 RPM SATA Hard Drive
- DVD±RW/CD-RW Drive (8x)
- 1 Gbps Ethernet Adapter
- 56 Kbps Phone Modem

Storage Speed, Size, Cost

Henry's Laptop	Registers (Pentium 4 Core Duo 1.83 GHz)	Cache Memory (L2)	Main Memory (667 MHz DDR2 SDRAM)	Hard Drive (SATA 7200 RPM)	Ethernet (1000 Mbps)	DVD±RW (8x)	Phone Modem (56 Kbps)
Speed (MB/sec) [peak]	359,792[7] (14,640 MFLOP/s*)	259,072 [8]	10,928 [9]	100 [10]	125	10.8 [11]	0.007
Size (MB)	304 bytes** [12]	2	2048	100,000	unlimited	unlimited	unlimited
Cost ($/MB)	–	$17 [13]	$0.04 [13]	$0.0002 [13]	charged per month (typically)	$0.00004 [13]	charged per month (typically)

* MFLOP/s: millions of floating point operations per second

** 8 32-bit integer registers, 8 80-bit floating point registers, 8 64-bit MMX integer registers, 8 128-bit floating point XMM registers

Storage Use Strategies

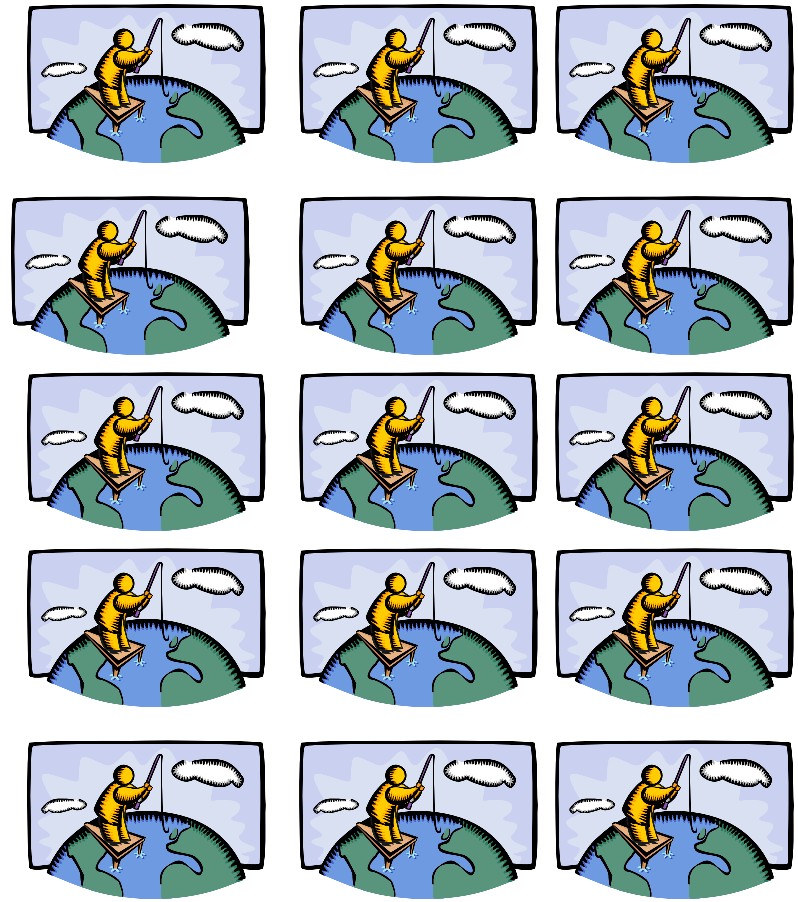
- ***Register reuse***: do a lot of work on the same data before working on new data.
- ***Cache reuse***: the program is much more efficient if all of the data and instructions fit in cache; if not, try to use what's in cache a lot before using anything that isn't in cache.
- ***Data locality***: try to access data that are near each other in memory before data that are far.
- **I/O efficiency**: do a bunch of I/O all at once rather than a little bit at a time; don't mix calculations and I/O.

Parallelism

Parallelism

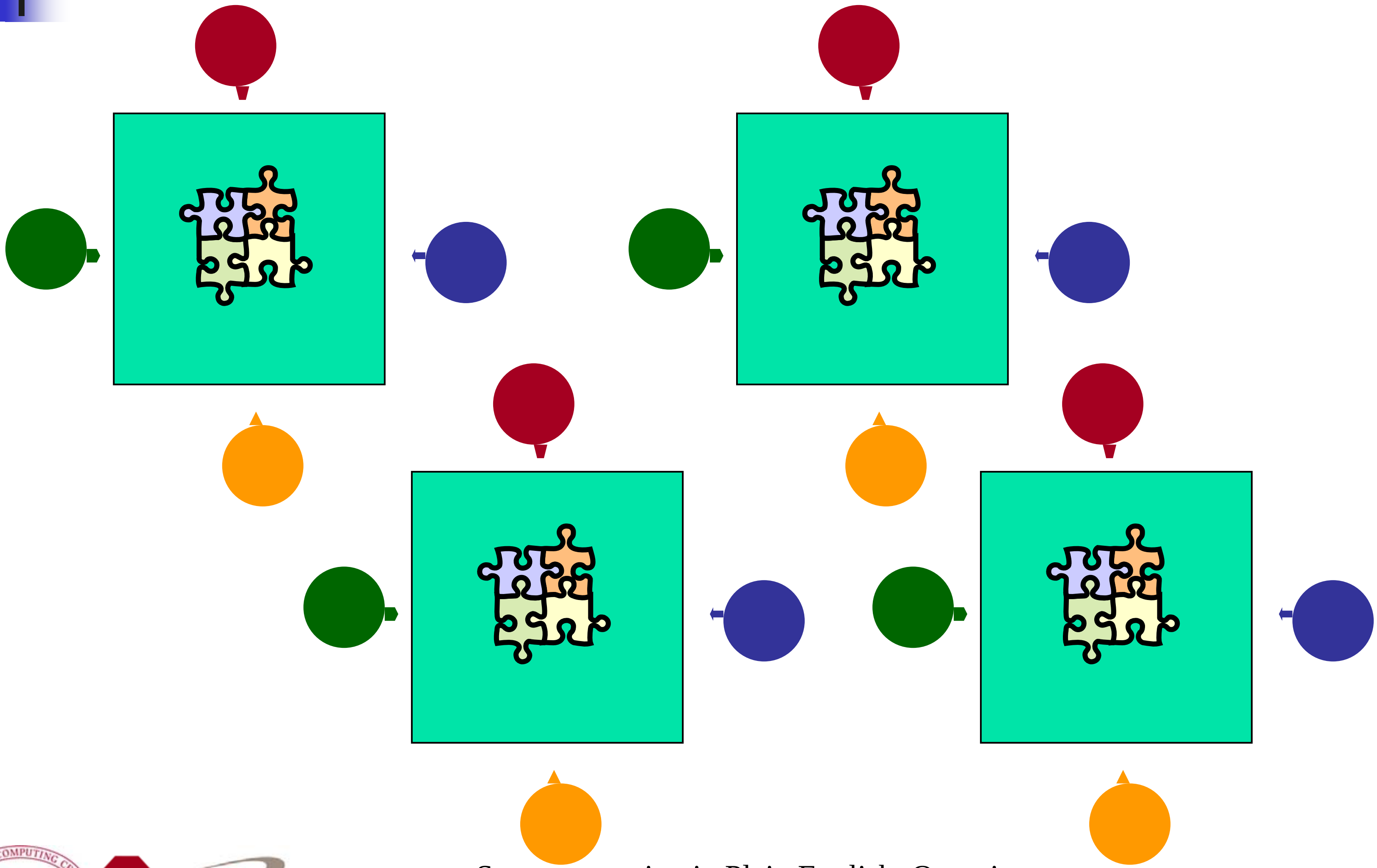
Parallelism means doing multiple things at the same time: you can get more work done in the same time.

Less fish …

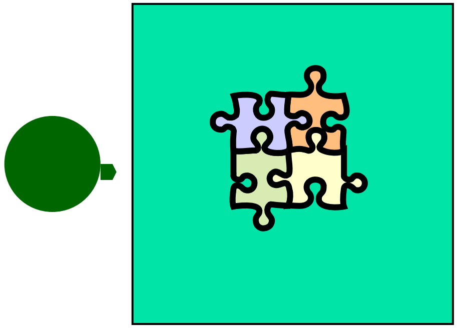


More fish!

The Jigsaw Puzzle Analogy



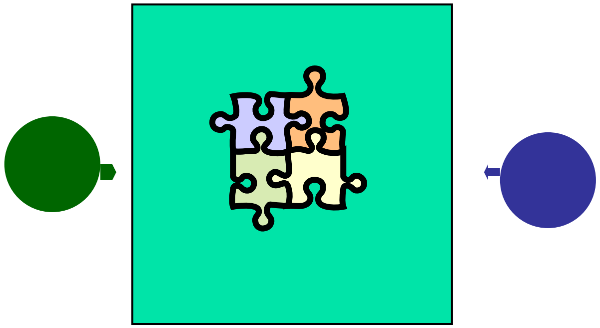
Serial Computing



Suppose you want to do a jigsaw puzzle that has, say, a thousand pieces.

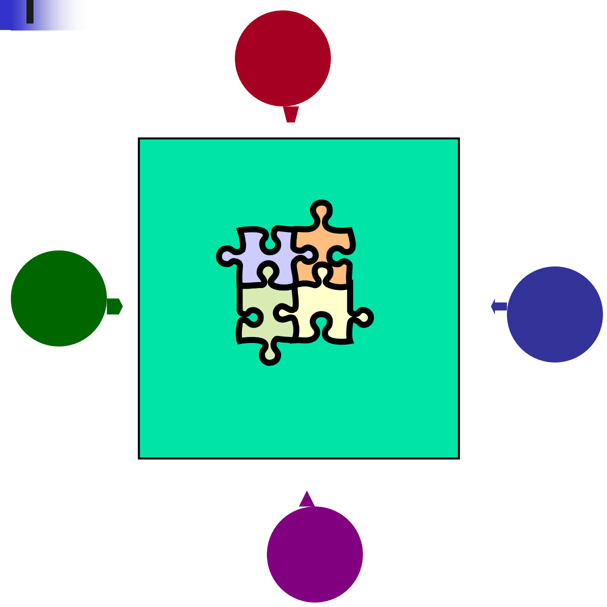
We can imagine that it'll take you a certain amount of time. Let's say that you can put the puzzle together in an hour.

Shared Memory Parallelism



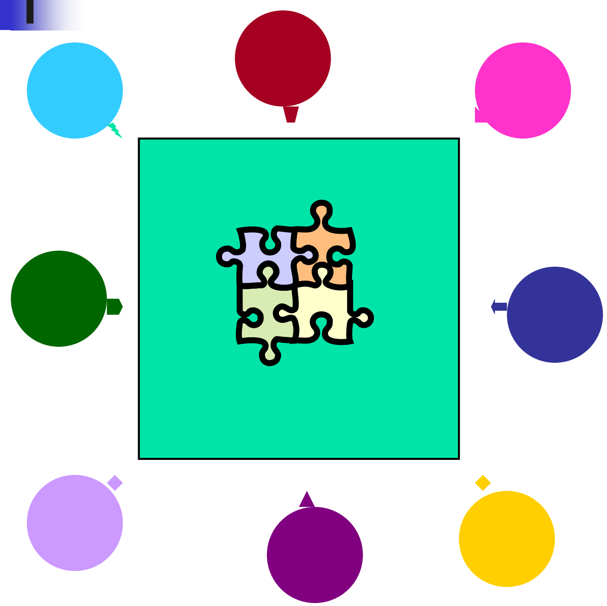
If Horst sits across the table from you, then he can work on his half of the puzzle and you can work on yours. Once in a while, you'll both reach into the pile of pieces at the same time (you'll ***contend*** for the same resource), which will cause a little bit of slowdown. And from time to time you'll have to work together (***communicate***) at the interface between his half and yours. The speedup will be nearly 2-to-1: y'all might take 35 minutes instead of 30.

The More the Merrier?



Now let's put Bruce and Dee on the other two sides of the table. Each of you can work on a part of the puzzle, but there'll be a lot more contention for the shared resource (the pile of puzzle pieces) and a lot more communication at the interfaces. So y'all will get noticeably less than a 4-to-1 speedup, but you'll still have an improvement, maybe something like 3-to-1: the four of you can get it done in 20 minutes instead of an hour.

Diminishing Returns



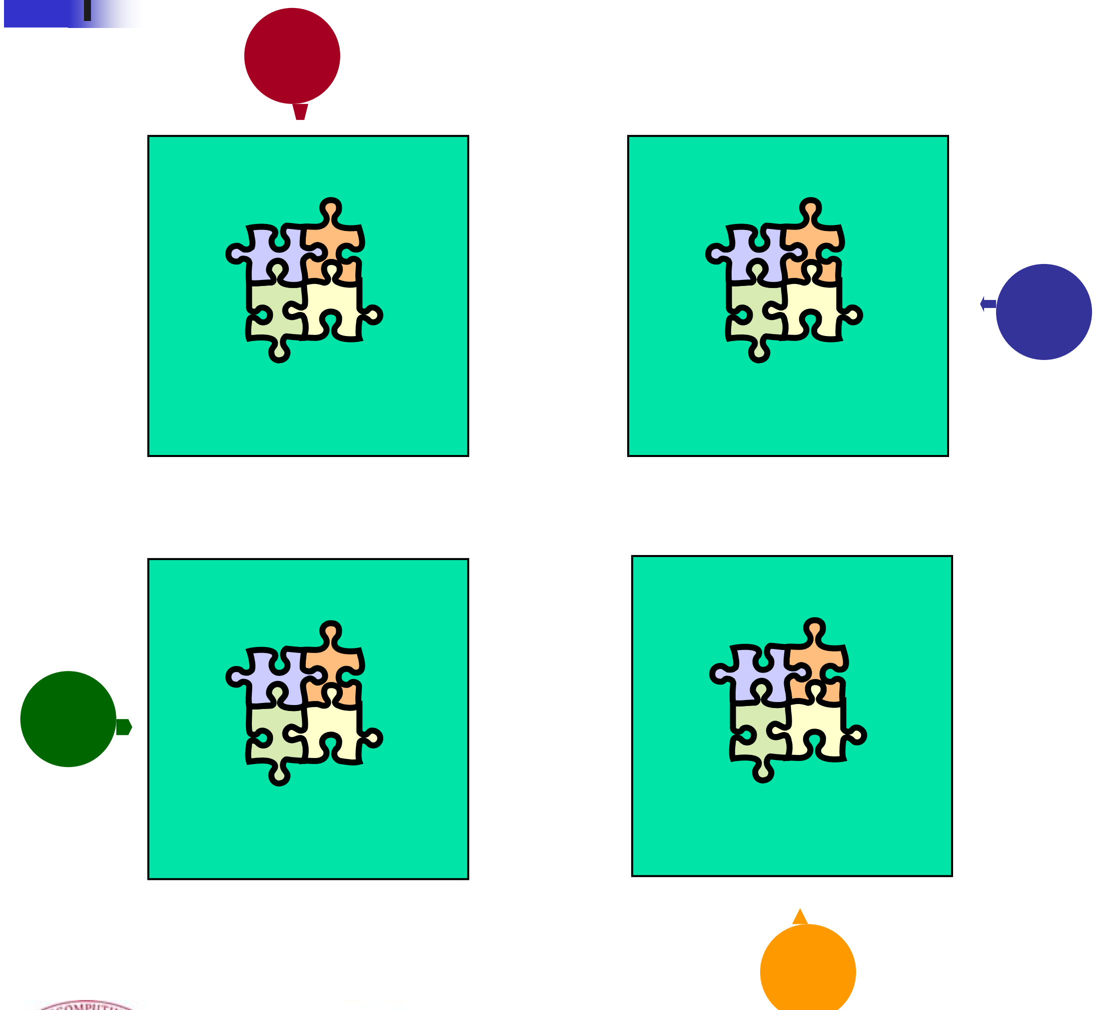
If we now put Rebecca and Jen and Alisa and Darlene on the corners of the table, there’s going to be a whole lot of contention for the shared resource, and a lot of communication at the many interfaces. So the speedup y’all get will be much less than we’d like; you’ll be lucky to get 5-to-1.

So we can see that adding more and more workers onto a shared resource is eventually going to have a diminishing return.

Distributed Parallelism

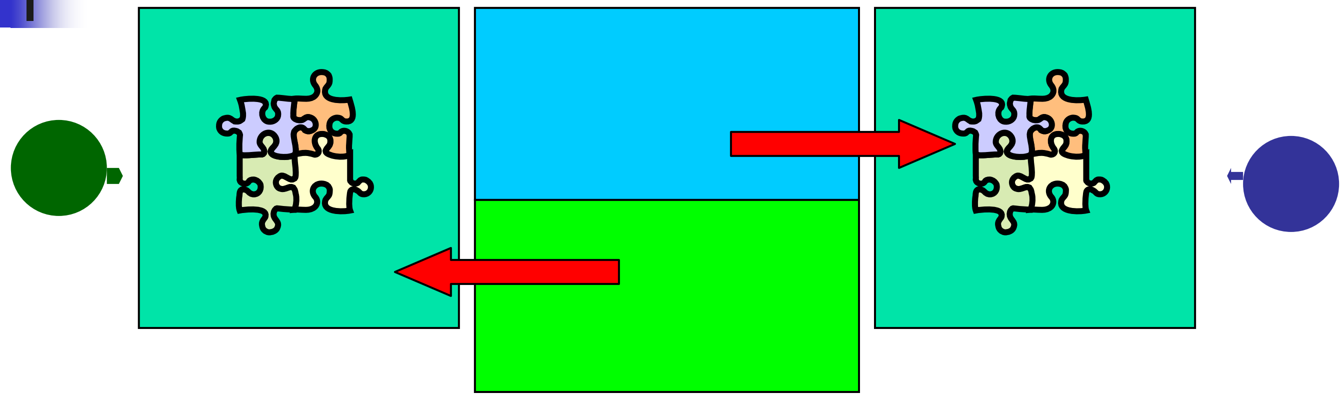
Now let's try something a little different. Let's set up two tables, and let's put you at one of them and Horst at the other. Let's put half of the puzzle pieces on your table and the other half of the pieces on Horst's. Now y'all can work completely independently, without any contention for a shared resource. **BUT**, the cost of communicating is **MUCH** higher (you have to scootch your tables together), and you need the ability to split up (***decompose***) the puzzle pieces reasonably evenly, which may be tricky to do for some puzzles.

More Distributed Processors



It’s a lot easier to add more processors in distributed parallelism. But, you always have to be aware of the need to decompose the problem and to communicate between the processors. Also, as you add more processors, it may be harder to ***load balance*** the amount of work that each processor gets.

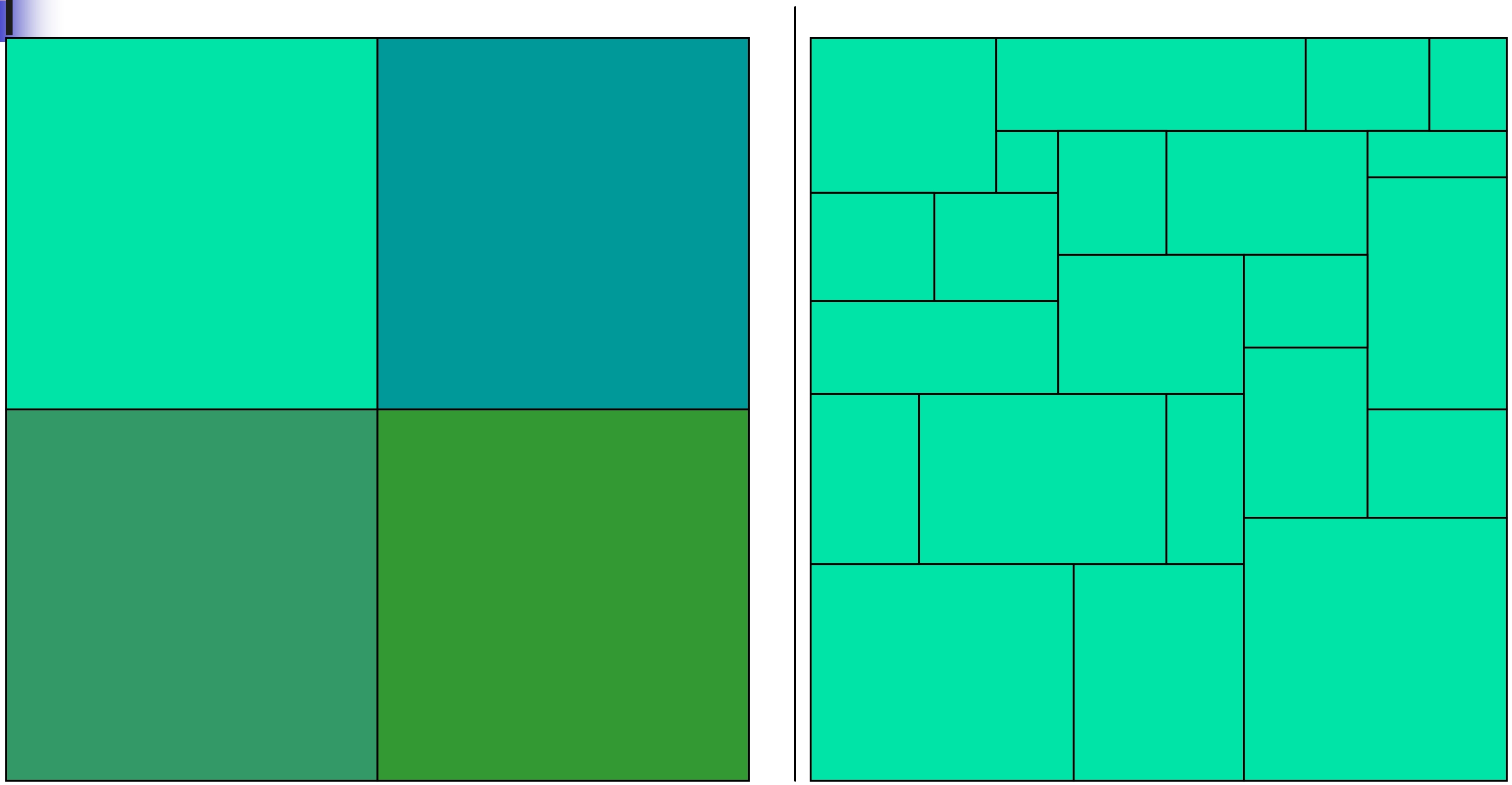
Load Balancing



Load balancing means giving everyone roughly the same amount of work to do.

For example, if the jigsaw puzzle is half grass and half sky, then you can do the grass and Julie can do the sky, and then y'all only have to communicate at the horizon – and the amount of work that each of you does on your own is roughly equal. So you'll get pretty good speedup.

Load Balancing



Load balancing can be easy, if the problem splits up into chunks of roughly equal size, with one chunk per processor. Or load balancing can be very hard.

Moore's Law

Moore's Law

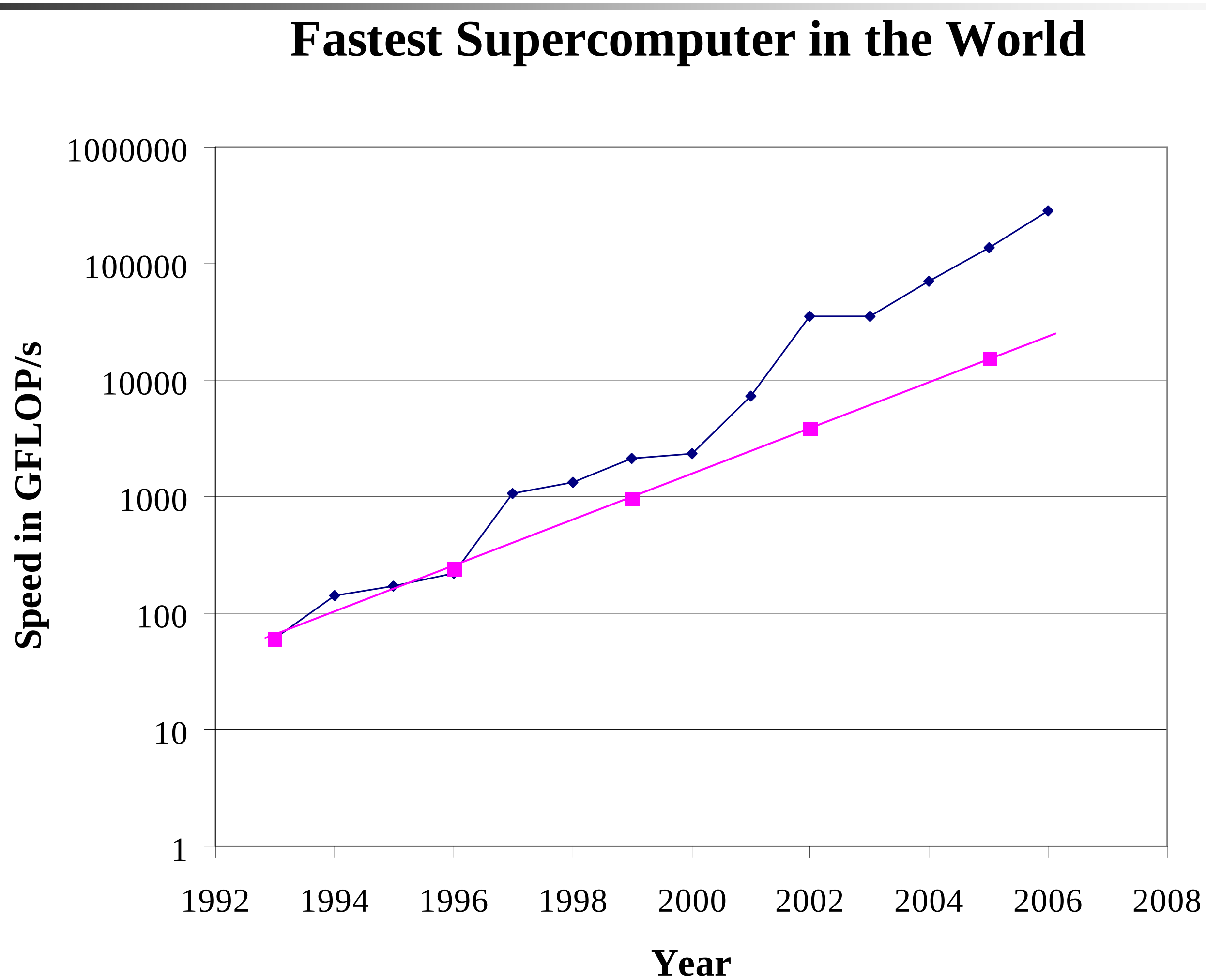
In 1965, Gordon Moore was an engineer at Fairchild Semiconductor.

He noticed that the number of transistors that could be squeezed onto a chip was doubling about every 18 months.

It turns out that computer speed is roughly proportional to the number of transistors per unit area.

Moore wrote a paper about this concept, which became known as ***"Moore's Law."***

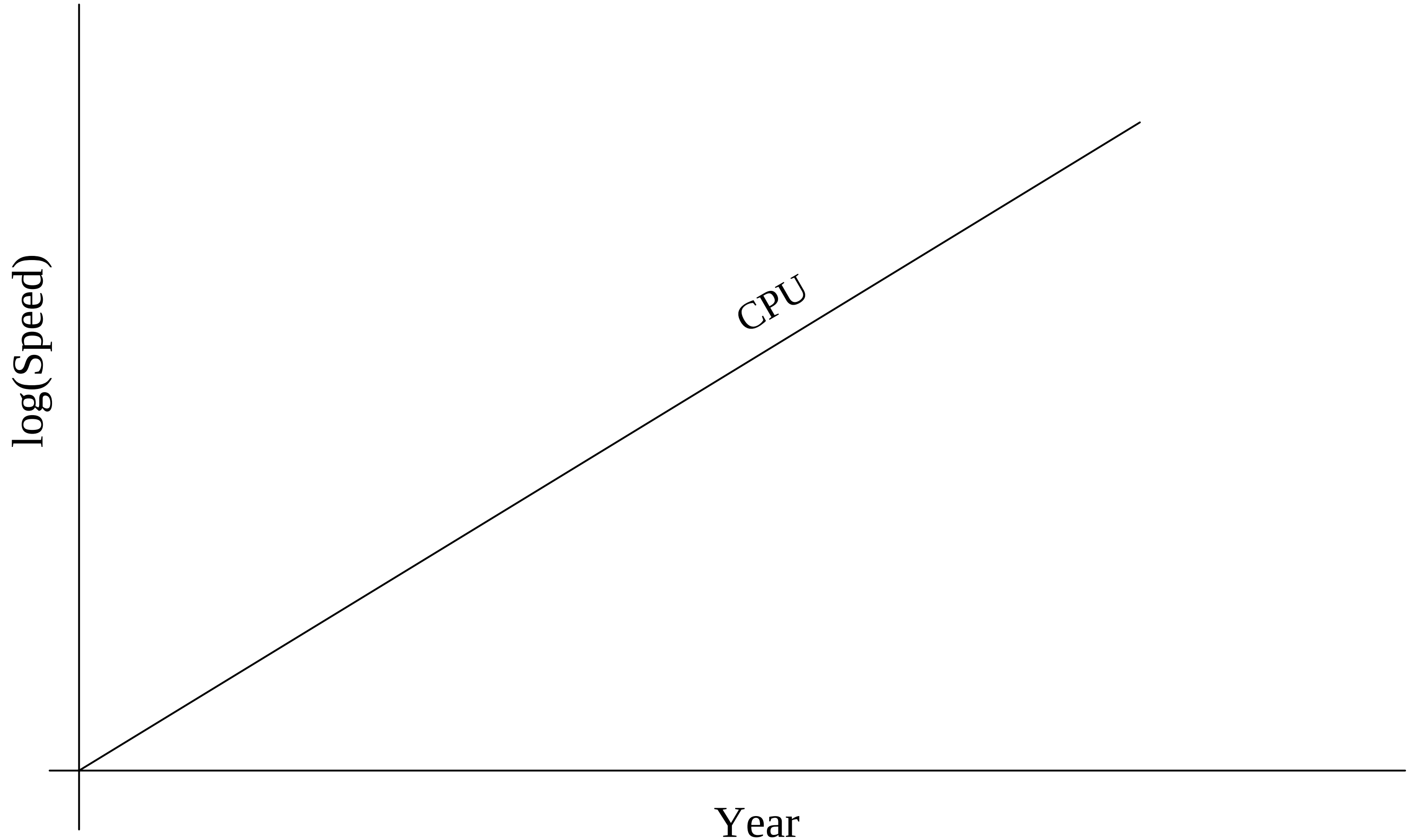
Fastest Supercomputer vs. Moore



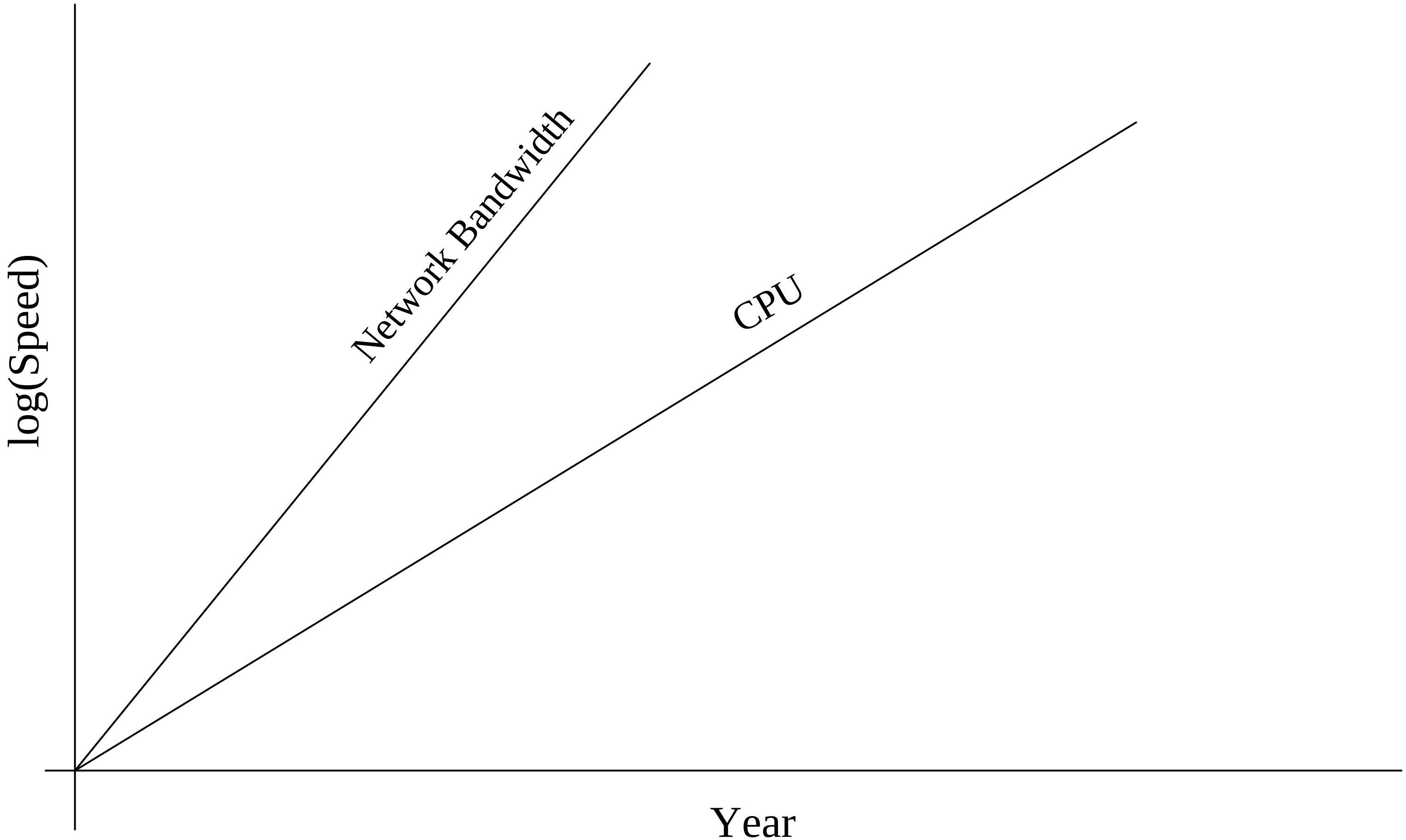
GFLOPs:

billions of calculations per second

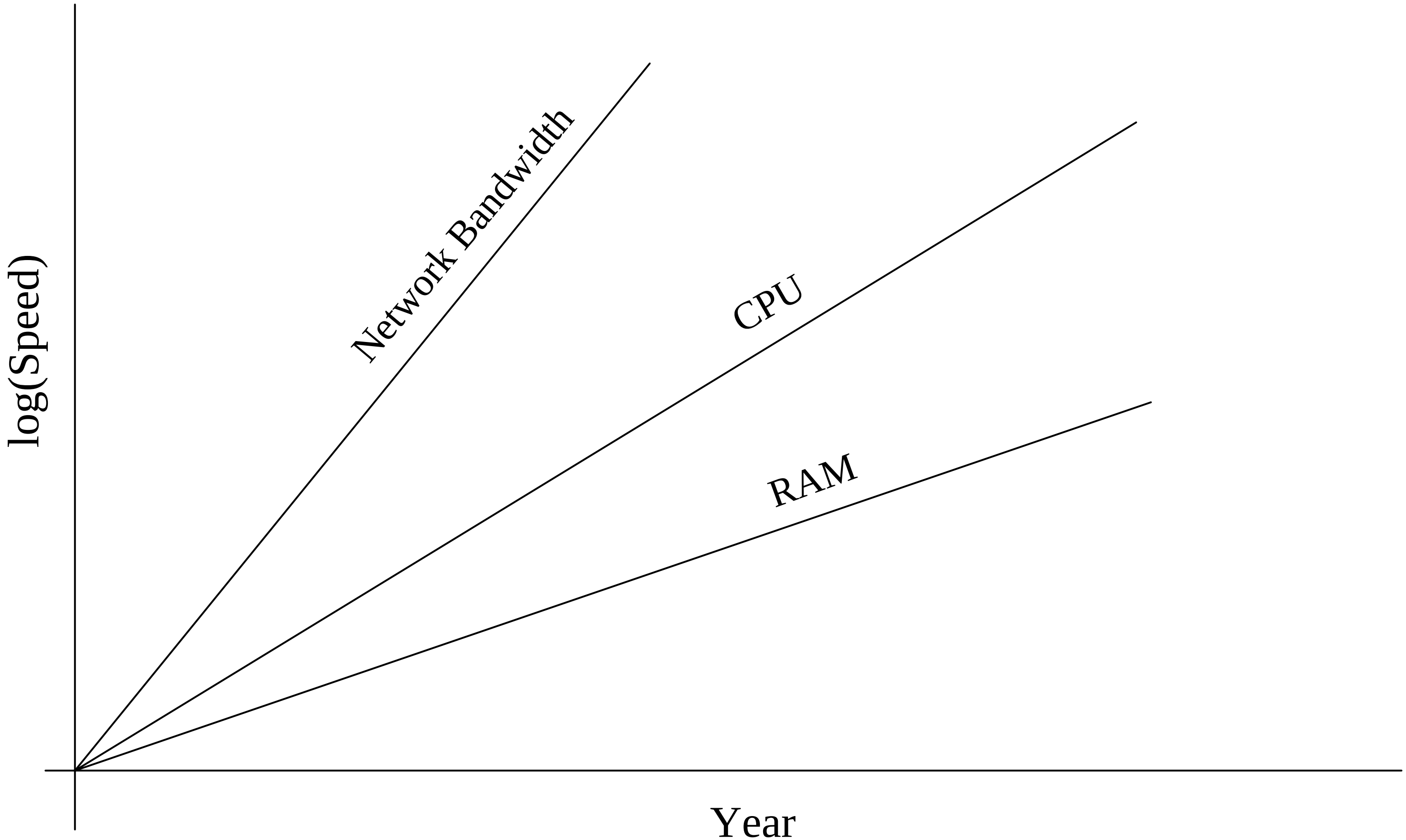
Moore’s Law in Practice



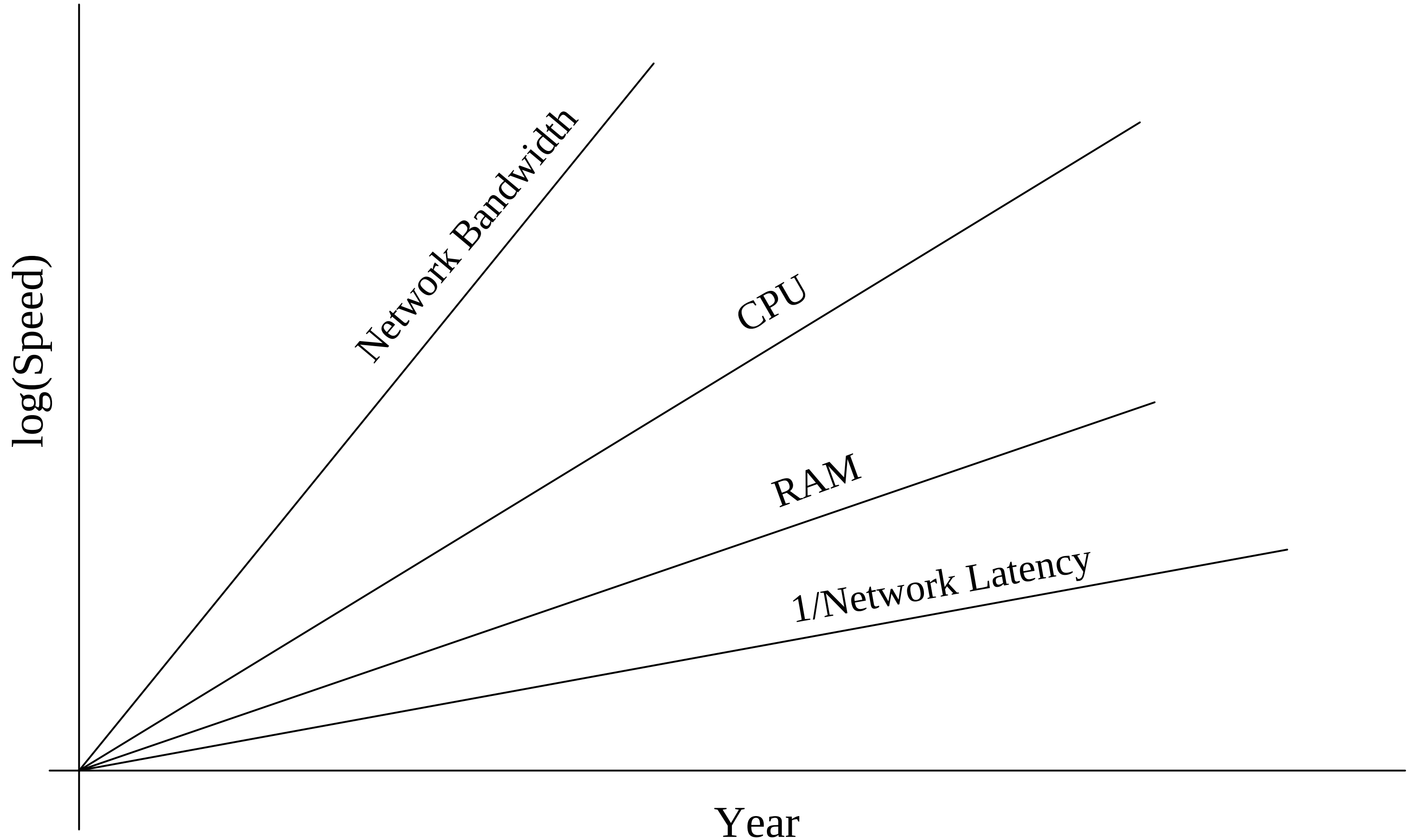
Moore's Law in Practice



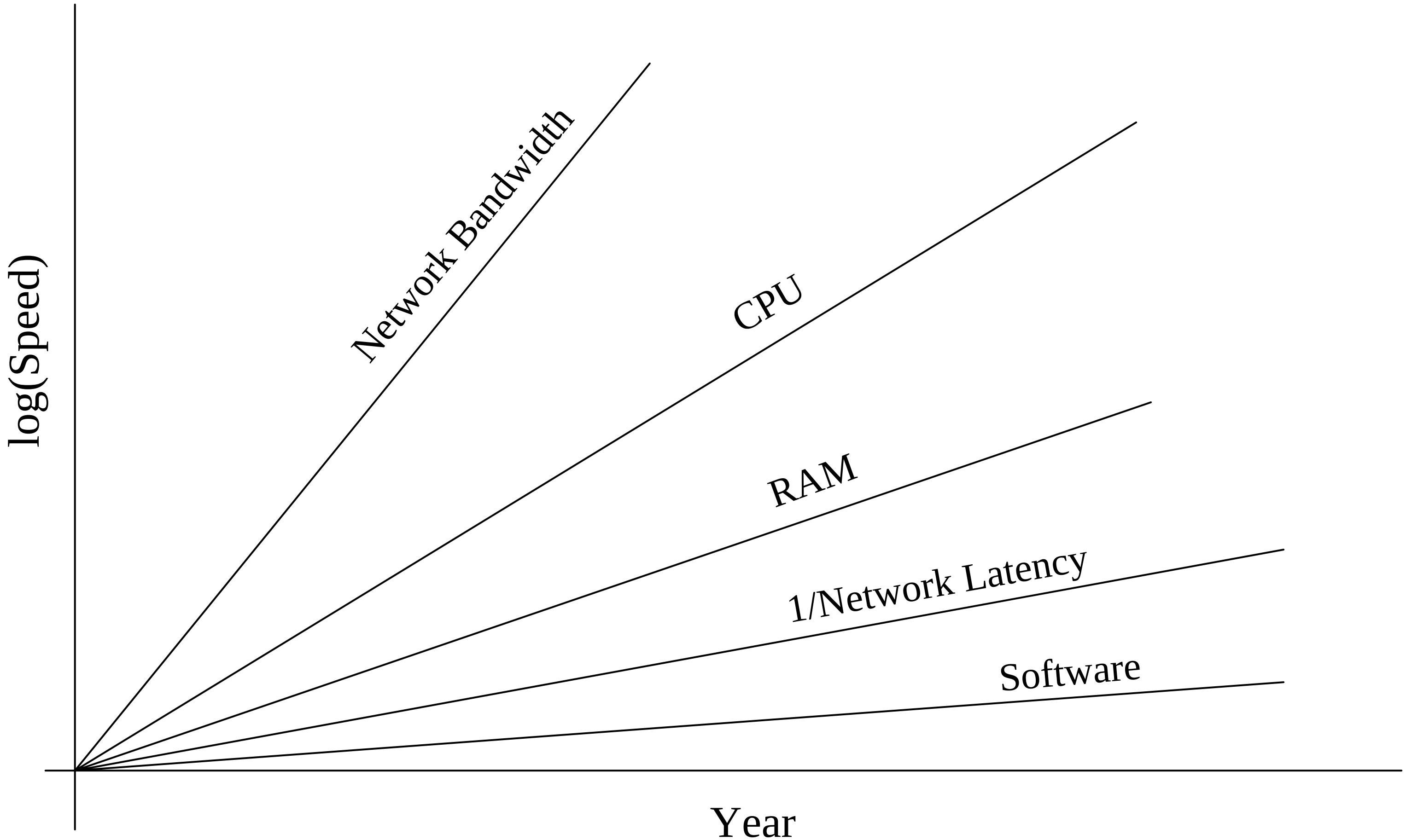
Moore’s Law in Practice



Moore’s Law in Practice



Moore's Law in Practice



Why Bother?

Why Bother with HPC at All?

It's clear that making effective use of HPC takes quite a bit of effort, both learning how and developing software.

That seems like a lot of trouble to go to just to get your code to run faster.

It's nice to have a code that used to take a day run in an hour. But if you can afford to wait a day, what's the point of HPC?

Why go to all that trouble just to get your code to run faster?

Why HPC is Worth the Bother

- What HPC gives you that you won't get elsewhere is the ability to do **bigger, better, more exciting science**. If your code can run faster, that means that you can tackle much bigger problems in the same amount of time that you used to need for smaller problems.
- HPC is important not only for its own sake, but also because what happens in HPC today will be on your desktop in about 15 years: it puts you **ahead of the curve**.

The Future is Now

Historically, this has always been true:

Whatever happens in supercomputing today will be on your desktop in 10 – 15 years.

So, if you have experience with supercomputing, you’ll be ahead of the curve when things get to the desktop.

To Learn More Supercomputing

http://www.oscer.ou.edu/education.php

http://symposium2007.oscer.ou.edu/

Thanks for your attention!

Questions?

References

[1] Image by Greg Bryan, MIT: http://zeus.ncsa.uiuc.edu:8080/chdm_script.html
[2] "Update on the Collaborative Radar Acquisition Field Test (CRAFT): Planning for the Next Steps." Presented to NWS Headquarters August 30 2001.
[3] See http://scarecrow.caps.ou.edu/~hneeman/hamr.html for details.
[4] http://www.dell.com/
[5] http://www.f1photo.com/
[6] http://www.vw.com/newbeetle/
[7] Richard Gerber, The Software Optimization Cookbook: High-performance Recipes for the Intel Architecture. Intel Press, 2002, pp. 161-168.
[8] http://www.anandtech.com/showdoc.html?i=1460&p=2
[9] ftp://download.intel.com/design/Pentium4/papers/24943801.pdf
[10] http://www.seagate.com/cda/products/discsales/personal/family/0,1085,621,00.html
[11] http://www.samsung.com/Products/OpticalDiscDrive/SlimDrive/OpticalDiscDrive_SlimDrive_SN_S082D.asp?page=Specifications
[12] ftp://download.intel.com/design/Pentium4/manuals/24896606.pdf
[13] http://www.pricewatch.com/
[14] Steve Behling et al, *The POWER4 Processor Introduction and Tuning Guide*, IBM, 2001, p. 8.
[15] Kevin Dowd and Charles Severance, *High Performance Computing*, 2nd ed. O'Reilly, 1998, p. 16.
[16] http://emeagwali.biz/photos/stock/supercomputer/black-shirt/