

Parallel & Cluster Computing Supercomputing Overview

National Computational Science Institute
August 8-14 2004

Paul Gray, University of Northern Iowa
David Joiner, Shodor Education Foundation
Tom Murphy, Contra Costa College
Henry Neeman, University of Oklahoma
Charlie Peck, Earlham College



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.

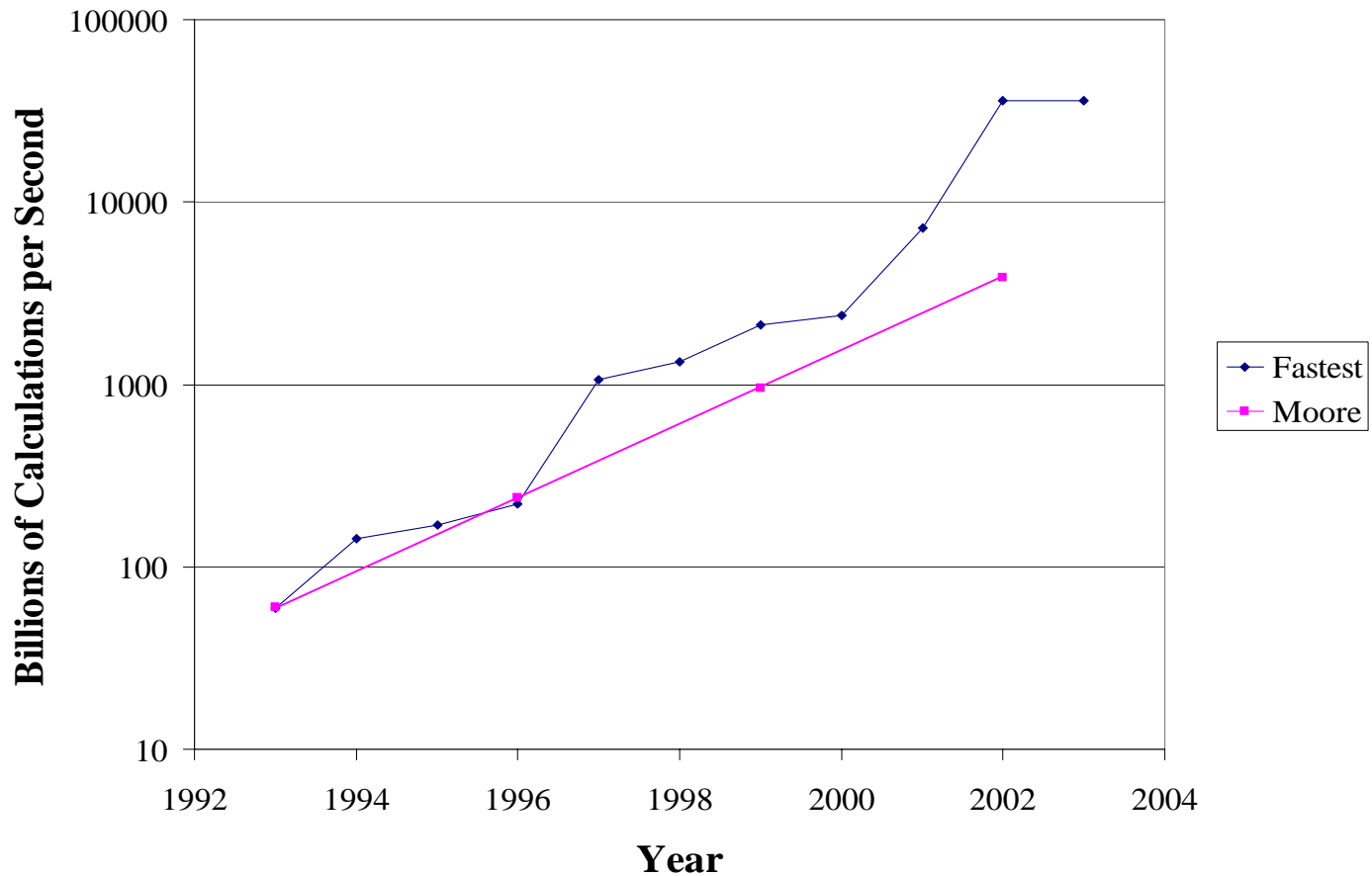
Rule of Thumb: a supercomputer is typically at least 100 times as powerful as a PC.

Jargon: supercomputing is also called High Performance Computing (HPC).



Fastest Supercomputer

Fastest Supercomputer in the World



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004



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**.

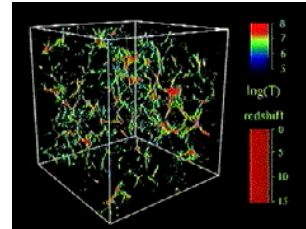


What is HPC Used For?

- **Simulation** of physical phenomena, such as

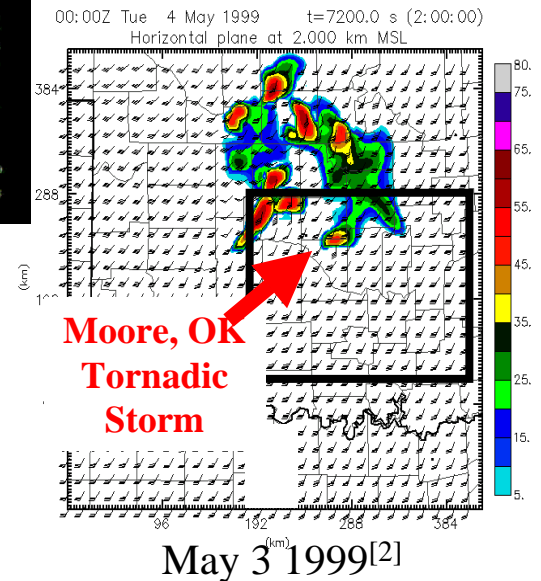
- Weather forecasting
- Galaxy formation
- Oil reservoir management

[1]



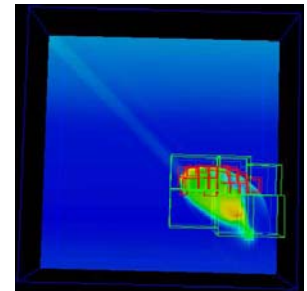
- **Data mining**: finding **needles** of information in a **haystack** of data, such as

- Gene sequencing
- Signal processing
- Detecting storms that could produce tornados



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

[3]



What is OSCER?

- Multidisciplinary center within OU's Department of Information Technology
- OSCER provides:
 - Supercomputing **education**
 - Supercomputing **expertise**
 - Supercomputing **resources**: hardware, storage, software
- OSCER is for:
 - Undergrad students
 - Grad students
 - Staff
 - Faculty



Who is OSCER? Academic Depts

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

Over 140 faculty & staff in 23 depts in Colleges of Arts & Sciences, Business, Engineering, Geosciences 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
- Department of Information Technology
- Fears Structural Engineering Laboratory
- Geosciences Computing Network
- Great Plains Network
- Human Technology Interaction Center
- Institute of Exploration & Development Geosciences
- Instructional Development Program
- Laboratory for Robotic Intelligence and Machine Learning
- Langston University Department of Mathematics
- Microarray Core Facility
- National Severe Storms Laboratory
- NOAA Storm Prediction Center
- Oklahoma EPSCoR



Expected Biggest Consumers

- Center for Analysis & Prediction of Storms:
daily real time weather forecasting 
- WeatherNews Inc: daily real time weather
forecasting 
- Oklahoma Center for High Energy Physics:
particle physics simulation and data analysis
using Grid computing
- Advanced Center for Genome Technology:
on-demand genomics



Who Are the Users?

Over 180 users so far:

- over 40 OU faculty
- over 40 OU staff
- over 80 students
- about a dozen off campus users
- ... more being added every month.

Comparison: National Center for Supercomputing Applications, with several tens of millions in annual funding and 18 years of history, has about **5400 users**.*

* Unique usernames on public-linux, public-sun, titan and cu



What Does OSCER Do? Teaching

Supercomputing in Plain English

An Introduction to High Performance Computing

Henry Neeman, Director
OU Supercomputing Center for Education & Research



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004



What Does OSCER Do? Rounds



From left: Civil Engr undergrad from Cornell; CS grad student; OSCER Director; Civil Engr grad student; Civil Engr prof; Civil Engr undergrad



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004





OSCER Hardware

- IBM Regatta p690 Symmetric Multiprocessor
- Aspen Systems Pentium4 Xeon Linux Cluster
- **Aspen Systems Itanium2 cluster: new arrival!**
- IBM FAStT500 FiberChannel-1 Disk Server
- Qualstar TLS-412300 Tape Library



Hardware: **IBM** p690 Regatta

32 POWER4 CPUs (1.1 GHz)

32 GB RAM

218 GB internal disk

OS: AIX 5.1

Peak speed: 140.8 GFLOP/s*

Programming model:

shared memory

multithreading (OpenMP)

(also supports MPI)

*GFLOP/s: billion floating point operations per second



sooner.oscer.ou.edu



NCSI Parallel & Cluster Computing Workshop @ OU

August 8-14 2004



Hardware: Pentium4 Xeon Cluster

270 Pentium4 XeonDP CPUs
270 GB RAM
8,700 GB disk
OS: Red Hat Linux Enterprise 3
Peak speed: 1.08 TFLOP/s*
Programming model:
distributed multiprocessing
(MPI)



*TFLOP/s: trillion floating point operations per second



boomer.oscer.ou.edu



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004



Hardware: Itanium2 Cluster

56 Itanium2 1.0 GHz CPUs

112 GB RAM

5,774 GB disk

OS: Red Hat Linux Enterprise 3

Peak speed: 224 GFLOP/s*

Programming model:
distributed multiprocessing
(MPI)



*GFLOP/s: billion floating point
operations per second

New arrival!



schooner.oscer.ou.edu



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004



National Lambda Rail at OU



For more information regarding NLR see <http://www.nlr.org>



What is National Lambda Rail?

The National Lambda Rail (NLR) is the next generation of high performance networking. You can think of it as “Internet3.”



For more information regarding NLR see <http://www.nlr.net> or contact info@nlr.net



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004





What is a Lambda Network?

A *lambda network* is a network that can use multiple wavelengths of light to carry multiple network signals on the same fiber.

Typically, in current lambda systems, each wavelength can carry up to 10 Gbps, and there can be as many as 40 lambda wavelengths per fiber.



What is NLR?

NLR is a consortium of academic and government organizations that are building a national lambda network.



For more information regarding NLR see <http://www.nlr.net> or contact info@nlr.net



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004



Who is NLR?

NLR participants include:

- [Corporation for Education Network Initiatives in California](#) (CENIC)
- [Pacific Northwest GigaPop](#) (PNWGP)
- [Pittsburgh Supercomputing Center](#)
- [Duke University](#), representing a coalition of North Carolina universities
- [Mid-Atlantic Terascale Partnership](#), MATP and the Virginia Tech Foundation
- [Cisco Systems](#)
- [Internet2](#)®
- [Florida LambdaRail, LLC](#)
- [Georgia Institute of Technology](#)
- [Committee on Institutional Cooperation](#) (CIC)
- [Cornell University](#)
- [Louisiana Board of Regents](#)
- [Oklahoma State Board of Regents](#)
- [Lonestar Education and Research Network](#) (LEARN)
- [University of New Mexico](#)
- [University Corporation for Atmospheric Research](#) (UCAR), representing a coalition of universities and government agencies from Colorado, Wyoming, and Utah





NLR in OK

In Oklahoma, the NLR will include a loop between Tulsa (the Point of Presence), Stillwater, OKC and Norman.

The main participants will be OU (Norman, HSC, Schusterman) and OSU.



How Will NLR Benefit OU?

- Cutting edge of high performance networking
- Enables large data transfers between OU and other NLR sites
- Specific projects that immediately need NLR:
 - Meteorology: LEAD, CASA (Droegemeier, Shapiro, Carr, Xue, Yeary, Yu)
 - Civil Engineering: NEES (Mish, Muralee, Zaman, Pulat, Pei, Ramseyer, Miller)
 - CS, ECE: high performance networking research (Atiquzzaman, Sluss, Verma, Tull, Hazem)
 - Physics: OCHEP (Skubic, Strauss, Gutierrez, Abbott, Kao, Milton)
- Long term: puts OU at the forefront of High End Computing research





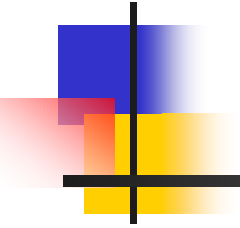
Supercomputing

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 C840^[4]



- Pentium 4 1.6 GHz w/512 KB L2 Cache
- 512 MB 400 MHz DDR SDRAM
- 30 GB Hard Drive
- Floppy Drive
- DVD/CD-RW Drive
- 10/100 Mbps Ethernet
- 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 **in the future**
- Secondary storage is non-volatile: values **don't** disappear when power is turned off.
- Examples: hard disk, CD, DVD, magnetic tape, Zip, Jaz
- Many are portable: 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

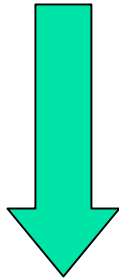


The Storage Hierarchy



[5]

Fast, expensive, few



Slow, cheap, a lot

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



[6]

RAM is Slow

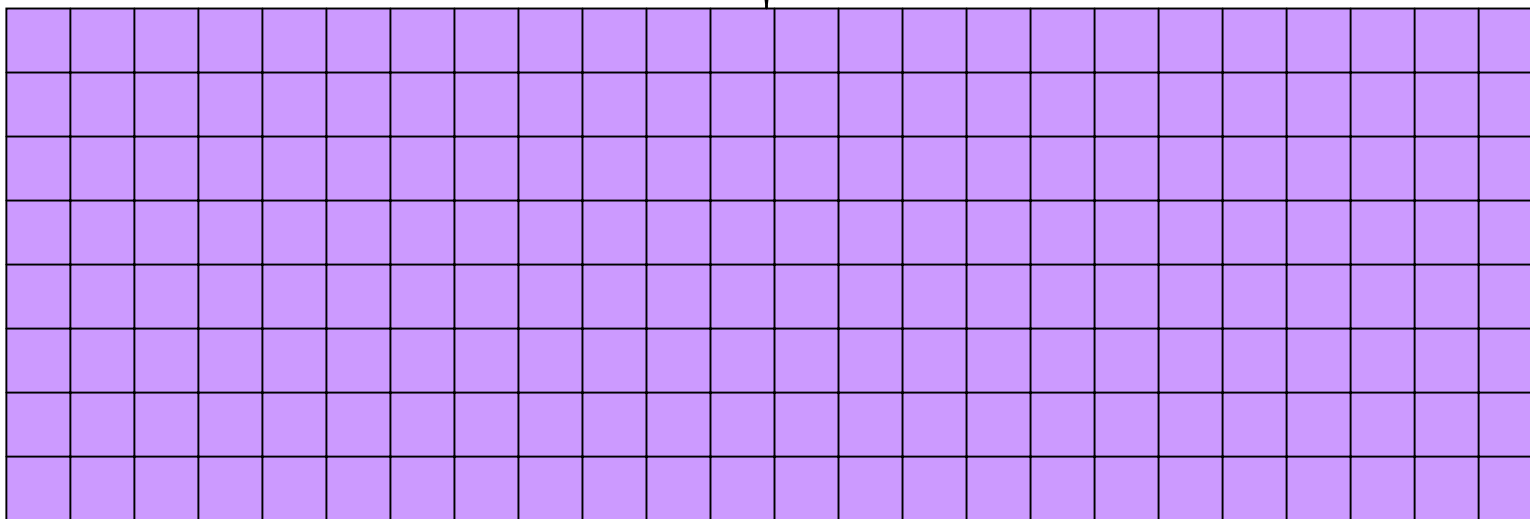
The speed of data transfer between Main Memory and the CPU is much slower than the speed of calculating, so the CPU spends most of its time waiting for data to come in or go out.

CPU

73.2 GB/sec^[7]

Bottleneck

3.2 GB/sec^[9]

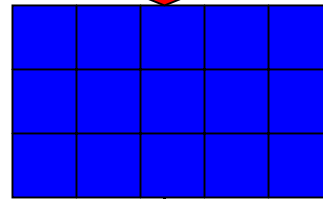


Why Have Cache?

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!

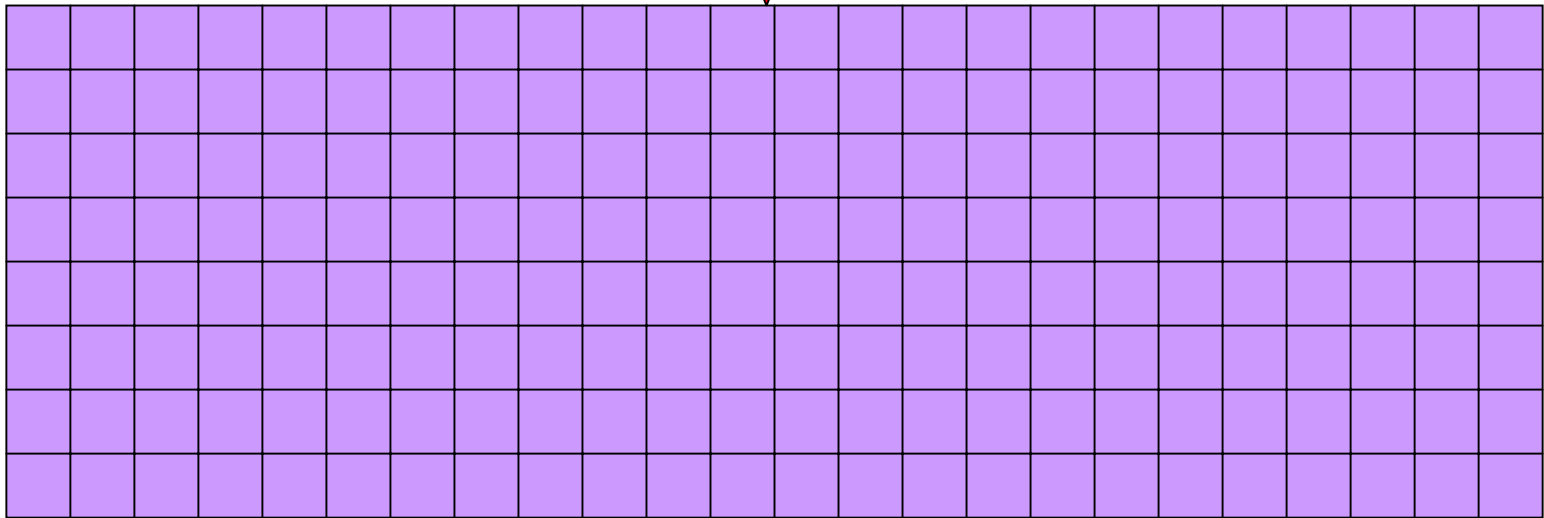
CPU

73.2 GB/sec^[7]



51.2 GB/sec^[8]

3.2 GB/sec^[9]



Henry's Laptop, Again

Dell Latitude C840^[4]



- Pentium 4 1.6 GHz w/512 KB L2 Cache
- 512 MB 400 MHz DDR SDRAM
- 30 GB Hard Drive
- Floppy Drive
- DVD/CD-RW Drive
- 10/100 Mbps Ethernet
- 56 Kbps Phone Modem

Storage Speed, Size, Cost

Henry's Laptop	Registers (Pentium 4 1.6 GHz)	Cache Memory (L2)	Main Memory (400 MHz DDR SDRAM)	Hard Drive	Ethernet (100 Mbps)	CD-RW	Phone Modem (56 Kbps)
Speed (MB/sec) [peak]	73,232 ^[7] (3200 MFLOP/s*)	52,428 ^[8]	3,277 ^[9]	100 ^[10]	12	4 ^[11]	0.007
Size (MB)	304 bytes** ^[12]	0.5	512	30,000	unlimited	unlimited	unlimited
Cost (\$/MB)	–	\$254 ^[13]	\$0.24 ^[13]	\$0.0005 ^[13]	charged per month (typically)	\$0.0015 ^[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.



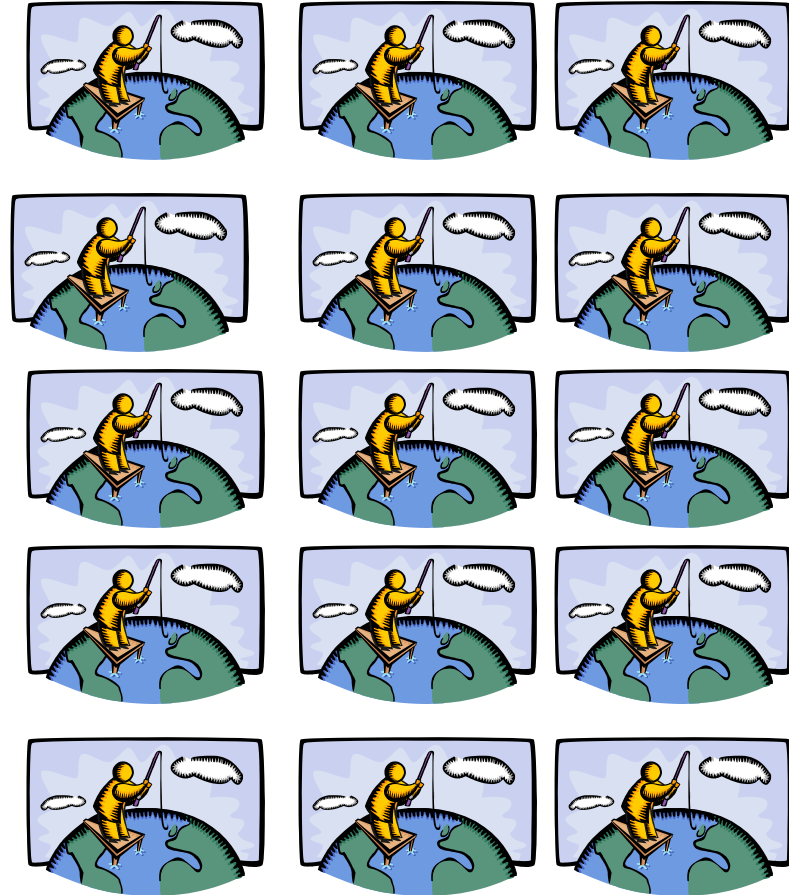


Parallelism

Parallelism

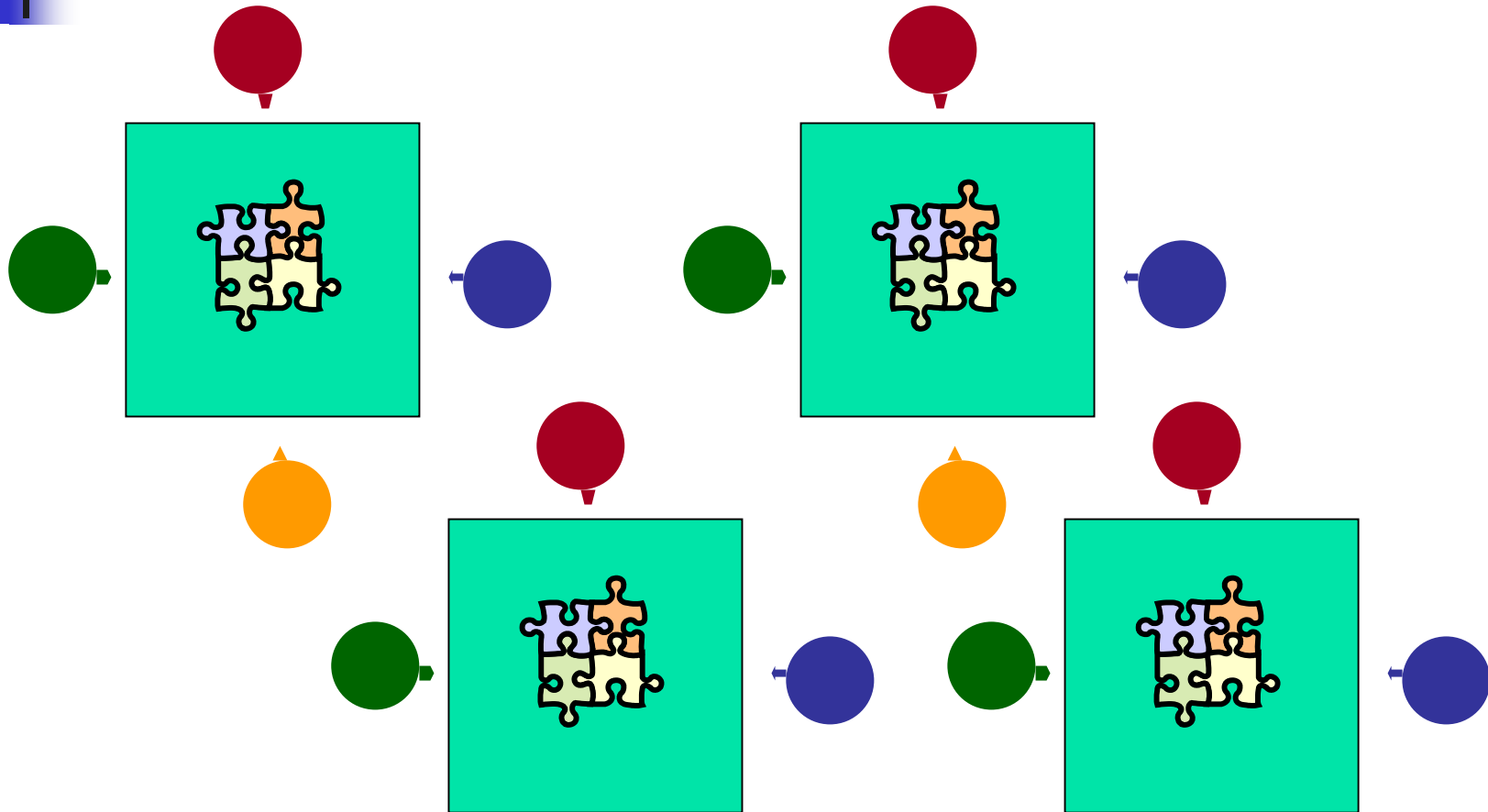
Parallelism means doing multiple things at the same time: you can get more work done in the same time.

Less fish ...



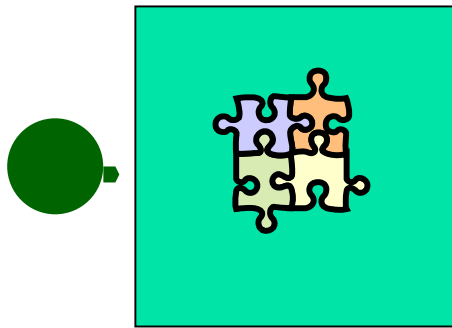
More fish!

The Jigsaw Puzzle Analogy



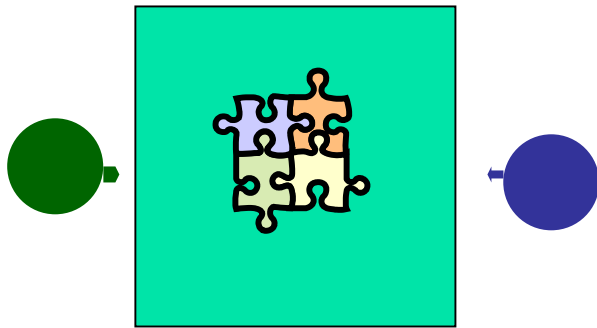
Serial Computing

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



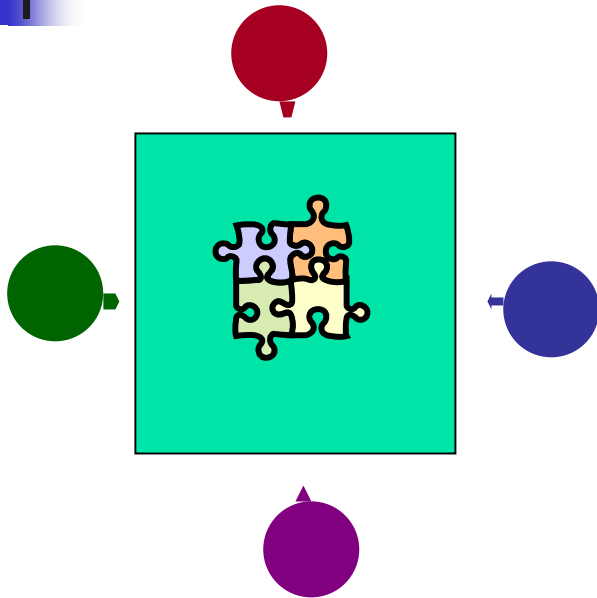
We can imagine that it'll take you a certain amount of time. Let's say that you can put the puzzle together in an hour.

Shared Memory Parallelism



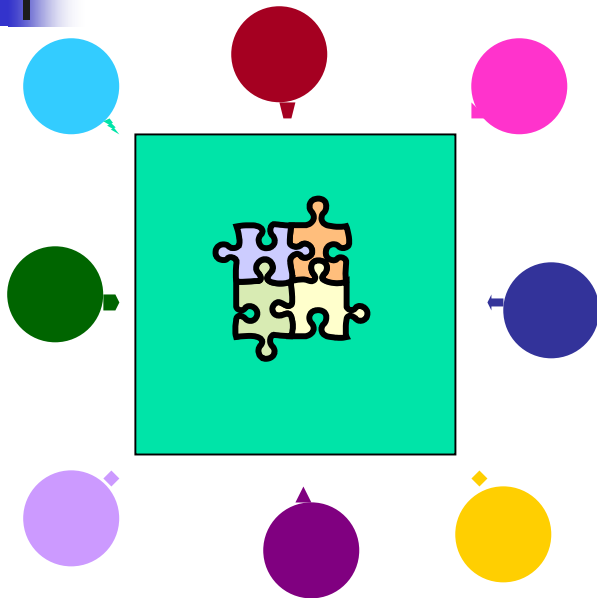
If Julie sits across the table from you, then she can work on her half of the puzzle and you can work on yours. Once in a while, you'll both reach into the pile of pieces at the same time (you'll contend for the same resource), which will cause a little bit of slowdown. And from time to time you'll have to work together (communicate) at the interface between her 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 Lloyd and Jerry 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 Dave and Paul and Tom and Charlie 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.

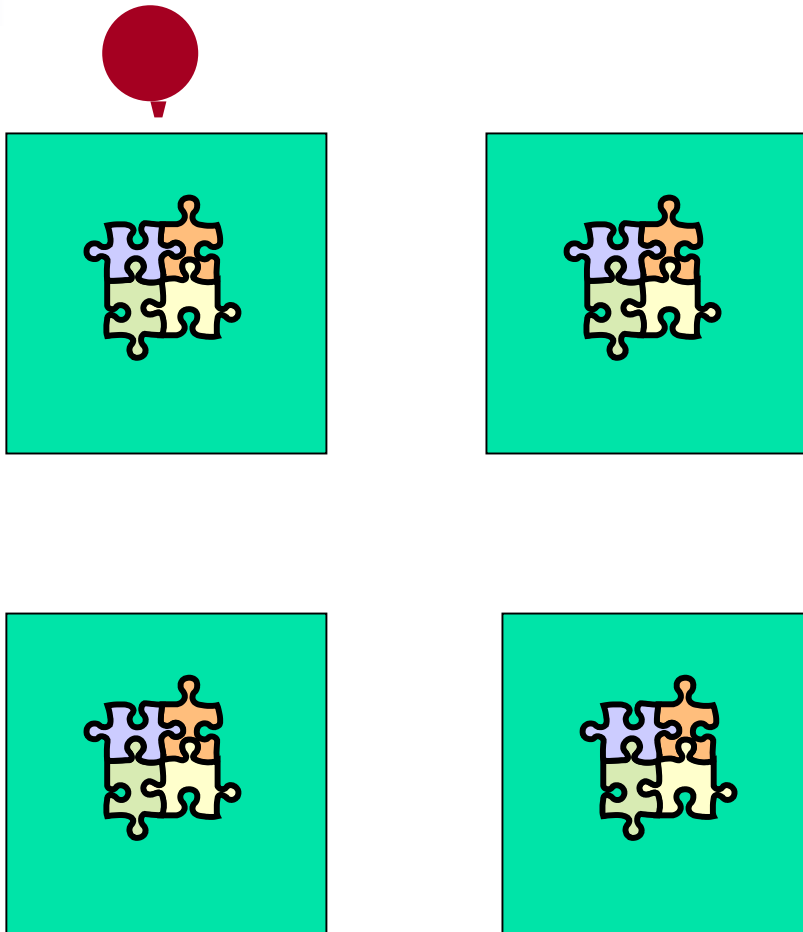
So we can see that adding more and more workers onto a shared resource is eventually going to have a diminishing return.

Distributed Parallelism



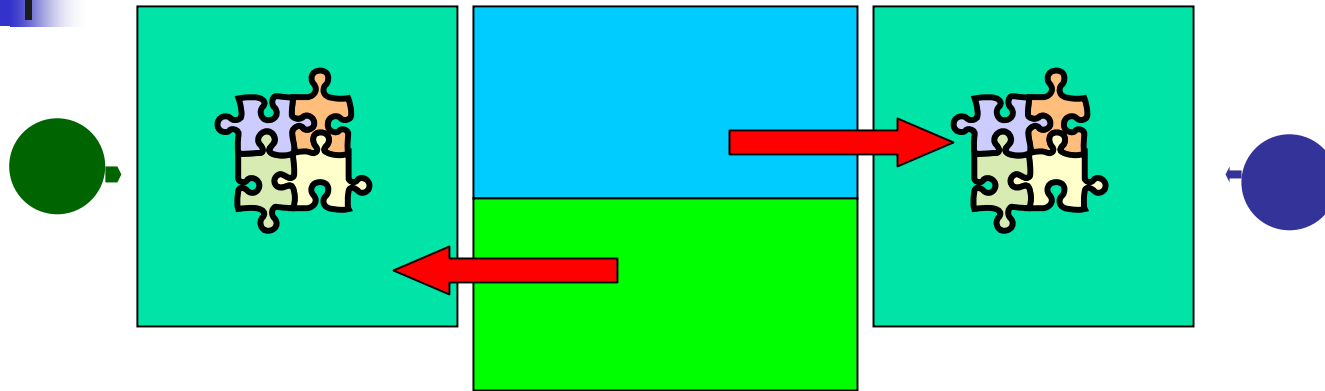
Now let's try something a little different. Let's set up two tables, and let's put you at one of them and Julie at the other. Let's put half of the puzzle pieces on your table and the other half of the pieces on Julie's. Now y'all can work completely independently, without any contention for a shared 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.

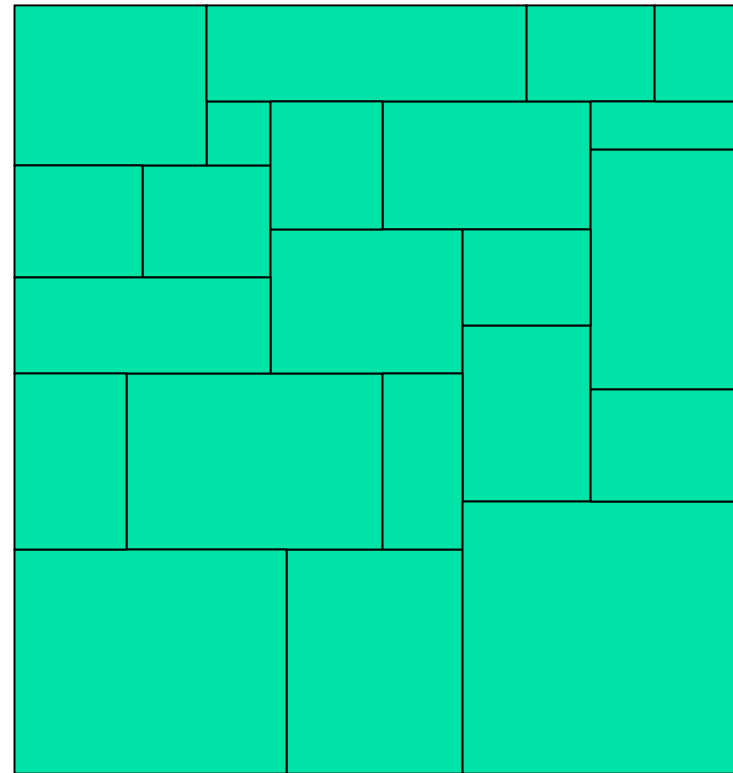
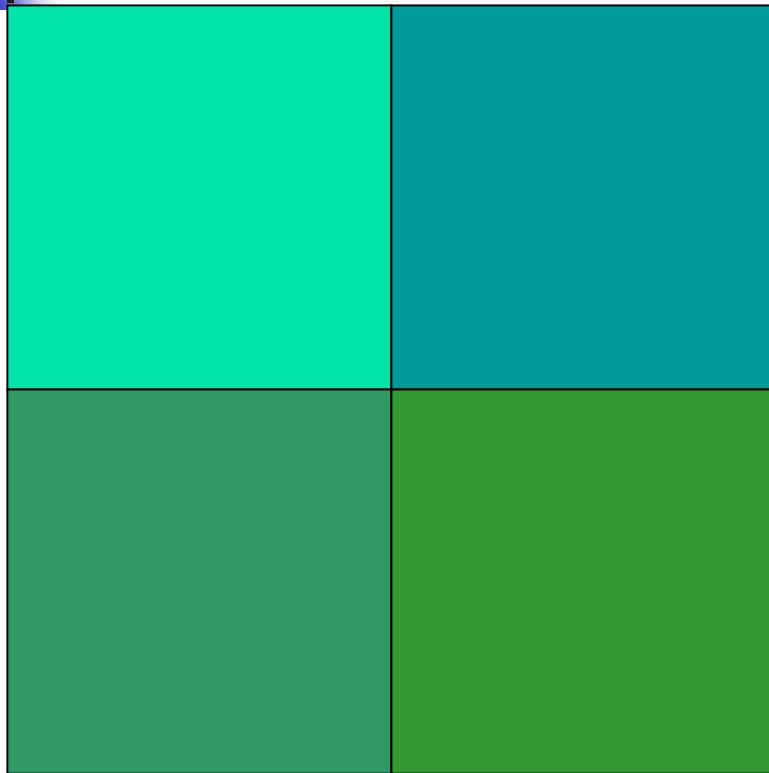
Load Balancing



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.

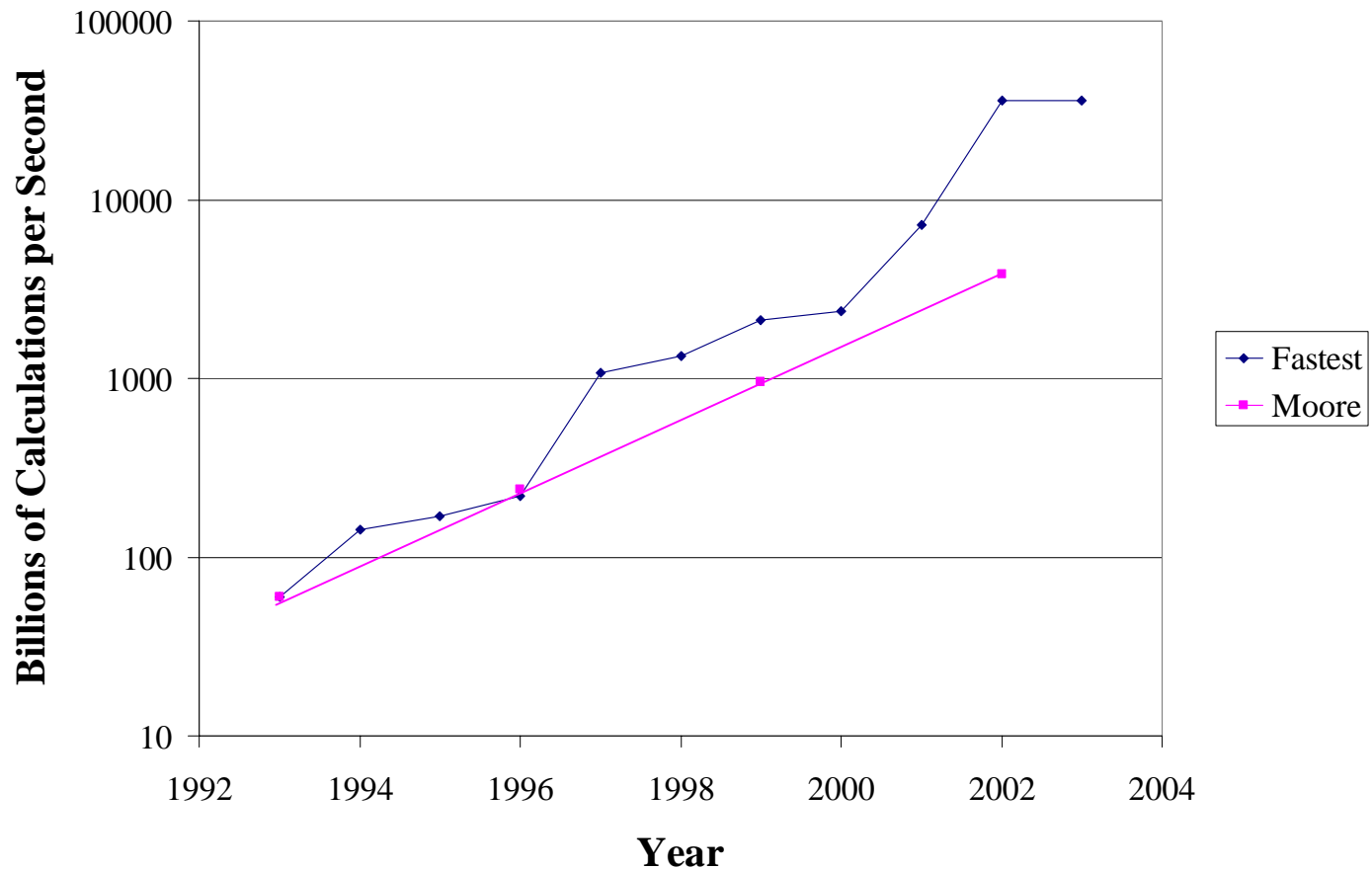
It turns out that computer speed is roughly proportional to the number of transistors per unit area.

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



Fastest Supercomputer

Fastest Supercomputer in the World



NCSI Parallel & Cluster Computing Workshop @ OU
August 8-14 2004





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.



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/us/en/bsd/products/model_latit_latit_c840.htm
- [5] <http://www.flphoto.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.toshiba.com/taecd/pd/products/features/MK2018gas-Over.shtml>
- [11] <http://www.toshiba.com/taecd/pd/techdocs/sdr2002/2002spec.shtml>
- [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/>

