Supercomputing in Plain English

Overview:
What the Heck is Supercomputing?

Henry Neeman, University of Oklahoma
Director, OU Supercomputing Center for Education & Research (OSCER)
Assistant Vice President, Information Technology – Research Strategy Advisor
Associate Professor, Gallogly College of Engineering
Adjunct Associate Professor, School of Computer Science
Tuesday January 23 2018

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.

If you lose your connection, you can retry the same kind of connection, or try connecting another way.

Remember, if all else fails, you always have the phone bridge to fall back on.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF

No matter how you connect, **PLEASE MUTE YOURSELF**, so that we cannot hear you.

At OU, we will turn off the sound on all conferencing technologies.

That way, we won't have problems with **echo cancellation**.

Of course, that means we cannot hear questions.

So for questions, you'll need to send e-mail:

supercomputinginplainenglish@gmail.com

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

Download the Slides Beforehand

Before the start of the session, please download the slides from the Supercomputing in Plain English website:

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

That way, if anything goes wrong, you can still follow along with just audio.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

YouTube

You can watch from a Windows, MacOS or Linux laptop or an Android or iOS handheld using YouTube.

Go to YouTube via your preferred web browser or app, and then search for:

`Supercomputing InPlainEnglish`

(**`InPlainEnglish`** is all one word.)

Many thanks to Skyler Donahue of OneNet for providing this.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

Twitch

You can watch from a Windows, MacOS or Linux laptop or an Android or iOS handheld using Twitch.

Go to:

http://www.twitch.tv/sipe2018

Many thanks to Skyler Donahue of OneNet for providing this.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

Wowza #1

You can watch from a Windows, MacOS or Linux laptop using Wowza from the following URL:

http://jwplayer.onenet.net/streams/sipe.html

If that URL fails, then go to:

http://jwplayer.onenet.net/streams/sipebackup.html

Many thanks to Skyler Donahue of OneNet for providing this.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

Wowza #2

Wowza has been tested on multiple browsers on each of:

- Windows 10: IE, Firefox, Chrome, Opera, Safari
- MacOS: Safari, Firefox
- Linux: Firefox, Opera

We've also successfully tested it via apps on devices with:

- Android
- iOS

Many thanks to Skyler Donahue of OneNet for providing this.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

Toll Free Phone Bridge

IF ALL ELSE FAILS, you can use our US TOLL phone bridge:

405-325-6688

684 684 #

NOTE: This is for **US** call-ins **ONLY**.

PLEASE MUTE YOURSELF and use the phone to listen.

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

Please use the phone bridge **ONLY IF** you cannot connect any other way: the phone bridge can handle only 100 simultaneous connections, and we have over 1000 participants.

Many thanks to OU CIO Eddie Huebsch for providing the phone bridge..

Please Mute Yourself

No matter how you connect, **PLEASE MUTE YOURSELF**, so that we cannot hear you.

(For YouTube, Twitch and Wowza, you don't need to do that, because the information only goes from us to you, not from you to us.)

At OU, we will turn off the sound on all conferencing technologies.

That way, we won't have problems with **echo cancellation**.

Of course, that means we cannot hear questions.

So for questions, you'll need to send e-mail.

PLEASE MUTE YOURSELF.

Questions via E-mail Only

Ask questions by sending e-mail to:

supercomputinginplainenglish@gmail.com

All questions will be read out loud and then answered out loud.

DON'T USE CHAT OR VOICE FOR QUESTIONS!

No one will be monitoring any of the chats, and if we can hear your question, you're creating an **echo cancellation** problem.

PLEASE MUTE YOURSELF.
PLEASE MUTE YOURSELF.

Onsite: Talent Release Form

If you're attending onsite, you **MUST** do one of the following:

- complete and sign the Talent Release Form,

OR

- sit behind the cameras (where you can't be seen) and don't talk at all.

If you aren't onsite, then **PLEASE MUTE YOURSELF.**

TENTATIVE Schedule

Tue Jan 23: Overview: What the Heck is Supercomputing?
Tue Jan 30: The Tyranny of the Storage Hierarchy
Tue Feb 6: Instruction Level Parallelism
Tue Feb 13: Stupid Compiler Tricks
Tue Feb 20: Shared Memory Multithreading
Tue Feb 27: Distributed Multiprocessing
Tue March   6: Applications and Types of Parallelism
Tue March 13: Multicore Madness
Tue March 20: **NO SESSION** (OU's Spring Break)
Tue March 27: High Throughput Computing
Tue Apr   3: GPGPU: Number Crunching in Your Graphics Card
Tue Apr 10: Grab Bag: Scientific Libraries, I/O Libraries, Visualization
Tue Apr 17: Topic to be announced
Tue Apr 24: Topic to be announced
Tue May 1: Topic to be announced

Thanks for helping!

- OU IT
  - OSCER operations staff (Dave Akin, Patrick Calhoun, Kali McLennan, Jason Speckman, Brett Zimmerman)
  - OSCER Research Computing Facilitators (Jim Ferguson, Horst Severini)
  - Debi Gentis, OSCER Coordinator
  - Kyle Dudgeon, OSCER Manager of Operations
  - Ashish Pai, Managing Director for Research IT Services
  - The OU IT network team
  - OU CIO Eddie Huebsch
- OneNet: Skyler Donahue
- Oklahoma State U: Dana Brunson

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.

If you lose your connection, you can retry the same kind of connection, or try connecting another way.

Remember, if all else fails, you always have the phone bridge to fall back on.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

PLEASE MUTE YOURSELF.

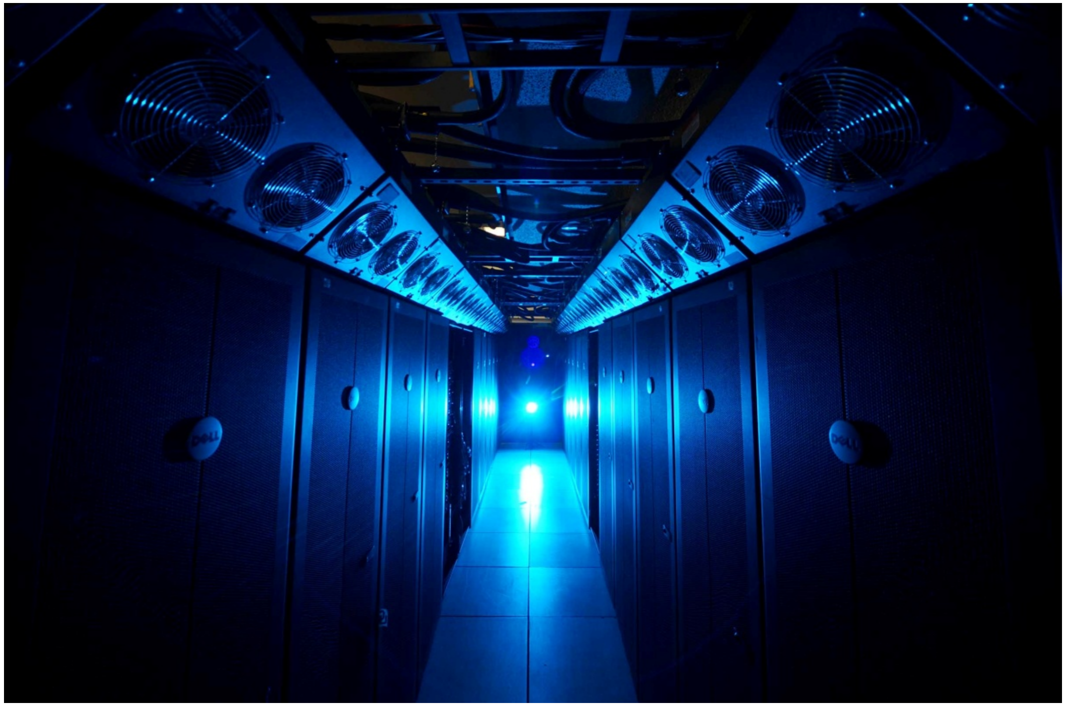
Coming in 2018!

- Coalition for Advancing Digital Research & Education (CADRE) Conference: Apr 17-18 2018 @ Oklahoma State U, Stillwater OK USA
  https://hpcc.okstate.edu/cadre-conference
- Linux Clusters Institute workshops
  http://www.linuxclustersinstitute.org/workshops/
  - Introductory HPC Cluster System Administration: May 14-18 2018 @ U Nebraska, Lincoln NE USA
  - Intermediate HPC Cluster System Administration: Aug 13-17 2018 @ Yale U, New Haven CT USA
- Great Plains Network Annual Meeting: details coming soon
- Advanced Cyberinfrastructure Research & Education Facilitators (ACI-REF) Virtual Residency Aug 5-10 2018, U Oklahoma, Norman OK USA
- PEARC 2018, July 22-27, Pittsburgh PA USA
  https://www.pearc18.pearc.org/
- IEEE Cluster 2018, Sep 10-13, Belfast UK
  https://cluster2018.github.io
- **OKLAHOMA SUPERCOMPUTING SYMPOSIUM 2018, Sep 25-26 2018 @ OU**
- SC18 supercomputing conference, Nov 11-16 2018, Dallas TX USA
  http://sc18.supercomputing.org/

People


EXIT
CHE5480 : A Course in Chemical Engineering &
Computer Science
Fall Semester 2003
Integration of HPC in
Nanotechnology
National Science Foundatior
Anita LaSalle, Program Manager (CISE
By Drs. Lloyd Lee, Gerald Newman, Chemical Engine
Dr. Henry Neeman, Computer Science

Things



Thanks for your attention!

Questions?

www.oscer.ou.edu

What is Supercomputing?

Supercomputing is the **biggest, fastest computing right this minute**.

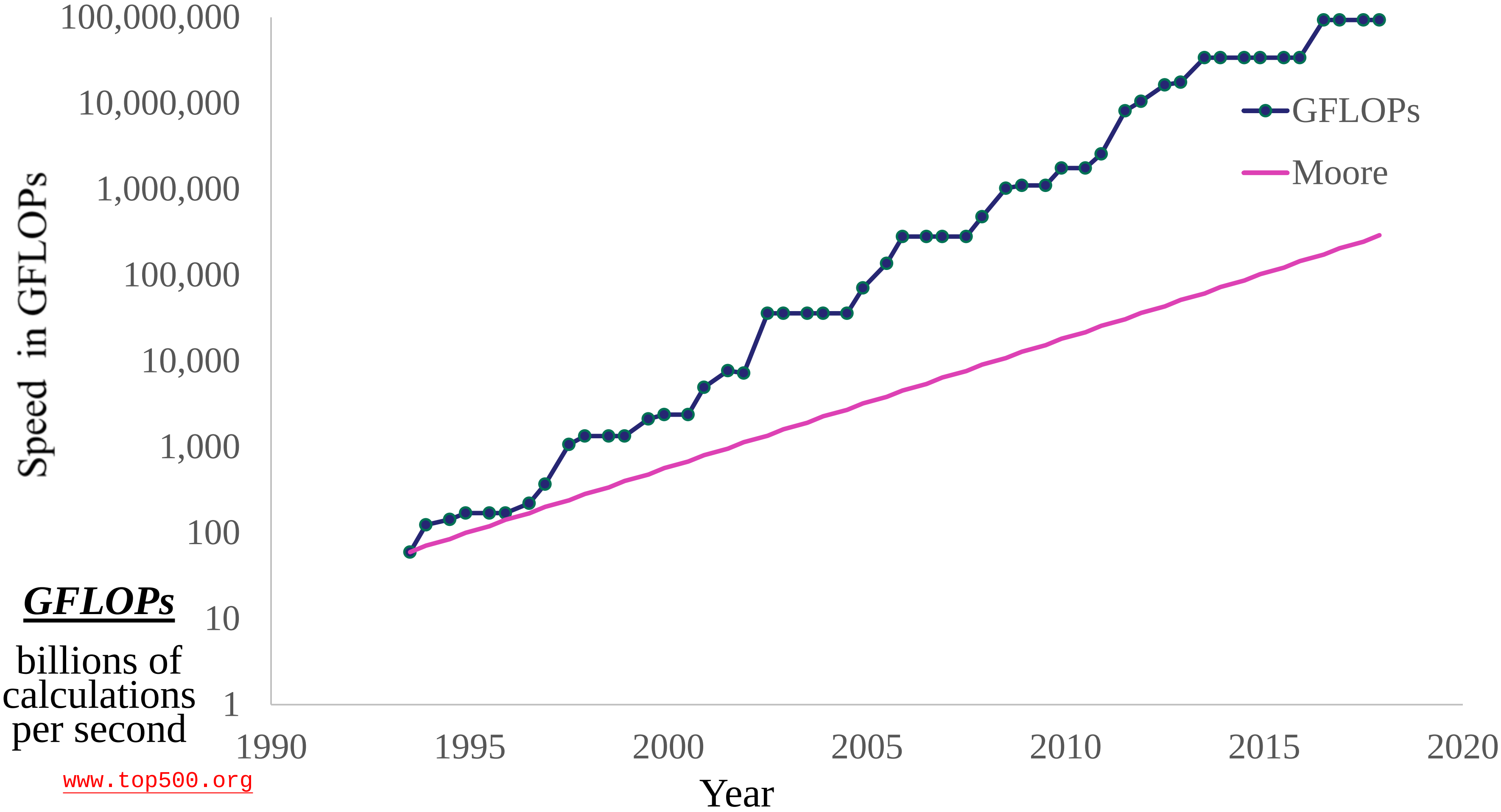
Likewise, a ***supercomputer*** is one of the biggest, fastest computers right this minute.

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



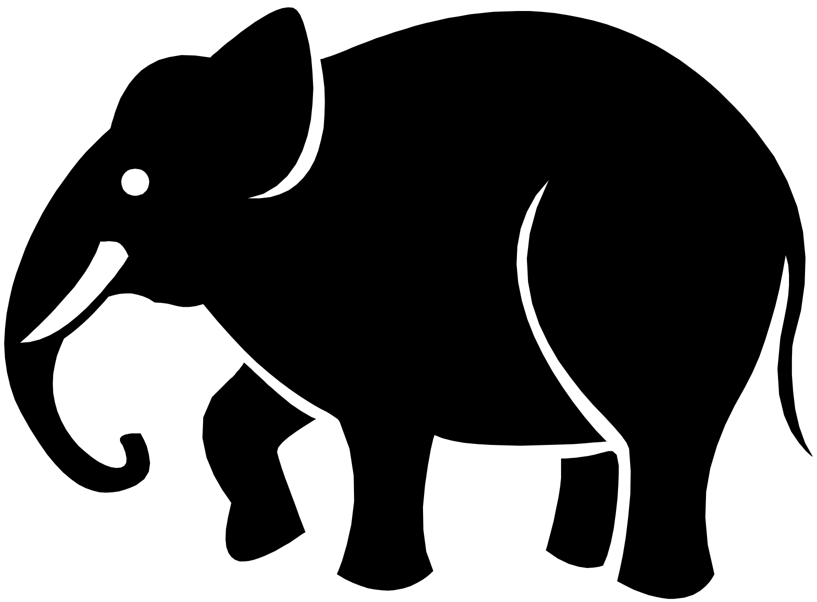
GFLOPs

billions of calculations per second

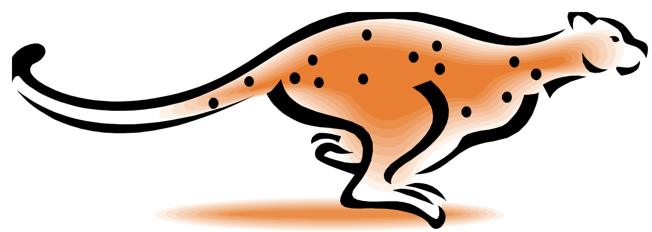
www.top500.org

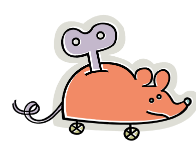
What is Supercomputing About?

Size



Speed

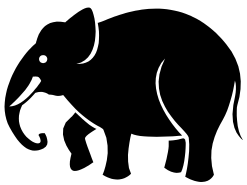




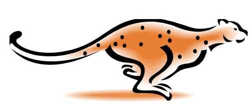
Laptop

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.



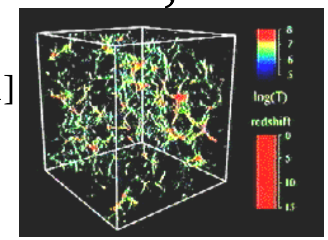
- **Speed**: Many problems that are interesting to scientists and engineers would take a very very long time to run on a PC: months or even years. But a problem that would take **a month on a PC** might take only **an hour on a supercomputer**.

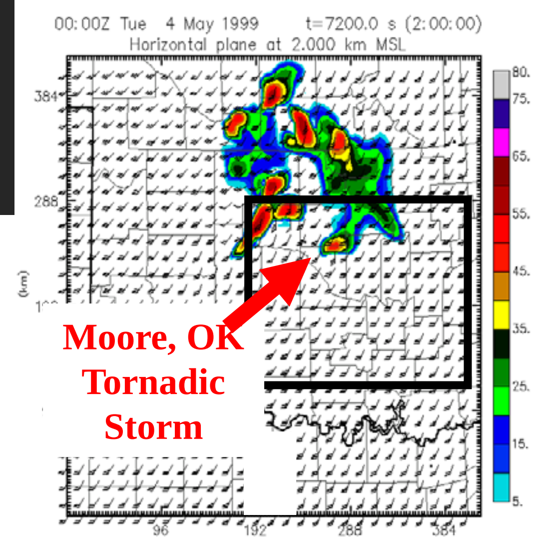


What Is HPC Used For?

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

[1]

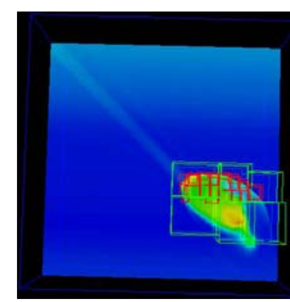




May 3 1999[2]

- ***Data mining***: finding **needles** of information in a **haystack** of data, such as
  - Gene sequencing
  - Signal processing
  - Detecting storms that might produce tornados
- ***Visualization***: turning a vast sea of **data** into **pictures** that a scientist can understand

[3]



Supercomputing Issues

- The tyranny of the ***storage hierarchy***
- ***Parallelism***: doing multiple things at the same time

What is a Cluster Supercomputer?

"… [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"

http://lh3.ggpht.com/_6hgSmco4R9M/SfpFA3057zI/AAAAAAAACSg/G-AGCgLrQOk/s1600-h/pirates%5B5%5D.jpg

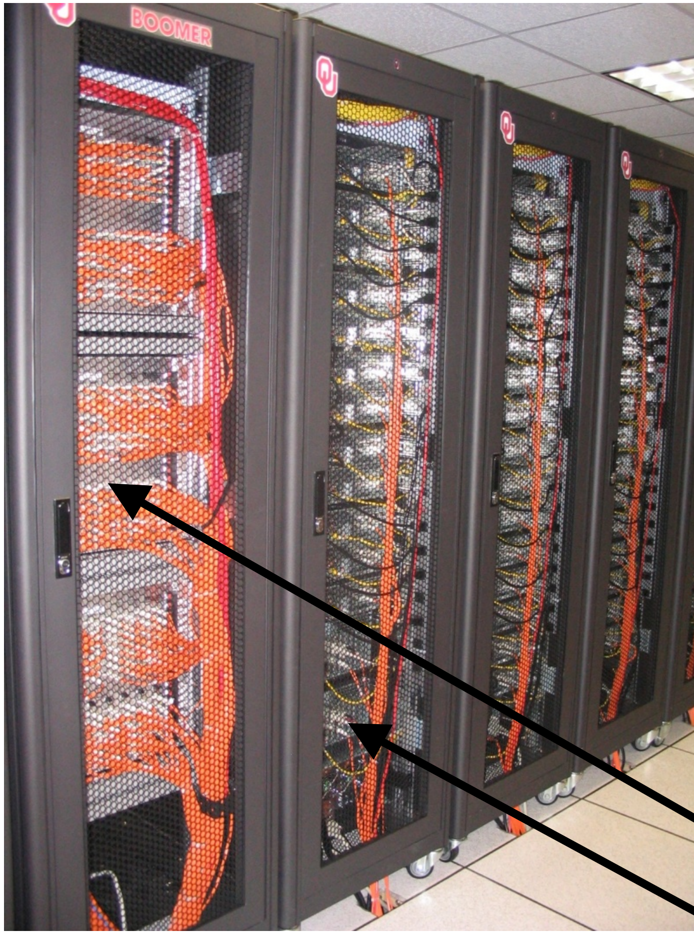
What a Cluster is ….

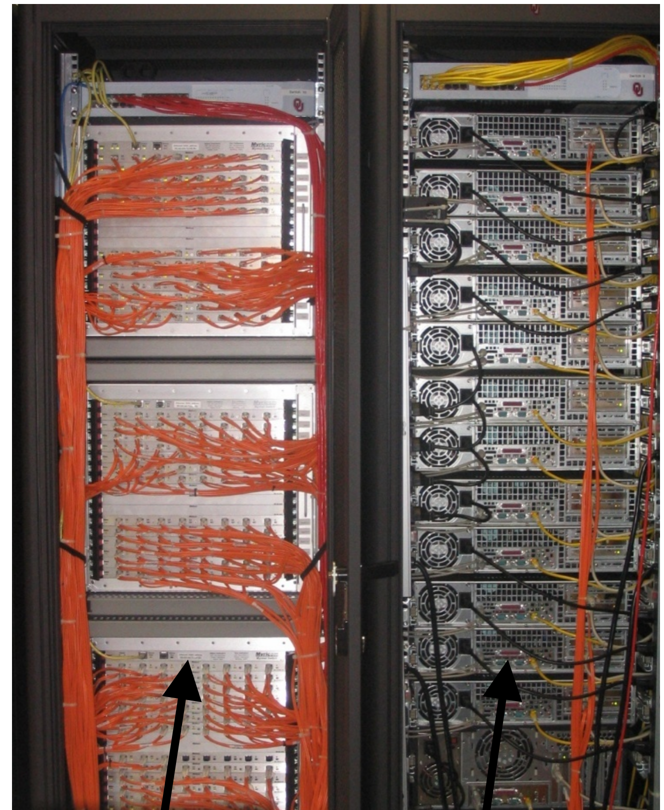
A cluster **<u>needs</u>** of a collection of small computers, called ***<u>nodes</u>***, hooked together by an ***<u>interconnection network</u>*** (or ***<u>interconnect</u>*** for short).

It also **<u>needs</u>** software that allows the nodes to communicate over the interconnect.

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

An Actual Cluster





Interconnect

Nodes

Boomer, in service 2002-5.

A Quick Primer on Hardware

Henry's Laptop

Dell Latitude E5540[4]

http://content.hwigroup.net/images/products/xl/204419/dell_latitude_e5540_55405115.jpg

- Intel Core i3-4010U
  dual core, 1.7 GHz, 3 MB L3 Cache
- 12 GB 1600 MHz DDR3L SDRAM
- 340 GB SATA 5400 RPM Hard Drive
- DVD±RW/CD-RW Drive
- 1 Gbps Ethernet Adapter

Typical Computer Hardware

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

Central Processing Unit

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

Components

- ***Control Unit***: figures out what to do next – for example, 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 – for example, 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, Blu-ray, magnetic tape, floppy disk
- Many are ***portable***: can pop out the CD/DVD/tape/floppy and take it with you

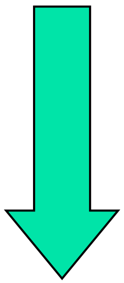
Input/Output

- Input devices – for example, keyboard, mouse, touchpad, joystick, scanner
- Output devices – for example, monitor, printer, speakers

The Tyranny of
the Storage Hierarchy

The Storage Hierarchy

Fast, expensive, few



Slow, cheap, a lot

- Registers
- Cache memory
- Main memory (RAM)
- Hard disk
- Removable media (CD, DVD etc)
- Internet

[5]

RAM is Slow

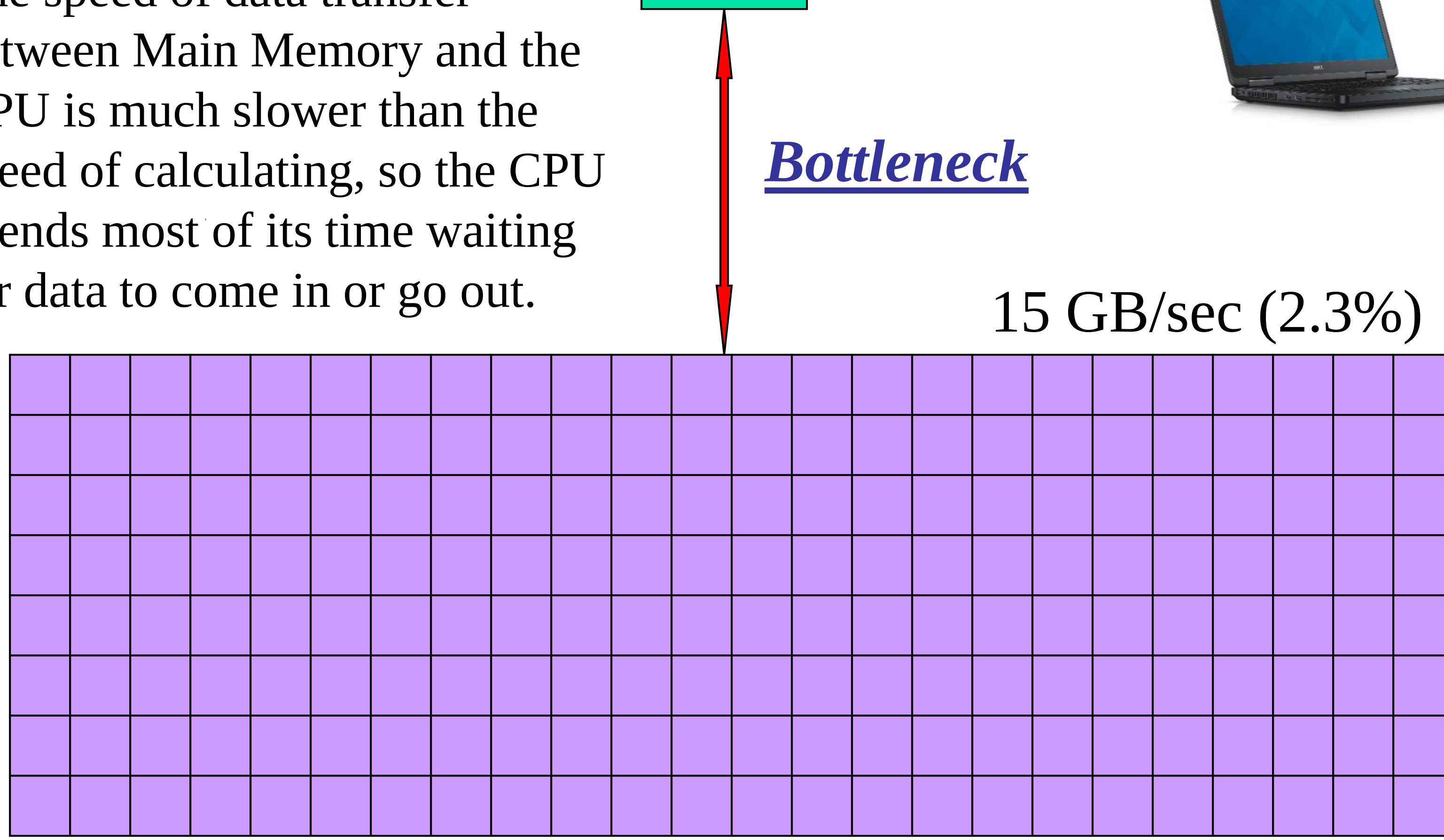
The speed of data transfer between Main Memory and the CPU is much slower than the speed of calculating, so the CPU spends most of its time waiting for data to come in or go out.

CPU

653 GB/sec

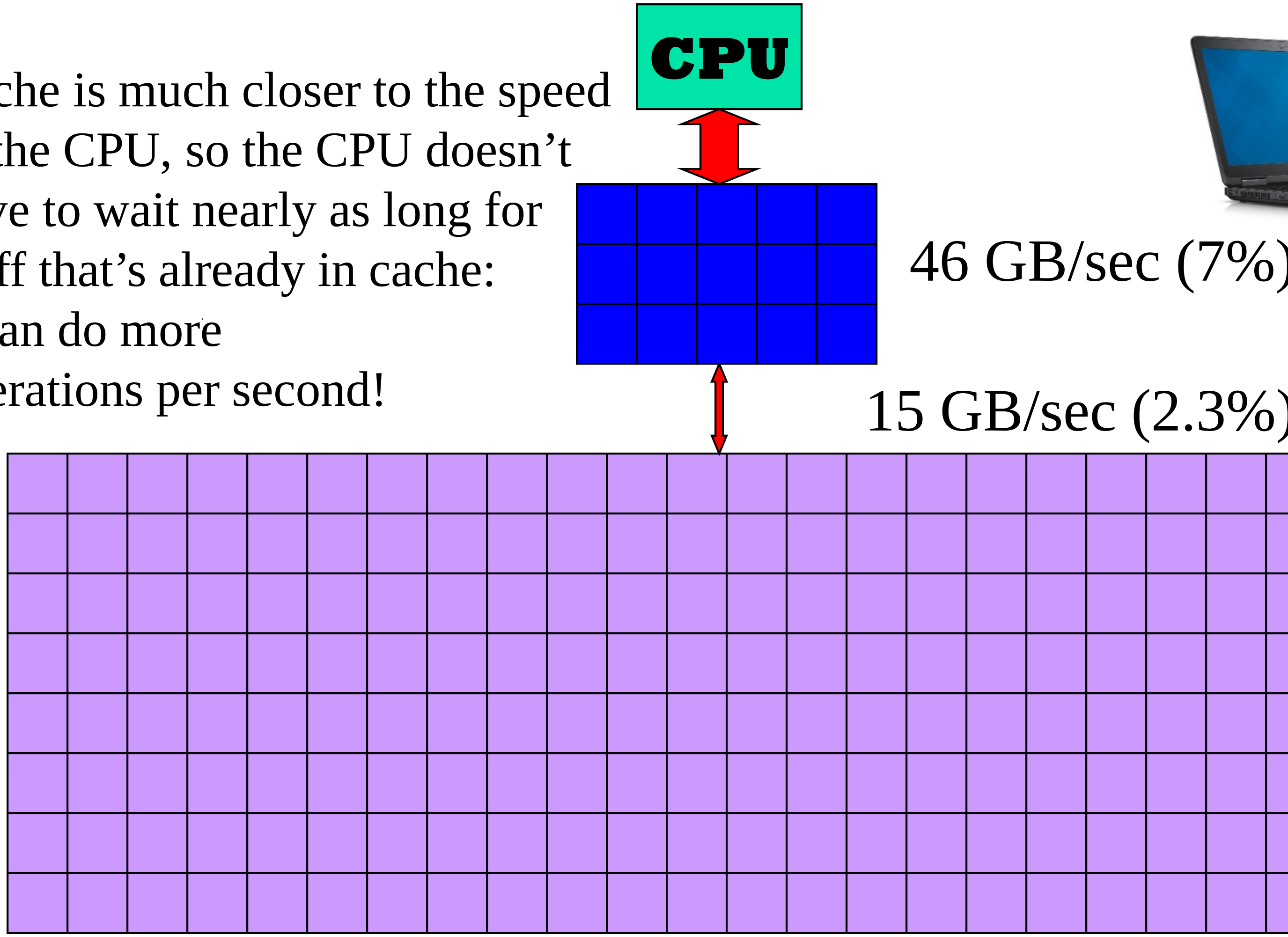
Bottleneck

15 GB/sec (2.3%)



Why Have Cache?

Cache is much closer to the speed of 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

Dell Latitude E5540[4]

http://content.hwigroup.net/images/products/xl/204419/dell_latitude_e5540_55405115.jpg

- Intel Core i3-4010U
  dual core, 1.7 GHz, 3 MB L3 Cache
- 12 GB 1600 MHz DDR3L SDRAM
- 340 GB SATA 5400 RPM Hard Drive
- DVD±RW/CD-RW Drive
- 1 Gbps Ethernet Adapter

Storage Speed, Size, Cost

Henry's Laptop	Registers (Intel Core2 Duo 1.6 GHz)	Cache Memory (L3)	Main Memory (1600MHz DDR3L SDRAM)	Hard Drive	Flash Thumb Drive (USB 3.0)	Ethernet (1000 Mbps)	Blu-Ray
Speed (MB/sec) [peak]	668,672[6] (16 GFLOP/s*)	46,000	15,000 [7]	100[9]	625	125	72 [10]
Size (MB)	10,752 bytes** [11]	3	12,288 4096 times as much as cache	340,000	1024	unlimited	unlimited
Cost ($/MB)	–	$20 [12]	$0.0093 [12] ~1/2000 as much as cache	$0.00003 [12]	$0.00018 [12]	charged per month (typically)	$0.00006 [12]

* GFLOP/s: billions of floating point operations per second

** 168 256-bit integer vector registers,
168 256-bit floating point vector registers

Why the Storage Hierarchy?

Why does the Storage Hierarchy always work? Why are faster forms of storage more expensive and slower forms cheaper?

Proof by contradiction:

Suppose there were a storage technology that was **slow** and **expensive**.

How much of it would you buy?

Comparison

- Floppy: 1.44 MB each, $0.69 ($0.48 per MB), speed 0.03 MB/sec
- Blu-Ray: 25 GB Disk ~$1 ($0.00006 per MB), speed 72 MB/sec

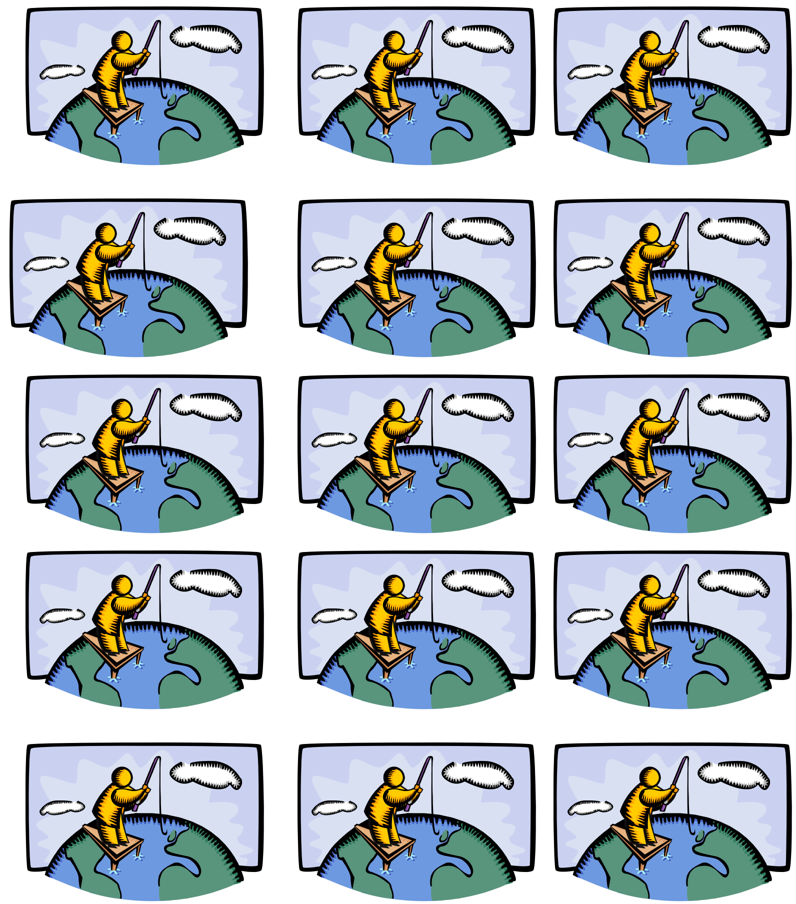
Not surprisingly, no one buys floppy disks any more.

Parallelism

Parallelism

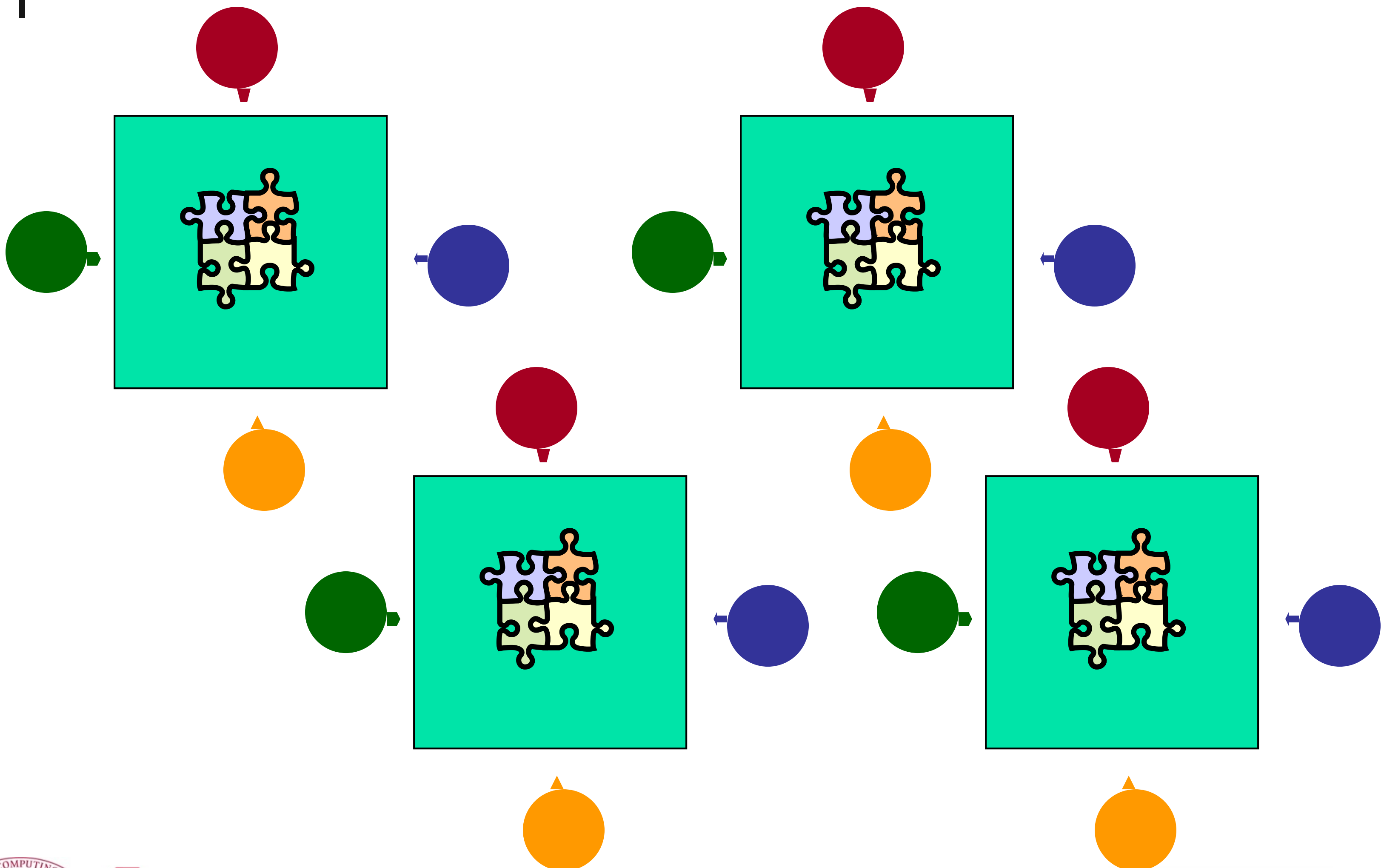
Parallelism means doing multiple things at the same time: you can get more work done in the same time.

Less fish …

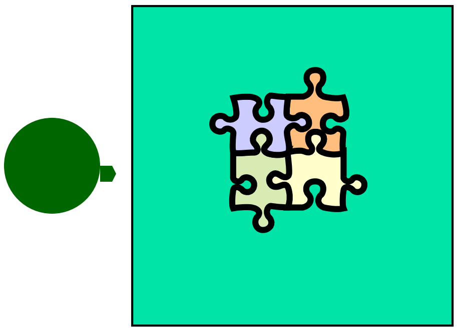


More fish!

The Jigsaw Puzzle Analogy



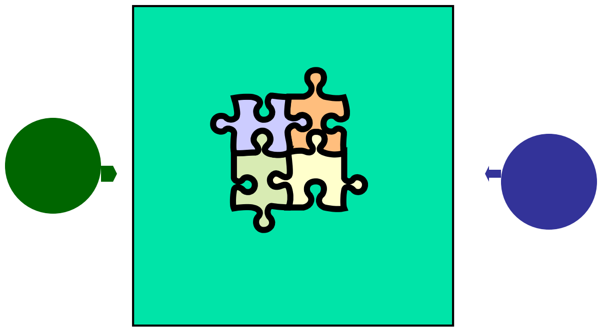
Serial Computing



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

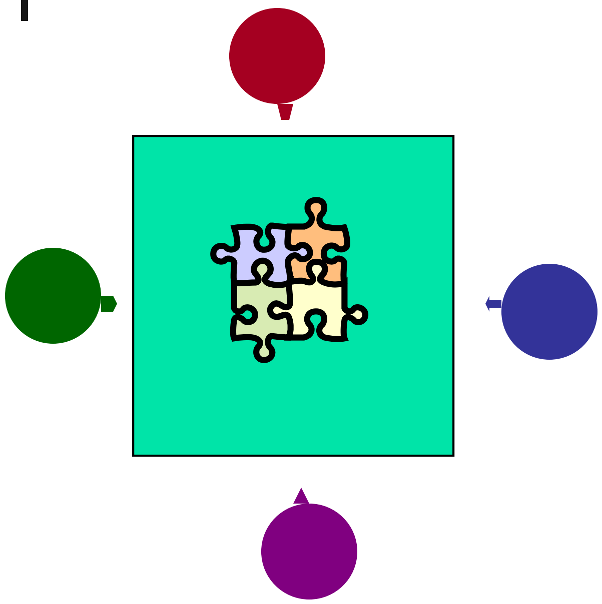
We can imagine that it'll take you a certain amount of time. Let's say that you can put the puzzle together in an hour.

Shared Memory Parallelism



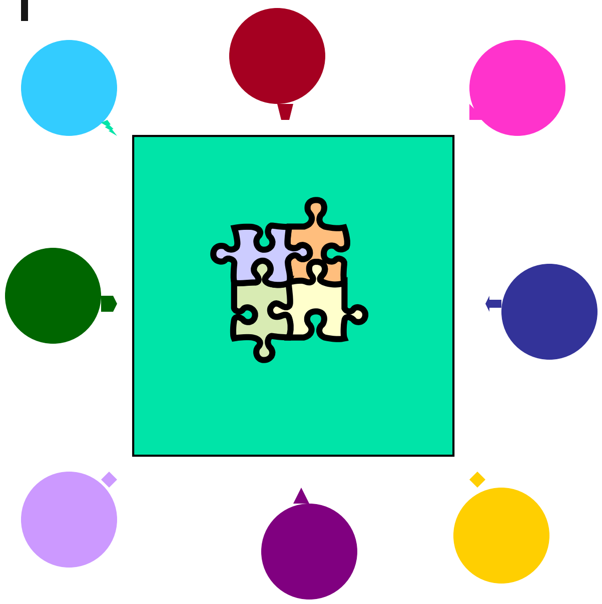
If Scott sits across the table from you, then he can work on his half of the puzzle and you can work on yours. Once in a while, you'll both reach into the pile of pieces at the same time (you'll ***contend*** for the same resource), which will cause a little bit of slowdown. And from time to time you'll 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 Paul and Charlie 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



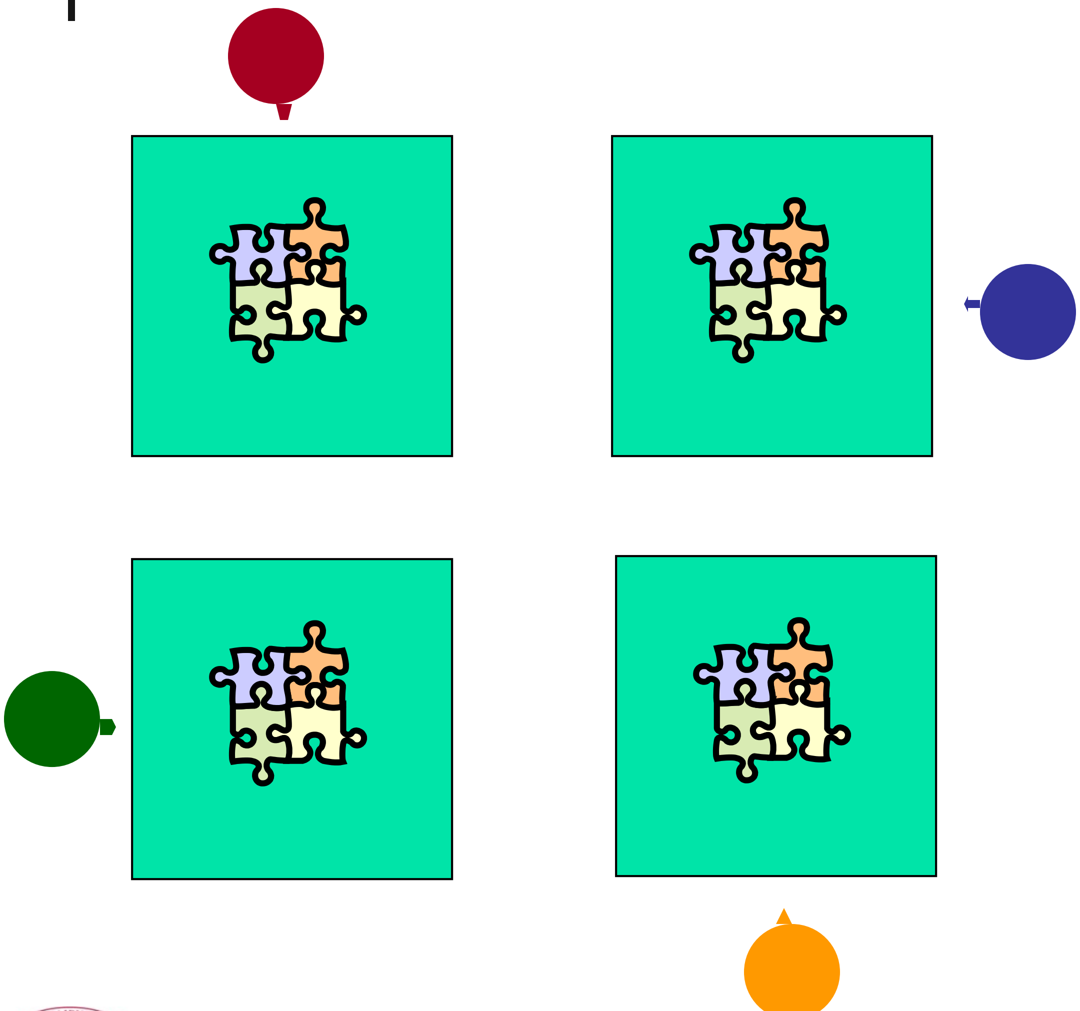
If we now put Dave and Tom and Horst and Brandon on the corners of the table, there's going to be a whole lot of contention for the shared resource, and a lot of communication at the many interfaces. So the speedup y'all get will be much less than we'd like; you'll be lucky to get 5-to-1.

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

Distributed Parallelism

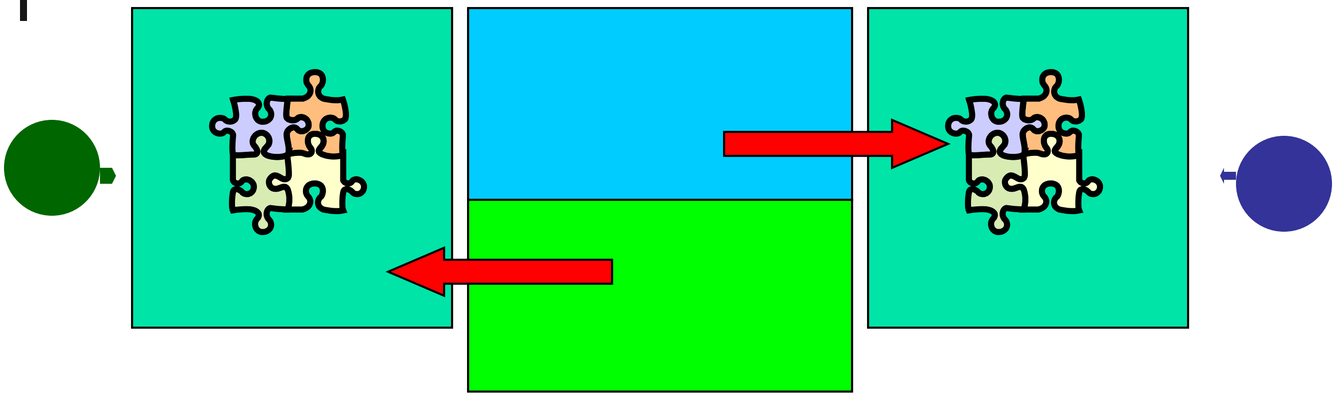
Now let's try something a little different. Let's set up two tables, and let's put you at one of them and Scott at the other. Let's put half of the puzzle pieces on your table and the other half of the pieces on Scott's. Now y'all can work completely independently, without any contention for a shared resource. **BUT**, the cost per communication 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 among the processors. Also, as you add more processors, it may be harder to ***load balance*** the amount of work that each processor gets.

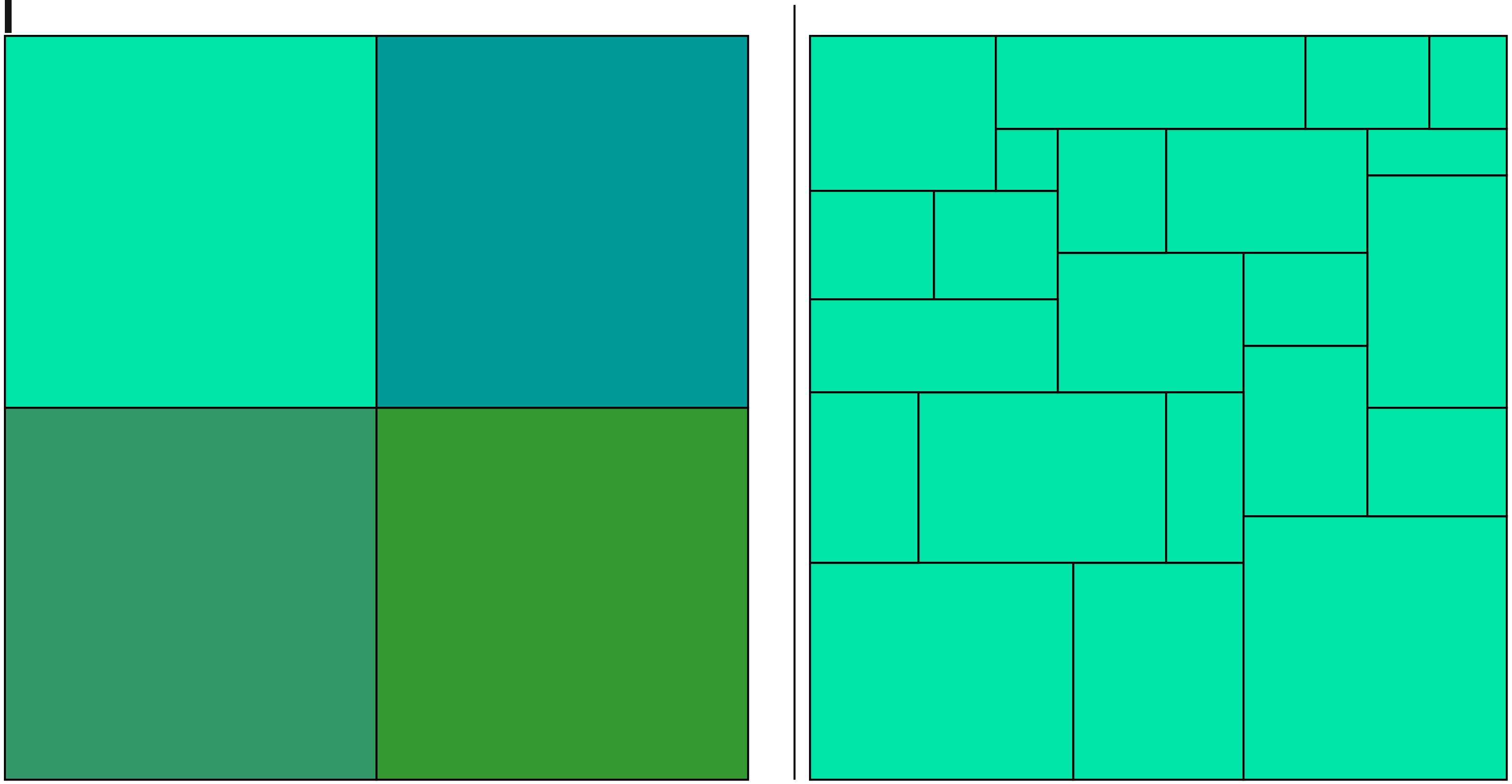
Load Balancing



Load balancing means ensuring that everyone completes their workload at roughly the same time.

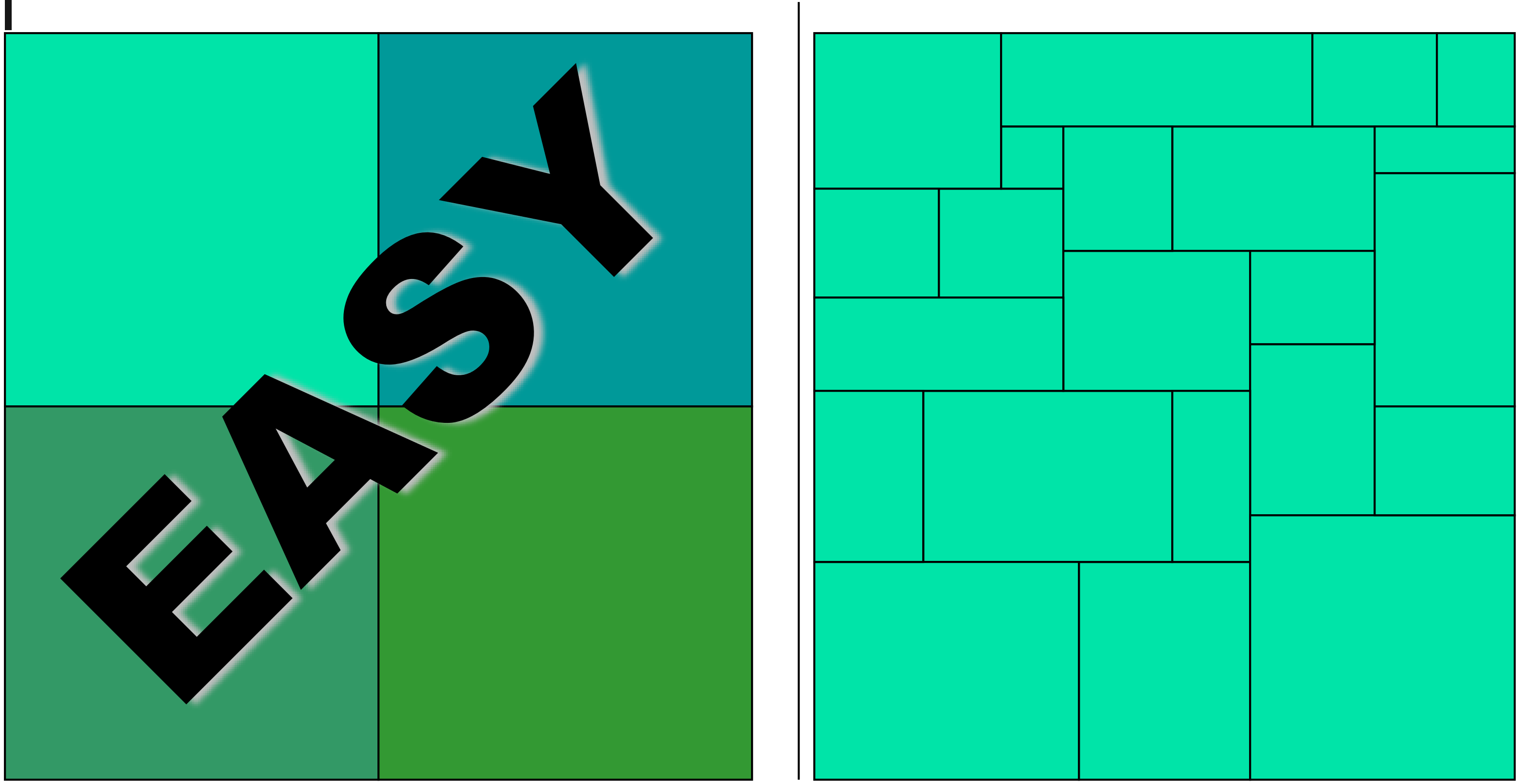
For example, if the jigsaw puzzle is half grass and half sky, then you can do the grass and Scott can do the sky, and then y'all only have to communicate at the horizon – and the amount of work that each of you does on your own is roughly equal. So you'll get pretty good speedup.

Load Balancing



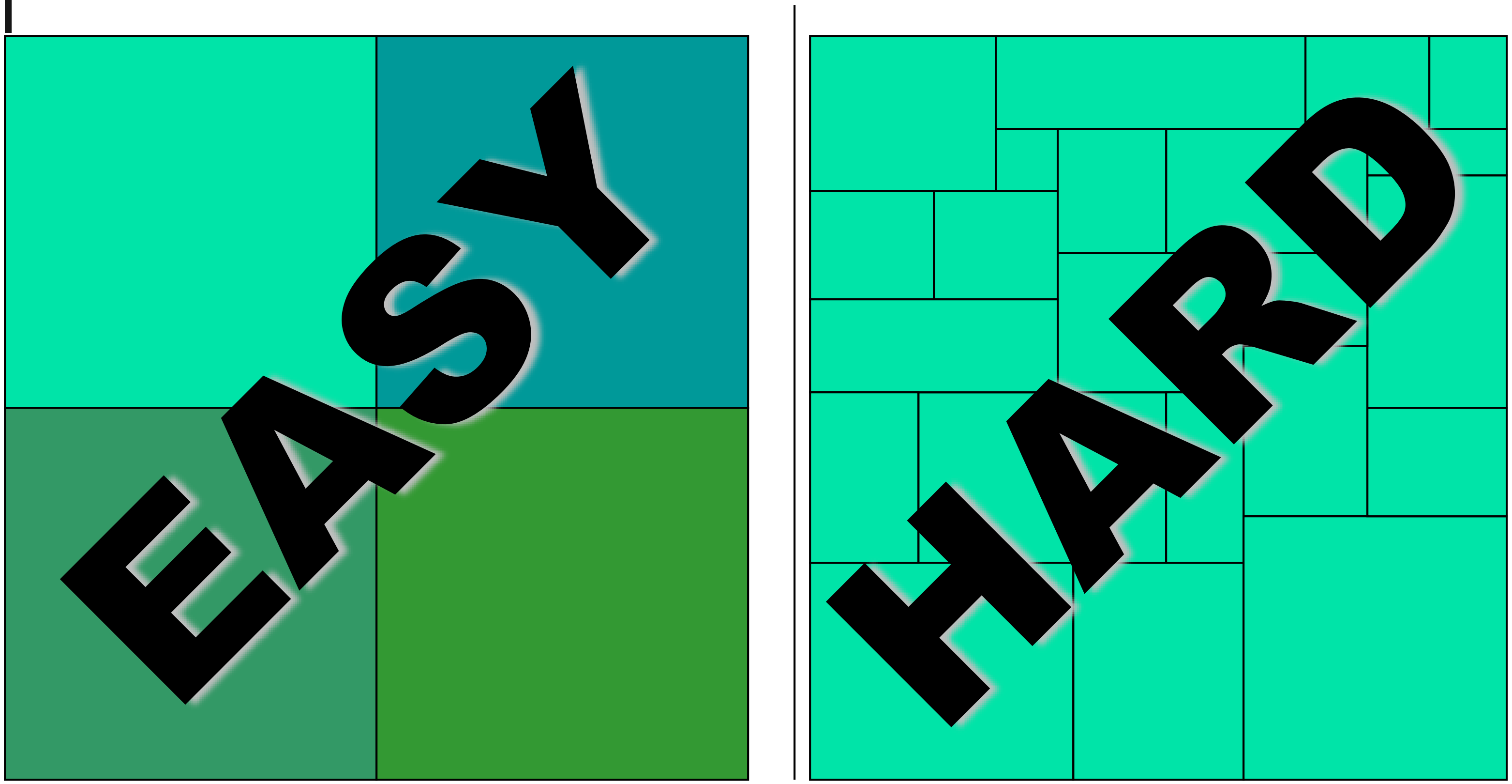
Load balancing can be easy, if the problem splits up into chunks of roughly equal size, with one chunk per processor. Or load balancing can be very hard.

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.

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 2 years.

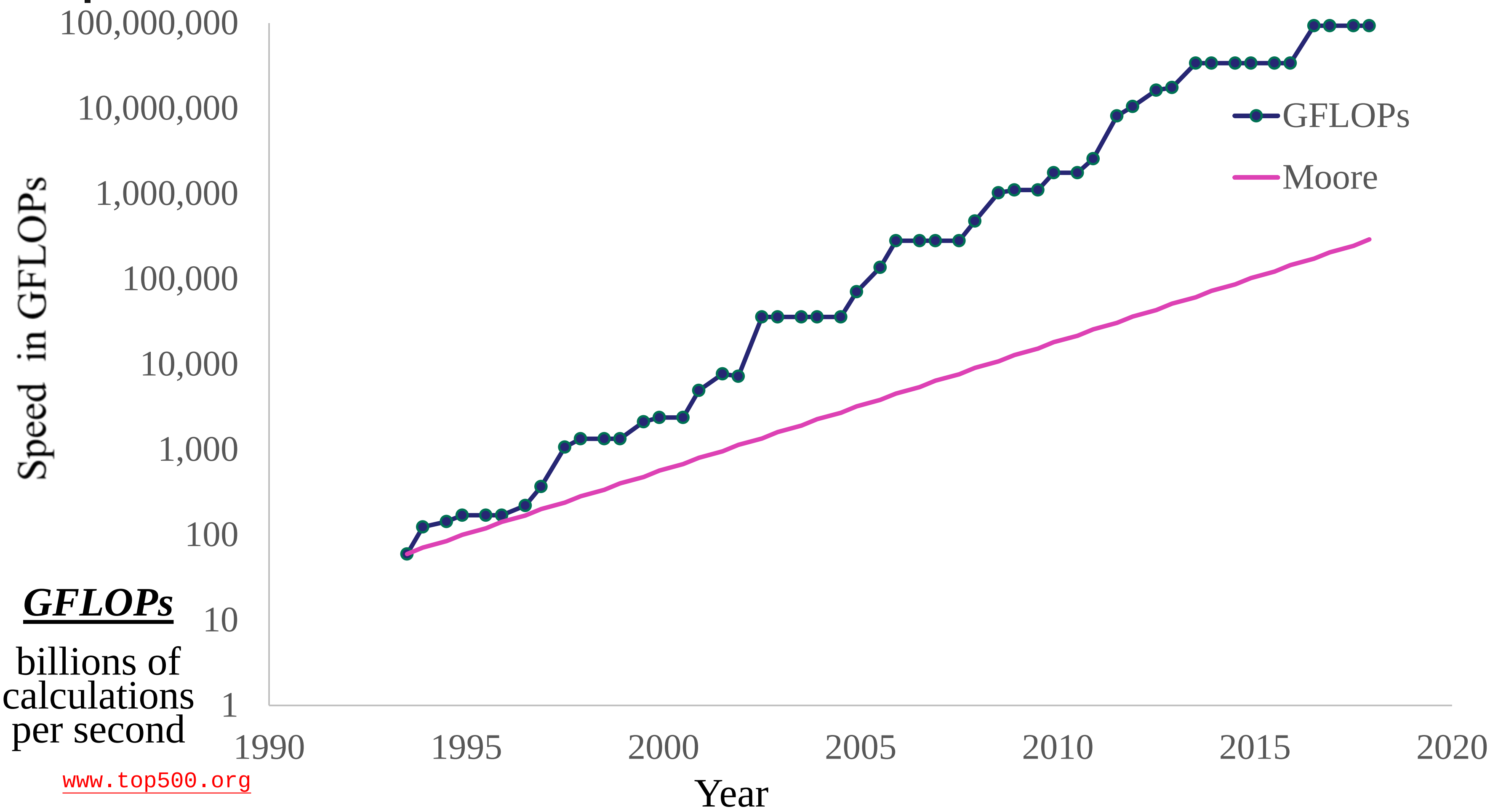
It turns out that computer speed, and storage capacity, is roughly proportional to the number of transistors per unit area.

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

(Originally, he predicted a doubling every year, but not long after, he revised that to every other year.)

G. Moore, 1965: "Cramming more components onto integrated circuits." *Electronics*, 38 (8), 114-117.

Fastest Supercomputer vs. Moore

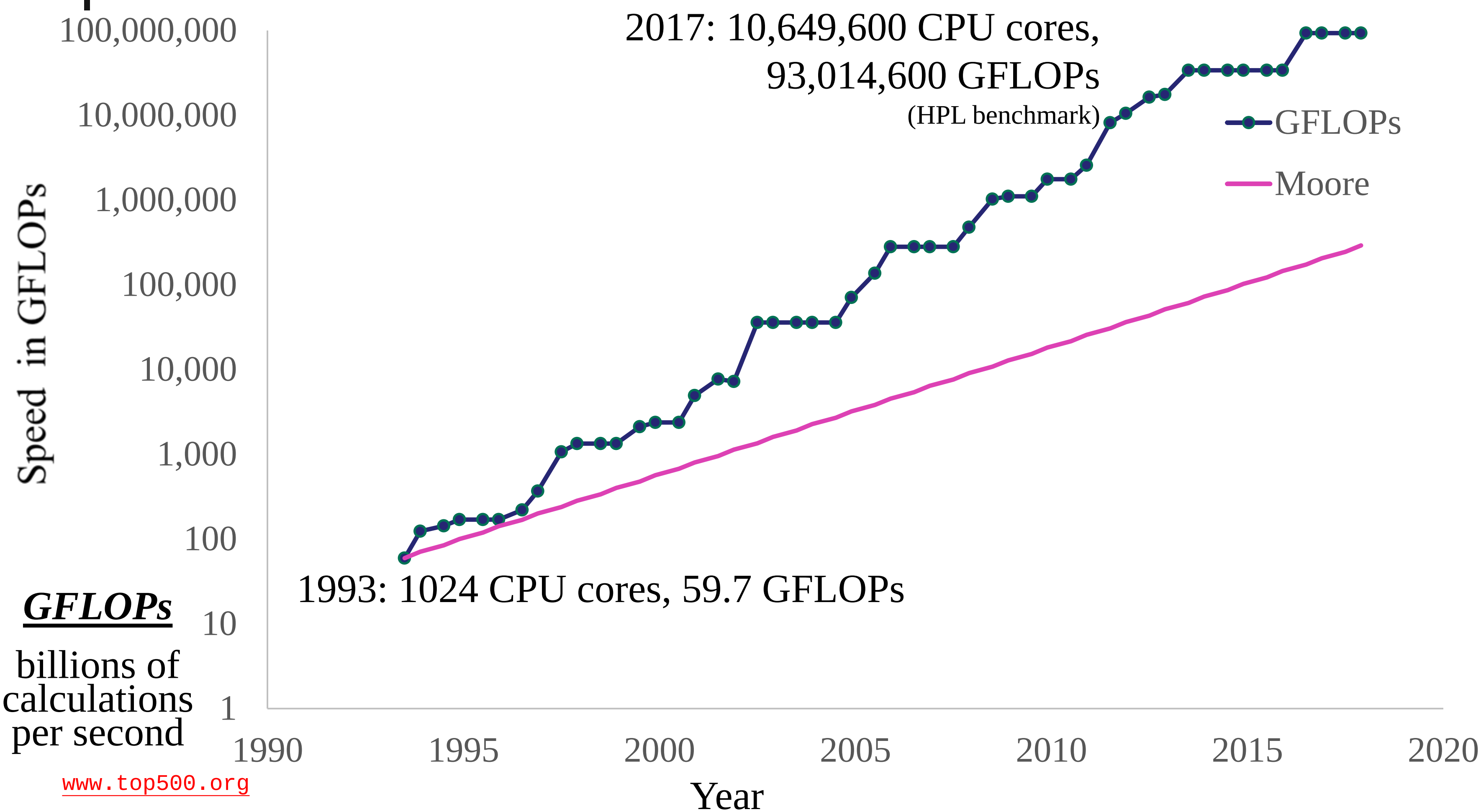


GFLOPs

billions of calculations per second

www.top500.org

Fastest Supercomputer vs. Moore



GFLOPs

billions of calculations per second

www.top500.org

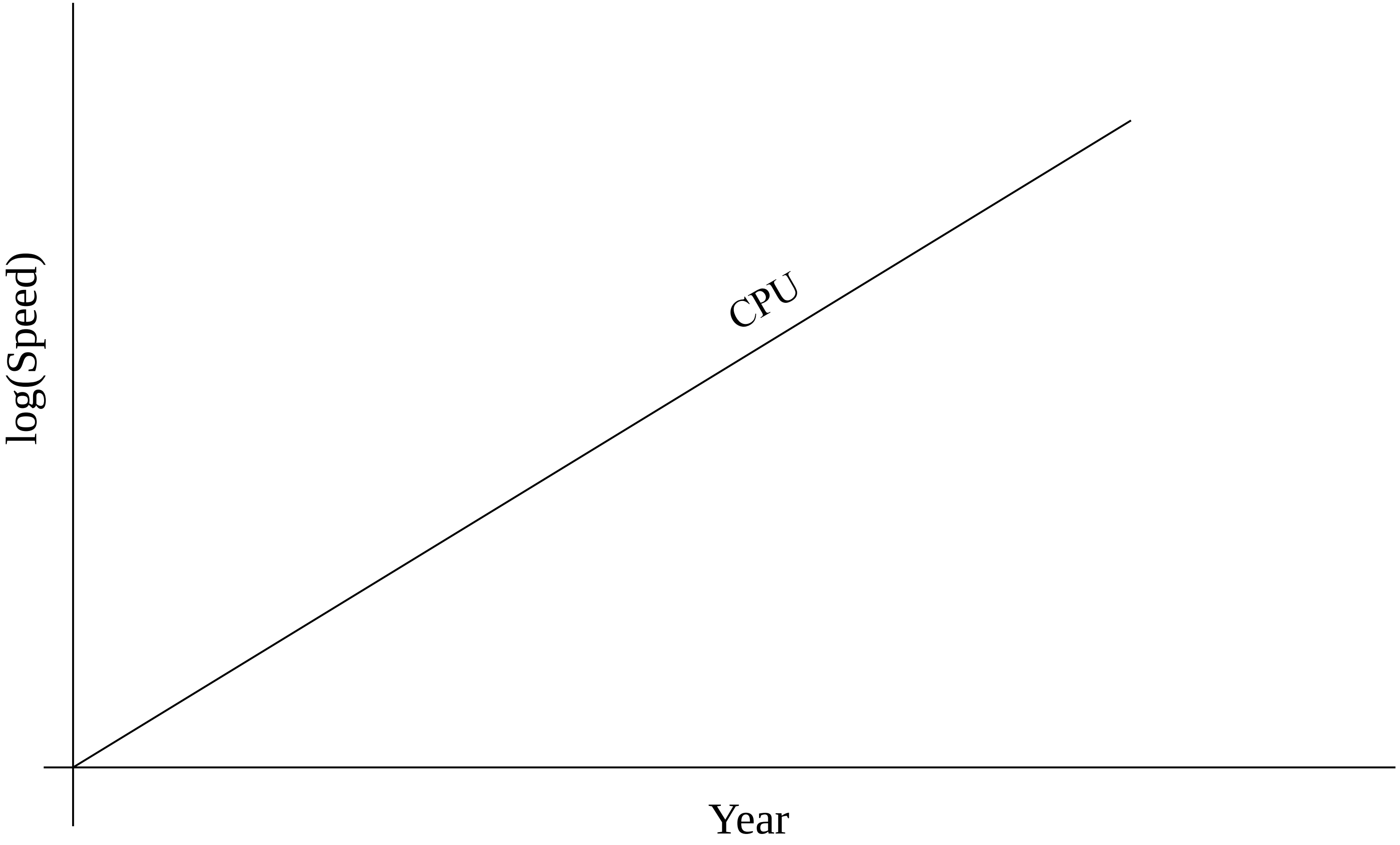
Moore: Uncanny!

- Nov 1971: Intel 4004 – 2300 transistors
- March 2010: Intel Nehalem Beckton – 2.3 billion transistors
- Factor of 1,000,000 improvement in 38 1/3 years
- $2^{(38.33 \text{ years} / 1.9232455)} = 1{,}000{,}000$

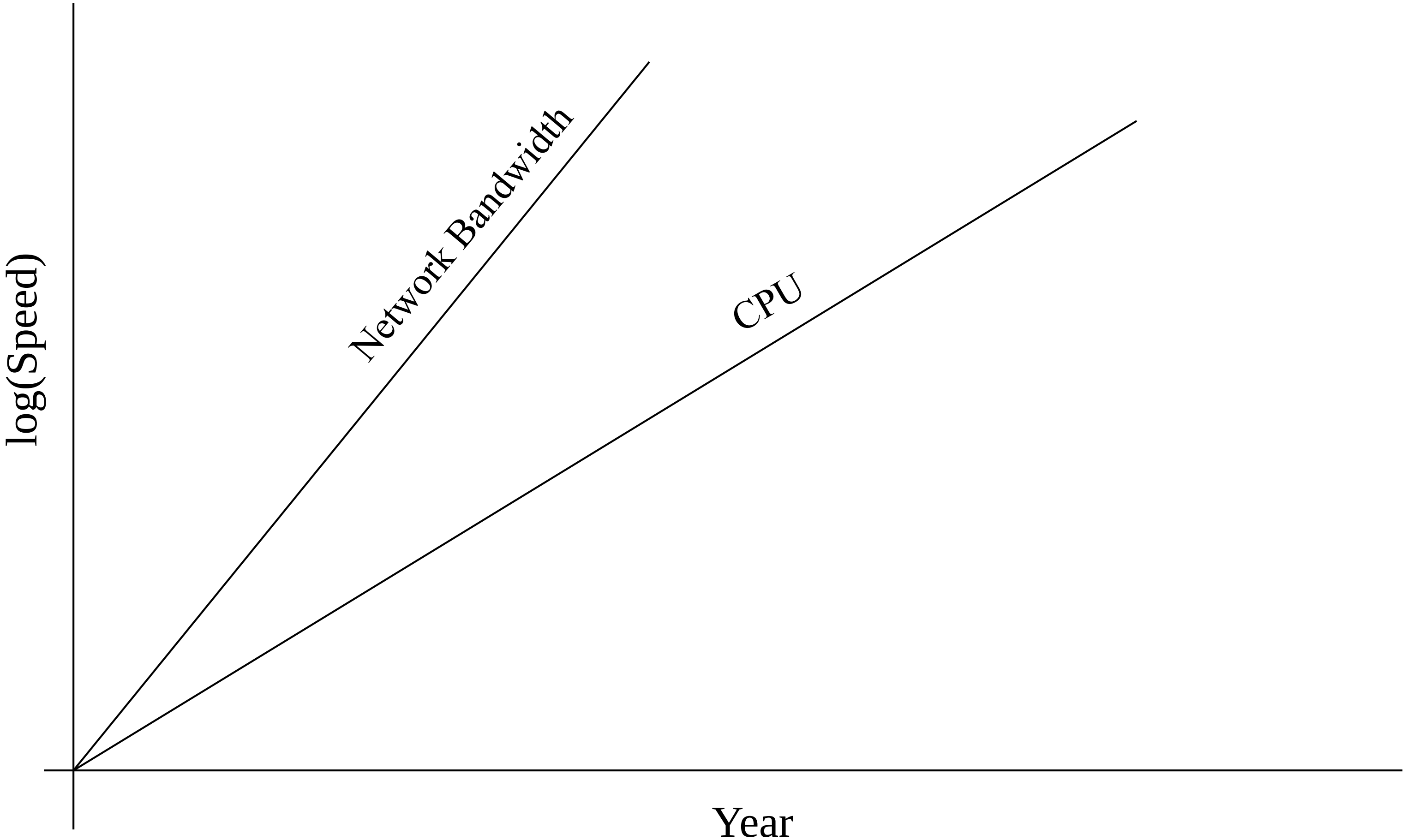
So, transistor density has doubled every 23 months:

UNCANNILY ACCURATE PREDICTION!

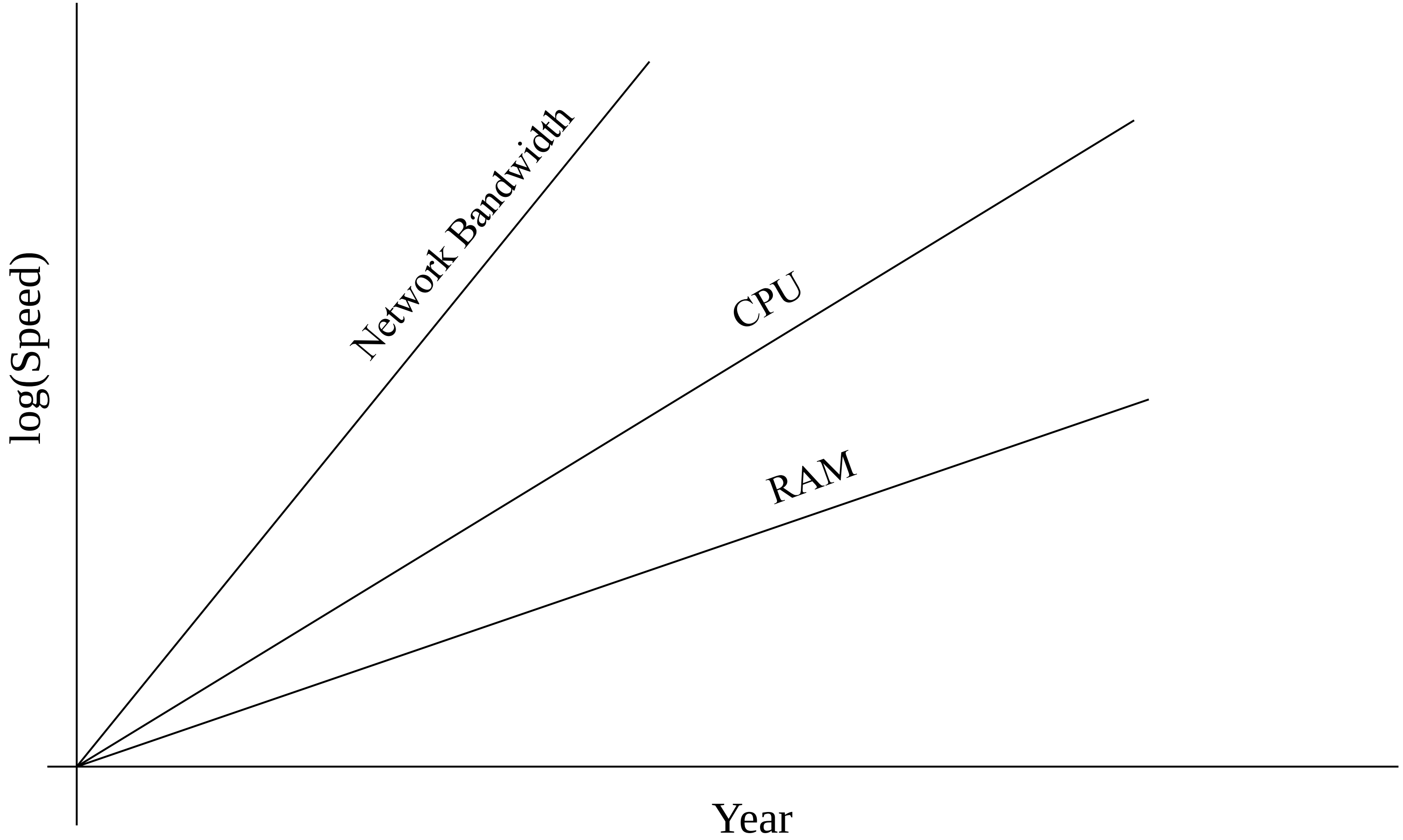
Moore's Law in Practice



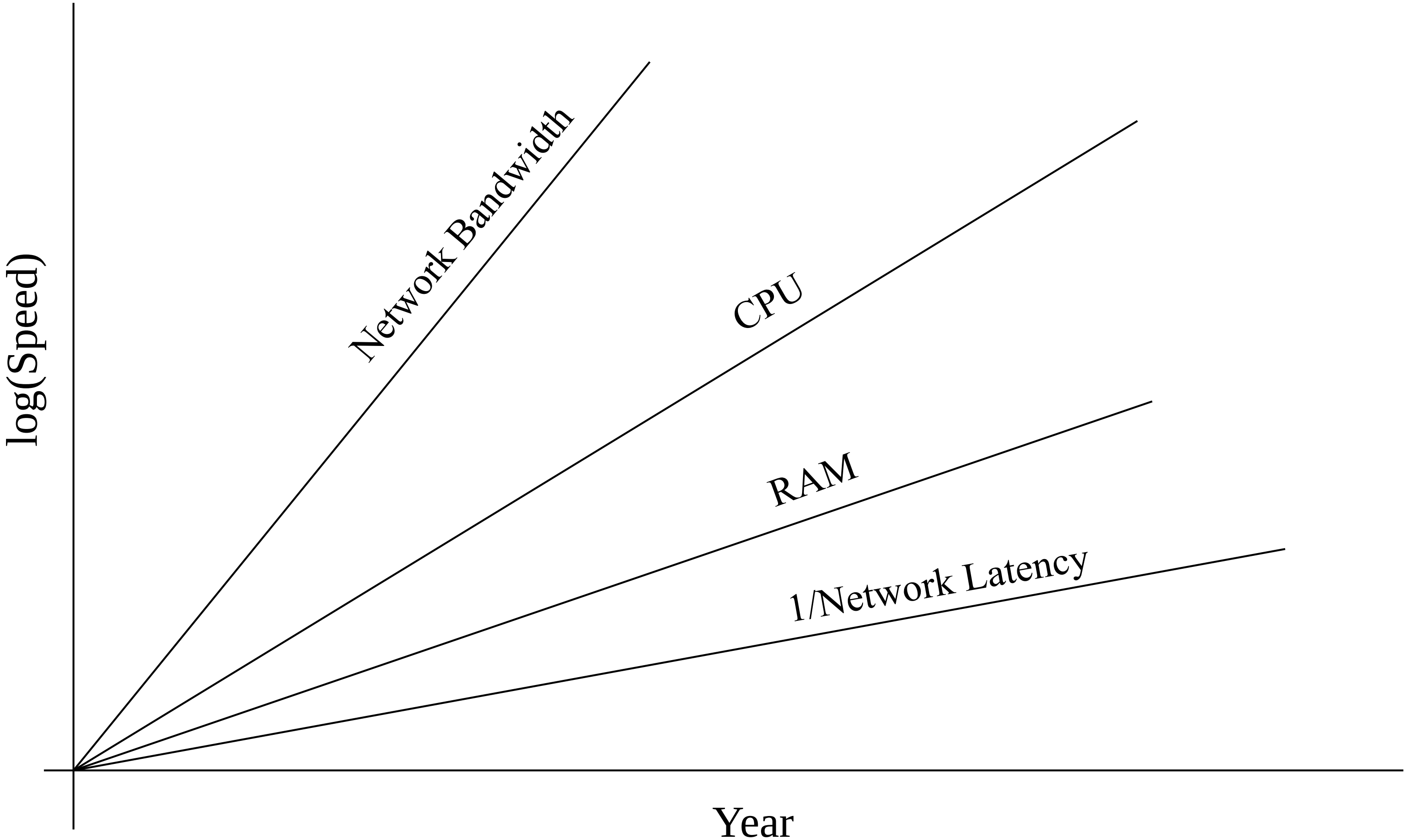
Moore's Law in Practice



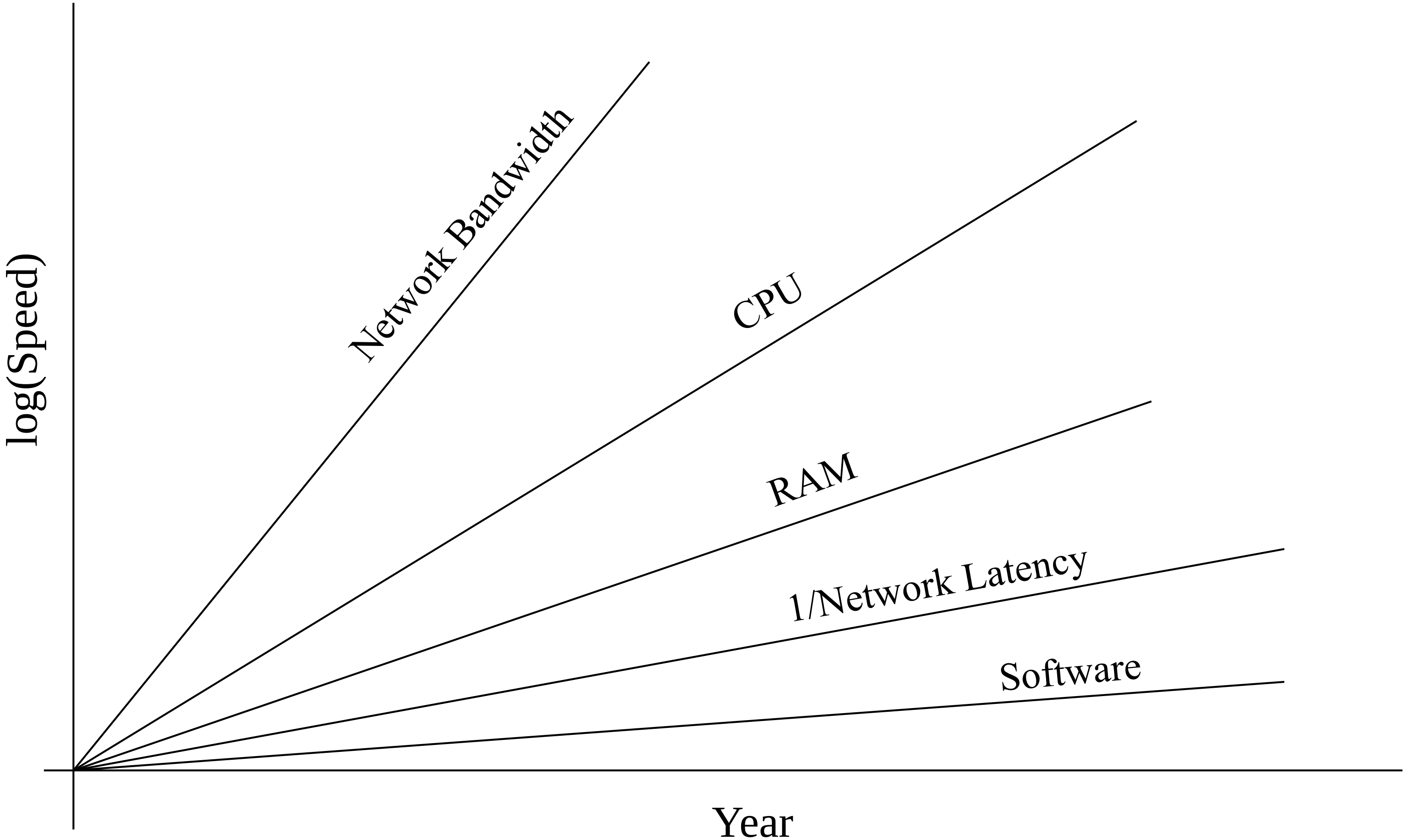
Moore's Law in Practice



Moore's Law in Practice

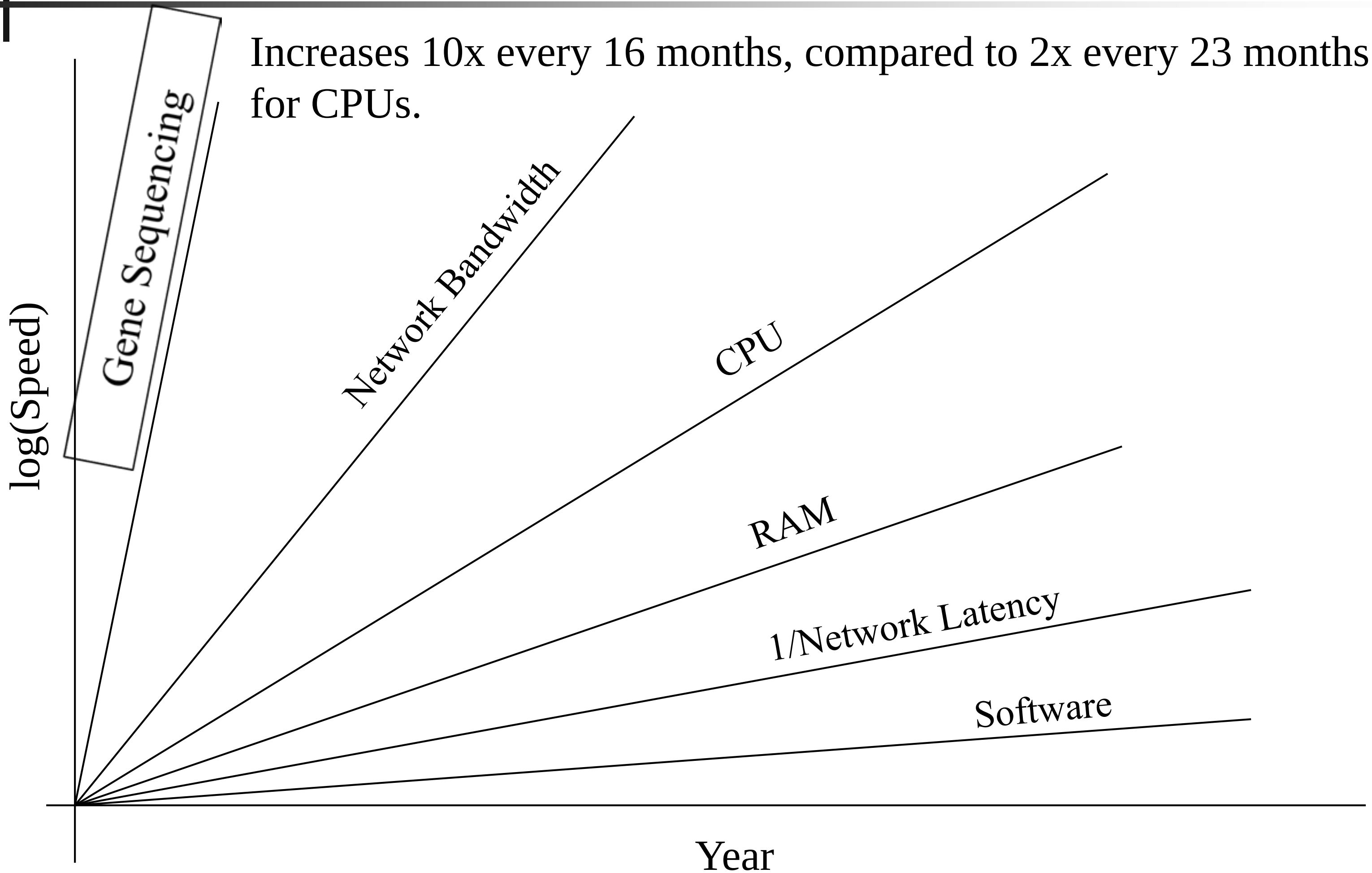


Moore's Law in Practice



Moore's Law on Gene Sequencers

Increases 10x every 16 months, compared to 2x every 23 months for CPUs.



What does 1 TFLOPs Look Like?

1 TFLOPs: trillion calculations per second

2002: Row → **2012: Card**

1997: Room

ASCI RED[13]
Sandia National Lab

`boomer.oscer.ou.edu`
In service 2002-5: 11 racks

AMD FirePro W9000[14]

NVIDIA Kepler K20[15]

Intel MIC Xeon PHI[16]

CPU Chip 2017

AMD EPYC

Intel Skylake

https://www.top500.org/static/media/uploads/.thumbnails/epyc-vs-xeon.jpg/epyc-vs-xeon-742x382.jpg

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, now 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 10 to 15 years and on your cell phone in 25 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.

Thanks for your attention!

Questions?

www.oscer.ou.edu

References

[1] Image by Greg Bryan, Columbia U.

[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://hneeman.oscer.ou.edu/hamr.html for details.

[4] http://www.dell.com/

[5] http://www.vw.com/newbeetle/

[6] Richard Gerber, The Software Optimization Cookbook: High-performance Recipes for the Intel Architecture. Intel Press, 2002, pp. 161-168.

[7] RightMark Memory Analyzer. http://cpu.rightmark.org/

[8] ftp://download.intel.com/design/Pentium4/papers/24943801.pdf

[9] http://www.samsungssd.com/meetssd/techspecs

[10] http://www.samsung.com/Products/OpticalDiscDrive/SlimDrive/OpticalDiscDrive_SlimDrive_SN_S082D.asp?page=Specifications

[11] https://www.realworldtech.com/haswell-cpu/3/

[12] http://www.pricewatch.com/