Fundamentals of Computational Science

Dr. Hyrum D. Carroll

August 23, 2016

Introductions

Each student:

- Name
- Undergraduate school & major
- Masters & major
- Previous research (if any)
- Why Computational Science PhD?

Introductions

- Hyrum D. Carroll
- Brigham Young University, Computer Engineering (B.S.)
- Brigham Young University, Computer Science (M.S.)
- Brigham Young University, Computer Science (Ph.D.)
- National Institutes of Health, Postdoc
- Research in Bioinformatics / Computational Biology
- Computational Science: Interdisciplinary research.

Introductions

- Hyrum D. Carroll
- Brigham Young University, Computer Engineering (B.S.)
- Brigham Young University, Computer Science (M.S.)
- Brigham Young University, Computer Science (Ph.D.)
- National Institutes of Health, Postdoc
- Research in Bioinformatics / Computational Biology
- Computational Science: Interdisciplinary research.
- https://cs.mtsu.edu/~hcarroll/personal/family/2015/IMG_4474edited.jpg

Syllabus

http://www.cs.mtsu.edu/~hcarroll/6100/syllabus.html

Lecture Notes

See `www.cs.mtsu.edu/~hcarroll/6100/assignments/latexLectureNotes.html`.

Each lecture period, a student will be responsible for taking lecture notes and posting them. The notes should cover the topics presented in class (whether or not they're on the slides), in the student's own words. Students will need to use LaTeXfor the notes.

Grading Rubric:
Completeness
Accuracy
Concise
Additional information
Overall appearance

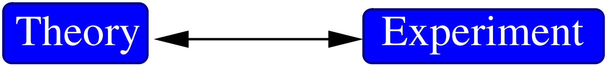
Email the `.tex` and `.pdf` files to me to be graded and posted.

COMS 6100

- Computational Tools and Techniques
- Scientific Programming and Optimization
- Numerical Methods and Libraries

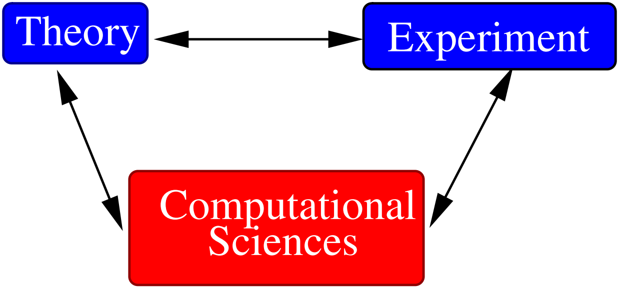
Science

Old-style



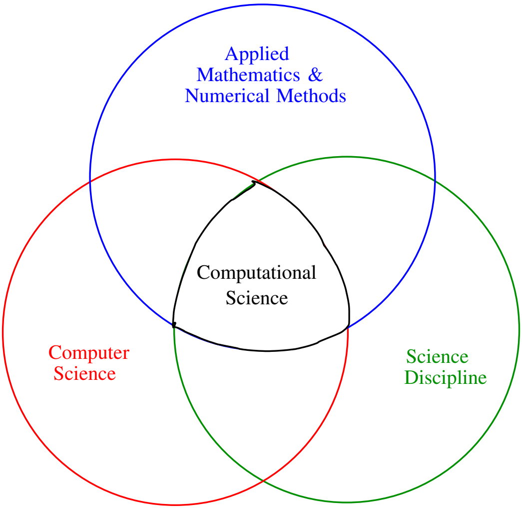
Computational Science

The Science Triad



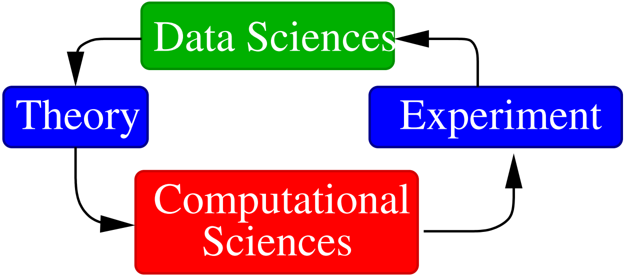
Computational Science

A blend of disciplines



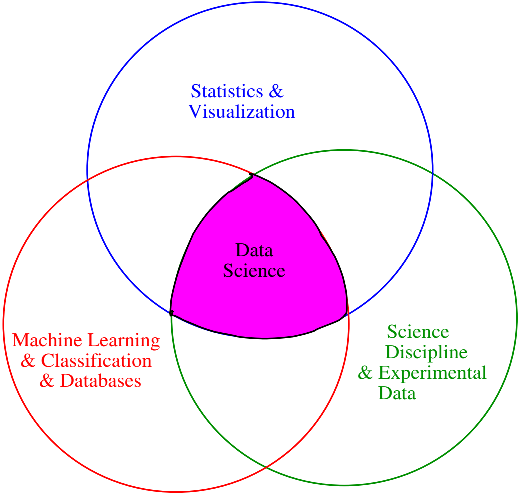
Data Science

Connecting Experiments Back to Theory



Data Sciences

A blend of disciplines



Why Do Scientist Use Computers?

Why Do Scientist Use Computers?

Experiments are impossible
Experiments are too expensive
Equations too difficult to be solved analytically
Experiments don't provide enough insight or accuracy
Data sets too complex to be analyzed by hand

Computers bridge the gap between experiments and theory

Are Algorithms Important?

Which is more important?

1. An efficient algorithm
2. A fast computer

FFT Algorithms

- The FFT algorithms changed the computational cost of calculation Fourier Transforms from $O(n^2)$ to $O(n \log n)$
- The particle mesh method and the hierarchical tree code changed the cost of solving n-body methods from $O(n^2)$ to $O(n \log n)$
- Assume it takes 1 second to calculate the Fourier transform of a given size. Approximately how much longer will it take to calculate the Fourier transform of a problem one thousand times larger?

Sorting Example

Human Sort

How long does it take you to sort the following list?

0.512514	0.978670
0.685045	0.475133
0.460972	0.872282
0.856403	0.874345
0.865597	0.071149
0.383721	0.298935
0.395536	0.771029
0.379287	0.744434
0.281189	0.034619
0.563337	0.282640

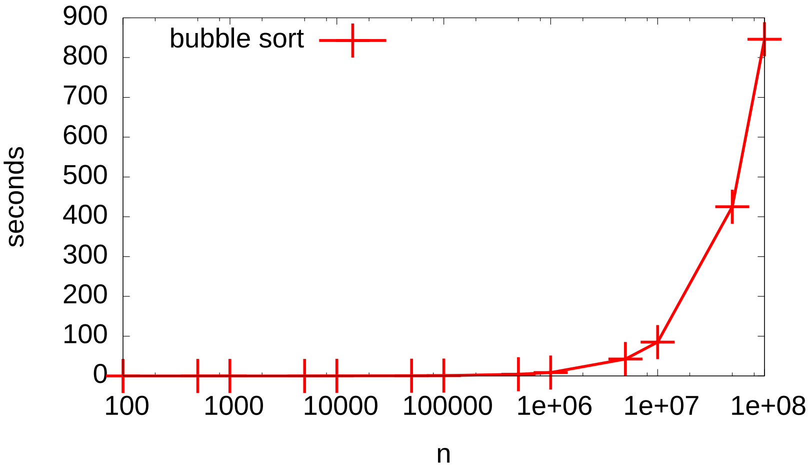
Sorting Example

Bubble Sort in Octave

```
n = [100, 500, 1000, 5000, 10000, 50000, 100000,
    500000, 1000000];
for j = n
  j
  st = time;
  a = rand(j,1);
  index1 = 1;
  while (index1 < j)
    for index2 = index1:n
      if (a(index1) > a(index2) )
        tmp = a(index1);
        a(index1) = a(index2);
        a(index2) = tmp;
      end
    end
    index1 = index1 + 1;
  end
  et = time;
  timeinterval = et - st
end
```

Sorting Example

Bubble Sort in Octave



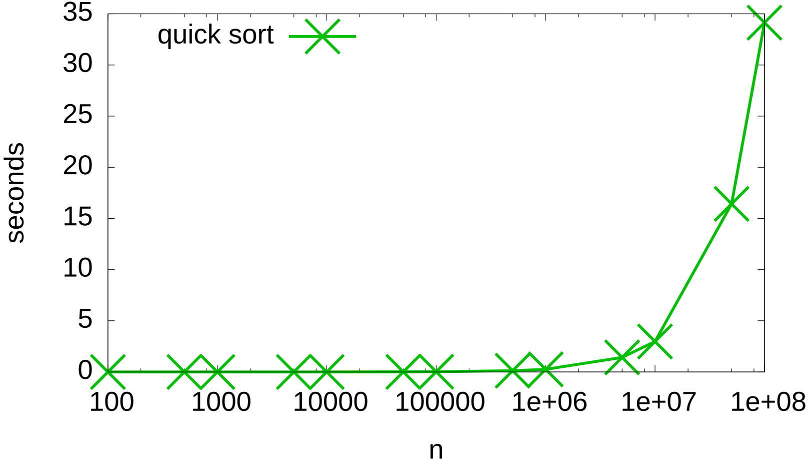
Sorting Example

Quicksort in Octave

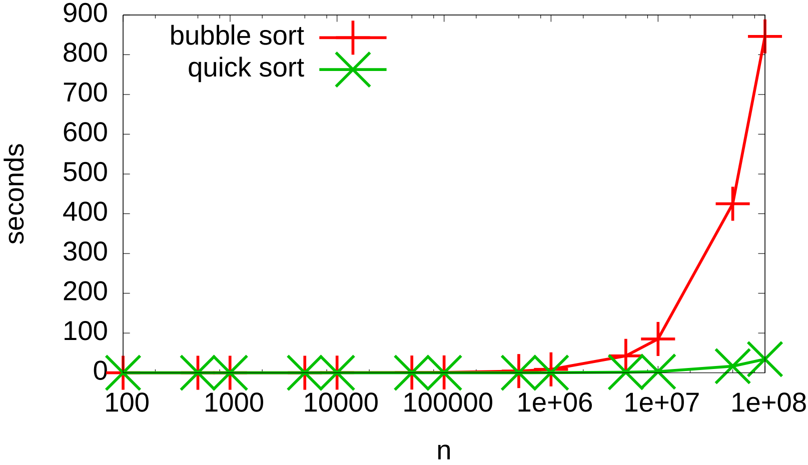
```
n = [100, 500, 1000, 5000, 10000, 50000, 100000,
    500000, 1000000];
for j = n
  j
  st = time;
  a = rand(j,1);
  b = sort(a);
  et = time;
  timeinterval = et - st
end
```

Sorting Example

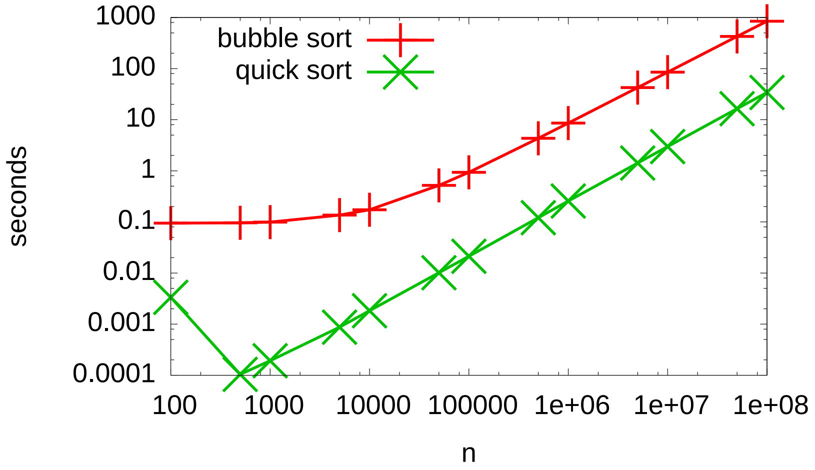
Quicksort in Octave



Sorting Example



Sorting Example



Algorithms Matter!

Algorithms Matter!

http://www.youtube.com/v/k4RRi_ntQc8

(Senator Obama Knows His Algorithms!)

The Atanasoff-Berry Computer

The earliest electronic digital computer was built in the basement of the Physics Department at Iowa State University by Atanasoff and Berry in 1937—1942.

It was a special purpose machine that was used to solve a 27x27 element linear equation. Even though its programming was limited to a single task, it contained all the elements (storage, digital logic, base-2) of modern machines.

Although this seems like a trivial problem now, solving this type of matrix problem is extremely difficult without a computer.

Supercomputer Speeds

Year	Computer	Speed	CPU
1947	Eniac	500 FLOPS	1
1955	IBM 704	10 kFLOPS	1
1964	CDC 6600	1.2 MFLOPS	1
1976	Cray 1	12.5 MFLOPS	1
1988	Cray Y-MP	2 GFLOPS	16
1993	Fujitsu Numerical Wind Tunnel	124.5 GFLOP	140
2000	ASCI Red (Sandia)	2.4 TFLOP	9632
2005	BlueGene/L (DOE/NNSA/LLNL)	280 TFLOP	131,072
2008	Road Runner (DOE/NNSA/LLNL)	1026 TFLOP	122,400

(Data from `http://home.earthlink.net/~mrob/pub/computer-history.html` and `http://www.top500.org`)

Supercomputer Speeds

Year	Computer	Speed	CPU
1947	Eniac	500 FLOPS	1
1955	IBM 704	10 kFLOPS	1
1964	CDC 6600	1.2 MFLOPS	1
1976	Cray 1	12.5 MFLOPS	1
1988	Cray Y-MP	2 GFLOPS	16
1993	Fujitsu Numerical Wind Tunnel	124.5 GFLOP	140
2000	ASCI Red (Sandia)	2.4 TFLOP	9632
2005	BlueGene/L (DOE/NNSA/LLNL)	280 TFLOP	131,072
2008	Road Runner (DOE/NNSA/LLNL)	1026 TFLOP	122,400
2011	Sequoia (DOE/NNSA/LLNL)	20 PFLOP	1,572,864

(Data from `http://home.earthlink.net/~mrob/pub/computer-history.html` and `http://www.top500.org`)

Supercomputer Speeds

Year	Computer	Speed	CPU
1947	Eniac	500 FLOPS	1
1955	IBM 704	10 kFLOPS	1
1964	CDC 6600	1.2 MFLOPS	1
1976	Cray 1	12.5 MFLOPS	1
1988	Cray Y-MP	2 GFLOPS	16
1993	Fujitsu Numerical Wind Tunnel	124.5 GFLOP	140
2000	ASCI Red (Sandia)	2.4 TFLOP	9632
2005	BlueGene/L (DOE/NNSA/LLNL)	280 TFLOP	131,072
2008	Road Runner (DOE/NNSA/LLNL)	1026 TFLOP	122,400
2011	Sequoia (DOE/NNSA/LLNL)	20 PFLOP	1,572,864
2012	Titan (DOE/SC/Oak Ridge)	27 PFLOP	560,640

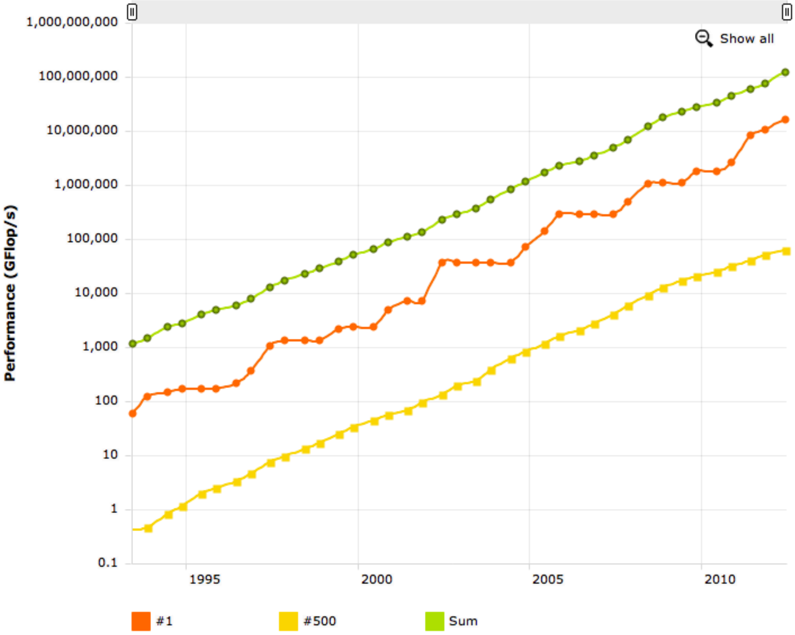
(Data from `http://home.earthlink.net/~mrob/pub/computer-history.html` and `http://www.top500.org`)

Supercomputer Speeds

Year	Computer	Speed	CPU
1947	Eniac	500 FLOPS	1
1955	IBM 704	10 kFLOPS	1
1964	CDC 6600	1.2 MFLOPS	1
1976	Cray 1	12.5 MFLOPS	1
1988	Cray Y-MP	2 GFLOPS	16
1993	Fujitsu Numerical Wind Tunnel	124.5 GFLOP	140
2000	ASCI Red (Sandia)	2.4 TFLOP	9632
2005	BlueGene/L (DOE/NNSA/LLNL)	280 TFLOP	131,072
2008	Road Runner (DOE/NNSA/LLNL)	1026 TFLOP	122,400
2011	Sequoia (DOE/NNSA/LLNL)	20 PFLOP	1,572,864
2012	Titan (DOE/SC/Oak Ridge)	27 PFLOP	560,640
2015	MilkyWay-2 (NUDT, China)	54 PFLOP	3,120,000

(Data from `http://home.earthlink.net/~mrob/pub/computer-history.html` and `http://www.top500.org`)

Supercomputer Speeds



Operating Systems

What percentage share do each of the major operating systems have in the Top 500 list?

OS Family	Count	System Share %	Rpeak (GFlops)	Cores

Operating Systems

What percentage share do each of the major operating systems have in the Top 500 list?

OS Family	Count	System Share %	Rpeak (GFlops)	Cores
Linux	476	95.2	217,932,444	18,700,112

Operating Systems

What percentage share do each of the major operating systems have in the Top 500 list?

OS Family	Count	System Share %	Rpeak (GFlops)	Cores
Linux	476	95.2	217,932,444	18,700,112
Unix	16	3.2	3,949,373	181,120

Operating Systems

What percentage share do each of the major operating systems have in the Top 500 list?

OS Family	Count	System Share %	Rpeak (GFlops)	Cores
Linux	476	95.2	217,932,444	18,700,112
Unix	16	3.2	3,949,373	181,120
Mixed	4	0.8	1,184,521	417,792
Windows	3	0.6	465,600	46,092
BSD Based	1	0.2	122,400	1,280

Operating Systems

What percentage share do each of the major operating systems have in the Top 500 list?

	June 2011		June 2012		June 2013		June 2014	
System	%	#	%	#	%	#	%	#
Linux	94.0	470	94.8	474	95.2	476	97.0	485
Unix	4.6	23	4.8	24	3.2	16	2.4	12
Windows	1.2	6	0.4	2	0.6	3	0.4	2

(http://www.top500.org/statistics/overtime/)

Linux

OS of Choice in Computational Science

Linux

OS of Choice in Computational Science

- Free, open source, and very stable
- Low OS overhead
- Extensive freeware and commercial development and analysis tools
- Easy access to low level analysis tools
- Excellent control of machine resources
- Works well, even with petascale computing

Using Linux for class

We'll be using `herschel.cs.mtsu.edu` for class.

- Accessible from Windows machines via `ssh` using PuTTY (www.putty.org)
- A virtual machine (e.g., VirtualBox, then `ssh`)
- X2Go (either web or desktop clients)
  - `http://wiki.x2go.org/doku.php/doc:installation:x2goclient`
- A linux machine via `ssh`

Access to Linux Machines

- Go to `http://www.cs.mtsu.edu/`
- Select CS Account Management from the menu on the right side
- Log with your pipeline account
- For account purpose - use "COMS 6100 and PhD program"

Virtual Machines

Virtual Machines

- A program that mimics a machine
- Programs running within this program behave like they are running on a different machine

VirtualBox (by Oracle):

- Free for academic and personal use
- Many versions for different operating systems
- `http://www.virtualbox.org`

Installing a Virtual Machine on Your Computer

- Download the VirtualBox software appropriate for your computer, and install it
- Download or load an ISO image of your favorite operating system to your hard drive
  - Ubuntu 32 bit works great
- Start up the VirtualBox Software - click new and follow the prompts in the Virtual Machine Wizard
  - You will want $>$ 50 Gb of disk space and $>$1Gb of RAM
- Select the ISO image within the Wizard and install it in your Virtual Machine

Installing a Virtual Machine (2)

- Boot the VM and log in
- Update the VM with any OS patches
- Add the Virtual Box additions to let the system go full screen
- Add any additional software you want - compilers, editors, etc using a package manager

Using X2Go to access `herschel.cs.mtsu.edu`

Two methods to access `herschel.cs.mtsu.edu`:

1. Web client
2. Downloaded client (Macs need the beta client)

Using X2Go to access `herschel.cs.mtsu.edu`

Two methods to access `herschel.cs.mtsu.edu`:

1. Web client (easier, Windows only)
2. Downloaded client (Macs need the beta client)

Basic Unix Skills

Basic Unix Skills

1. Creating and moving files and directories
2. Editing files
3. Compressing & decompressing files and directories
4. File/directory permissions
5. Modifying your path
6. Using find and grep (or similar tools) for basic tasks
7. Some scripting language (perl, python, etc.)

Many tutorials are available - but you MUST learn the command line interface

hwkLinux1

https://www.cs.mtsu.edu/~hcarroll/6100/assignments/hwkLinux1.html