18-613 Lecture 2:

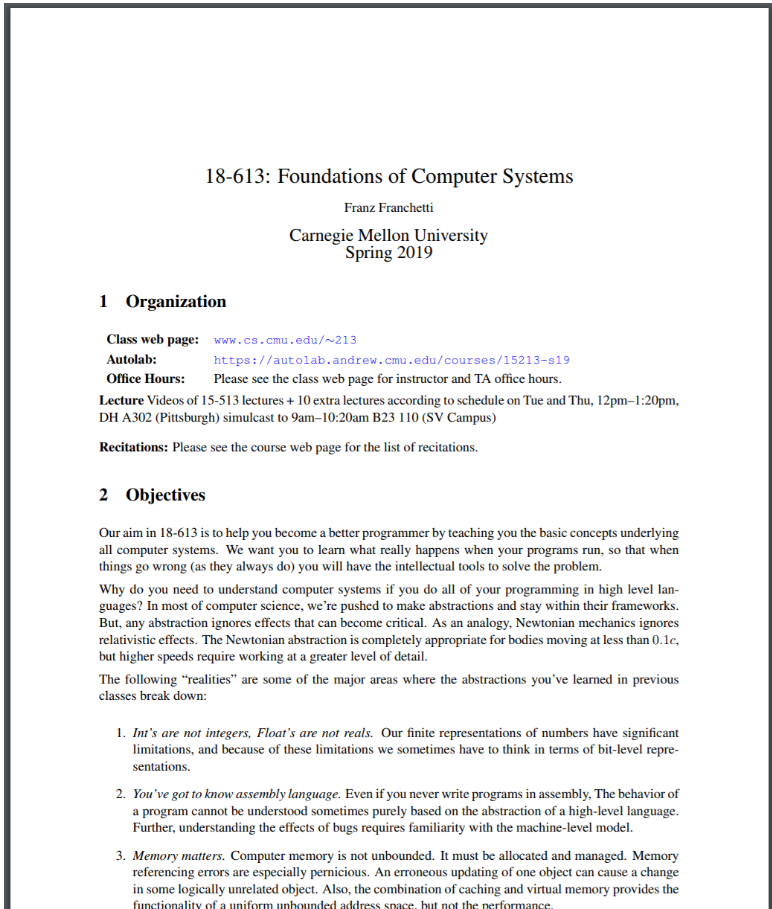
State-of-the Art in Computing

January 22, 2019

Franz Franchetti
Instructor

Questions from Piazza

- 18-613 Syllabus is the authorative source for information http://www.cs.cmu.edu/afs/cs/academic/class/15213-s19/www/syllabus613/syllabus.pdf
- Labs: 50%, Exams: 50%
- Extra exam question beyond 15-513
- Extra Credit lab LX (ArchLab)



18-613: Foundations of Computer Systems

Franz Franchetti

Carnegie Mellon University
Spring 2019

1 Organization

Class web page: www.cs.cmu.edu/~213
Autolab: https://autolab.andrew.cmu.edu/courses/15213-s19
Office Hours: Please see the class web page for instructor and TA office hours.

Lecture Videos of 15-513 lectures + 10 extra lectures according to schedule on Tue and Thu, 12pm–1:20pm, DH A302 (Pittsburgh) simulcast to 9am–10:20am B23 110 (SV Campus)

Recitations: Please see the course web page for the list of recitations.

2 Objectives

Our aim in 18-613 is to help you become a better programmer by teaching you the basic concepts underlying all computer systems. We want you to learn what really happens when your programs run, so that when things go wrong (as they always do) you will have the intellectual tools to solve the problem.

Why do you need to understand computer systems if you do all of your programming in high level languages? In most of computer science, we're pushed to make abstractions and stay within their frameworks. But, any abstraction ignores effects that can become critical. As an analogy, Newtonian mechanics ignores relativistic effects. The Newtonian abstraction is completely appropriate for bodies moving at less than $0.1c$, but higher speeds require working at a greater level of detail.

The following "realities" are some of the major areas where the abstractions you've learned in previous classes break down:

1. *Int's are not integers, Float's are not reals.* Our finite representations of numbers have significant limitations, and because of these limitations we sometimes have to think in terms of bit-level representations.
2. *You've got to know assembly language.* Even if you never write programs in assembly, The behavior of a program cannot be understood sometimes purely based on the abstraction of a high-level language. Further, understanding the effects of bugs requires familiarity with the machine-level model.
3. *Memory matters.* Computer memory is not unbounded. It must be allocated and managed. Memory referencing errors are especially pernicious. An erroneous updating of one object can cause a change in some logically unrelated object. Also, the combination of caching and virtual memory provides the functionality of a uniform unbounded address space, but not the performance.

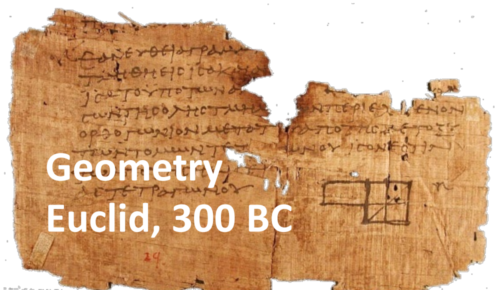
Difference from 15-213/18-213/15-513

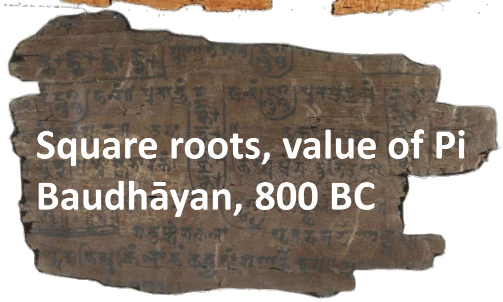
- NO Problem Sets
- NO In-Class Quizzes
- NO Bonus points

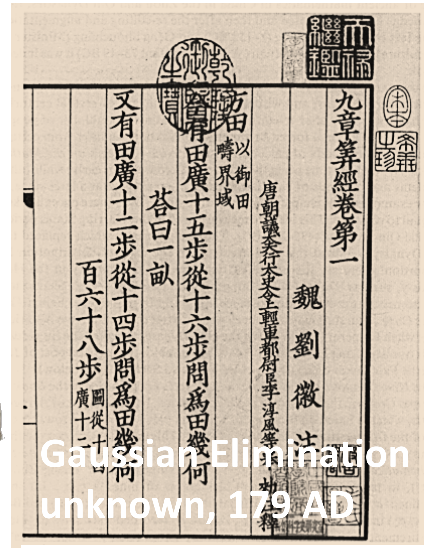
Careful: Look at the correct syllabus

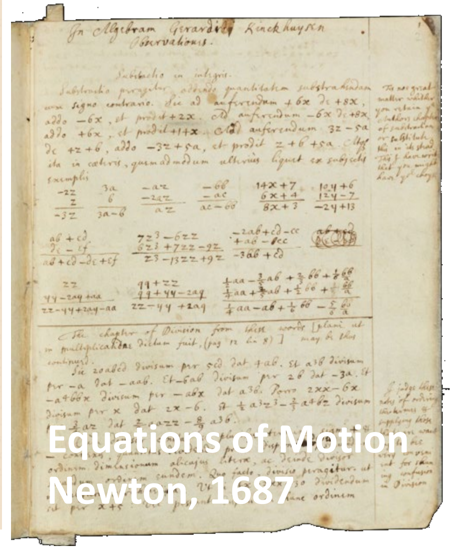
Lab 0 ?

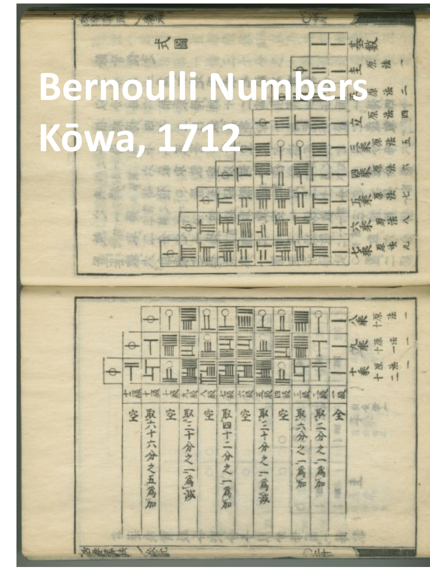
Algorithms and Mathematics: 2,500+ Years



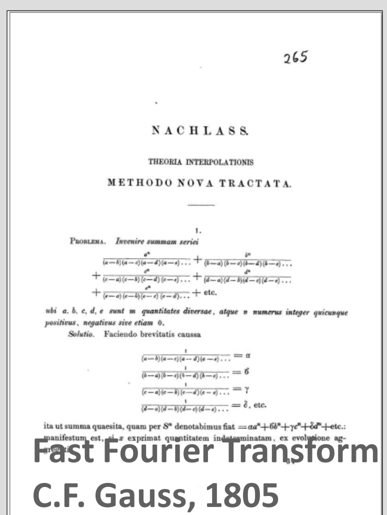


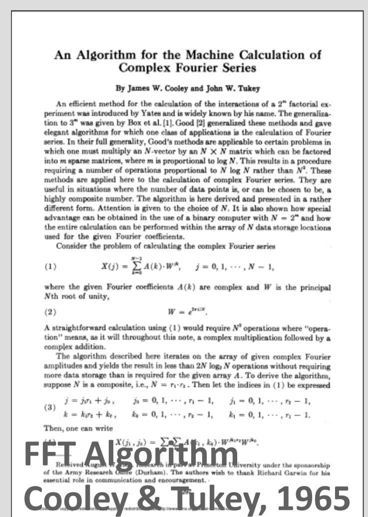


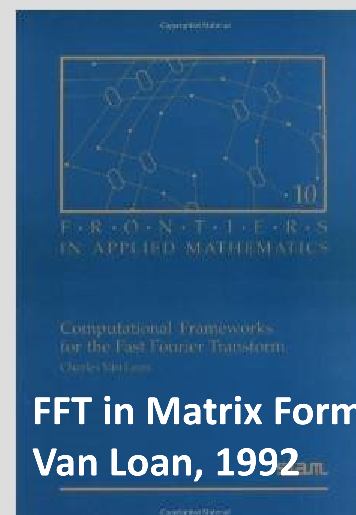




Fast Fourier Transform







Computers I have Used: 10^{10}x Gain in 35 Years

The first computer I...

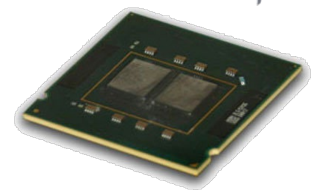
"My" first...

10 kflop/s

...programmed
Commodore VIC20
1MHz MOS 6502
5 kB RAM, TV
1985

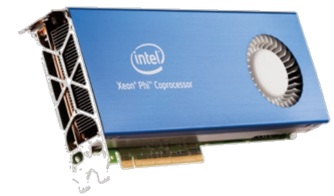
...owned
IBM PC/XT compatible
8088 @ 8 MHz, 640kB RAM
360 kB FDD, 720x348 mono,
1989

...telnet'ed into
IBM RS/6000-390
256 MB RAM, 6GB HDD
67 MHz Power2+, AIX
1994



...multicore
Pentium D
2 cores, 3.6 GHz
2/4-way SIMD
2005

...GPGPU
GeForce 8800
1.3 GHz, 128 shaders
16-way SIMT
2006



...manycore
Xeon Phi
1.3 GHz, 60 cores
8/16-way SIMD
2011

Computers I use, circa 2019

110 Tflop/s FP16 (ML)

Summit
2,282,544 cores @ 3.07 GHz
200 Pflop/s, #1 in Top500

Dell Power Edge
80 cores @ 3GHz
3 TB RAM

Dell Precision 3620
3.7 GHz Xeon Quad-core
Nvidia TITAN V, 64 GB RAM

Lenovo X270
2.8 GHz Core i7 Dual-core
Mobile GPU, 16GB RAM

Sony Xperia XZ1
2.5 GHz Octa-core
Mobile GPU, 4GB RAM

1 Gflop/s = one billion floating-point operations (additions or multiplications) per second

Computing Platforms Over The Years

F-16A/B, C/D, E/F, IN, IQ, N, V

B52 Stratofortress

Compare: Desktop/workstation class CPUs/machines

Assembly code compatible !!

x86 binary compatible, but 500x parallelism ?!

1972
Intel 8008
0.2—0.8 MHz
Intelligent terminal

1989
IBM PC/XT compatible
8088 @ 8 MHz, 640kB RAM
360 kB FDD, 720x348 mono

1994
IBM RS/6000-390
256 MB RAM, 6GB HDD
67 MHz Power2+, AIX

2006
GeForce 8800
1.3 GHz, 128 shaders
16-way SIMT

2011
Xeon Phi
1.3 GHz, 60 cores
8/16-way SIMD

2019
Xeon Platinum 8180M
28 cores, 2.5-3.6 GHz
2/4/8/16-way SIMD

x86 ISA: hiding 10^7 – 10^8 compounded performance gain over 45 years

Programming/Languages Libraries Timeline

Popular performance programming languages

- 1953: Fortran
- 1973: C
- 1985: C++
- 1997: OpenMP
- 2007: CUDA
- 2009: OpenCL

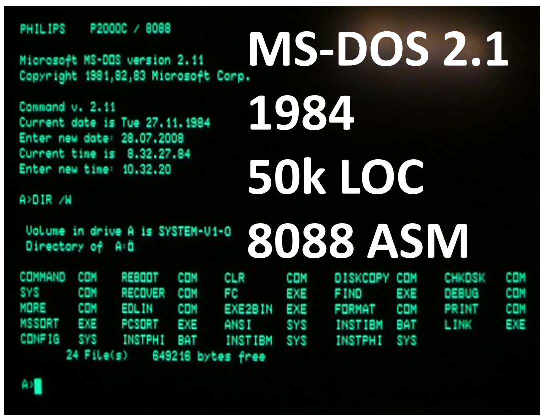
Popular performance libraries

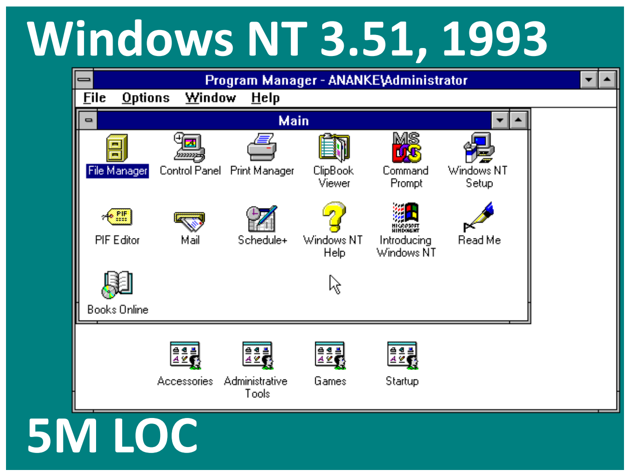
- 1979: BLAS
- 1992: LAPACK
- 1994: MPI
- 1995: ScaLAPACK
- 1995: PETSc
- 1997: FFTW

Popular productivity/scripting languages

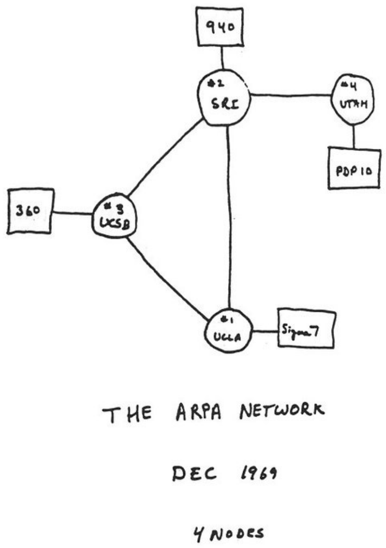
- 1987: Perl
- 1989: Python
- 1993: Ruby
- 1995: Java
- 2000: C#

Growth of Source Code





From ARPA Net to the Cloud



50 years…

Giving Moore's Law a Face

1 flop/s = one floating-point operation (addition or multiplication) per second
mega (M) = 10^6, giga (G) = 10^9, tera (T) = 10^{12}, peta (P) = 10^{15}, exa (E) = 10^{18}

In 2019…

Cell phone
15 Gflop/s

Laptop
90 Gflop/s

Workstation
1.5 Tflop/s (CPU)
170 Tflop (GPU)

HPC
80 Tflop/s (CPU)
1.7 Pflop/s (GPU)

#1 supercomputer
200 Pflop/s

…would have been the #1 supercomputer back in…

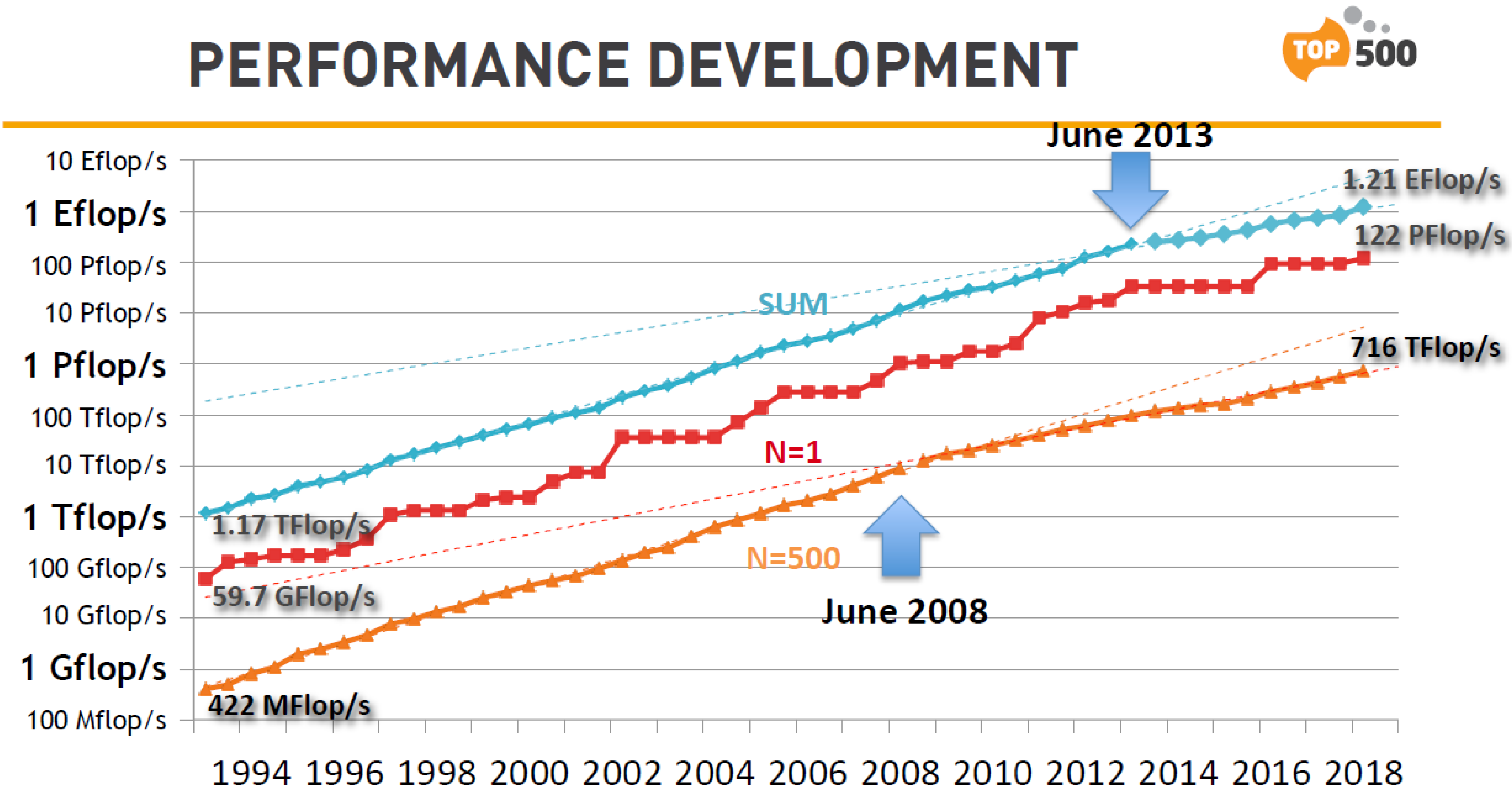
Cray Y-MP C90
16 Gflop/s
1991

CM-5/1024
131 Gflop/s
1993

Intel ASCI Red
1.338 Tflop/s
1997

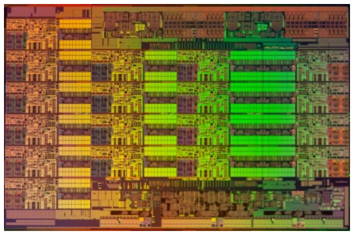
BlueGene/L
70.72 Tflop/s
2004

The Evolution of Performance: Top500.org

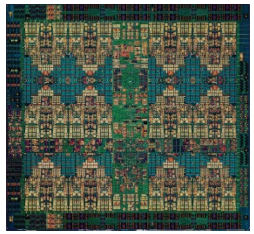


Today’s Computing Landscape

1 Gflop/s = one billion floating-point operations (additions or multiplications) per second

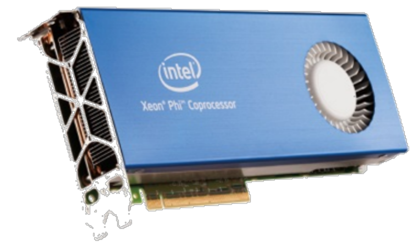


Intel Xeon 8180M
2.25 Tflop/s, 205 W
28 cores, 2.5—3.8 GHz
2-way—16-way AVX-512



IBM POWER9
768 Gflop/s, 300 W
24 cores, 4 GHz
4-way VSX-3

Nvidia Tesla V100
7.8 Tflop/s, 300 W
5120 cores, 1.2 GHz
32-way SIMT



Intel Xeon Phi 7290F
1.7 Tflop/s, 260 W
72 cores, 1.5 GHz
8-way/16-way AVX512

Snapdragon 835
15 Gflop/s, 2 W
8 cores, 2.3 GHz
A540 GPU, 682 DSP, NEON

Intel Atom C3858
32 Gflop/s, 25 W
16 cores, 2.0 GHz
2-way/4-way SSSE3

Dell PowerEdge R940
3.2 Tflop/s, 6 TB, 850 W
4x 24 cores, 2.1 GHz
4-way/8-way AVX

Summit
200 Pflop/s, 13 MW
9,216 x 22 cores POWER9
+ 27,648 V100 GPUs

What Is High Performance Computing?

1 flop/s = one floating-point operation (addition or multiplication) per second
mega (M) = 10^6, giga (G) = 10^9, tera (T) = 10^{12}, peta (P) = 10^{15}, exa E) = 10^{18}

Computing systems in 2019 available to ECE students

Your laptop
4 cores
100 Gflop/s

Your desktop
48 cores + GPU
1.2 Tflop/s +
7.5 Tflop/s GPU

ECE Cluster
1k cores + GPUs
32 Tflop/s

PSC HPC
11k cores
200 GPUs
21.35 Pflop/s

amazon web services
HTCondor
Windows Azure
Google Cloud Platform

Cloud
1,000s of cores
many small
machines

18-613 Labs ⟷ High Performance Computing

Your own machines
- Laptop: quad core, 10s of GB
- Desktop: 1—2 sockets, 6—28 cores/socket, GPU, 64 GB or more

Larger than your own machines
- Large shared memory machines: up to 12 TB
- Parallel HPC machines: up to 10k cores
- Cloud and high throughput systems

What Problems Can You Solve?

1 flop/s = one floating-point operation (addition or multiplication) per second
mega (M) = 10^6, giga (G) = 10^9, tera (T) = 10^{12}, peta (P) = 10^{15}, exa (E) = 10^{18}

Matrix-matrix multiplication…

+= ×

```
for i=1:n
  for j=1:n
    for k=1:n
      C[i,j] +=
        A[i,k]*B[k,j]
```

…running on…

Cell phone
15 Gflop/s
12k × 12k
3.5 GB, 4 min

Laptop
90 Gflop/s
25k × 25k
14 GB, 5min

Workstation
1.5 Tflop/s
50k × 50k
64 GB, 3min

HPC
80 Tflop/s
1.5M × 1.5M
80 TB, 22h

#1 supercomputer
200 Pflop/s
10M × 10M
2.8 PB, 1.8h

Looking into the Future: In 2009…

…the performance of the #1 supercomputer of 2009…

#1 supercomputer
1 Pflop/s

…could be available as

HPC
1 Pflop/s
2018

Workstation
1 Pflop/s
2023

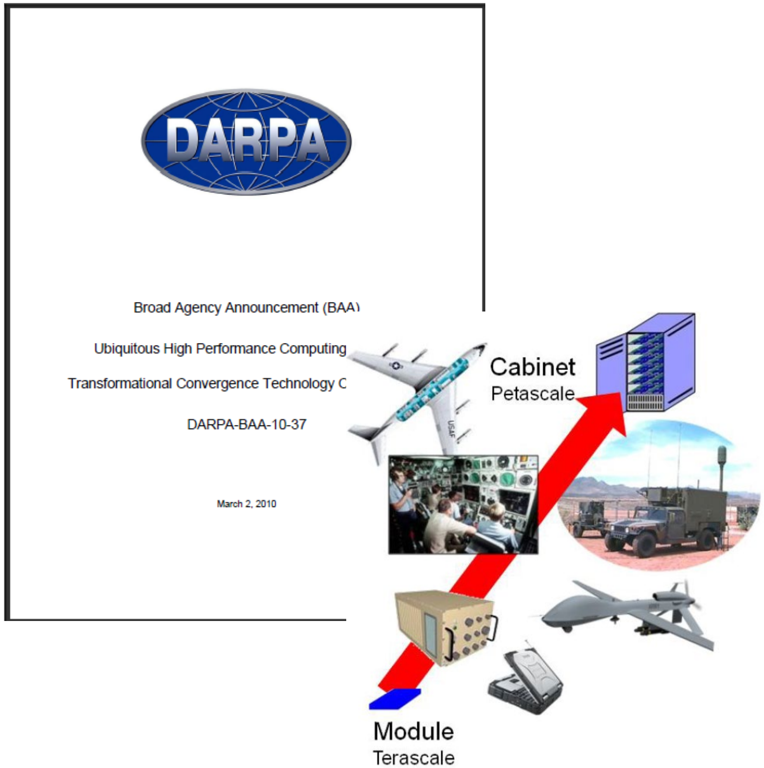
Laptop
1 Pflop/s
2030

Cell phone
1 Pflop/s
2036

We are actually ahead of the prediction. Surprising every time I see this.

Dream to Reality: Petaflop in a Rack

In 2010: hard research goal
DARPA UHPC ExtremeScale



Goal: In 2018 have
1 Pflop/s in 57kW, single rack, air cooled

In 2019: commercially available
2 Pflop/s, 10U, 10kW

Reality: 04/2018
Nvidia DGX-2: 2 Pflop/s
16 V100 GPUs, 2x24-core Xeon
1.5 TB CPU, 512 GB GPU
2 Pflop/s, 10 kW in ¼ rack

Grand Callenges, 50 Years Apart…

Name your project carefully…

2019: Computing on your Wrist

Display: 1.2" full circle AMOLED, 390 x 390 (327 ppi)

OS: Wear OS by Google

Processor: Qualcomm Snapdragon Wear 3100

Compatibility: iOS 9.3+ and Android 4.4+

Power: 340 mAh - one full day, extra 4-5 days in Time Only Mode

Memory: 1 GB

Storage: 8 GB

Sensors: Heart rate, Microphone, Altimeter, Accelerometer, Gyroscope, Ambient light

Connectivity: Bluetooth 4.2, Wi-Fi (802.11b/g/n 2.4GHz), GPS, NFC payment

Snapdragon Wear 3100: 1 GB RAM, 8 GB SSD, quad core, 1.2 GHz, Wi-Fi,…

2019: A $1M COTS* System

Dell PowerEdge R940
3.2 Tflop/s, 6 TB, 850 W
4x 24 cores, 2.1 GHz

OSS FSAn-4
200 TB PCIe NVMe flash
80 GB/s throughput

BittWare TeraBox
18M logic elements, 4.9 Tb/sec I/O
8 FPGA cards/16 FPGAs, 2 TB DDR4

AberSAN ZXP4
90x 12TB HDD, 1 kW
1PB raw

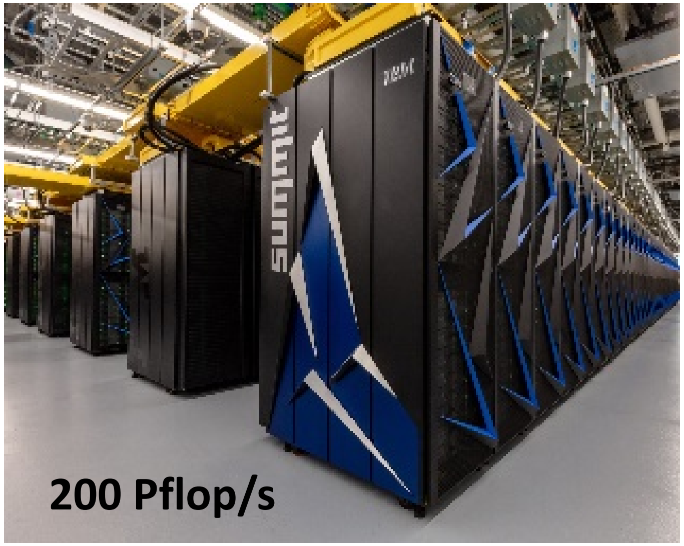
Nvidia DGX-1
8x Tesla V100, 3.2 kW
170 Tflop/s FP64, 1Pflop/s FP16,
128 GB

*COTS = Commercial off-the-shelf

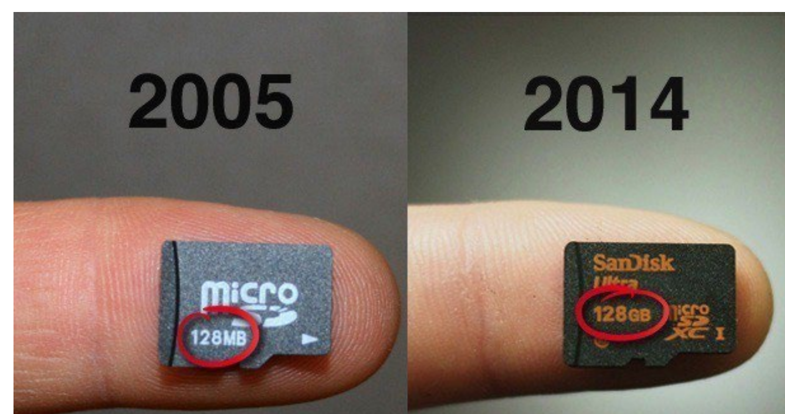
At the Top End: 138x in 9 Years…

#1 supercomputer of 2009: RoadRunner

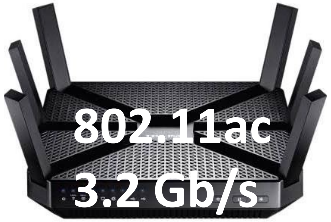
#1 supercomputer of 2018…

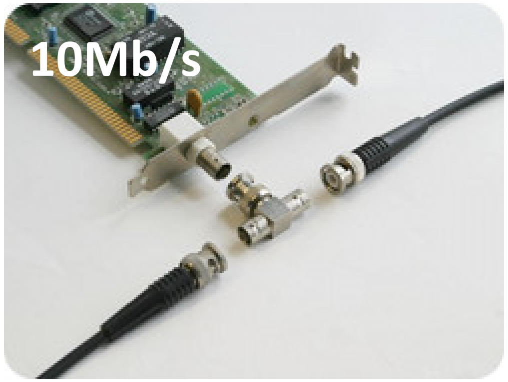


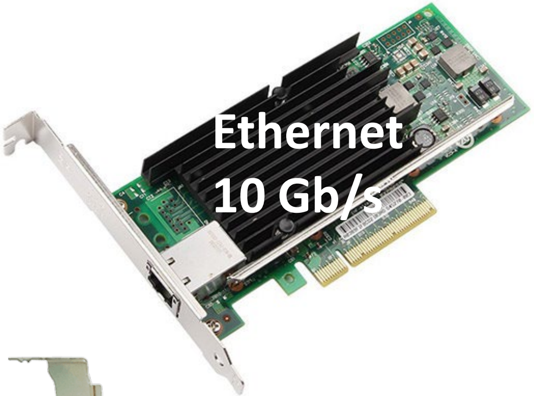
Storage: Big Data

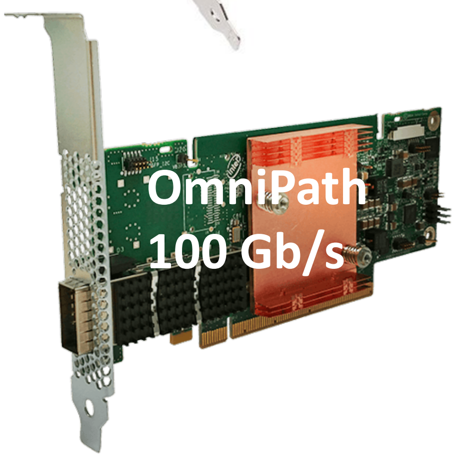


Communication: Bandwidth

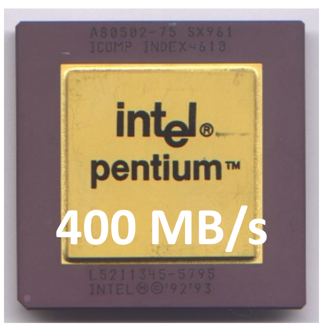






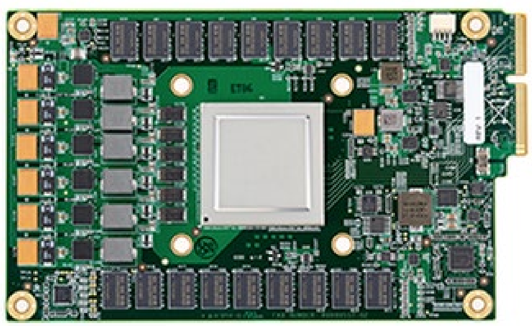


Memory Bandwidth

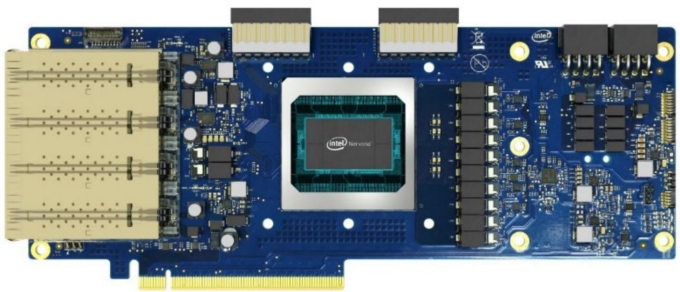


Accelerators

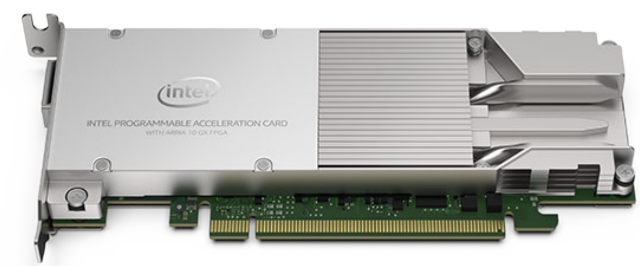
Google TPU 2.0
600 GB/s, 45 Tflop/s



Intel Nervana Neural Network Processor
aka Lake Crest, release date 2019



Nvidia V100 GPU
100 Tflop/s tensor operations
7.5 Tflop/s FP64 SIMT

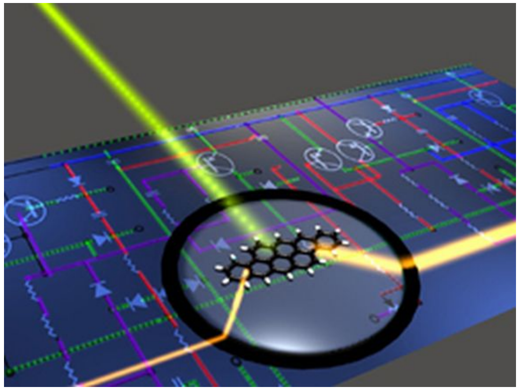


Intel Arria 10 GX FPGA
15Gbps SERDES transceivers
1,150K logic elements, 53 Mb
8 GB DDR4 ECC

Remember: A'int Talkin' 'Bout 2055…*

Quantum computing

DNA computing



Optical computing

Other far-out technologies

* With apologies to Van Halen

Gordon Bell 2016: Climate Simulation

770B unknowns, 7.95 Pflop/s, 488m resolution,
0.07 simulated-years/compute-day

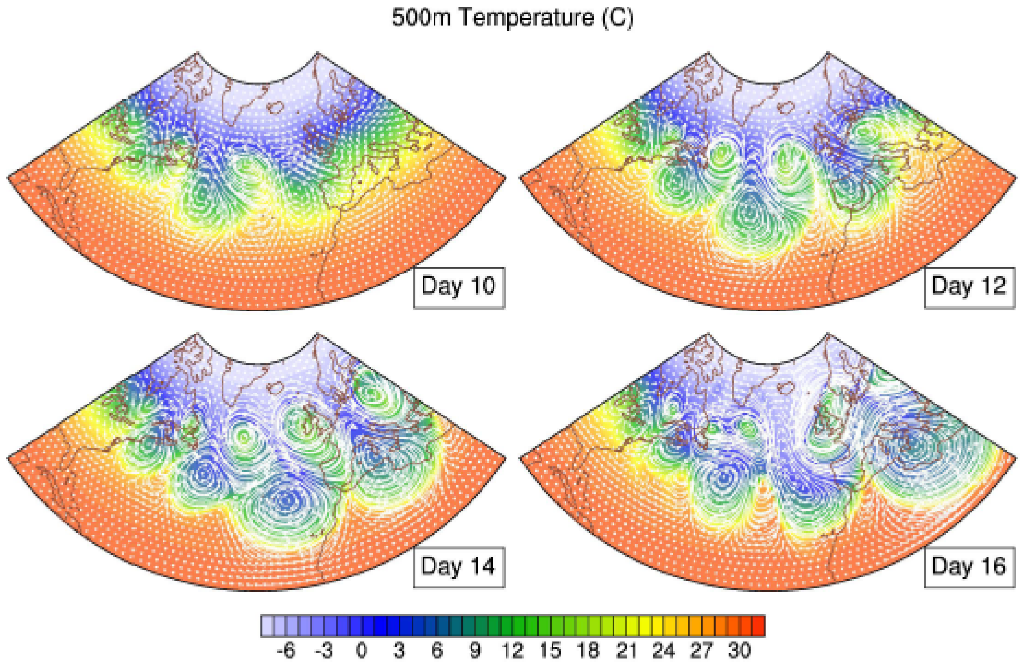
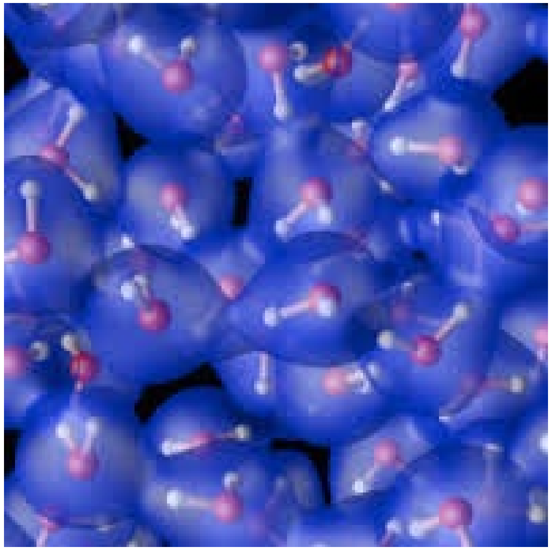


Fig. 5. The 500 m level simulation results at day 10, 12, 14 and 16 for the baroclinic instability test. Shown here are the temperature contours with the overlaid horizontal wind field, for which only around 1/3 of the whole computational domain is drawn to see more details of the baroclinc jets and eddies.

Gordon Bell 2006: Material Science

First principle molecular dynamics
1,000 molybdenum atoms, 12,000 electrons, 108 Tflop/s



2013 Science Run on BlueGene/Q

Simulate universe since Big Bang, trillions of particles, billions of years

Large Scale Structure Simulation Requirements

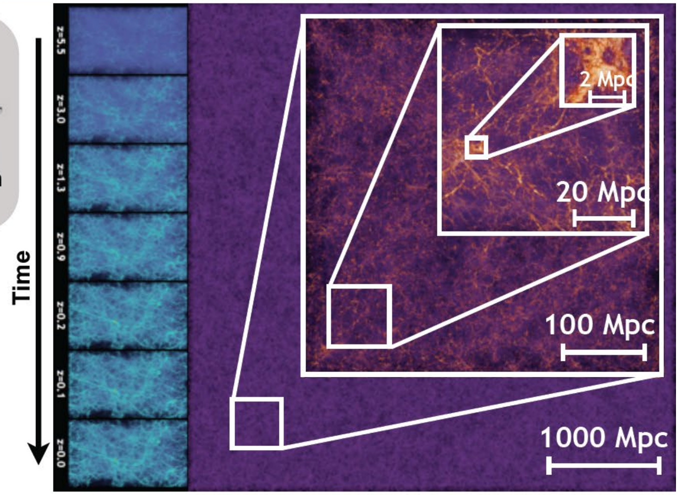
$$\frac{\partial f_i}{\partial t} + \dot{\mathbf{x}}\frac{\partial f_i}{\partial \mathbf{x}} - \nabla\phi\frac{\partial f_i}{\partial \mathbf{p}} = 0, \qquad \mathbf{p} = a^2\dot{\mathbf{x}},$$

$$\nabla^2\phi = 4\pi G a^2(\rho(\mathbf{x},t) - \langle\rho_{\rm dm}(t)\rangle) = 4\pi G a^2 \Omega_{\rm dm}\delta_{\rm dm}\rho_{\rm cr},$$

$$\delta_{\rm dm}(\mathbf{x},t) = (\rho_{\rm dm} - \langle\rho_{\rm dm}\rangle)/\langle\rho_{\rm dm}\rangle,$$

$$\rho_{\rm dm}(\mathbf{x},t) = a^{-3}\sum_i m_i \int d^3\mathbf{p} f_i(\mathbf{x},\dot{\mathbf{x}},t).$$

Cosmological Vlasov-Poisson Equation



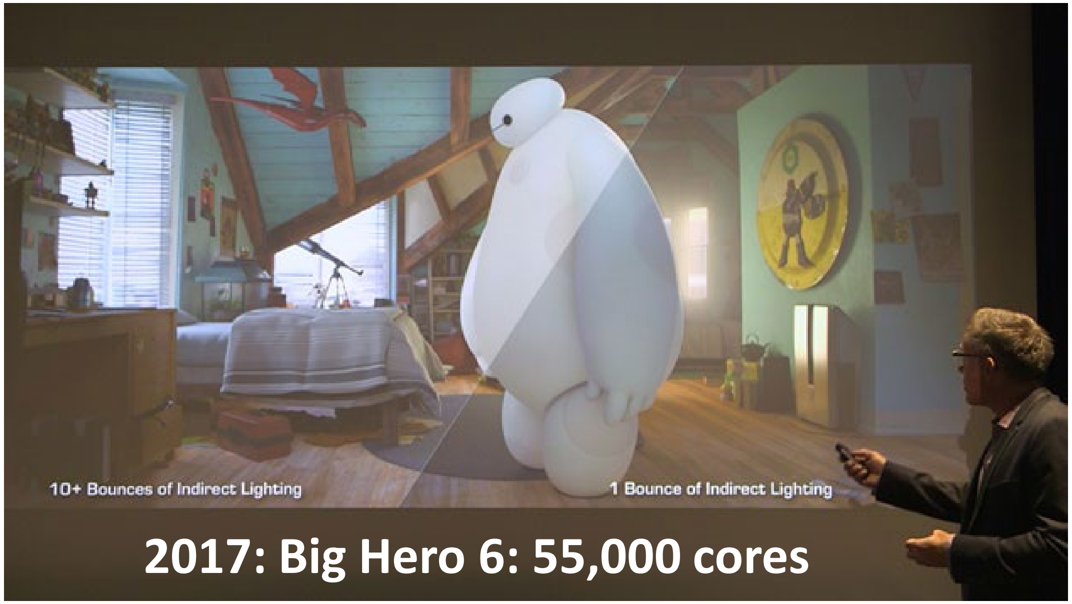
Gravitational Jeans Instablity

- **Resolution:**
  - Force resolution has to be ~kpc, a dynamic range of a million to one, also controls time-stepping
  - Local overdensity variation is ~million to one
- **Physics:**
  - Gravity dominates at scales greater than ~Mpc
  - At small scales: galaxy distribution modeling
- **Computing 'Boundary Conditions':**
  - Total memory in the PB+ class
  - Performance in the 10 PFlops+ class
  - Wall-clock of ~days/week, in situ analysis

Can the Universe be run as a short computational 'experiment'?

Wednesday, May 22, 13

Computer Graphics

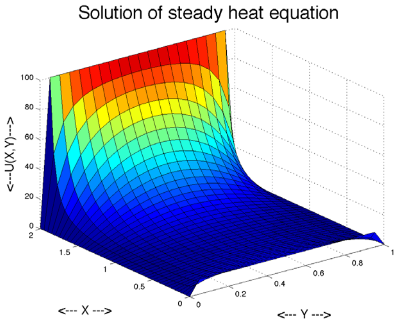


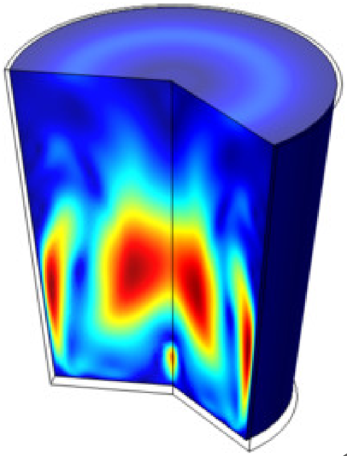
Example	Particles	Grid	min/frame
Snowball drop	3.0×10^5	$600 \times 300 \times 600$	5.2
Snowball smash	3.0×10^5	$200 \times 240 \times 600$	7.3
Double smash	6.0×10^5	$800 \times 300 \times 800$	13.3
Snowplow	3.9×10^6	$150 \times 50 \times 300$	2.1
Rolling snowball	7.2×10^6	$200 \times 240 \times 470$	35.7
SIGGRAPH	7.5×10^5	$780 \times 120 \times 220$	4.7
The end	5.8×10^5	$700 \times 120 \times 210$	3.8
Castle destruction	1.6×10^6	$360 \times 160 \times 560$	6.0
Walking character	3.0×10^6	$370 \times 120 \times 300$	15.7
Character digging	3.1×10^6	$280 \times 110 \times 340$	25.8

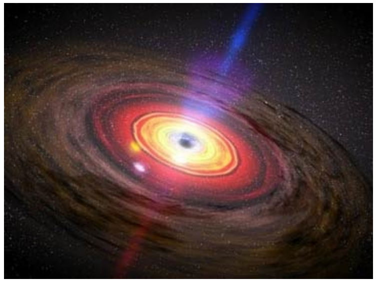
Table 3: *Example particle counts, resolutions and simulation times. Simulations were performed on an 8-core Intel Xeon X5550 2.67GHz machine.*

Self-Driving Cars: Compute, Sensors, etc.

Simulation Scaling







unsolvable

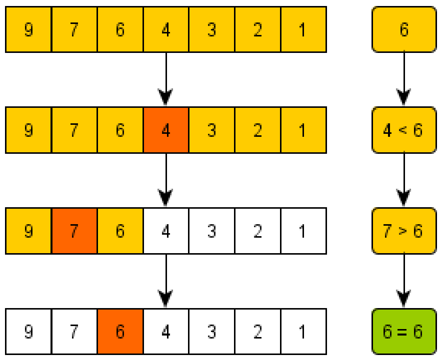
HPC

Laptop/desktop

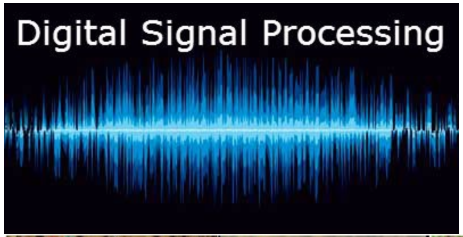
Toy problem

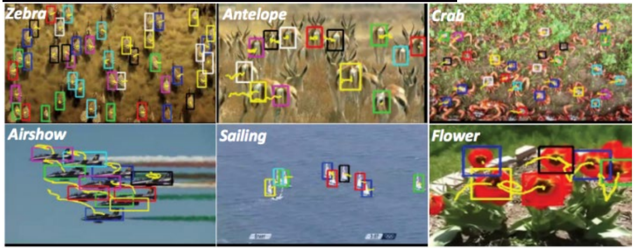
Scalability

Sub-linear

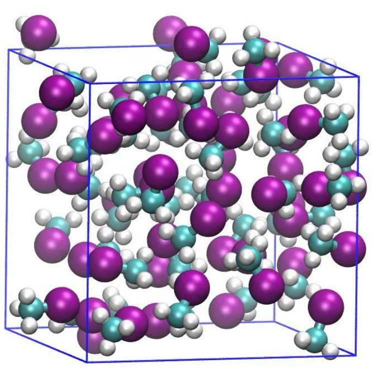


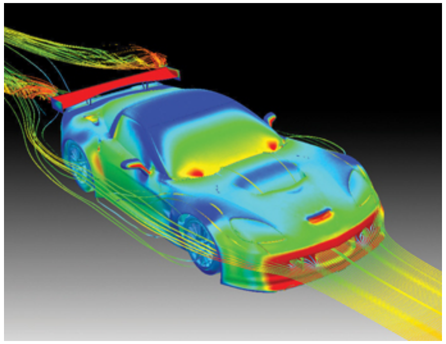
Linear, *N* log *N*





Polynomial

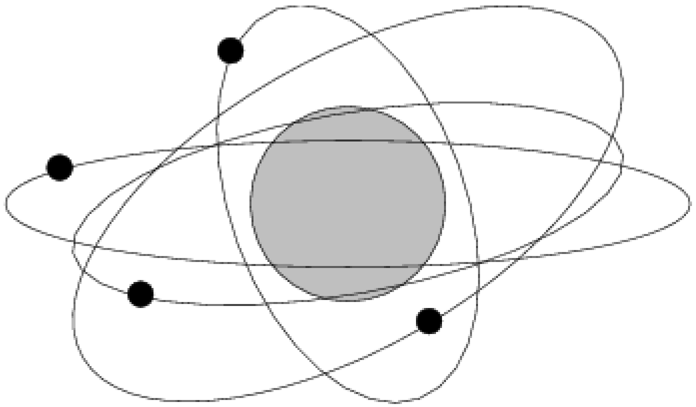




Exponential

Approximations, Models, and Heuristics

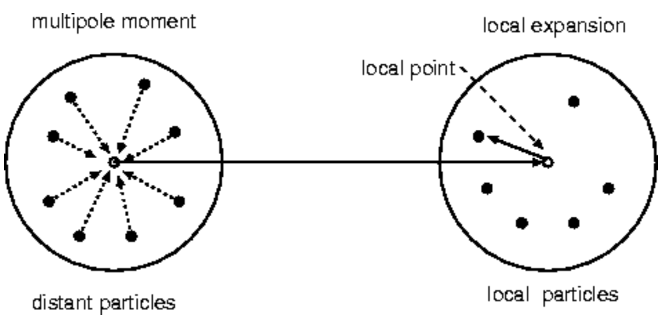
Physics: n-body problem



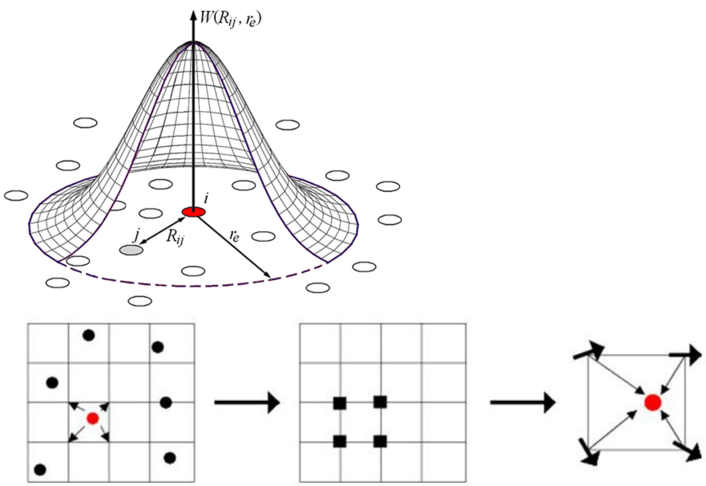
Direct solver: O(n²)

$$U_\varepsilon = \sum_{1\le i<j\le n} \frac{G m_i m_j}{\sqrt{\left\|\mathbf{q}_j - \mathbf{q}_i\right\|^2 + \varepsilon^2}}.$$

Fast multipole: O(n log n)



Particle/mesh: O(k log k)

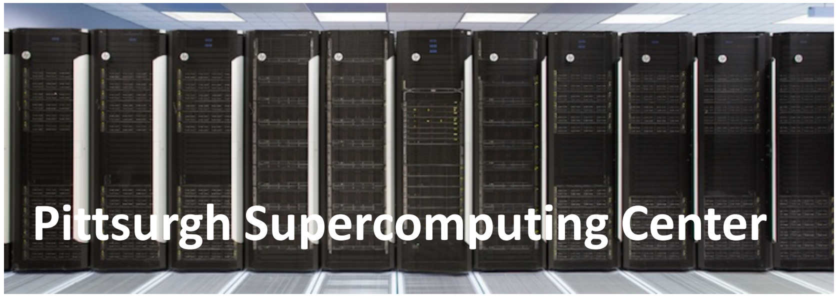


Take Home Lesson

Your laptop = #1 supercomputer from 1993

=

You have access to 10,000x your laptop's power



amazon web services

Google Cloud Platform

Windows Azure

HTCondor High Throughput Computing