# CS267 / E233
# Applications of Parallel Computers

**https://sites.google.com/lbl.gov/cs267-spr2019/**

**Video of lectures available the day after classroom lecture**

# Lecture 1: Introduction

Kathy Yelick
EECS Department
yelick@berkeley.edu

# Course Staff

**Instructors**

Kathy Yelick
EECS and LBNL

Aydın Buluç
EECS and LBNL

Jim Demmel
EECS

**GSIs**

Ben Brock
EECS and LBNL

Kevin Laeufer
EECS

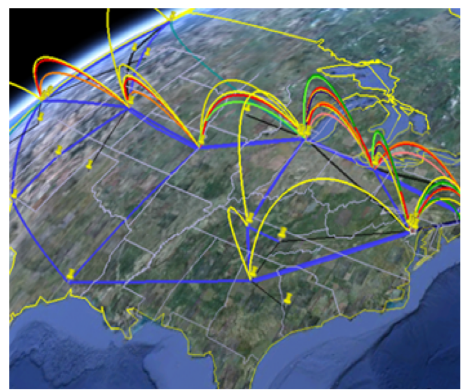
*See website for contact information and office hours. Email us at* cs267.sp19@gmail.com

# My day job

Computing Sciences at LBNL

Management job

- Computational Research
  - Applied Math
  - Computer Science
  - Data Science
  - Science Partnerships
- ESnet Facility
- NERSC Facility




My research

# Who is in this class?

- Applied Mathematics
- Applied Science & Tech
- Astrophysics
- Biostatistics
- Business Administration
- Chemical Engineering
- Chemistry
- Civil & Environmental
- Computer Science
- Earth & Planetary Sci
- EECS
- Environ. Sci, Pol & Mgmt
- Industrial Eng & Ops Res
- Info Mgmt & Sys (MIMS)
- Landscape Architecture
- Materials Science & Eng
- Mathematics
- Mechanical Engineering

26% CS
50% EECS
24% other

22% UG
35% MS/MEng
43% PhD

# Universities Offering CS267 in Spring 2018

## NSF XSEDE Program (using 2018 Berkeley lectures / material)

- Austin Peay
- Florida Polytechnic
- Franklin and Marshall
- George Mason University
- Indiana University Southeast
- Lamar University
- McNeese State University
- Morgan State
- North Carolina Central
- Oneonta University
- Penn State
- Prairie View A&M
- Rutgers University
- San Francisco State
- South Dakota State
- University of Houston- Clear Lake
- University of Kentucky

# Some of the World's Fastest Computer

The Top500 List

# Units of Measure for HPC

- **High Performance Computing (HPC) units are:**
  - Flop: floating point operation, usually double precision unless noted
  - Flop/s: floating point operations per second
  - Bytes: size of data (a double precision floating point number is 8 bytes)
- **Typical sizes are millions, billions, trillions…**

| | | |
|---|---|---|
| Kilo | Kflop/s = $10^3$ flop/sec | Kbyte = $10^3 \sim 2^{10}$ = 1024 bytes (KiB) |
| Mega | Mflop/s = $10^6$ flop/sec | Mbyte = $10^6 \sim 2^{20}$ bytes (MiB) |
| Giga | Gflop/s = $10^9$ flop/sec | Gbyte = $10^9 \sim 2^{30}$ bytes (GiB) |
| Tera | Tflop/s = $10^{12}$ flop/sec | Tbyte = $10^{12} \sim 2^{40}$ bytes (TiB) |
| Peta | Pflop/s = $10^{15}$ flop/sec | Pbyte = $10^{15} \sim 2^{50}$ bytes (PiB) |
| Exa | Eflop/s = $10^{18}$ flop/sec | Ebyte = $10^{18} \sim 2^{60}$ bytes (EiB) |
| Zetta | Zflop/s = $10^{21}$ flop/sec | Zbyte = $10^{21} \sim 2^{70}$ bytes (ZiB) |
| Yotta | Yflop/s = $10^{24}$ flop/sec | Ybyte = $10^{24} \sim 2^{80}$ bytes (YiB) |

- **Current fastest (public) machines are petaflop systems**
  - Up-to-date list at www.top500.org

# The TOP500 Project

- Listing the 500 most powerful computers in the world
- Yardstick: Rmax of Linpack
  - Solve Ax=b, Matrix A is dense with random entries
  - Dominated by dense matrix-matrix multiply
- Updated twice a year:
  - ISC'xy in June in Germany
  - SCxy in November in the U.S.
- All information available from the TOP500 web site at: www.top500.org

# The TOP12 of the Top500, November 2018

| # | Site | Manufacturer | Computer | Country | Cores | Rmax [Pflops] | Power [MW] |
|---|---|---|---|---|---|---|---|
| 1 | Oak Ridge National Laboratory | IBM | Summit<br>IBM Power System,<br>P9 22C 3.07GHz, Mellanox EDR, NVIDIA GV100 | USA | 2,397,824 | 143.5 | 9.8 |
| 2 | Lawrence Livermore National Laboratory | IBM | Sierra<br>IBM Power System,<br>P9 22C 3.1GHz, Mellanox EDR, NVIDIA GV100 | USA | 1,572,480 | 94.6 | 7.4 |
| 3 | National Supercomputing Center in Wuxi | NRCPC | Sunway TaihuLight<br>NRCPC Sunway SW26010,<br>260C 1.45GHz | China | 10,649,600 | 93.0 | 15.4 |
| 4 | National University of Defense Technology | NUDT | Tianhe-2A<br>ANUDT TH-IVB-FEP,<br>Xeon 12C 2.2GHz, Matrix-2000 | China | 4,981,760 | 61.4 | 18.5 |
| 5 | Swiss National Supercomputing Centre (CSCS) | Cray | Piz Daint<br>Cray XC50,<br>Xeon E5 12C 2.6GHz, Aries, NVIDIA Tesla P100 | Switzerland | 387,872 | 21.23 | 2.38 |
| 6 | Los Alamos NL / Sandia NL | Cray | Trinity<br>Cray XC40,<br>Intel Xeon Phi 7250 68C 1.4GHz, Aries | USA | 979,072 | 20.16 | 7.58 |
| 7 | National Institute of Advanced Industrial Science and Technology | Fujitsu | AI Bridging Cloud Infrastructure (ABCI)<br>PRIMERGY CX2550 M4,<br>Xeon Gold 20C 2.4GHz, IB-EDR, NVIDIA V100 | Japan | 391,680 | 19.9 | 1.65 |
| 8 | Leibniz Rechenzentrum | Lenovo | SuperMUC-NG<br>ThinkSystem SD530,<br>Xeon Platinum 8174 24C 3.1GHz, Intel Omni-Path | Germany | 305,856 | 19.5 | |
| 9 | Oak Ridge National Laboratory | Cray | Titan<br>Cray XK7,<br>Opteron 16C 2.2GHz, Gemini, NVIDIA K20x | USA | 560,640 | 17.6 | 8.21 |
| 10 | Lawrence Livermore National Laboratory | IBM | Sequoia<br>BlueGene/Q,<br>Power BQC 16C 1.6GHz, Custom | USA | 1,572,864 | 17.2 | 7.89 |
| 11 | Lawrence Livermore National Laboratory | IBM | Lassen<br>Cray XC40 | USA | 248,976 | 15.4 | |
| 12 | Lawrence Berkeley National Laboratory / NERSC | Cray | Cori<br>Cray XC40,<br>Intel Xeons Phi 7250 68C 1.4 GHz, Aries | USA | 622,336 | 14.0 | 3.94 |

# Summit (#1 machine) System Overview

## System Performance

- Peak performance of 200 petaflops for modeling & simulation
- Peak of 3.3 ExaOps for data analytics and artificial intelligence

## Each node has

- 2 IBM POWER9 processors
- 6 NVIDIA Tesla V100 GPUs
- 608 GB of fast memory
- 1.6 TB of NVMe memory

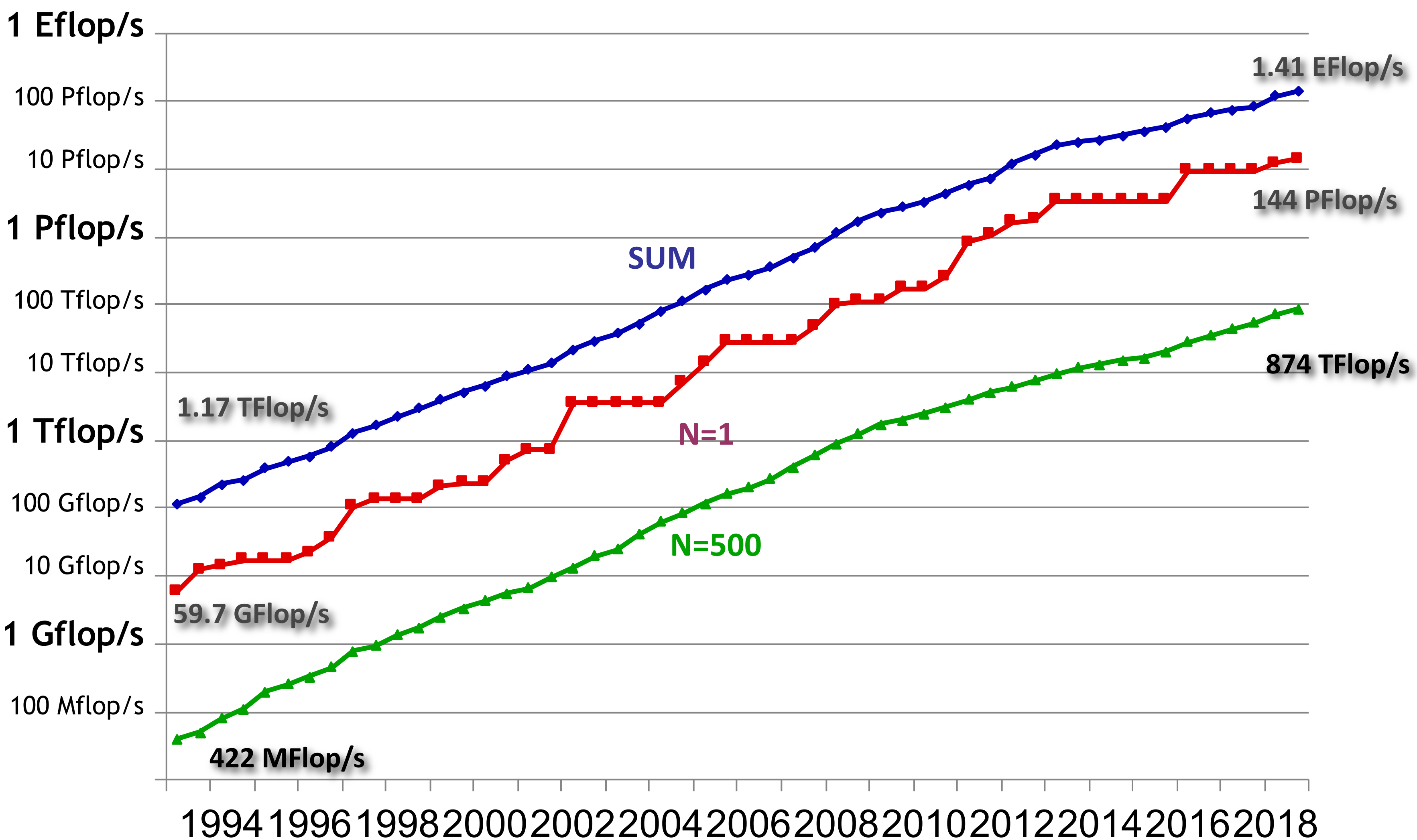
## The system includes

- 4608 nodes
- Dual-rail Mellanox EDR InfiniBand network
- 250 PB IBM Spectrum Scale file system transferring data at 2.5 TB/s

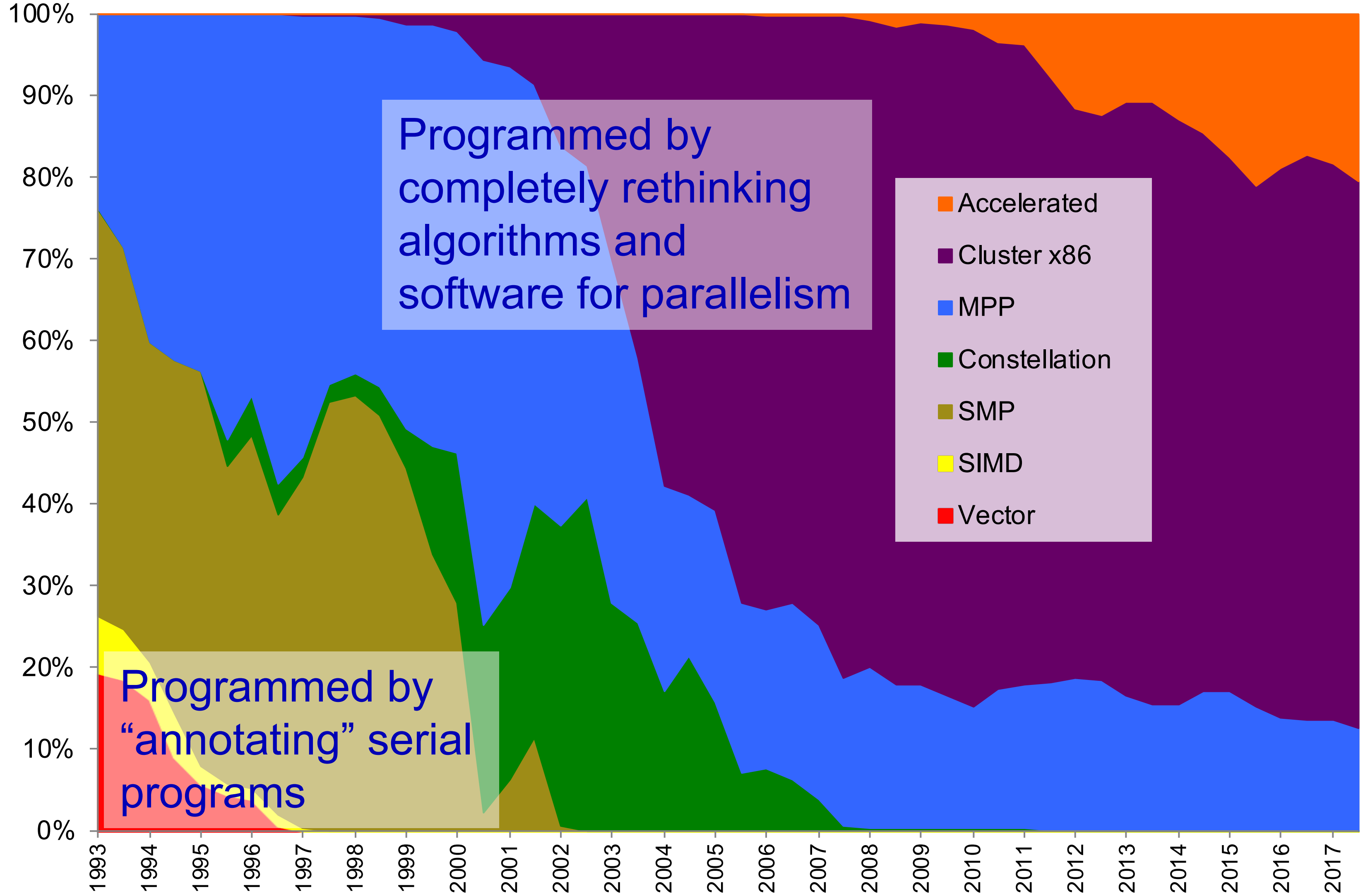
# Cori at NERSC in Berkeley, CA


| | | | |
|---|---|---|---|
| Phase 1 | **Peak:** 2.8 PFlop/s | **Proc:** 3800 Haswell (ph 1)<br>**Cores:** 2.3 GHz, 16/proc<br>**Proc Peak**: 0.6 TFlop/s | **Memory:** 0.3 PB<br>**Storage:** shared |
| Phase 2 | **Peak:** 28PFlop/s<br>**LinPack: 14 Pflops/s**<br>**Power: 4 MW** | **Processor:** 9300 KNL<br>**Cores:** 1.4 GHz, 68/proc<br>**Node Peak**: 3 TFlop/s | **Memory:** 1.31 PB<br>**MemBW:** 1 PB/s HBM,DDR<br>**Storage:** 28 PB |

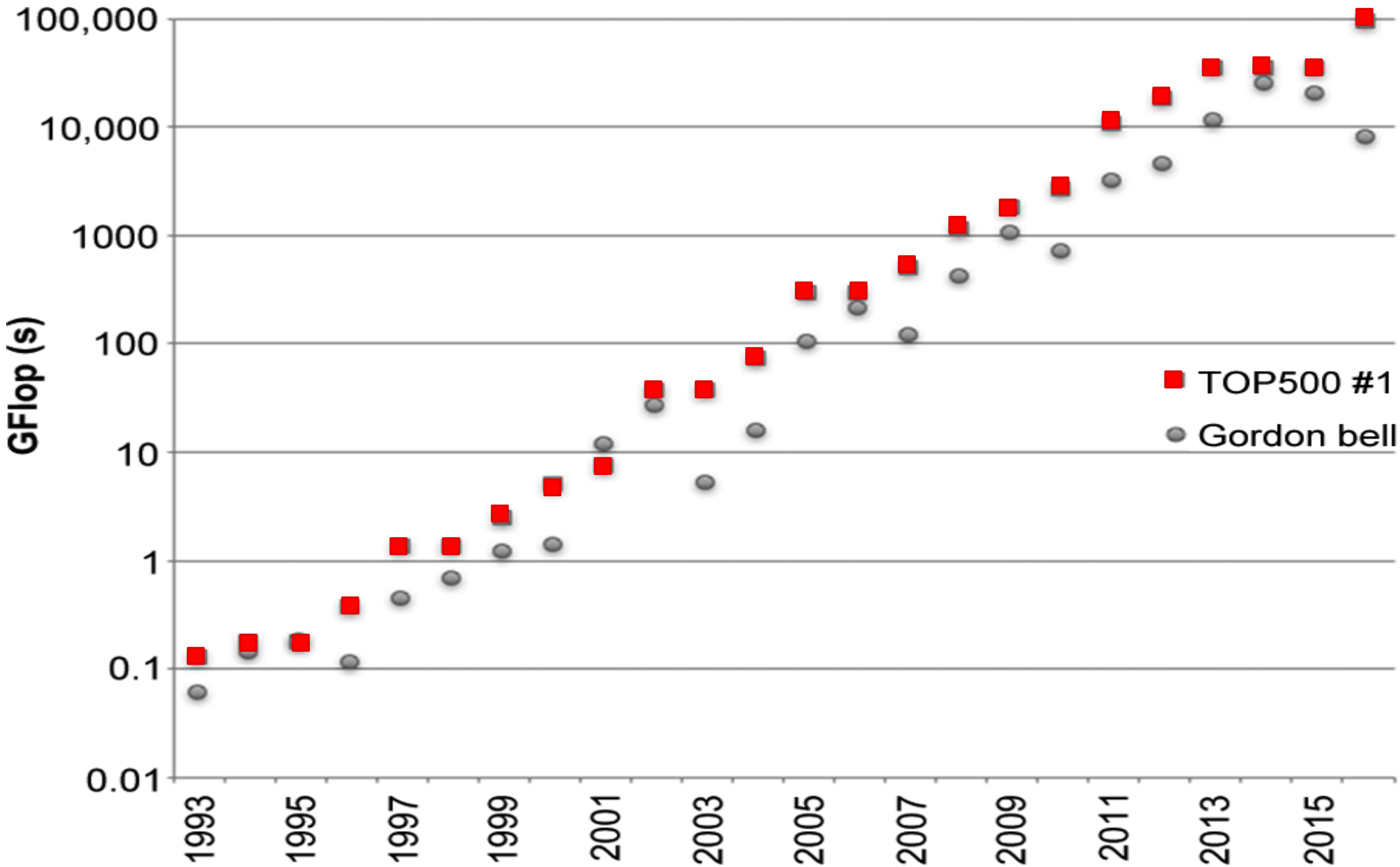
# Performance Development (2018)



# From Vector Supercomputers to Massively Parallel Accelerator Systems



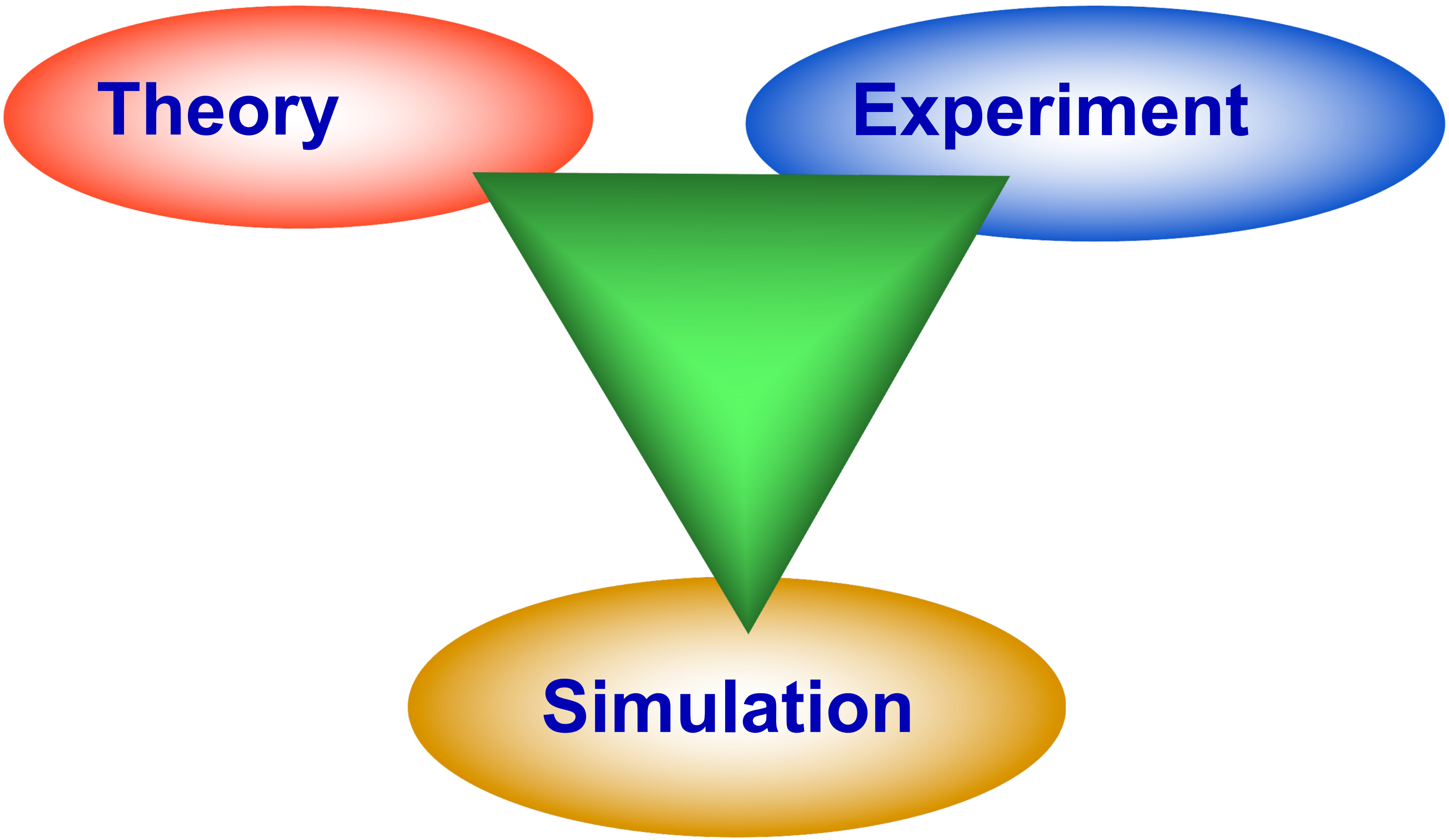
# Science: Gordon Bell Prizes vs Top 500



# Science using High Performance Computing

Today and tomorrow, mostly from the DOE Exascale Project

# The “Third Pillar” of Science



# Simulation in Science and Engineering

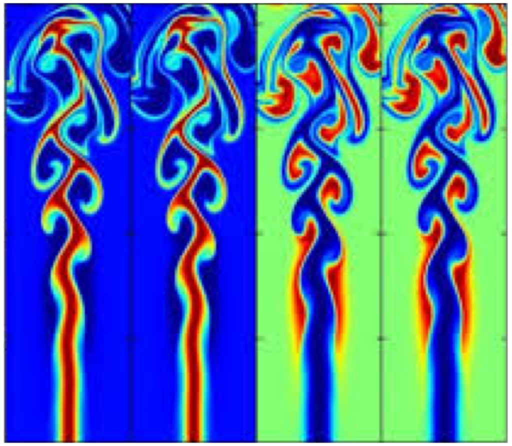
High performance computing (HPC) simulation to understand things that are:

- too big
- too small
- too fast
- too slow
- too expensive or
- too dangerous

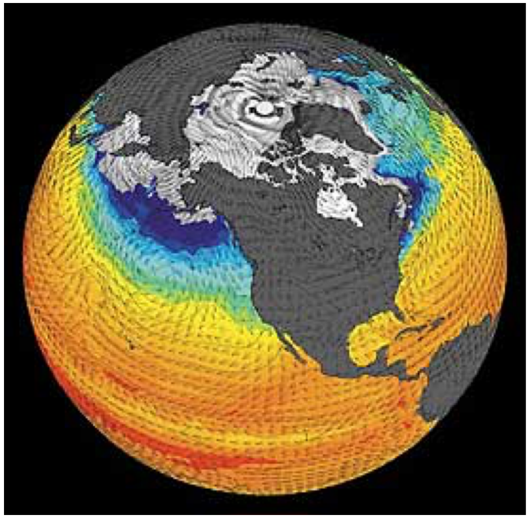
for experiments

Understanding the universe

Proteins and diseases

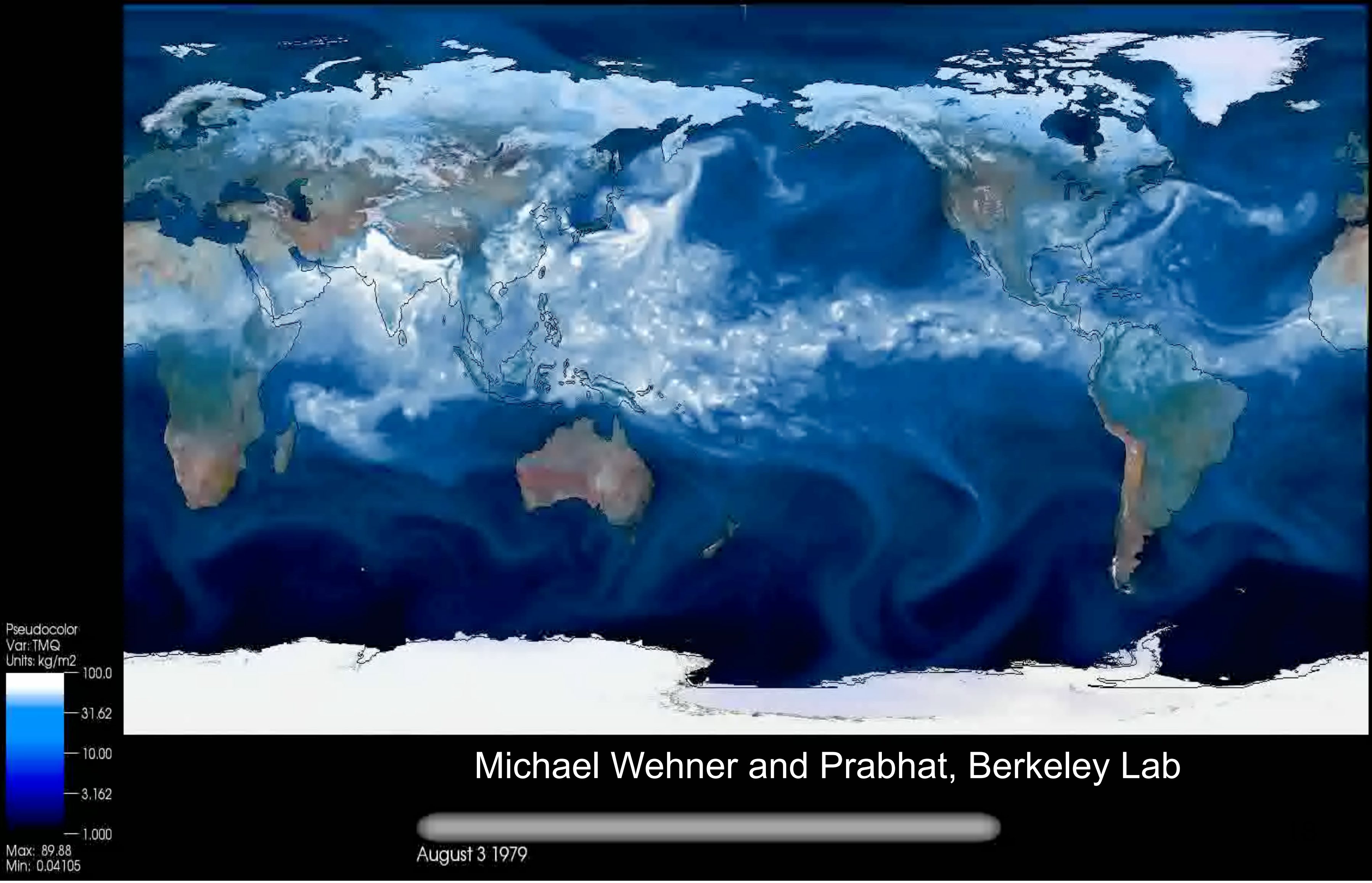


Energy-efficient jet engines



Climate change

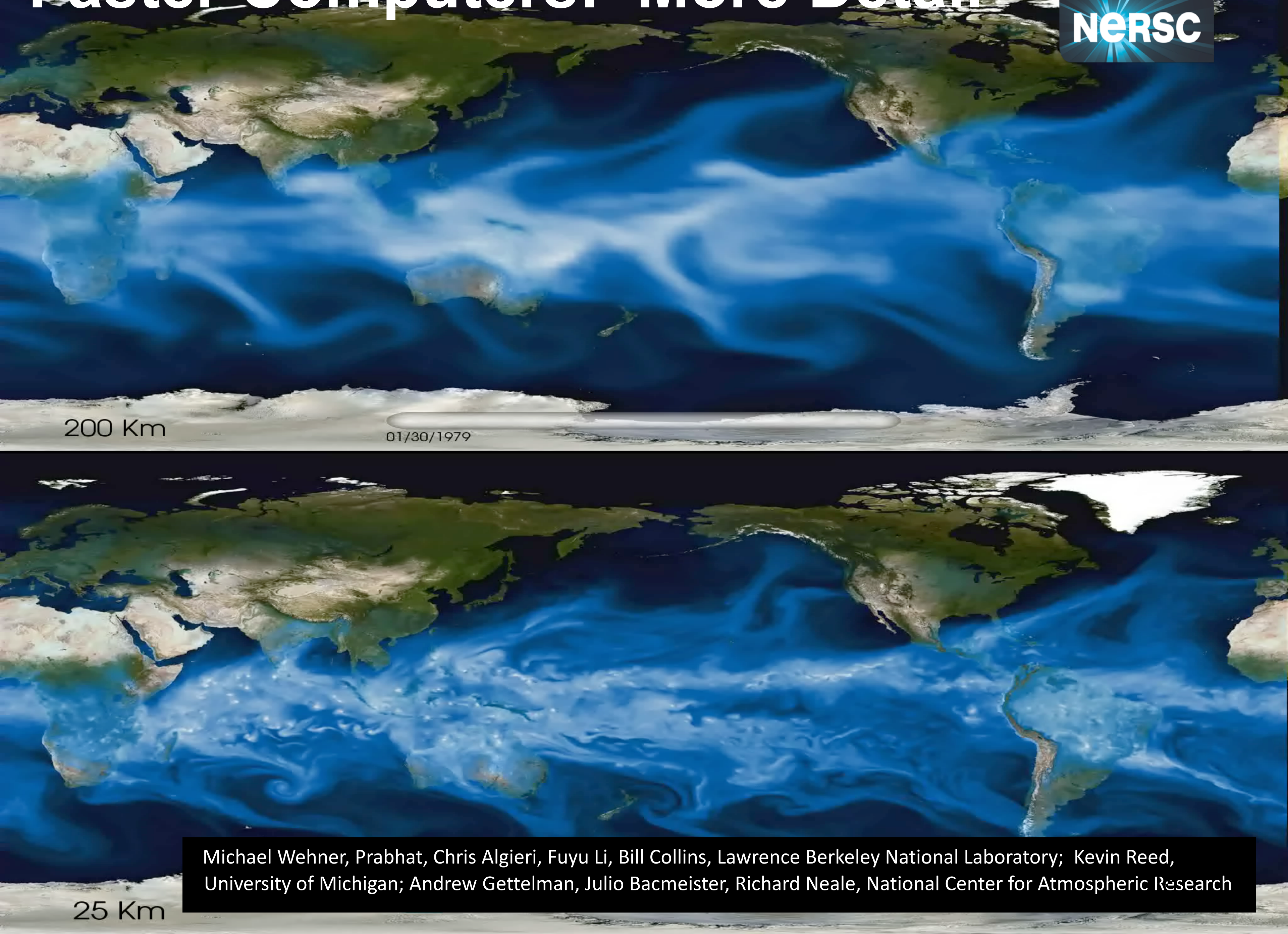
# Simulations Show the Effects of Climate Change in Hurricanes




Michael Wehner and Prabhat, Berkeley Lab

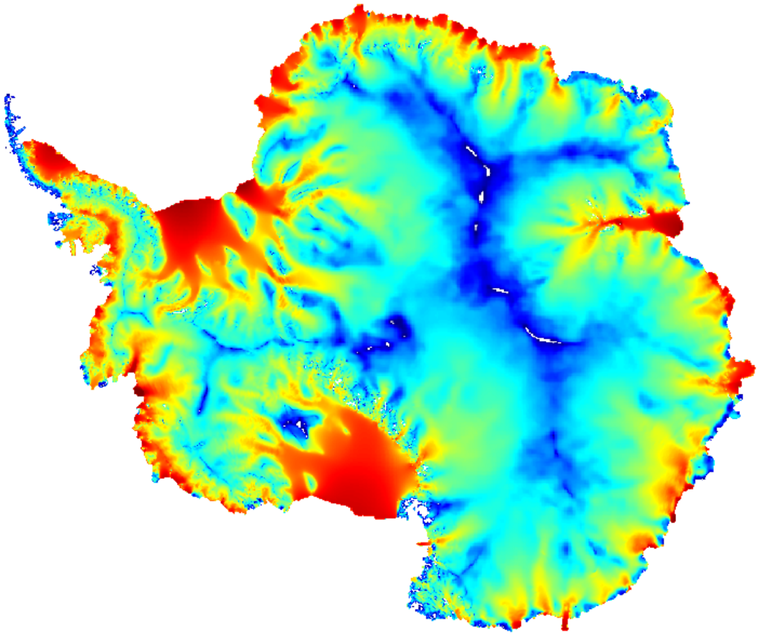
# Faster Computers: More Detail

NERSC




Michael Wehner, Prabhat, Chris Algieri, Fuyu Li, Bill Collins, Lawrence Berkeley National Laboratory; Kevin Reed, University of Michigan; Andrew Gettelman, Julio Bacmeister, Richard Neale, National Center for Atmospheric Research

# Extreme Scale Climate Science



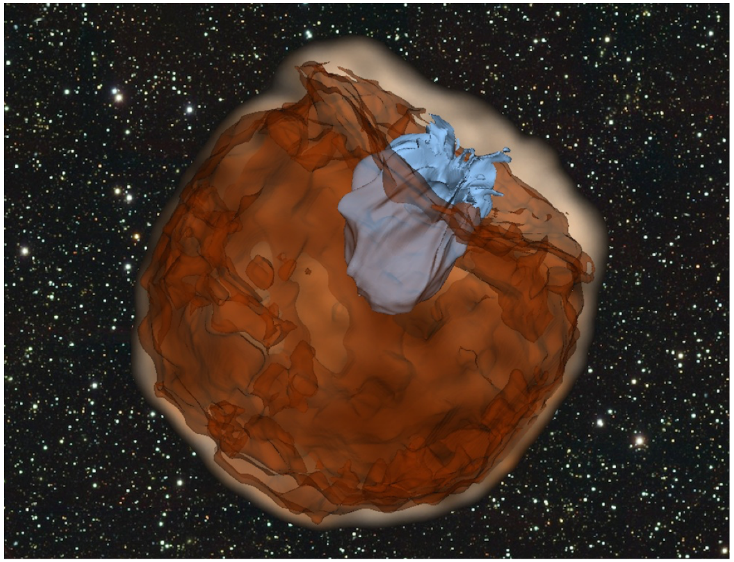
*Melting of West Antarctic Ice Sheet using Adaptive Mesh Refinement (AMR) Dan Martin, LBNL (BISICLES/E3SM)*

*Mathematical model for clouds David Romps et al, UCB*

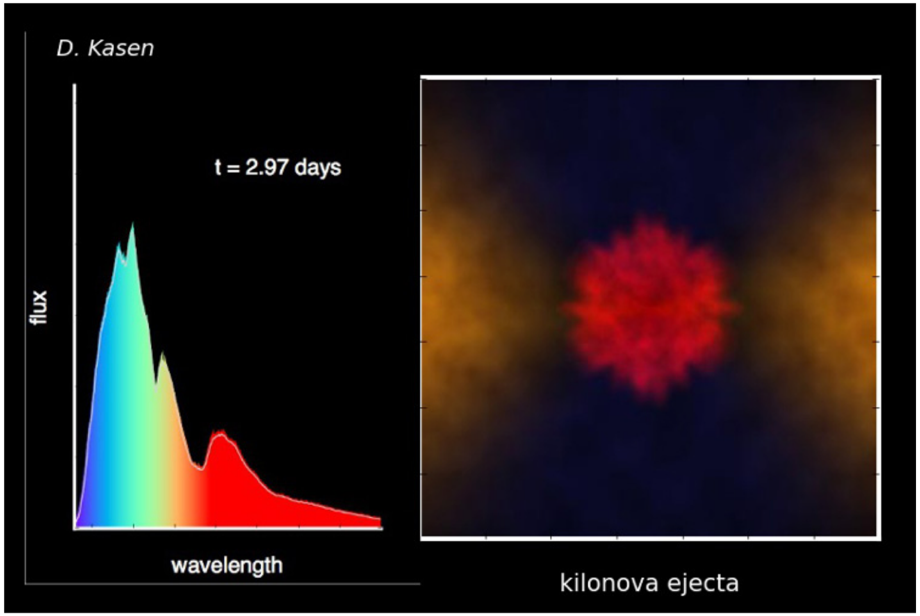
**Exascale is needed to simulate climate and analyze impacts**

Resolve clouds, predict sea level rise, quantify extreme events and model precipitation and ground water levels

# Extreme Scale Astrophysics (Dan Kasen, LBNL/UCB)



*Expanding debris from a supernova explosion (red) running over and shredding a nearby star (blue)*

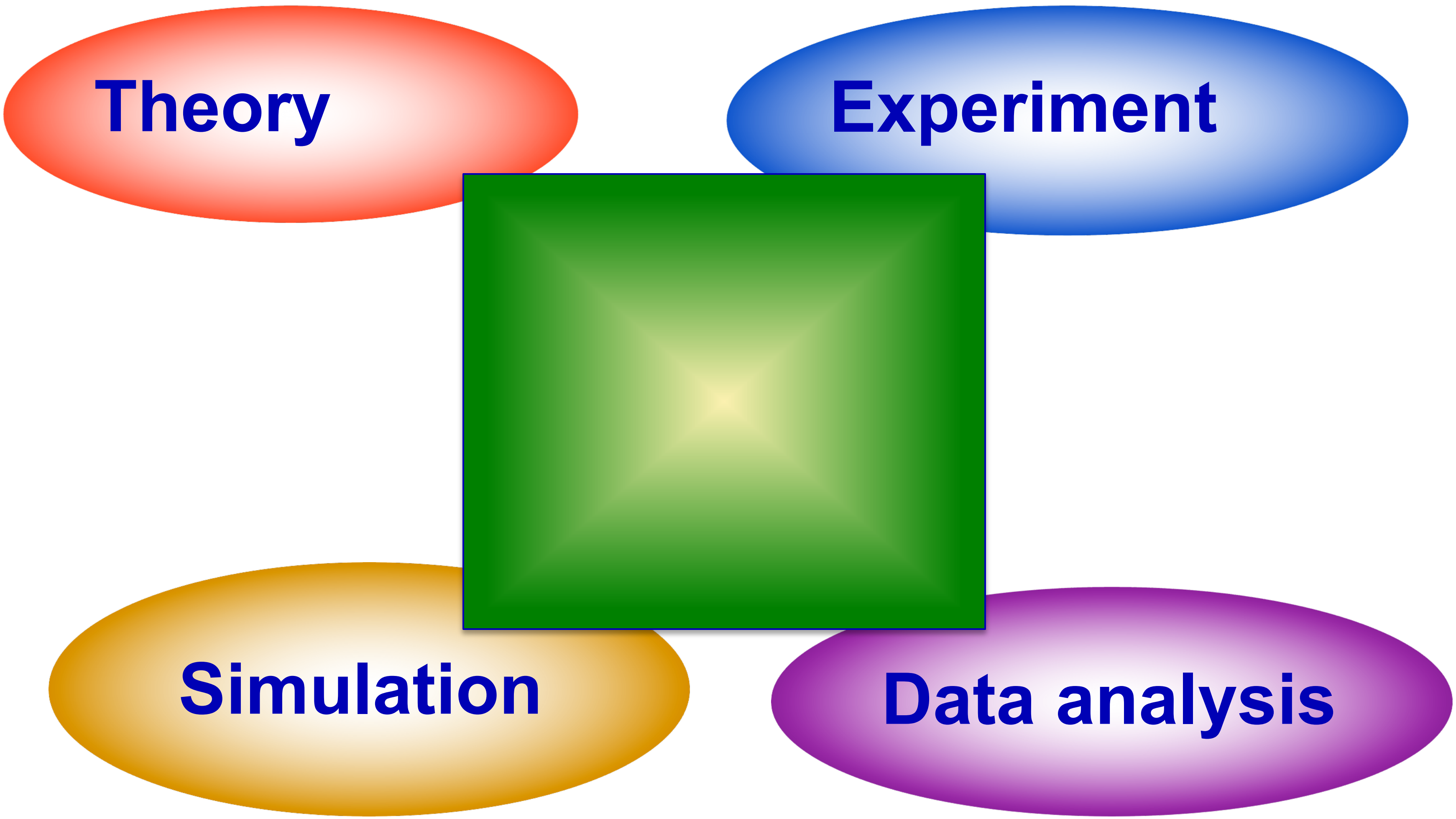



*Ligo and Virgo observations match earlier simulations of gravitational waves from neutron star merger. Simulations predict* ~200 earth masses of gold; ~500 of platinum

**Exascale is needed to identify the source of the heaviest elements**

Understand rapid neutron capture process (r-process) by simulating scenarios: core-collapse supernovae, neutron star mergers, and accreting black holes

# The Fourth Paradigm of Science



# Data analytics in science and engineering

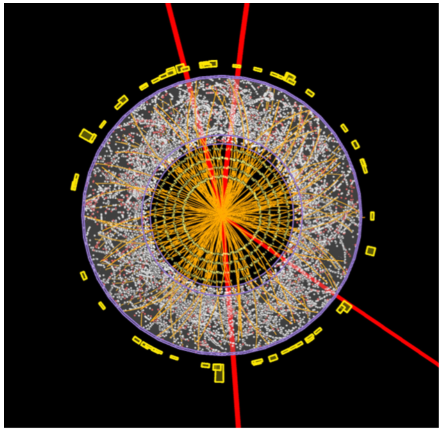
High Performance Data Analytics (HPDA) is used for data sets that are:

- too big
- too complex
- too fast (streaming)
- too noisy
- too heterogeneous

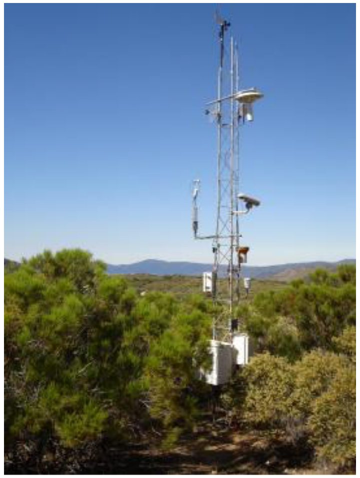
for measurement alone

Images from telescopes

Genomes from sequencers

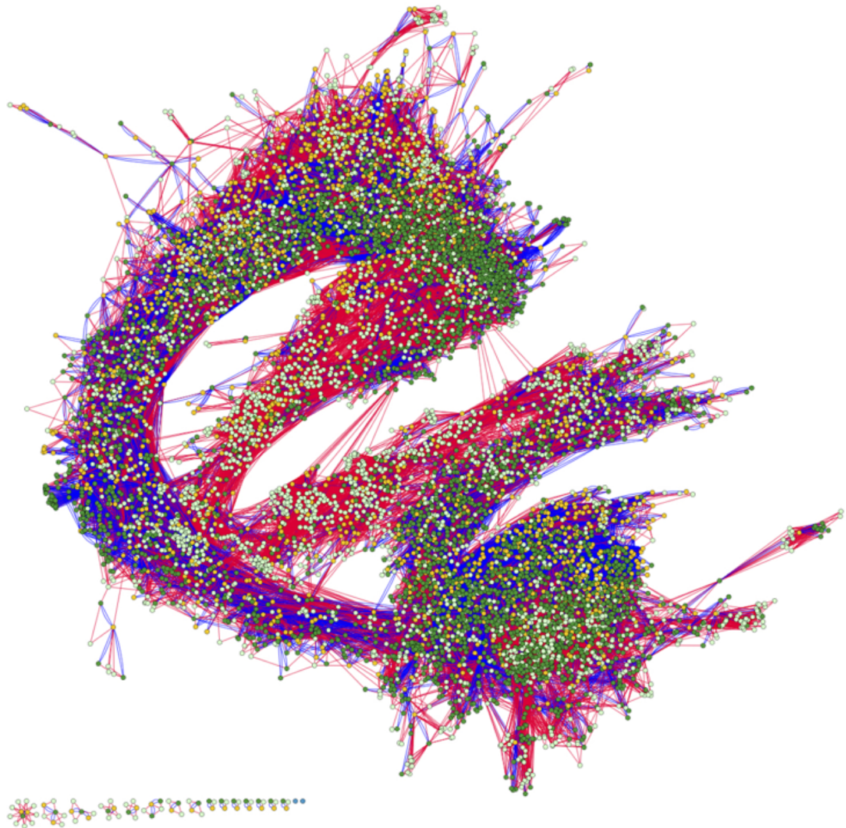


Particle from detectors



Sensor data

# Analysis of Genomic Data



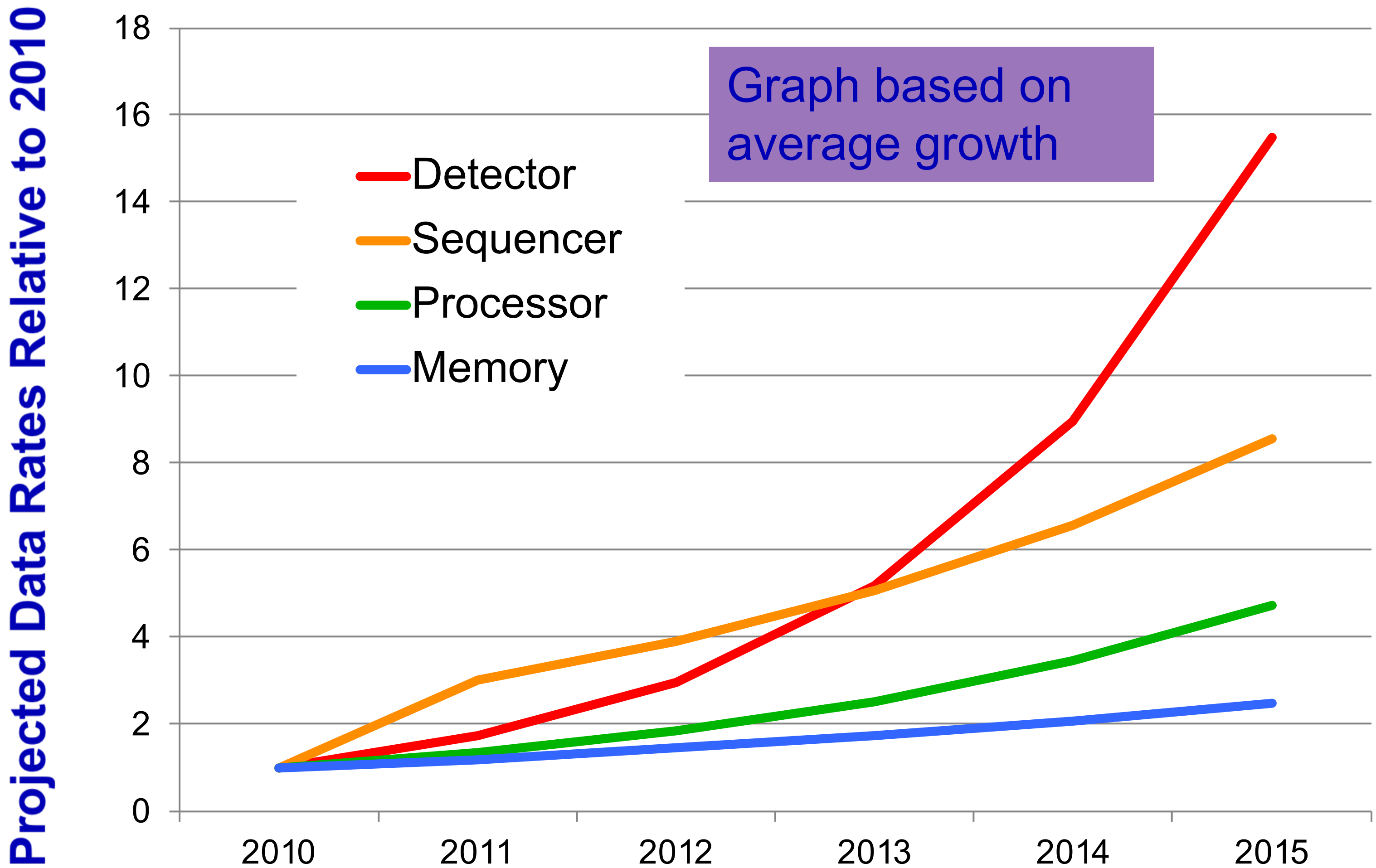
Dark green nodes: Kalanchoë genes
Yellow nodes: pineapple genes
Light green: model plant that uses a different photosynthesis strategy.

Blue edges: positive correlations
Red edges show negative correlations a

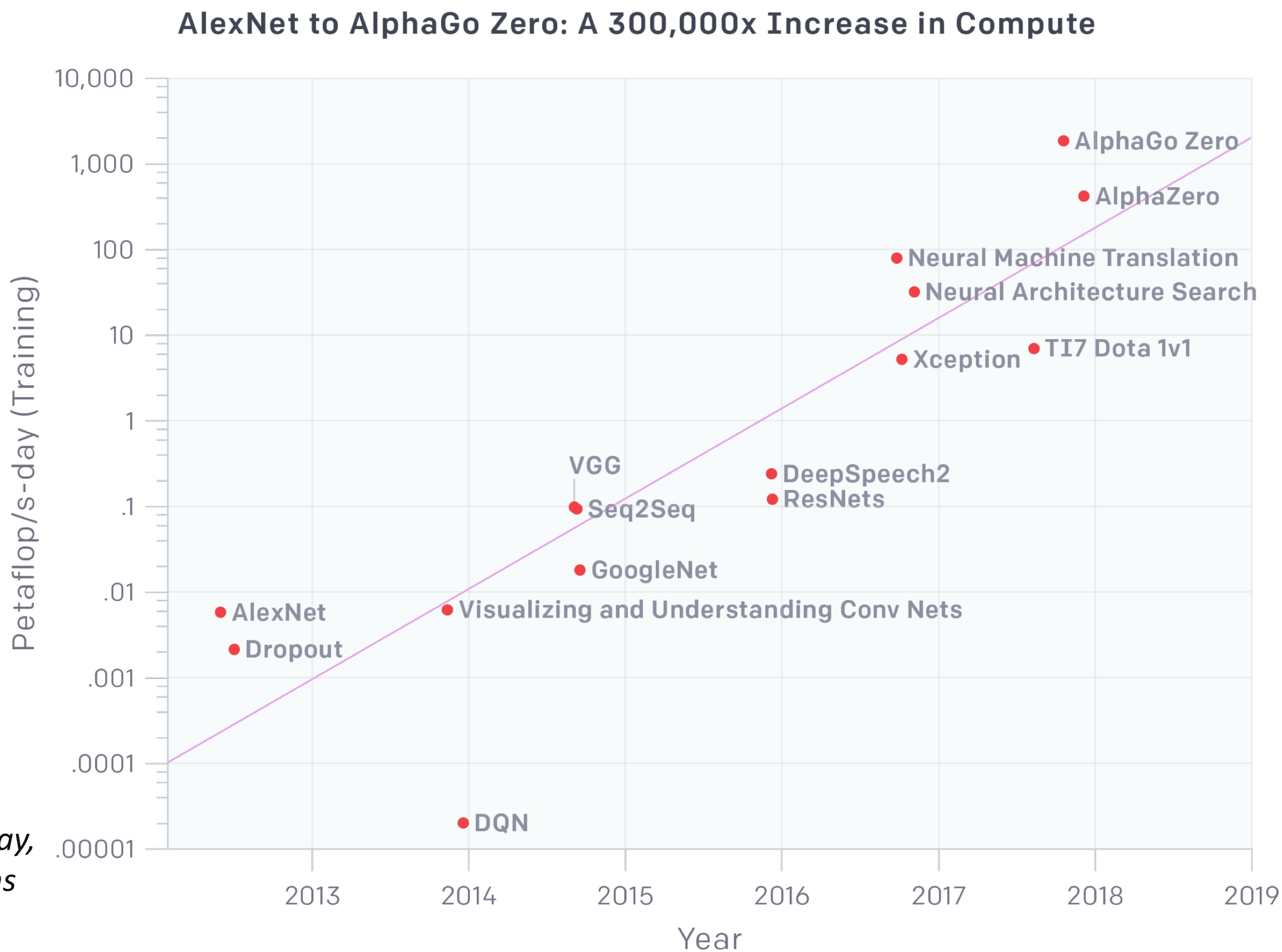
- Correlations of gene expression in plants that use different photosynthesis strategies.
- Kalanhoë and pineapple both use water-sparing photosynthesis
- 2.36 exaops / second on Summit computer
- (exaop = $10^{18}$ 16-bit ops)

Wayne Joubert, Dan Jacobson et al, Gordon Bell Prize (1 of 2) at SC18

# Data Growth is Outpacing Computing Growth



# Growth of Computing in Machine Learning

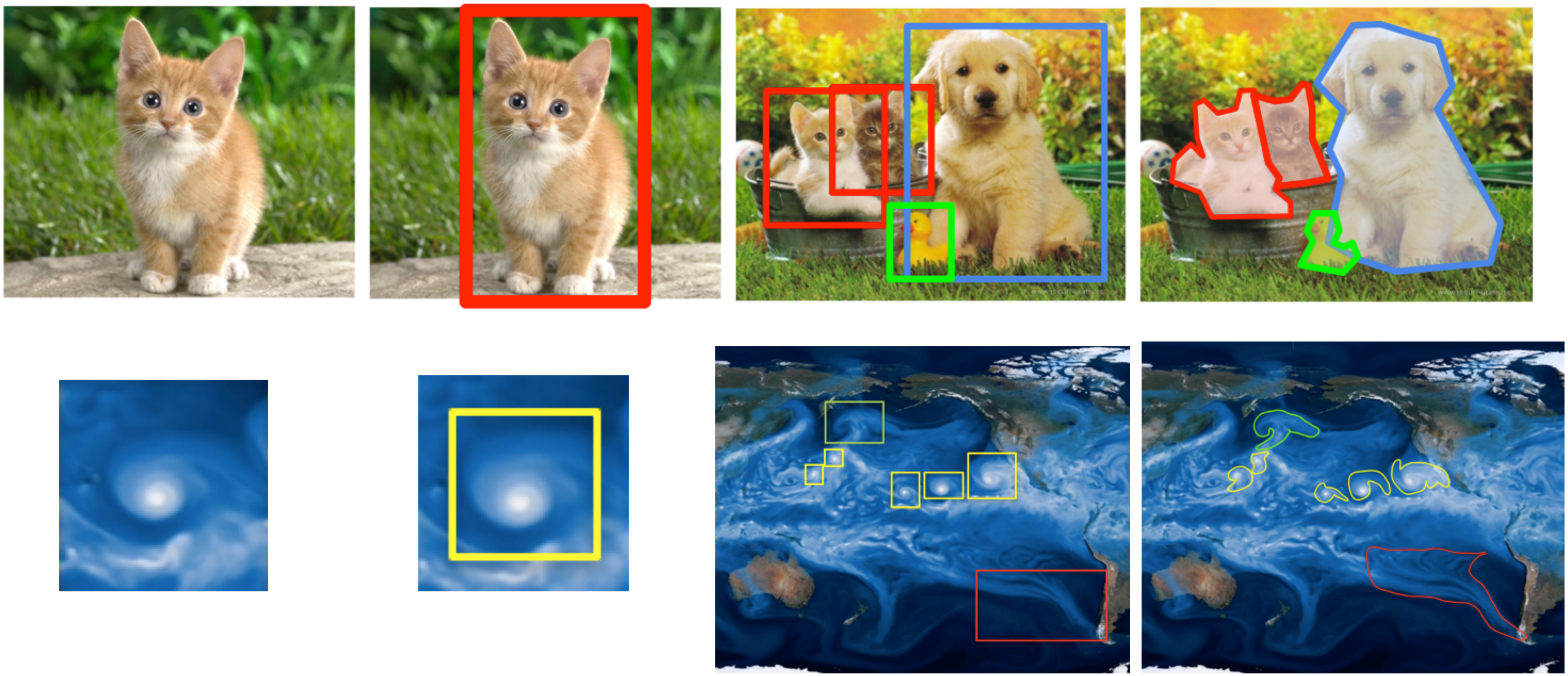



*A petaflop/s-day = $10^{15}$ neural net operations per second for one day, ~= $10^{20}$ operations*

https://blog.openai.com/ai-and-compute/

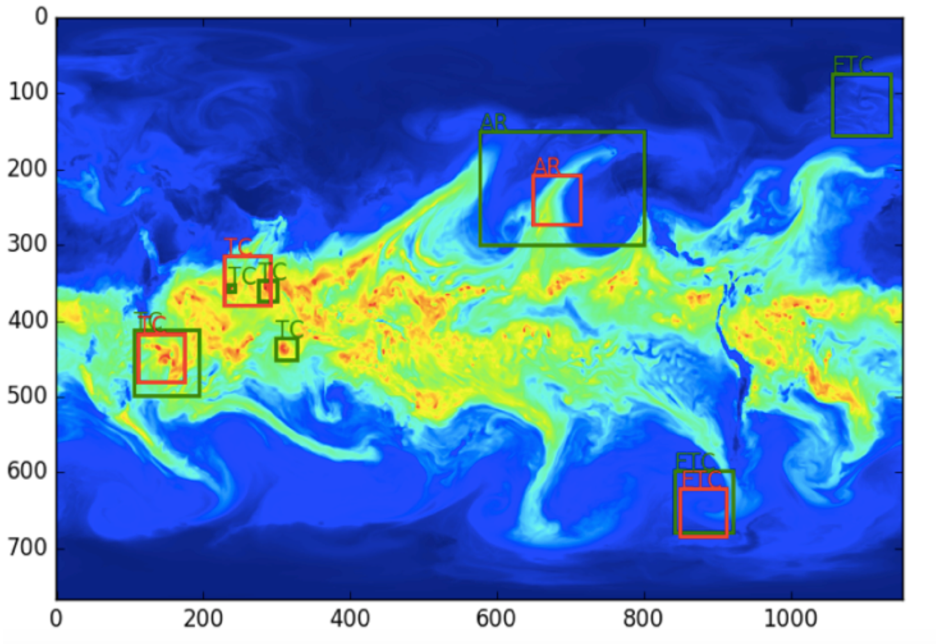
# Machine Learning in Climate Data

Classification | Classification + Localization | Object Detection | Instance Segmentation

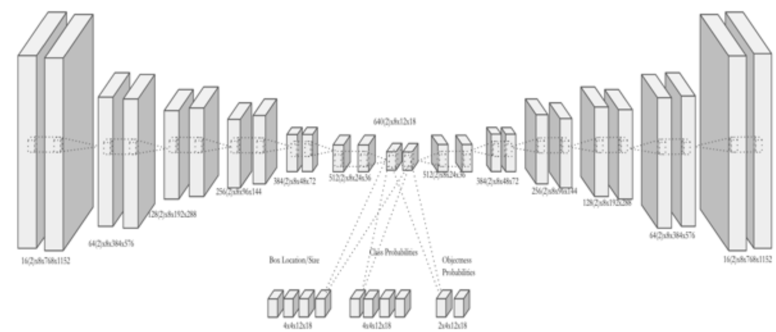


Contributors: Prabhat, Thorsten Kurth, Jian Yang, Ioannis Mitliagkas, Chris Pal, Nadathur Satish, Narayanan Sundaram, Amir Khosrowshahi, Michael Wehner, Bill Collins.

# Deep Learning for Extreme Weather Events




Ground Truth vs Prediction



Use of deep learning (CNNs)

- Supervised and semi-supervised learning on CAM5 data
- 85-99% accuracy at identifying extreme climate events
- 1 ExaOp (16-bit) on Summit at ORNL; trained in 100 minutes

# Outline

- ✓ Some of the World's Fastest (Parallel) Computers
    - Top500 List highlights
- ✓ Why science needs high performance computing (HPC)
    - Simulation and data analysis
- Why high performance machines are parallel
    - Including your laptops and handhelds
- Performance from parallelism (and locality)
    - Measuring, understanding, and reporting performance
- About this course
    - Topics, structure and objectives

# Why the ~~Fastest~~ all Computers are Parallel Computers (since 2005)

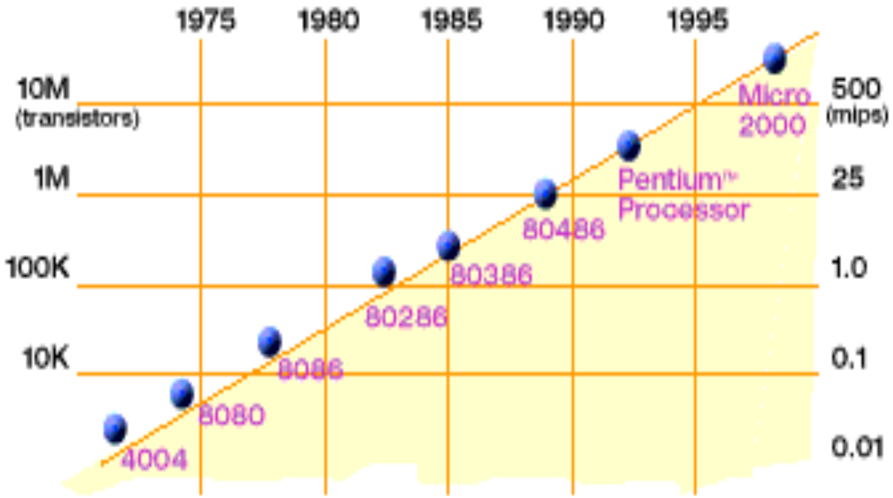
Including laptops and handhelds

# Tunnel Vision by Experts

- "I think there is a world market for maybe five computers."
  - Thomas Watson, chairman of IBM, 1943.
- "There is no reason for any individual to have a computer in their home"
  - Ken Olson, president and founder of Digital Equipment Corporation, 1977.
- "640K [of memory] ought to be enough for anybody."
  - Bill Gates, chairman of Microsoft,1981.
- "On several recent occasions, I have been asked whether parallel computing will soon be relegated to the trash heap reserved for promising technologies that never quite make it."
  - Ken Kennedy, CRPC Directory, 1994


Slide source: Warfield et al.

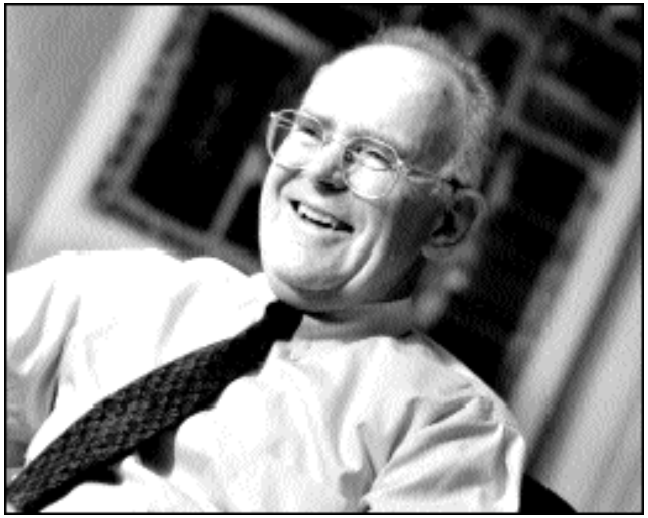
# Technology Trends: Microprocessor Capacity




2X transistors/Chip Every 1.5 years

**Called "Moore's Law"**

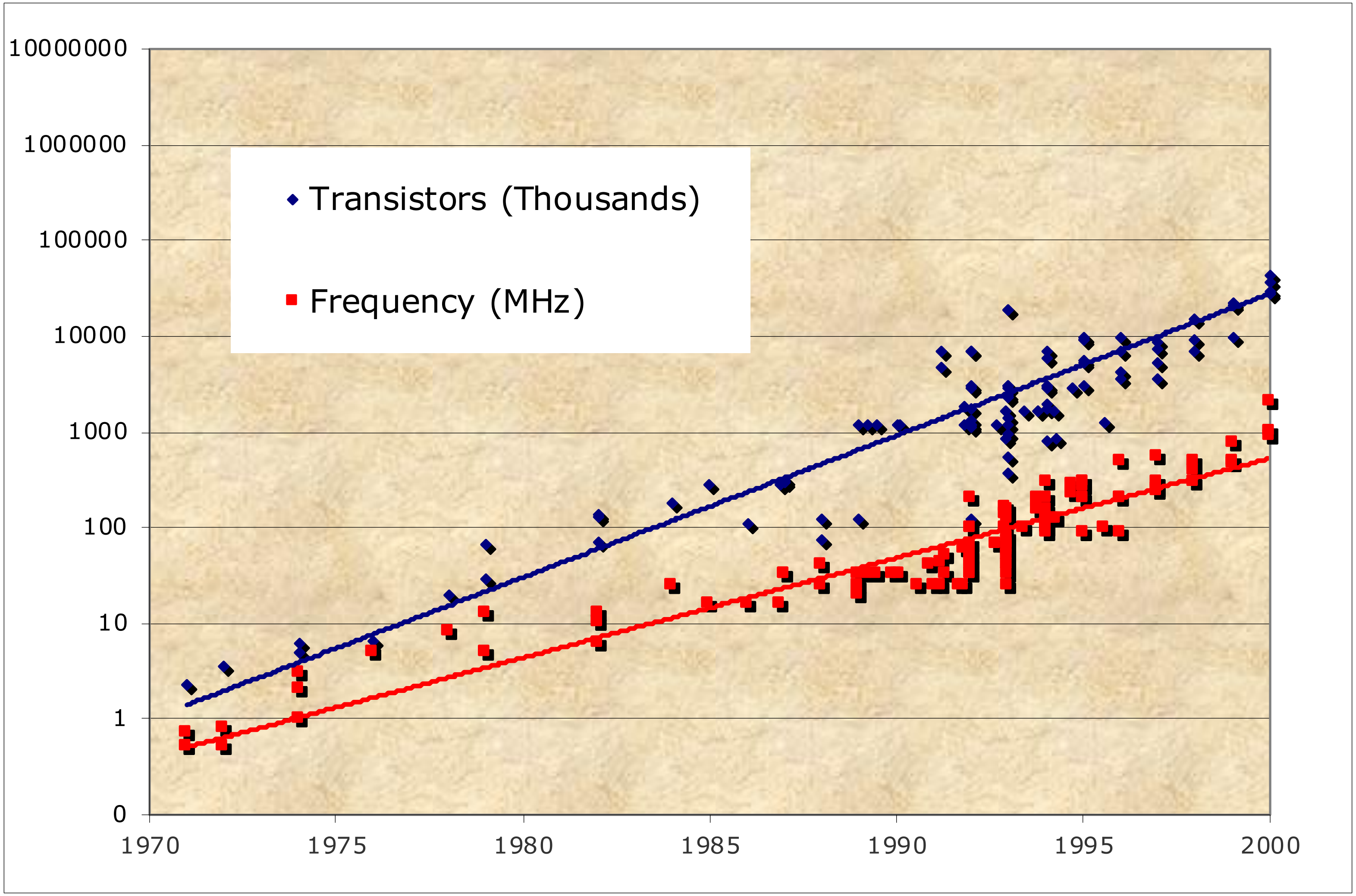
**Microprocessors have become smaller, denser, and more powerful.**



**Gordon Moore (co-founder of Intel) predicted in 1965 that the transistor density of semiconductor chips would double roughly every 18 months.**

Slide source: Jack Dongarra

# Microprocessor Transistors / Clock (1970-2000)

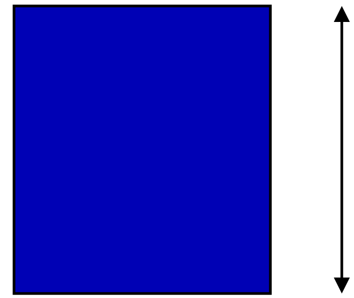


# Historical Impact of Device Shrinkage

- What happens when the feature size (transistor size) shrinks by a factor of x ?
- Clock rate goes up by x because wires are shorter
  - actually less than x, because of power consumption
- Transistors per unit area goes up by $x^2$
- Die size has also increased
  - typically another factor of ~x
- Raw computing power of the chip goes up by ~ $x^4$ !
  - typically $x^3$ is devoted to either on-chip
    - parallelism: hidden parallelism such as ILP
    - locality: caches
- So most programs $x^3$ times faster, without changing them

# Limits: How fast can a serial computer be?

1 Tflop/s, 1 Tbyte sequential machine



r = 0.3 mm

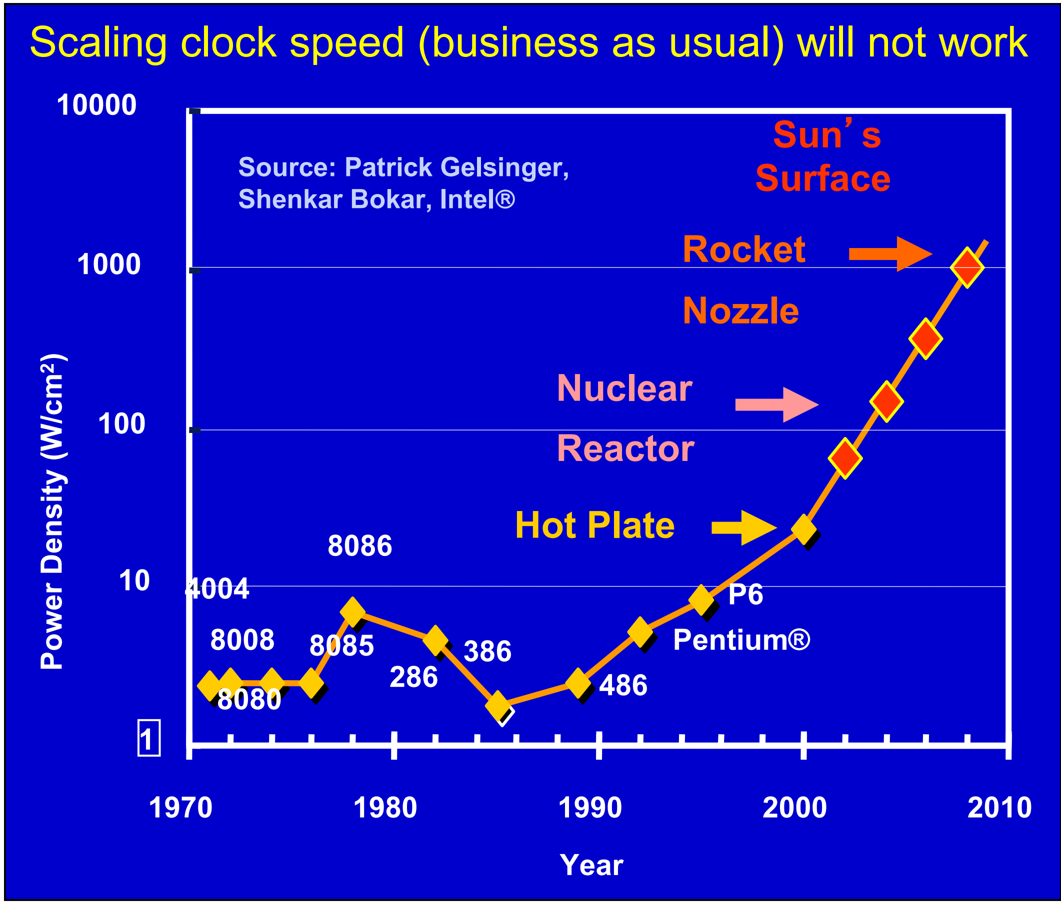
- Consider the 1 Tflop/s sequential machine:
  - Data must travel some distance, r, to get from memory to processor.
  - To get 1 data element per cycle, this means $10^{12}$ times per second at the speed of light, c = $3 \times 10^8$ m/s. Thus r < $c/10^{12}$ = 0.3 mm.
- Now put 1 Tbyte of storage in a 0.3 mm x 0.3 mm area:
  - Each bit occupies about 1 square Angstrom, or the size of a small atom.
- No choice but parallelism

# But What about Heat Density


Image by Sam Spratt

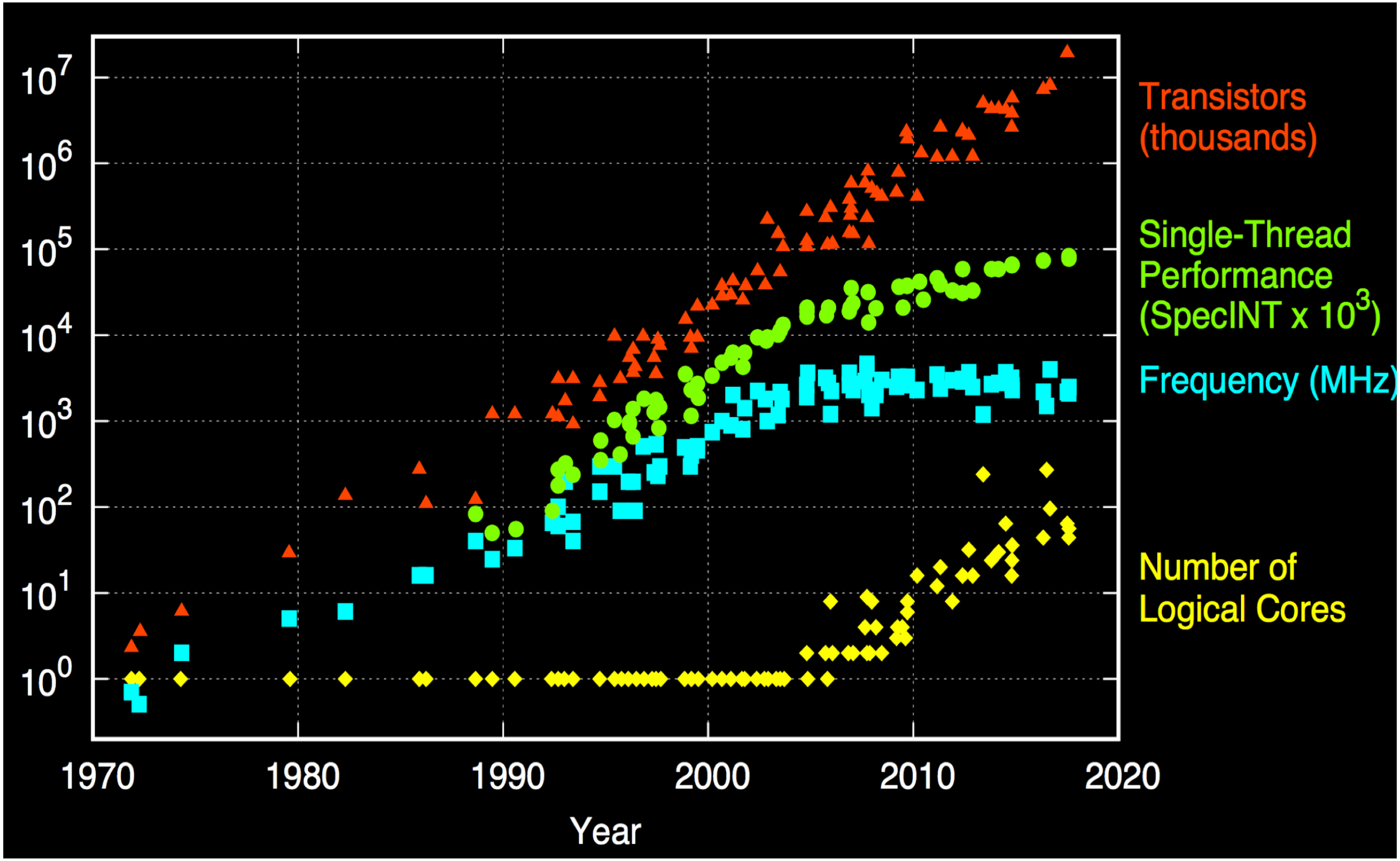
# Power Density Limits Serial Performance

- Concurrent systems are more power efficient
  - Dynamic power is proportional to $V^2fC$
  - Increasing frequency (f) also increases supply voltage (V) → cubic effect
  - Increasing cores increases capacitance (C) but only linearly
  - Save power by lowering clock speed




- High performance serial processors waste power
  - Speculation, dynamic dependence checking, etc. burn power
  - Implicit parallelism discovery
- More transistors, but not faster serial processors

# The Rest of the World Adopts Parallelism

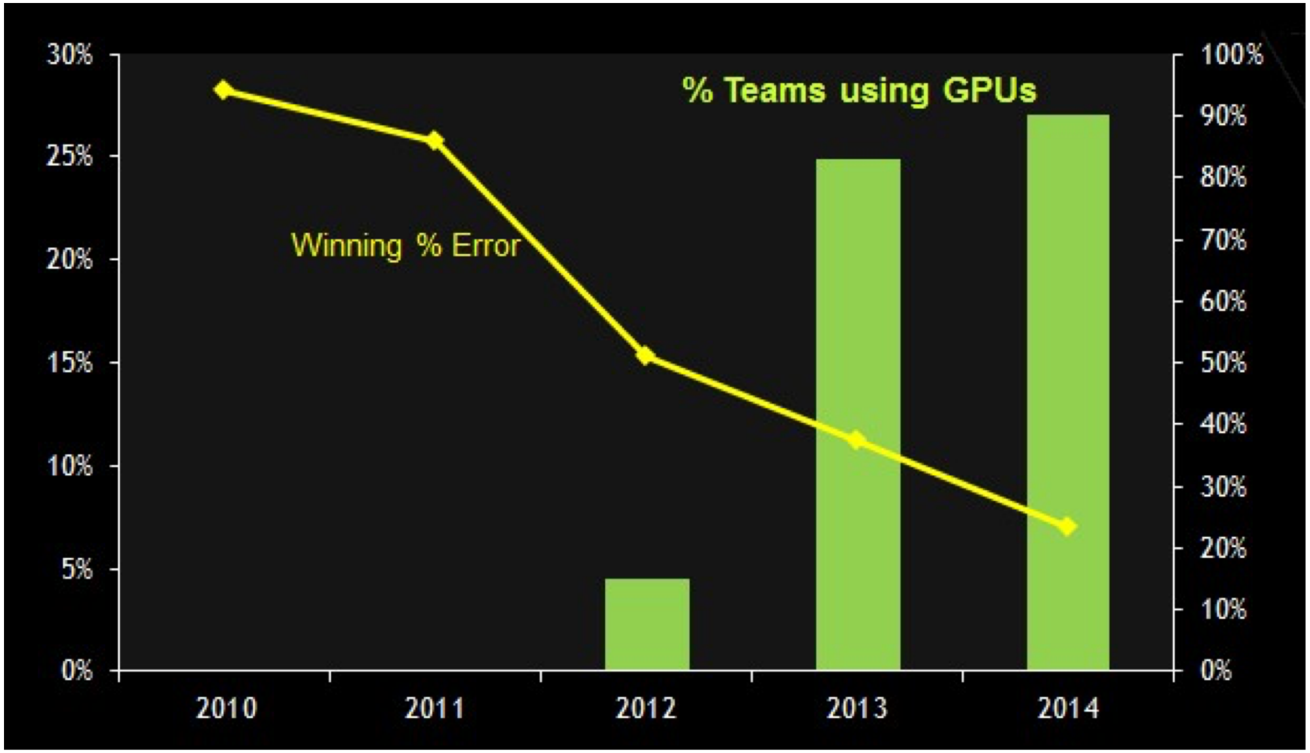



M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, C. Batten, and K. Rupp

Cores rather than clock frequency is doubling

# Disruptions from commercial data analytics

ImageNet Large Scale Visual Recognition Challenge (image from NVIDIA)




In the second quarter of 2016, NVIDIA attributed nearly half its $151 million in datacenter sales to GPUs intended for deep learning.

Applications in self-driving cars, etc.

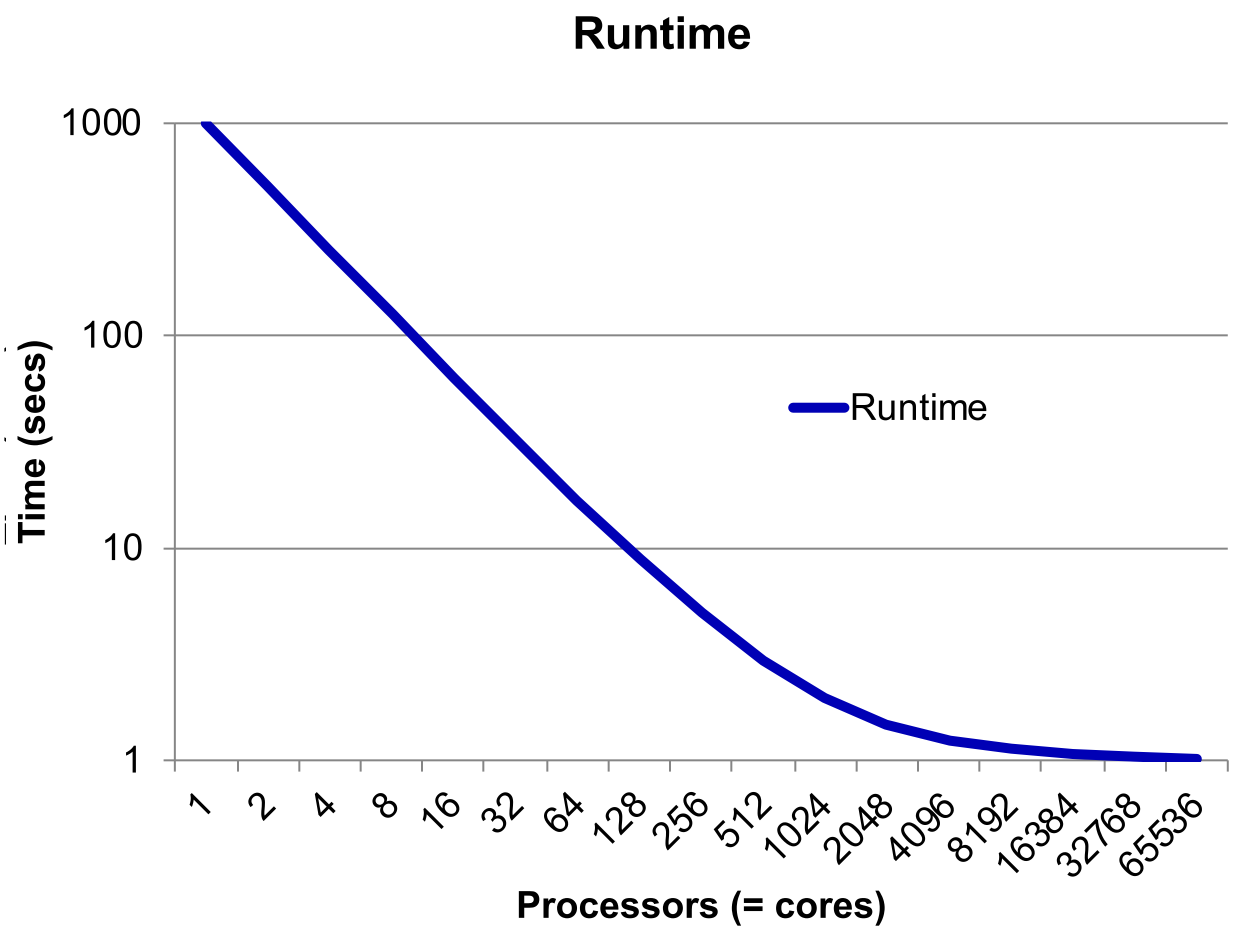
# Moore's Law reinterpreted

- Number of cores per chip can double every two years
- Clock speed will not increase (possibly decrease)
- Need to deal with systems with millions of concurrent threads
- Need to deal with inter-chip parallelism as well as intra-chip parallelism
- But Moore's Law is not forever… industry consortium predicts end in 2021

# Measuring Performance

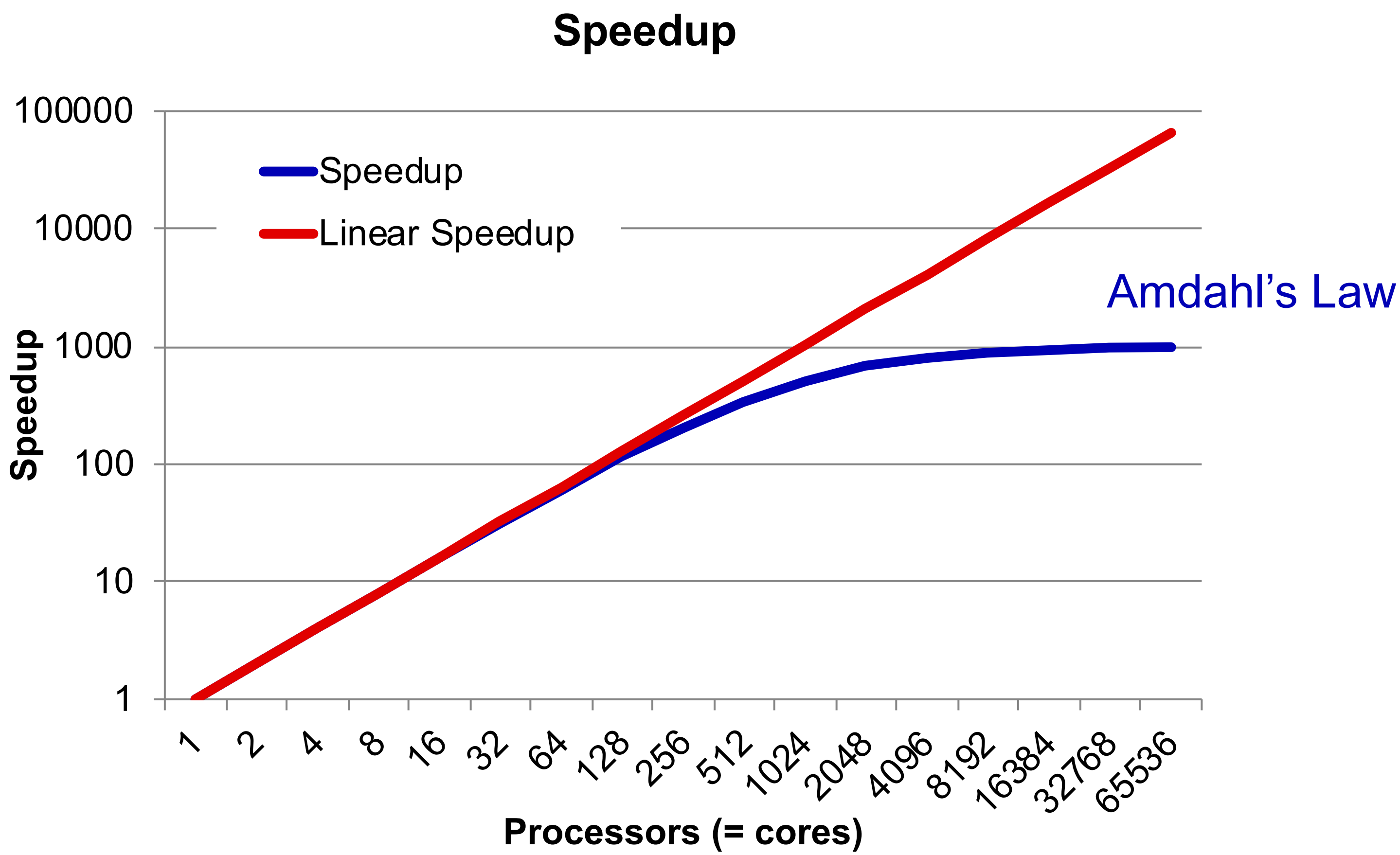
Tips, tricks, and cheats

# Goal of Parallelism: Decrease Running Time




A bit better, but the point of this is to show the value of parallelism

# Reporting Speedup (Strong Scaling)




Speedup(P) = Time(1)/Time(P)

This is a **strong scaling** plot: fixed problem size, vary number of processors

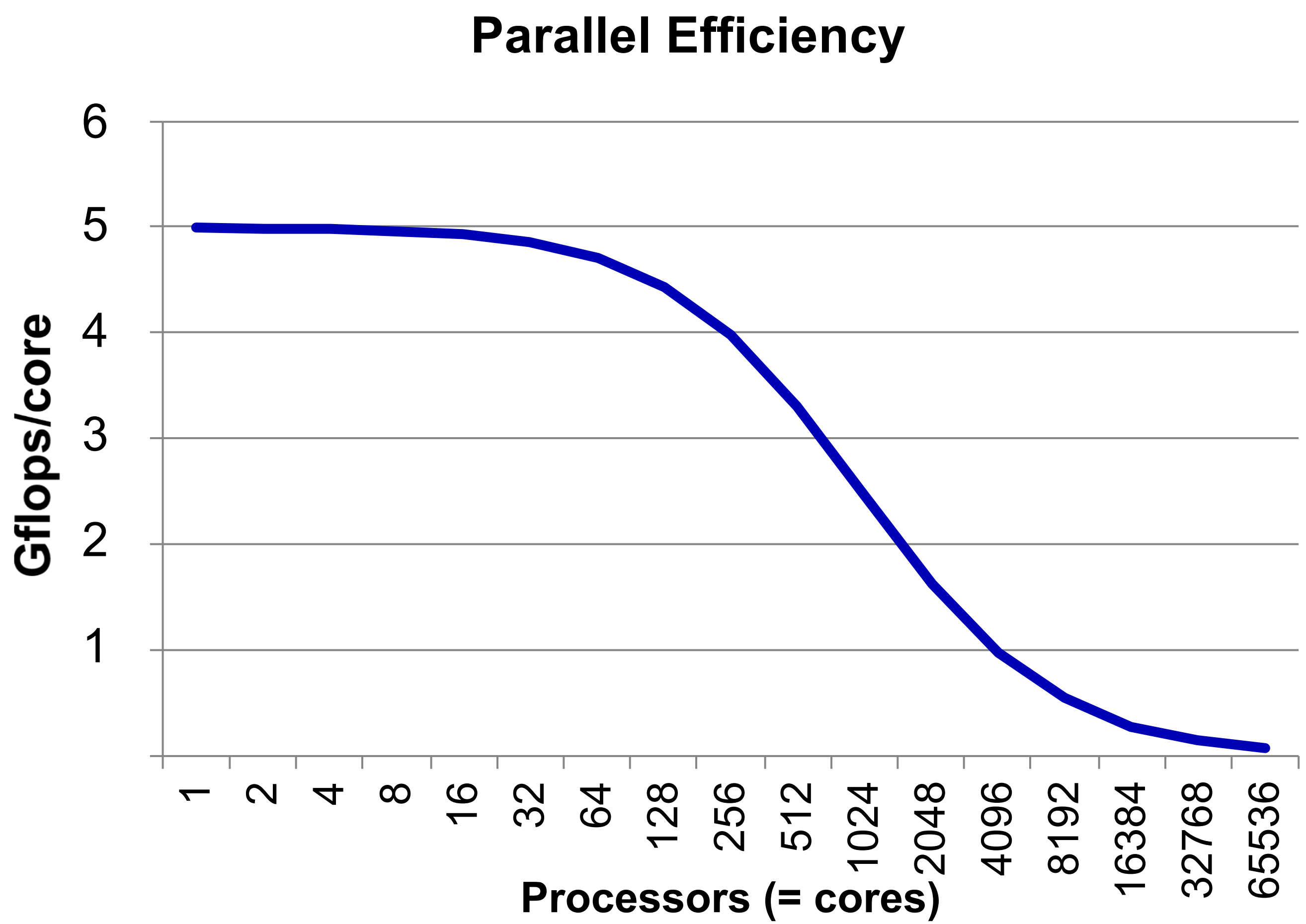
# Amdahl's Law

- Suppose only part of an application is parallel
- Amdahl's law
  - **s** = fraction of work done sequentially (Amdahl fraction), so (1-s) is fraction parallelizable
  - **P** = number of processors

$$\text{Speedup}(P) = \text{Time}(1)/\text{Time}(P) \le 1/(s + (1-s)/P) \le 1/s$$

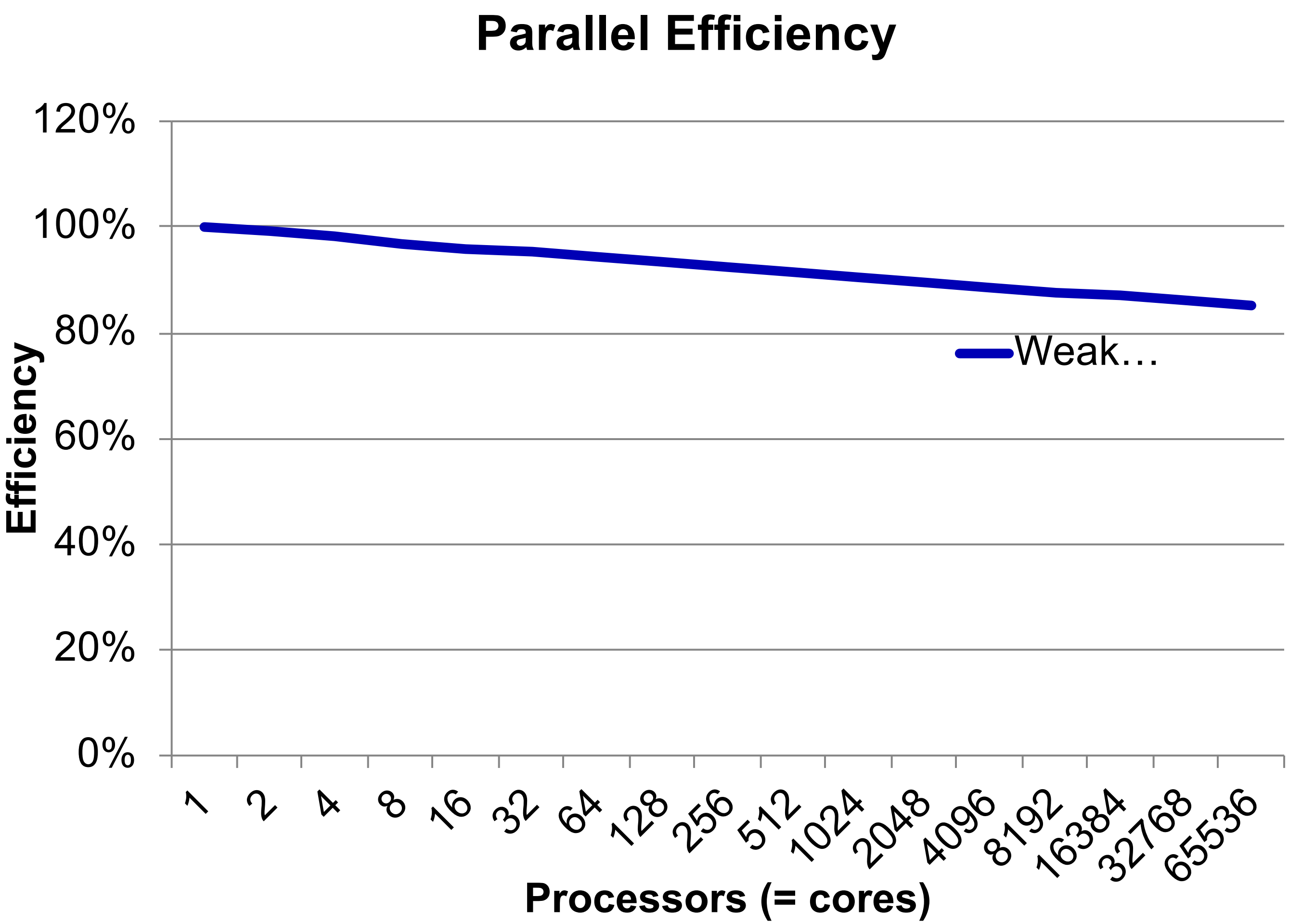
- Even if the parallel part speeds up perfectly performance is limited by the sequential part
- 1/10$^{th}$ of your code's runtime is serial → max speedup is 10x (Cori has 65K cores)

# Parallel Efficiency




Same **strong scaling** results shown as efficiency

# Parallel Efficiency (Weak Scaling)




**Weak scaling** uses a fixed problem size **per processor.** Can report as:

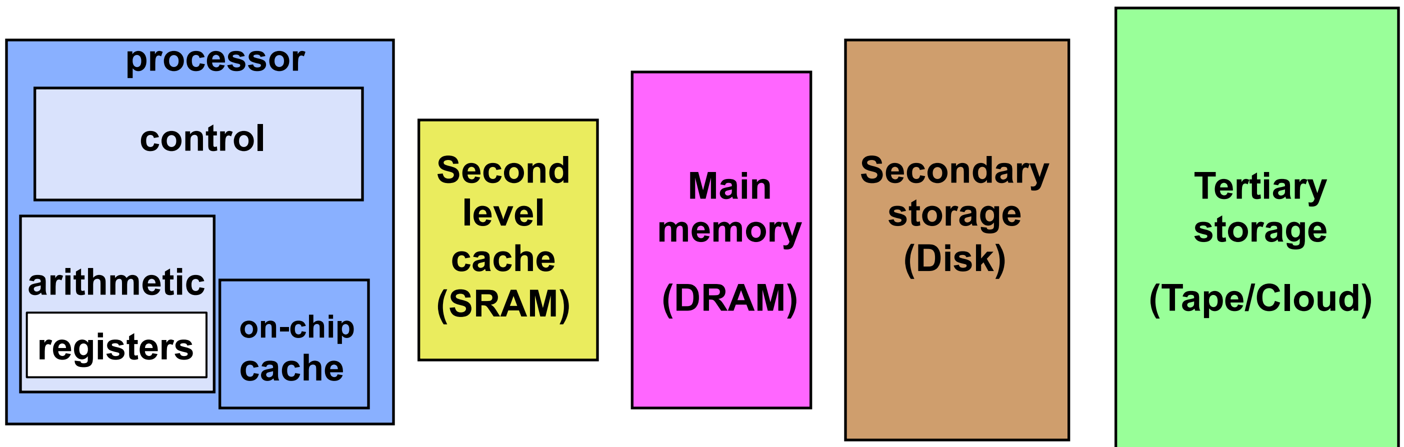
- Flop/s (or other rate) per processor; Efficiency based on rate per processor
- Time (if algorithm is linear in data size)

# Limits to Parallelism

- Besides Amdahl's Law, here are some other limits
- Parallelism overheads include:
  - cost of creating parallelism (starting a thread/process)
  - cost of communicating shared data
  - cost of synchronizing
  - extra (redundant) computation
- Overheads can be in milliseconds..
- Tradeoff:
  - Algorithm needs sufficiently large units of work to run fast in parallel (i.e. large granularity),
  - But not so large that there is not enough parallel work

# Limits to Performance Start at Home
## (Single Processor Memory Hierarchy)

- Large memories are slow; fast memories are small
- Most programs have a things nearby previous accesses
  - **temporal locality:** reusing an item that was previously accessed
- Memory hierarchy use this to improve *average case*




| | | | | | |
|---|---|---|---|---|---|
| Speed | 1ns | 10ns | 100ns | 10ms | 10sec |
| Size | KB | MB | GB | TB | PB |

# What’s in this course?

What should you learn + some mechanics

# Overview of the course (not in order)

- Parallel Programming Models and Machines (plus some architecture, e.g., caches)

| Algorithm/machine model | Language / Library skills |
|---|---|
| Shared memory | OpenMP |
| | PGAS |
| Distributed memory | MPI |
| Data parallel | SPARK |
| | CUDA |

- Parallelization Strategies for the "Motifs" of Scientific Computing (and Data)

| | |
|---|---|
| Dense Linear Algebra | Monte Carlo |
| Sparse Linear Algebra | Spectral Methods |
| Particle Methods | Graphs |
| Structured Grids | Sorting |
| Unstructured Grids | Hashing |

- Performance models: Roofline, α-β (latency/bandwidth), LogP
- Cross-cutting: Communication avoiding, load balancing, hierarchical algorithms, autotuning, Moore's Law, Amdahl's Law, Little's Law

# Course Mechanics

- **Web page:** https://sites.google.com/lbl.gov/cs267-spr2019/
  - Office hours will be posted there
- **Bcourses:** https://bcourses.berkeley.edu/courses/1480197
  - for homework submissions and grades
- **Piazza**: https://piazza.com/class/jr7v03k3lcd62s
  - for questions (XSEDE students should use Moodle)
- **Grading:**
  - Three team-based programming assignments (50% of grade)
    - HW0 (1%) Write a description of a parallel application of interest (see Website) To Do
    - HW1 (10%), HW2 (10+10+9%), HW3 (10%)
  - Final projects (50% of grade)
    - Could be parallelizing an application, building or evaluating a tool, etc.
    - We encourage interdisciplinary teams, as with much parallel scientific software
- **Fill out survey** (see Piazza) To Do
- **Computer accounts at NERSC** (Edison and Cori) To Do
  - https://nim.nersc.gov/nersc_account_request.php
    - Description: Course work in CS267; institution University of California, Berkeley)

# Reading Materials

- Must read:
  - The Landscape of Parallel Processing Research: The View from Berkeley
    - http://www.eecs.berkeley.edu/Pubs/TechRpts/2006/EECS-2006-183.pdf
- Some on-line texts:
  - Demmel's notes from CS267 Spring 1999, also linked to 1996 html notes.
    - http://www.cs.berkeley.edu/~demmel/cs267_Spr99/
  - Ian Foster's book, "Designing and Building Parallel Programming".
    - http://www-unix.mcs.anl.gov/dbpp/
- Potentially useful texts:
  - *Sourcebook for Parallel Computing*, by Dongarra, Foster, Fox, ..
    - A general overview of parallel computing methods
  - *Performance Optimization of Numerically Intensive Codes* by Goedecker and Hoisie
    - Practical guide to performance optimization
  - *Petascale Computing, Algorithms and Applications*, Bader et al, Chapman & Hall/CRC, 2007
  - *Parallel Processing for Scientific Computing*, ed. by Heroux, Ragahvan, Simon, SIAM, 2006.

# What you should get out of the course

**Understanding of:**

- Motivation for parallel computing
- Performance analysis and performance programming
- Parallel computing hardware options
- Importance of locality in performance and parallelism
- Overview of programming models (software) and tools, and experience using some of them
- Overview of parallel applications in simulation and data analysis
- Exposure to various open research questions

**Terminology from Today:**

- Speedup, Strong/Weak Scaling, Amdahl, Top500