

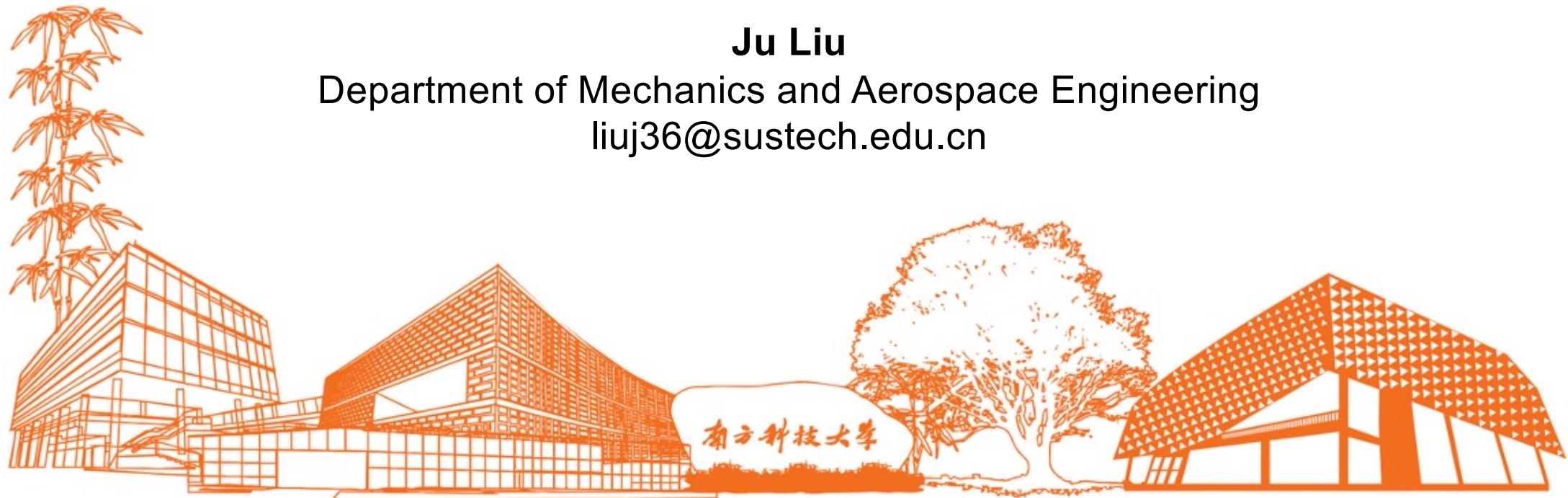
MAE 5032 High Performance Computing: Methods and Applications

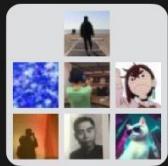
Lecture 1: Introduction

Ju Liu

Department of Mechanics and Aerospace Engineering

liuj36@sustech.edu.cn





群聊: MAE5032 HPC



该二维码7天内(2月24日前)有效, 重新进入将更新

What is it?

HPC

- High Performance Computing (HPC) refers to the practice of aggregating computing power in a way that delivers much higher performance than one could get out of a typical desktop computer or workstation in order to solve large problems in science, engineering, medicine, or business.
- It involves knowledges on hardware, software, applied mathematics, and the application background.

The need for speed



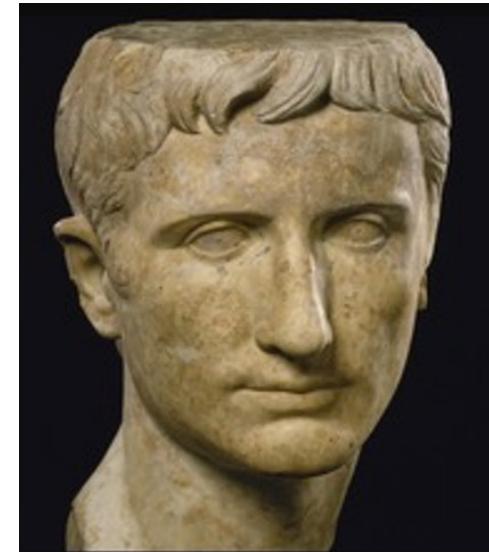
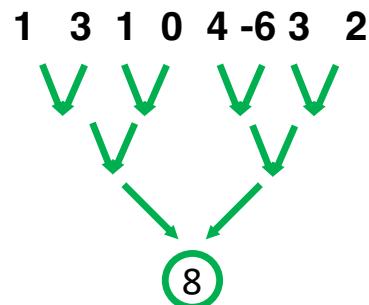
The need for speed

- Add n values

Serial: $O(n)$

Parallel: $O(\log n)$

Uses n processors!



- Takes advantage of associativity in +

Well done is quickly done.
- Augustus

The need for speed

Training compute (FLOPs) of milestone Machine Learning systems over time

n = 121

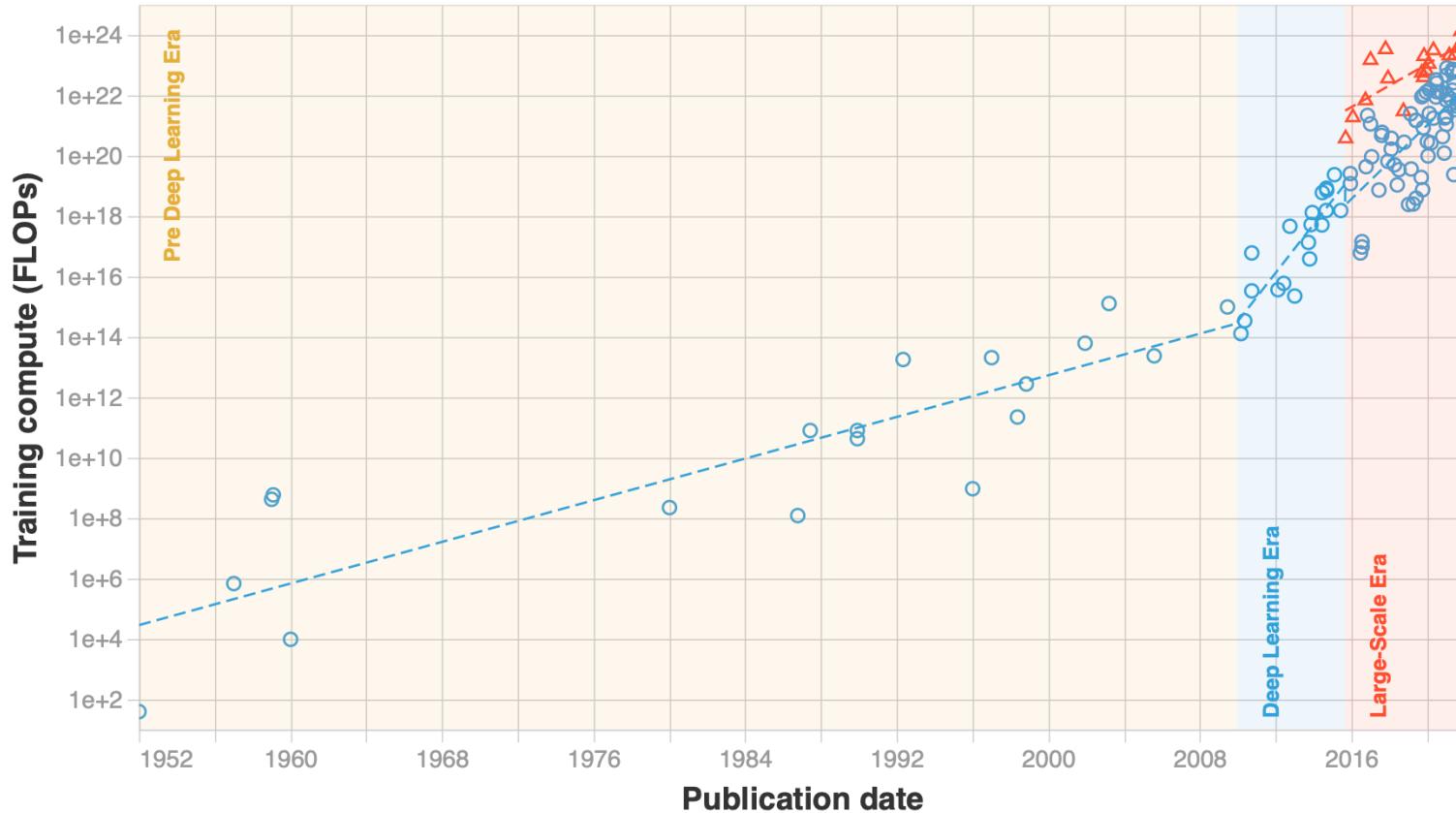
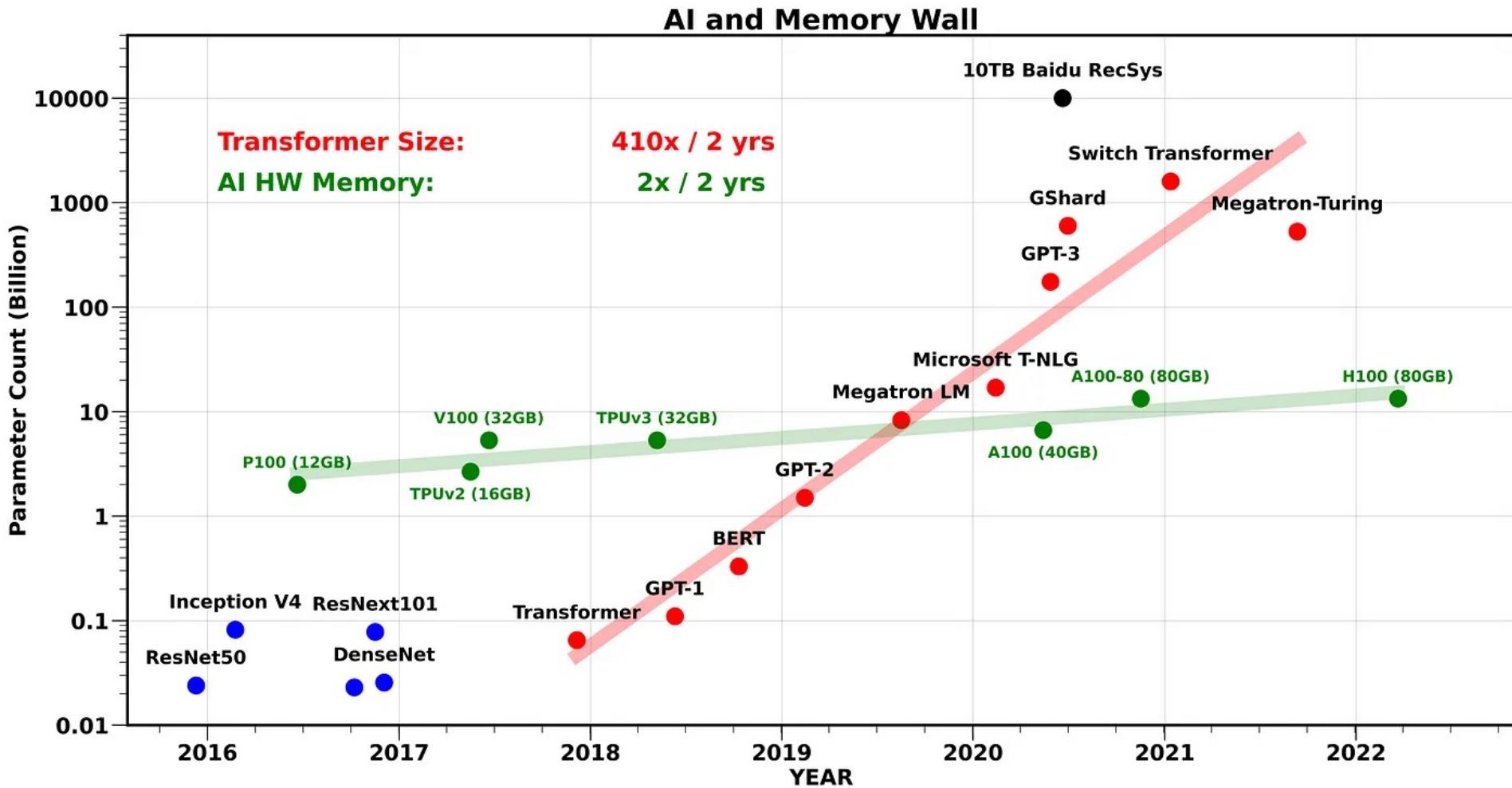


Figure 1: Trends in $n = 121$ milestone ML models between 1952 and 2022. We distinguish three eras. Notice the change of slope circa 2010, matching the advent of Deep Learning; and the emergence of a new large-scale trend in late 2015.

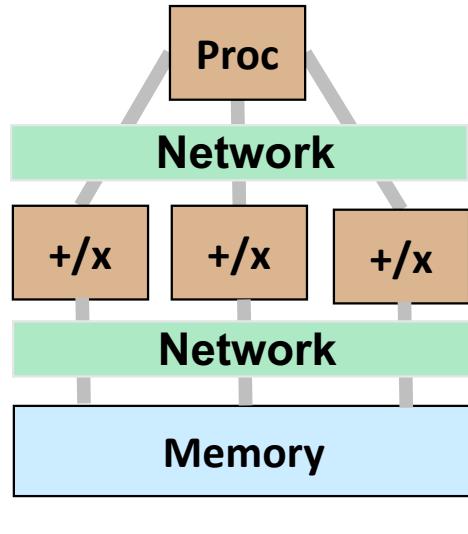
Source: OpenAI

The need for memory



Source: <https://medium.com/riselab/ai-and-memory-wall-2cb4265cb0b8>

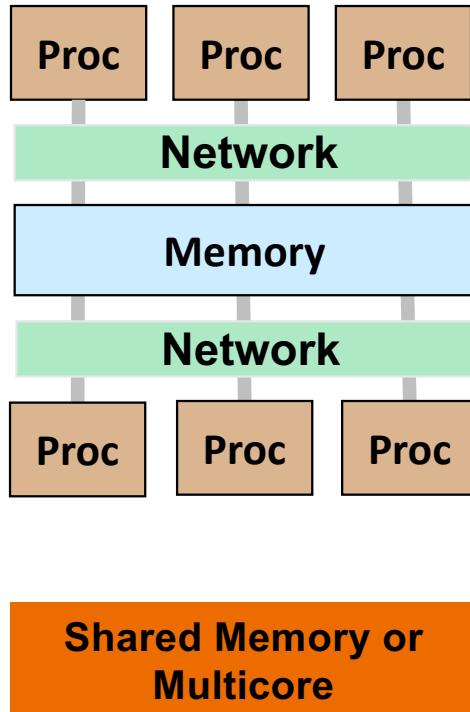
What is a parallel computer



**Single Instruction
Multiple Data (SIMD)**

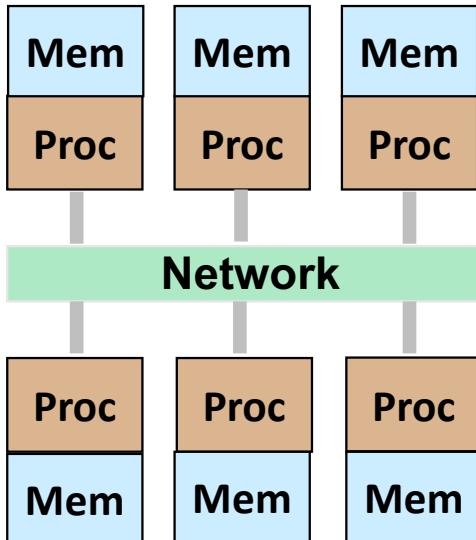
- A **Single Instruction Multiple Data (SIMD)** computer has multiple processors (or functional units) that perform the same operation on multiple data elements at once
- Most single processors have **SIMD units** with ~2-8 way parallelism
- Vector machine and Graphics processing units (**GPUs**) use this

What is a parallel computer



- A **shared memory multiprocessor** (SMP) by connecting multiple processors to a single memory system
- A **multicore processor** contains multiple processors (cores) on a single chip

What is a parallel computer



- A **distributed memory multiprocessor** has processors with their own memories connected by a high speed network
- Also called a **cluster**
- A **high performance computing (HPC)** system contains 100s or 1000s of such processors (nodes)

Distributed Memory

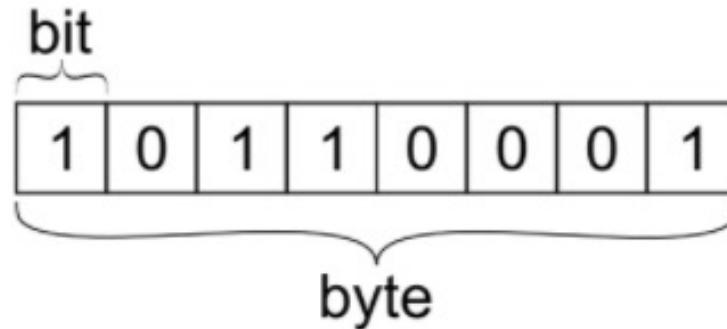
An Overview of the current fastest computers

The Top500 List



Units of measure in 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;



Units of measure in 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; one byte contains 8 bits);
- Typical sizes are millions, billions, trillions...

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

Current fastest machines are eta flop systems.

The TOP500 Project

- Listing the 500 most powerful (public) computers in the world
- Yardstick: Floating Point Operations per Second (FLOP/s) Rmax of Linpack
 - Solve $Ax = b$, with the matrix A being a dense matrix with random entries
 - Dominated by dense matrix-matrix multiply
- Updated twice a year:
 - ISC'xy in June in Germany
 - SCxy in November in US
- All information available from the TOP500 website at www.top500.org



Top 10 of the Top500, Nov. 2024

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)	Clock rate: the frequency at which the transistor switches on and off. Typically between 1 an 3 GHz.					
1	El Capitan - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	11,039,616	1,742.00	2,746.38	29,581	6	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
2	Frontier - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE Cray OS, HPE DOE/SC/Oak Ridge National Laboratory United States	9,066,176	1,353.00	2,055.72	24,607	7	Alps - HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11, HPE Cray OS, HPE Swiss National Supercomputing Centre (CSCS) Switzerland	2,121,600	434.90	574.84	7,124
3	Aurora - HPE Cray EX - Intel Exascale Compute Blade, Xeon CPU Max 9470 52C 2.4GHz, Intel Data Center GPU Max, Slingshot-11, Intel DOE/SC/Argonne National Laboratory United States	9,264,128	1,012.00	1,980.01	38,698	8	LUMI - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,752,704	379.70	531.51	7,107
4	Eagle - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure Microsoft Azure United States	2,073,600	561.20	846.84		9	Leonardo - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, EVIDEN EuroHPC/CINECA Italy	1,824,768	241.20	306.31	7,494
5	HPC6 - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, RHEL 8.9, HPE Eni S.p.A. Italy	3,143,520	477.90	606.97	8,461	10	Tuolumne - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, TOSS, HPE DOE/NNSA/LLNL United States	1,161,216	208.10	288.88	3,387

More of the Top500, Nov. 2021

433	AFI-NITY - PRIMERGY CX2550 M4, Xeon Gold 6148 20C 2.4GHz, Infiniband EDR, Fujitsu Tohoku University, Institute of Fluid Science Japan	35,200	1,691.0	2,703.4		
434	TaiYi - ThinkSystem SD530, Xeon Gold 6148 20C 2.4GHz, Intel Omni-Path, Lenovo Southern University of Science and Technology China	32,400	1,686.5	2,488.3		
435	NA-IT1 - ZettaScaler3.0, AMD EPYC 7702P 64C 1.5GHz, PEZY-SC3, Infiniband EDR, PEZY Computing / Exascaler Inc. NA Simulation Japan	822,400	1,684.8	2,353.8	22	

Frontier (#2 machine) system overview

- Peak performance of 2.1 double precision exaFLOPS Measured Top500
- performance (Rmax) was 1.35 exaFLOPS

- Each node has
- 3rd Gen AMD CPU with 64 cores
 - 4 AMD Instinct 250X GPUs
 - 4X128 GB of fast memory, 1 per GPU
 - 5 Terabytes of flash memory

- The system has
- 9472 nodes
 - Dual-rail Mellanox EDR InfiniBand network
 - 250 PB IBM Spectrum Scale file system transferring data at 2.5 TB/s



TianHe-2A (#24 machine) system overview

- Peak performance of 100 petaflops for modeling & simulation
- Power 18482 kW
- OS: Kylin Linux

- Each node has
- 24 Xeon E5 CPU 2.2GHz
 - 3 x 57 coprocessors (Xeon phi / Matrix-2000)
 - $64 + 3 \times 8 = 88$ GB Memory

- The system has
- 16000 compute nodes
 - Self-developed TH express-2
 - 12 PB file system



TaiYi system overview

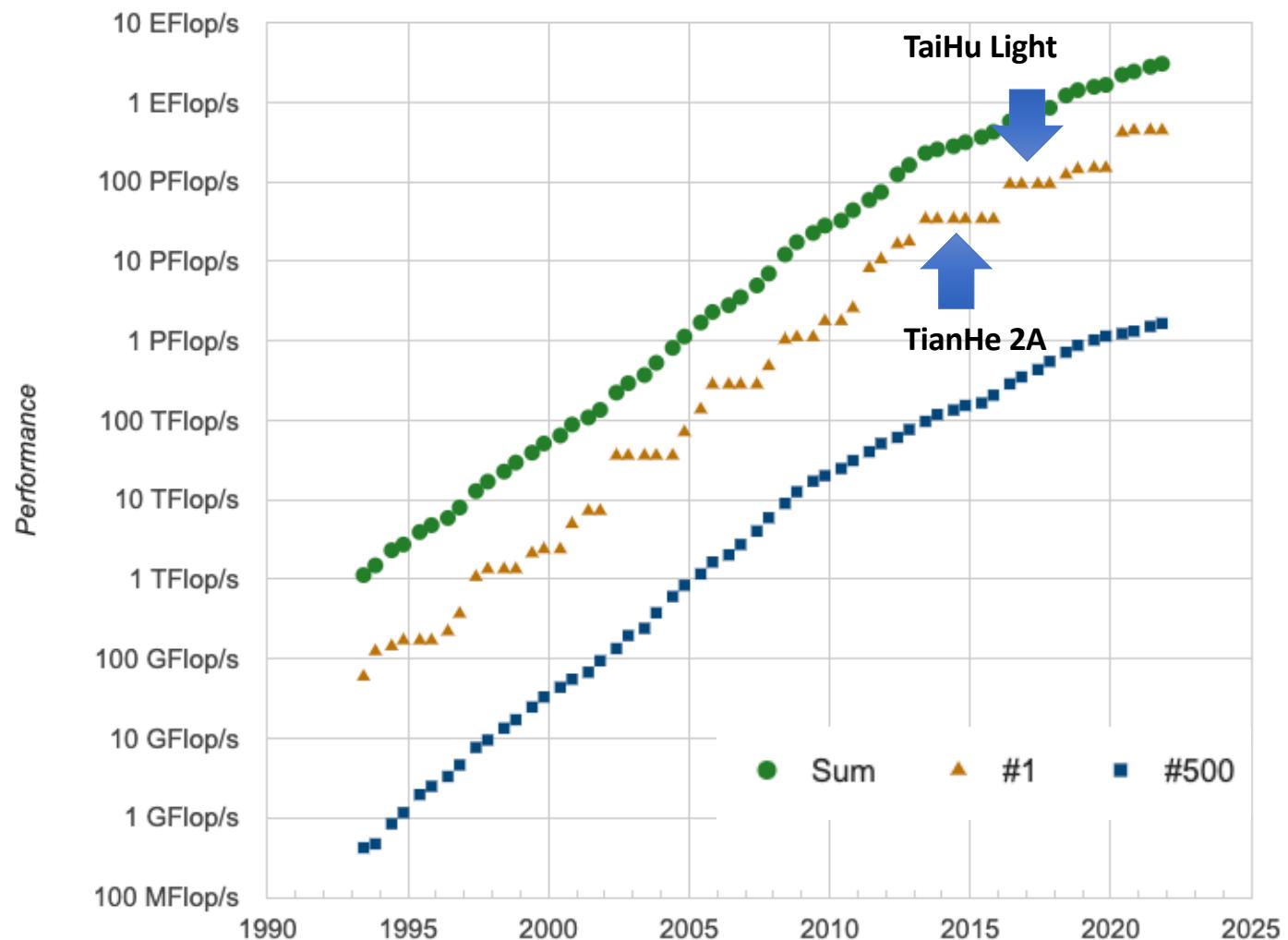
- Peak performance of 2.5 petaflops for modeling & simulation

- Each node has
- 2 Xeon Gold 6140 CPU
 - 384 GB Memory

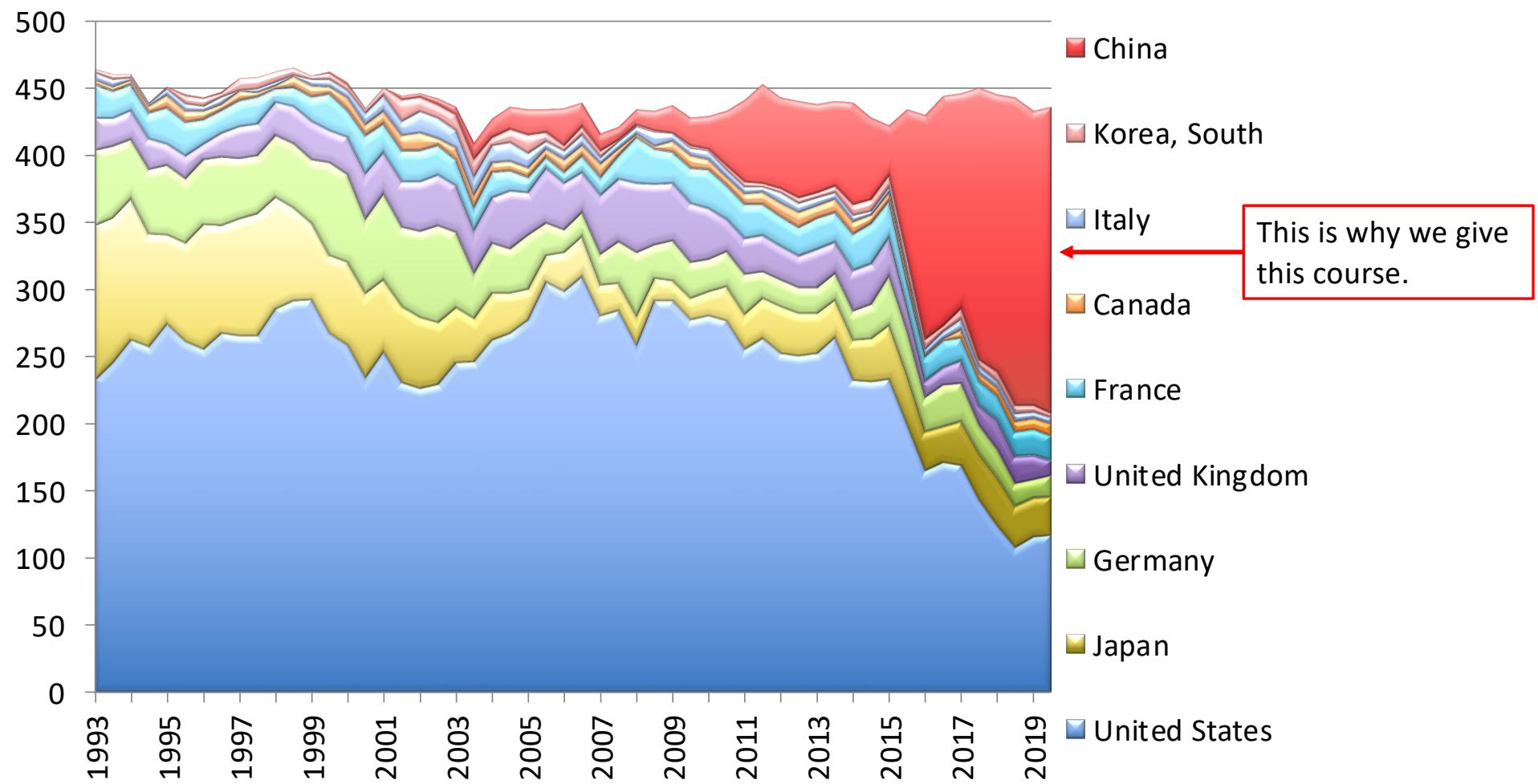
- The system has
- 815 compute nodes
 - Intel Omni-path network with 100 Gbps



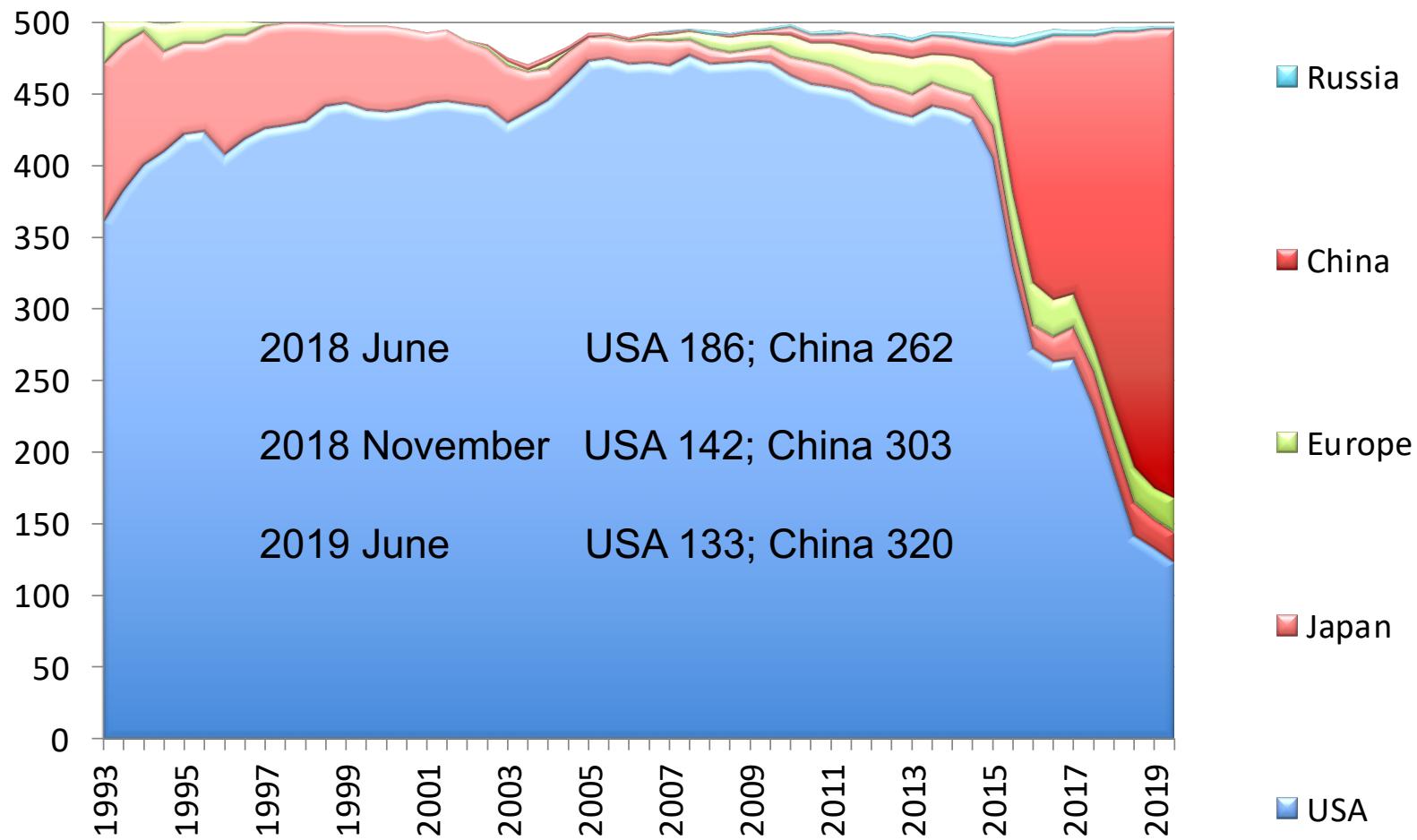
Performance over time



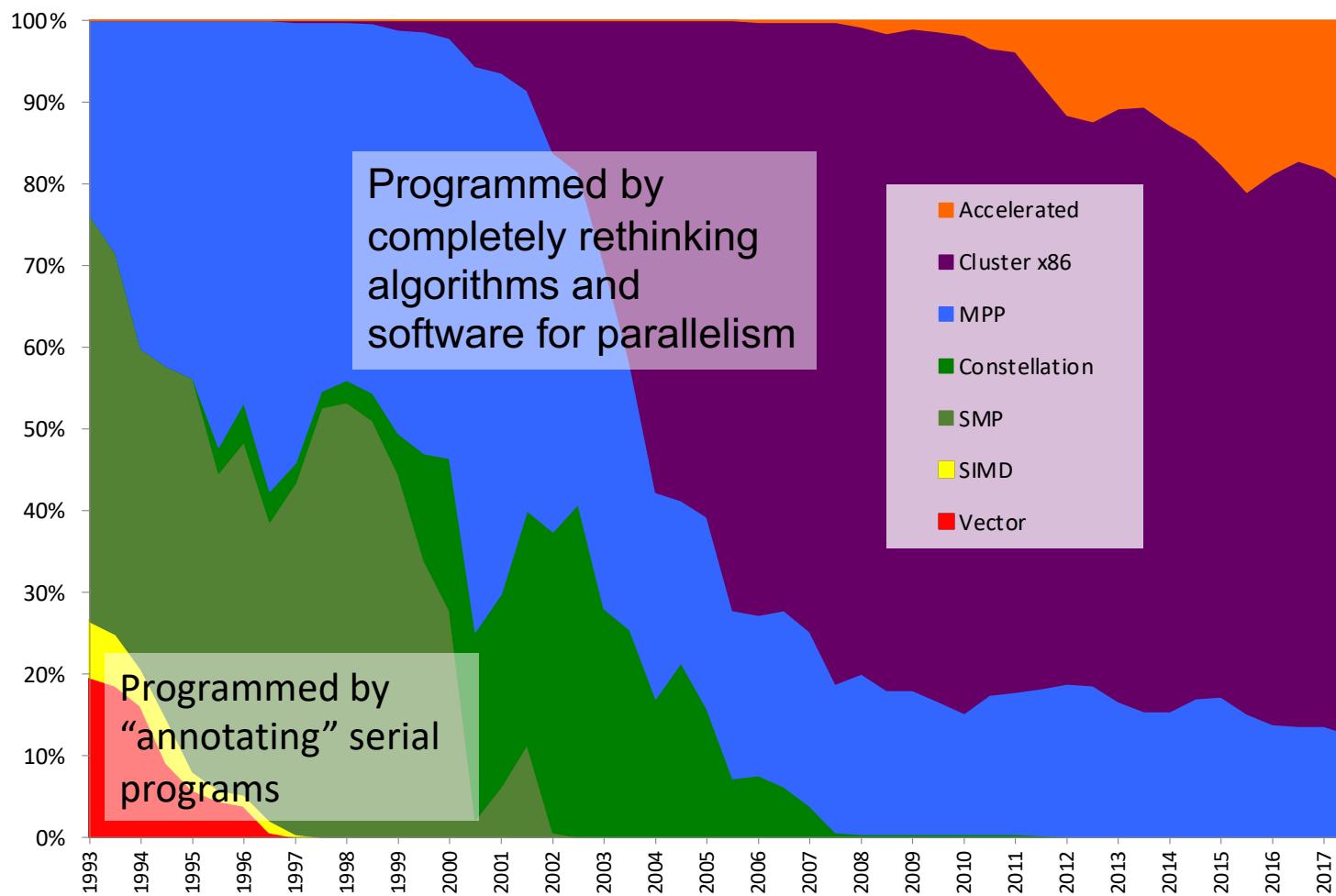
Countries



Producers

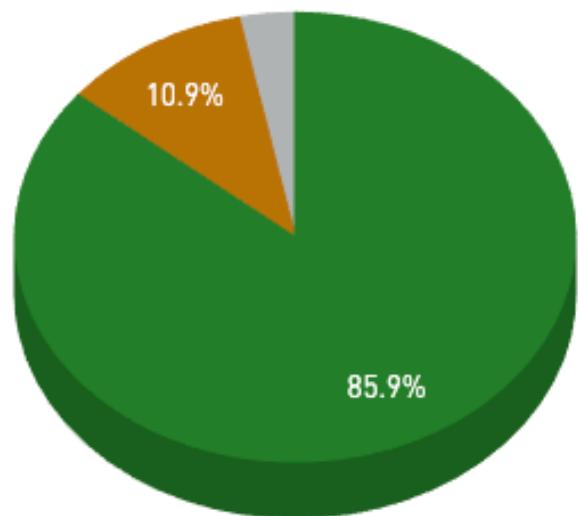


From vector supercomputers to massively parallel accelerator systems



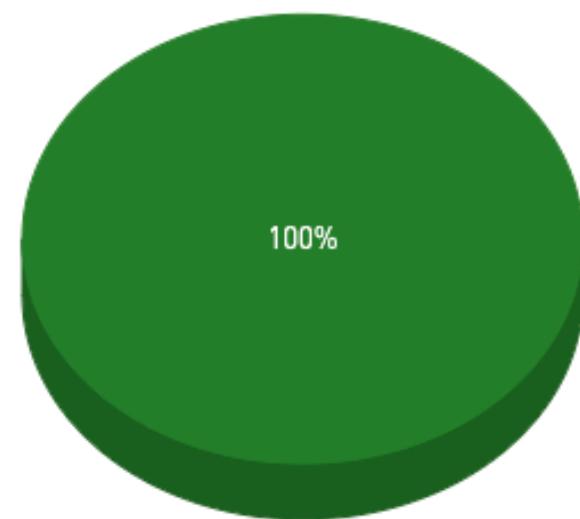
Operating system

Operating system Family System Share



Nov. 2000

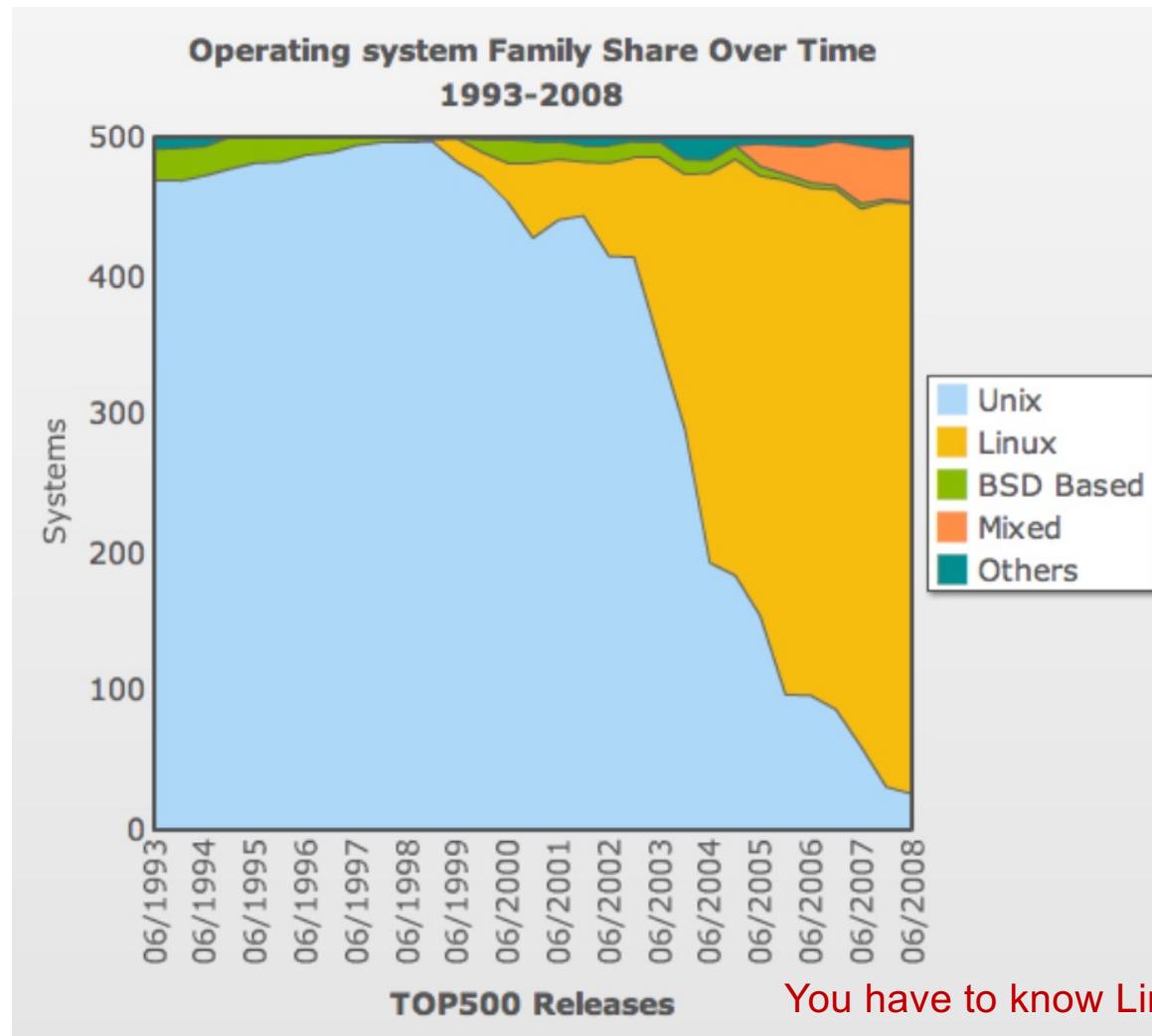
Operating system Family System Share



Nov. 2024

You have to know Linux for technical computing!

Operating system



Computer accounts

- Tai-Yi
 - We have a powerful academic system on campus
 - Linux system
 - Get an account on it.
- You own machine
 - More convenient for development
 - You need a Linux installed on it
 - Mac system is fine.

Homework 0:

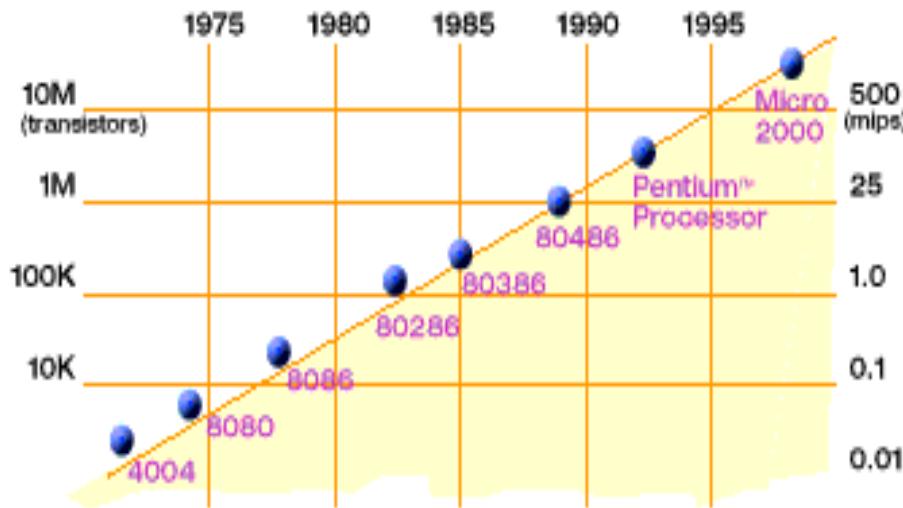
Install Linux Ubuntu 18.04 on your computer, and get access to TaiYi.
If you use Mac, let me know.

**Why all computers are parallel
computers (since 2005)**

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

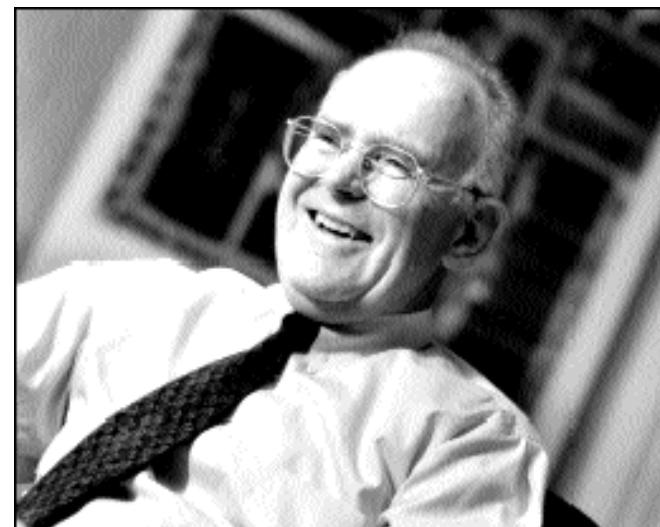
Microprocessor capacity



2X transistors/Chip Every 1.5 years

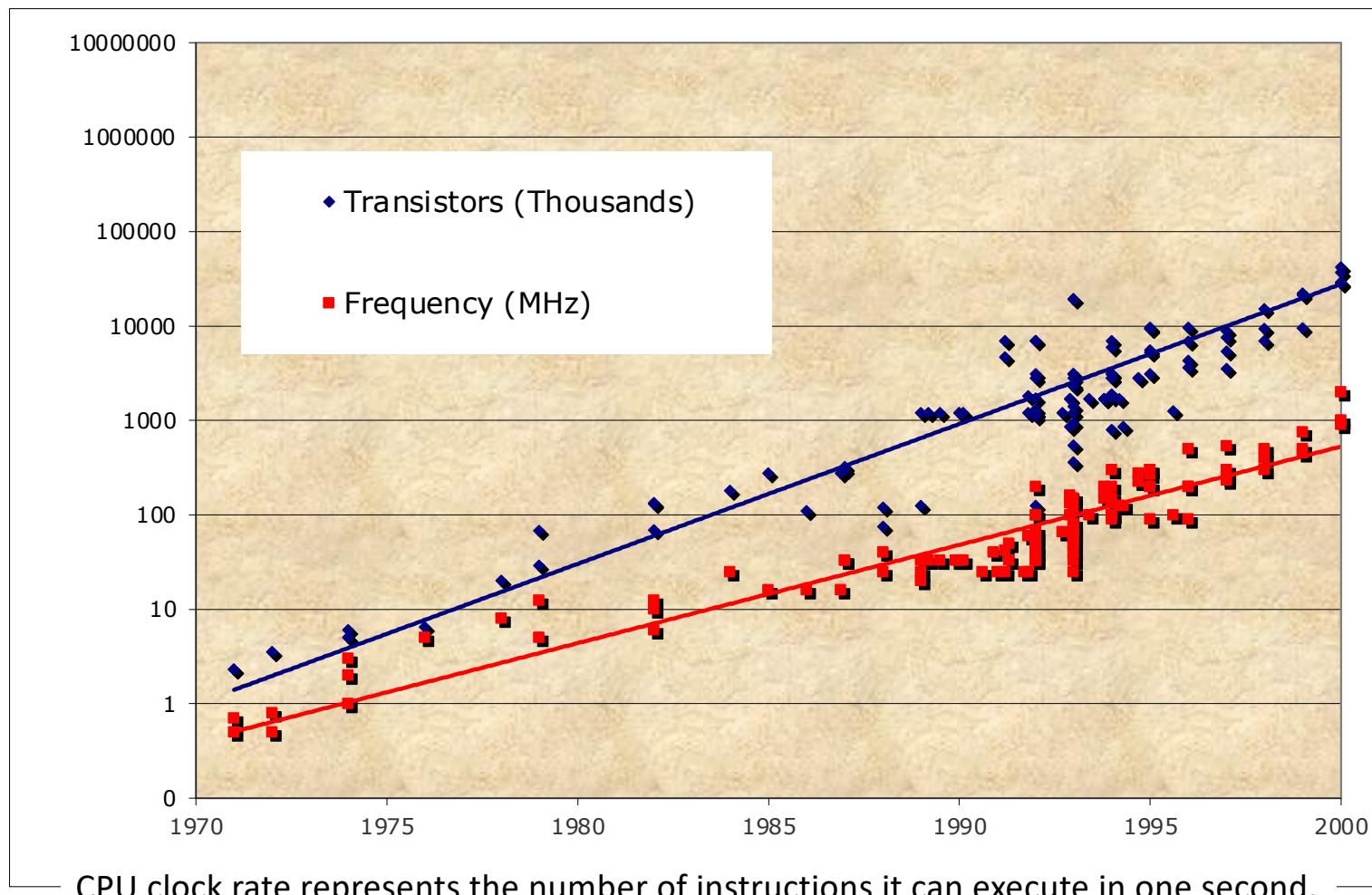
Called "Moore's Law"

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.

Microprocessor 1970 - 2000



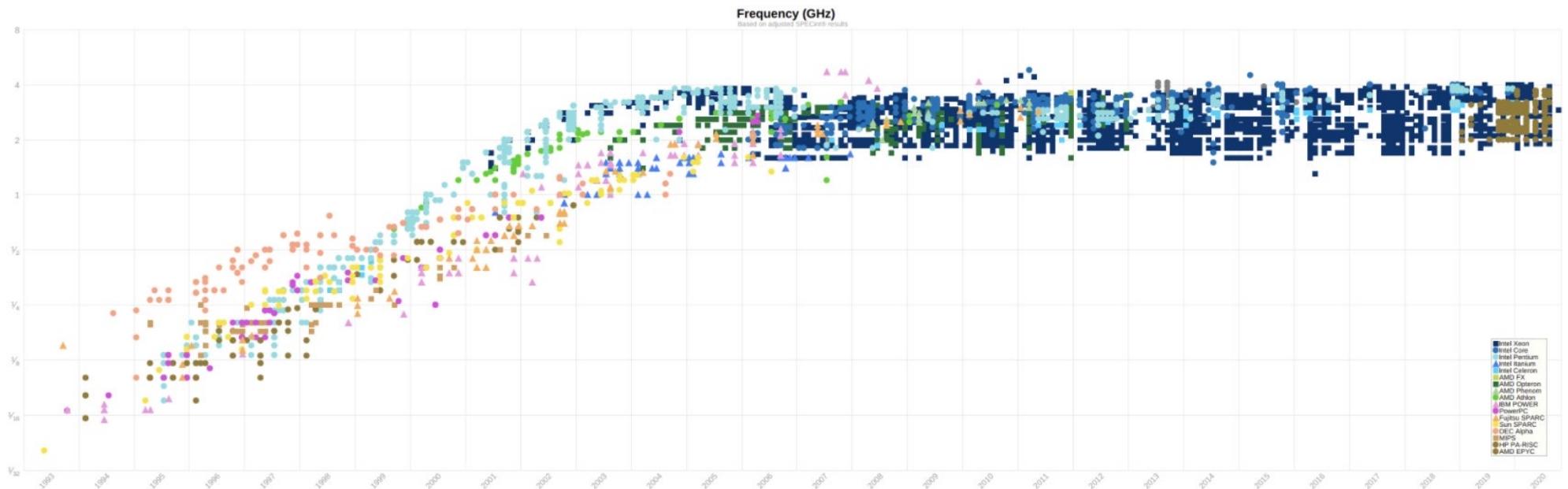
Beach strategy

Go to the beach and when you return,
your program runs faster.



Microprocessor: beyond 2000

This was not always the case; in the 80s and 90s, as transistors shrunk, they got faster according to [Dennard scaling](#). The physics is not too relevant, but the trends are.

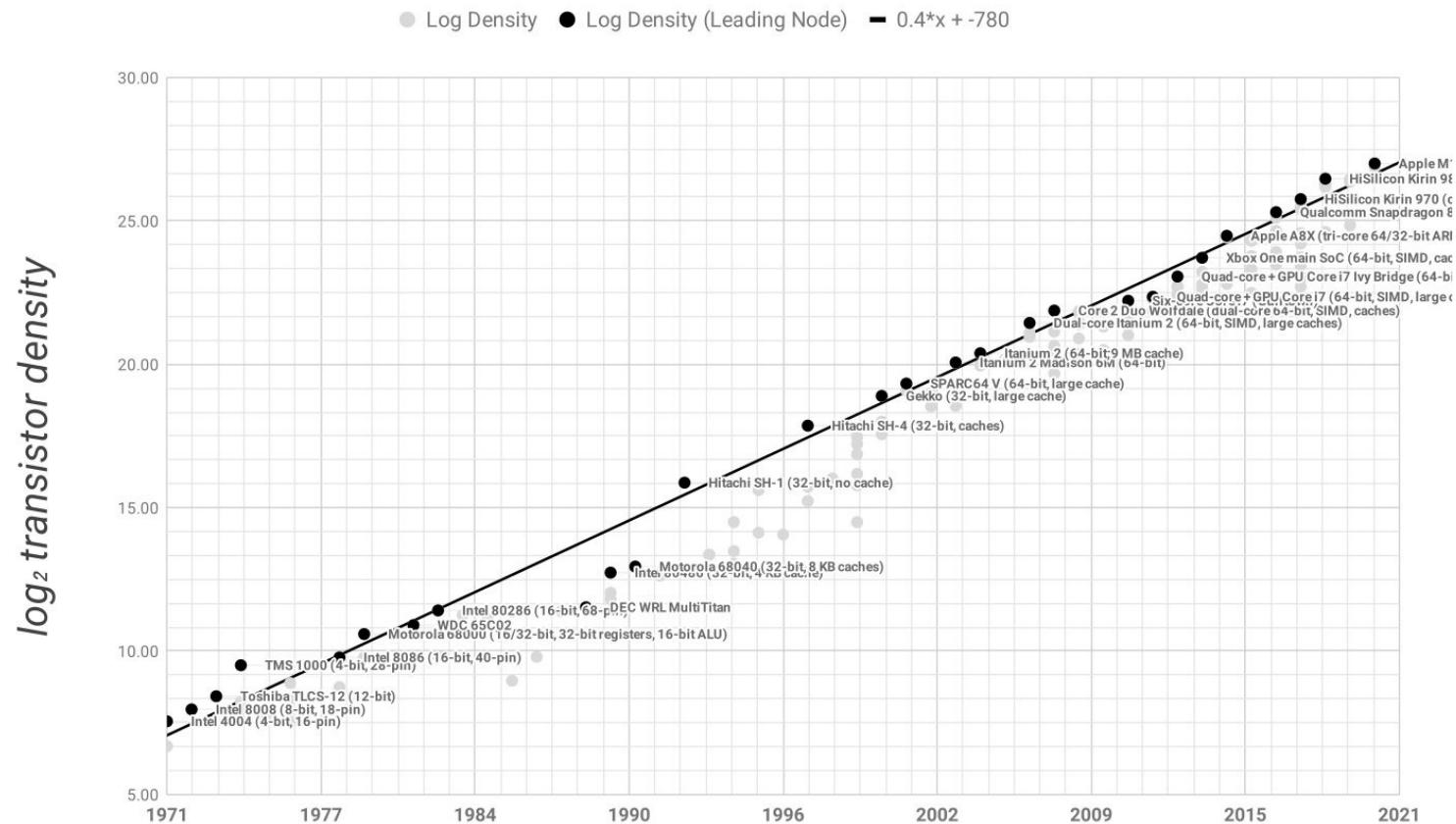


Transistors got exponentially faster until the moment they didn't.

Microprocessor: beyond 2000

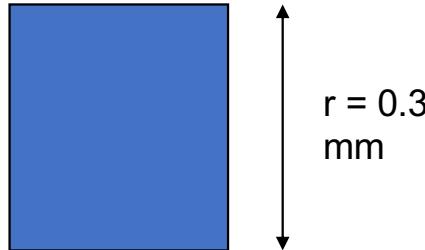
Transistor density improvements over time (CPU)

source: https://en.wikipedia.org/wiki/Transistor_count



Limits: How fast can a serial computer be?

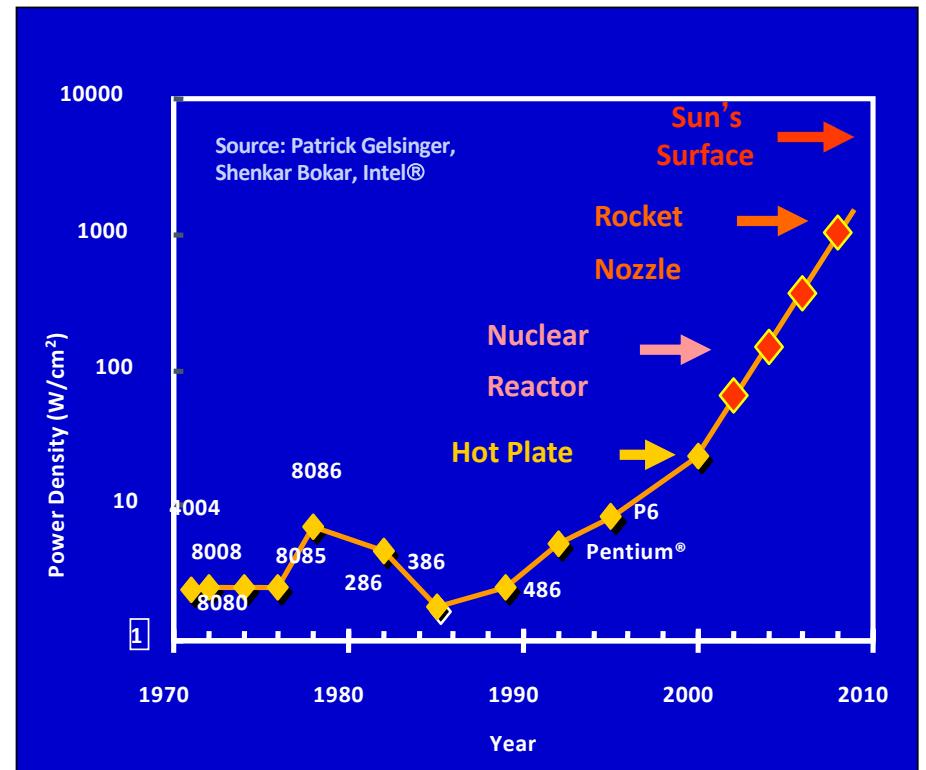
1 Tflop/s,
1 Tbyte
sequential
machine



- Consider the 1 Tflop/s (10^{12}) sequential machine:
 - Data must travel distance, r , 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

Microprocessor 1970 - 2000

- Concurrent systems are more power efficient
 - 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

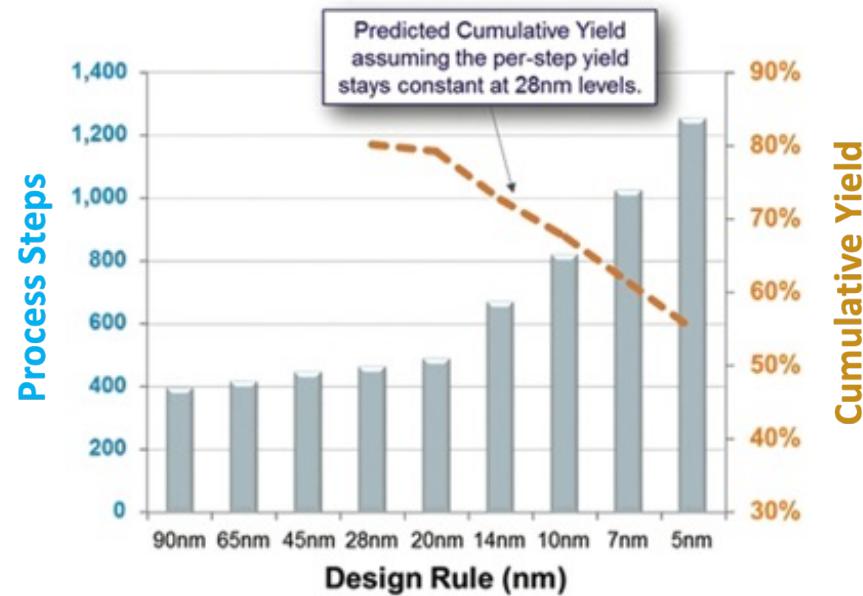


- High performance serial processors waste power
- More transistors, but not faster serial processors

Microprocessor 1970 - 2000

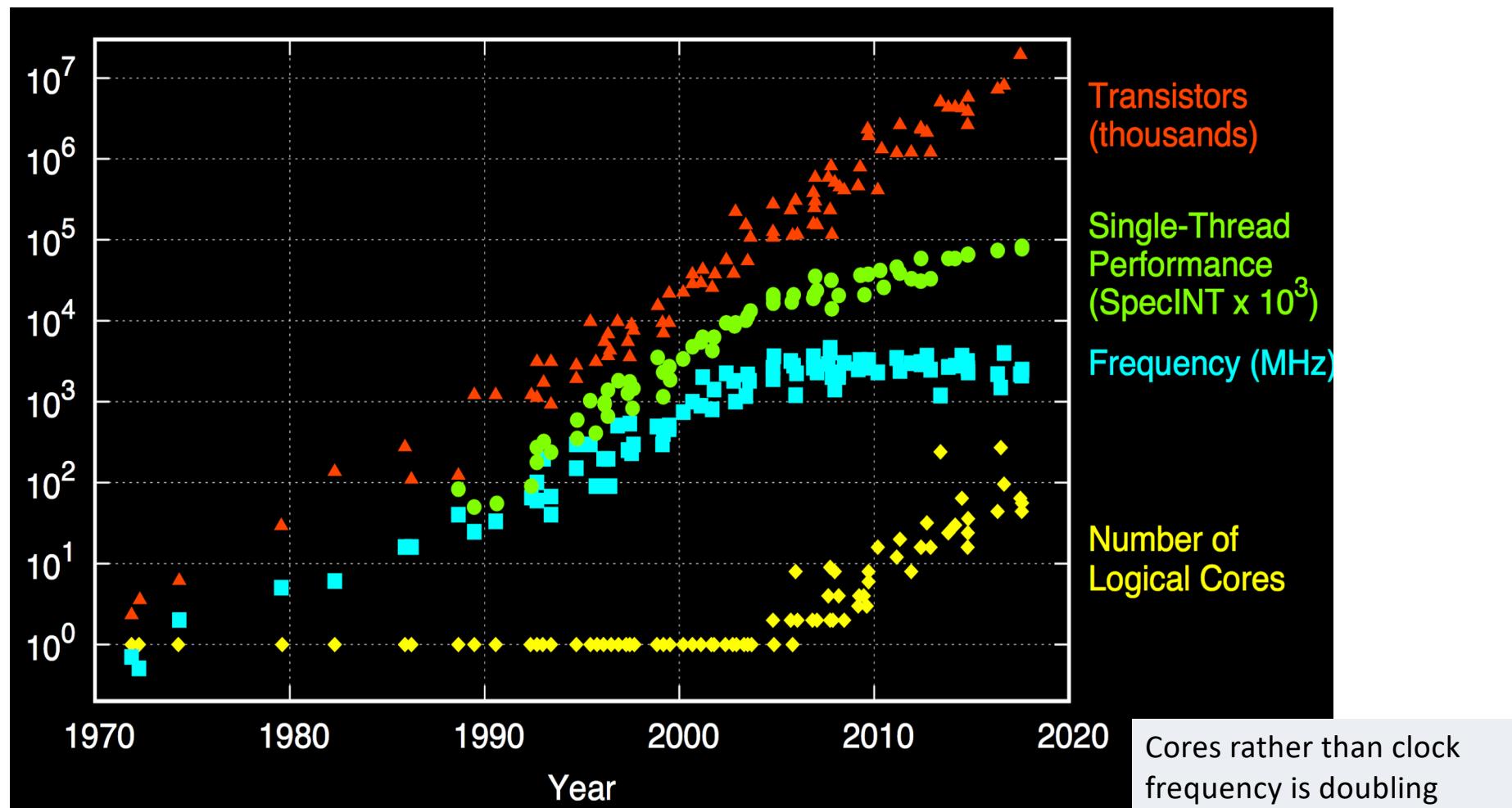
Yield

- What % of chips are usable?
- Complexity of fabrication (decreased size and number of steps) increases errors
- Parallelism helps, e.g., KNL (in Cori) sold with only 68 out of 76 “on” to improve yield
 - Serial processors are over-engineered!

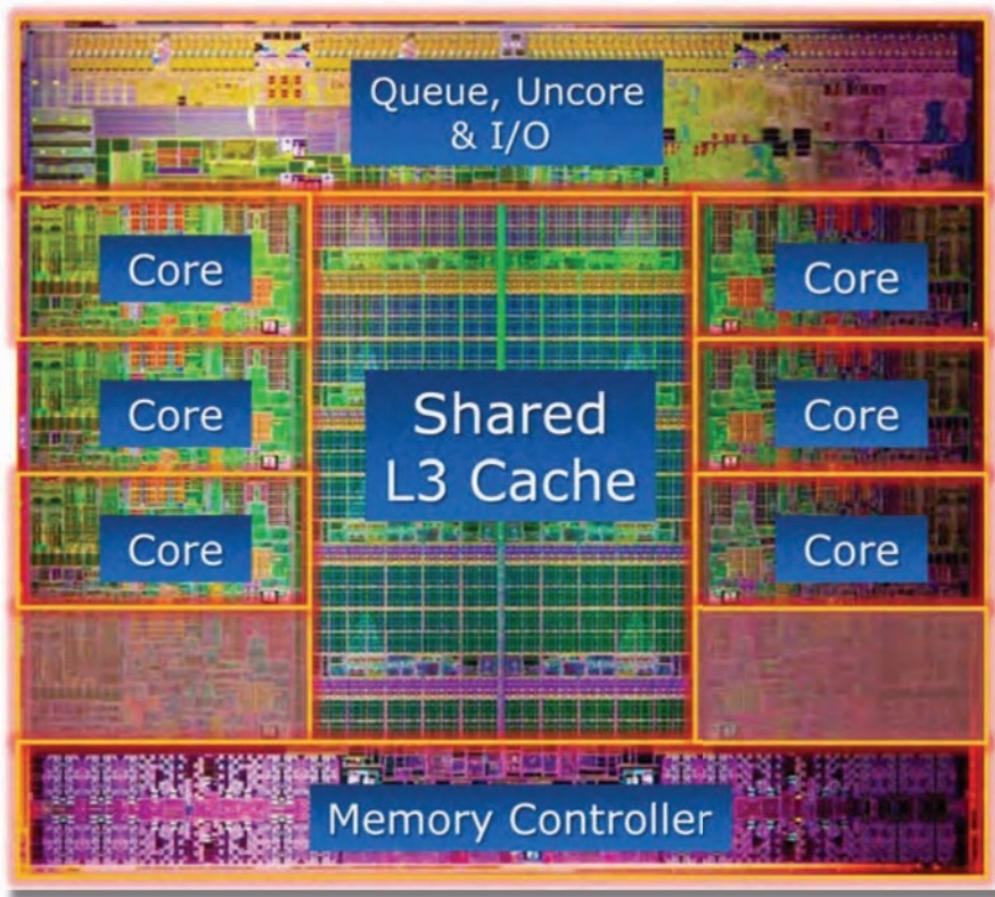


<http://electroiq.com/blog/2016/02/yield-and-cost-challenges-at-16nm-and-beyond/>

Parallelism



Multicore architecture



- Intel Core i7 Sandy Bridge E 2011
 - 6 cores
 - 3.3 GHz
 - 15 MB L3 Cache

To scale performance, manufacturer put many processing cores on the microprocessor chip.

Each generation of Moore's law potentially doubles the number of cores.

Software must be adapted to utilize this hardware efficiently.

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...

You need to know how to
write software

Gordon Bell Prizes

Established in 1987 with a cash award of \$10,000 (since 2011), funded by Gordon Bell, a pioneer in HPC.
For innovation in applying *HPC to applications in science, engineering, and data analytics.*

In 2016, 2020, and 2021, Gordon Bell prize were awarded to research teams from China.



Gordon Bell Prizes

The 2016 award goes to Chao Yang, et al. They simulated weather at 0.5 kilometer resolution with 770 billion unknowns.

Calculations were done on TaiHu Light with 10.4 million processors at a rate of 7.95 Pflops/s, versus 93 Pflops/s for Linpack.



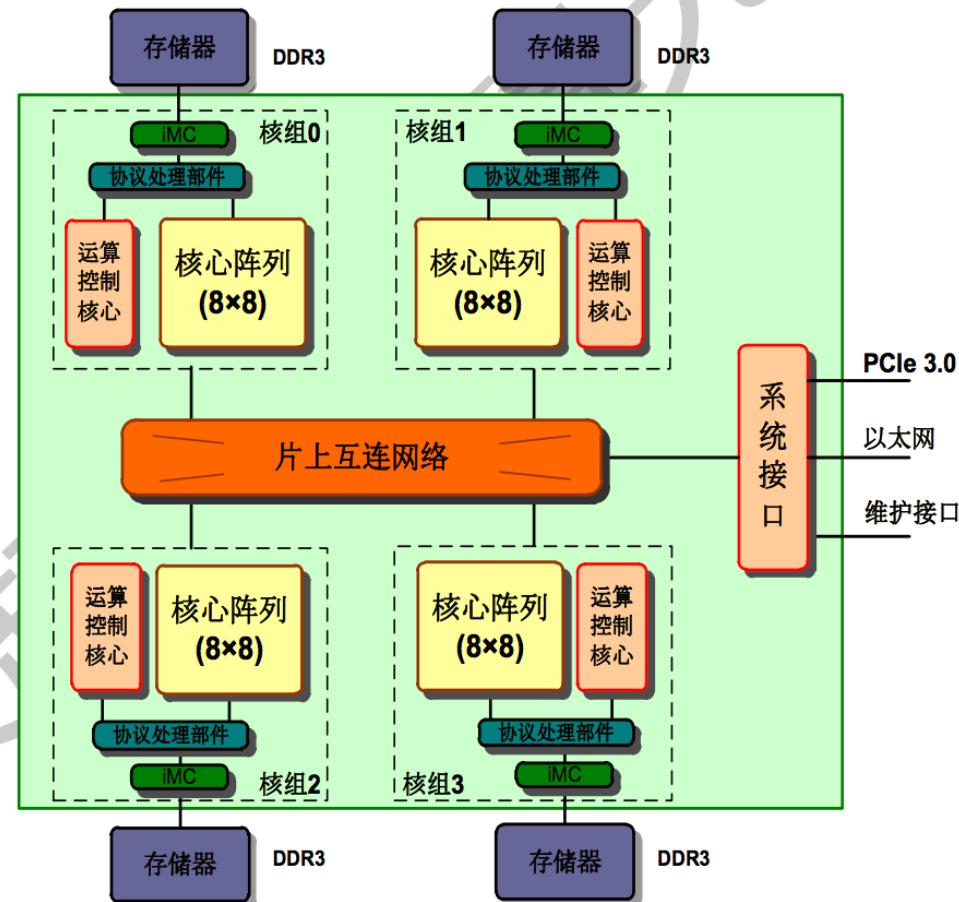
Gordon Bell Prizes

申威26010众核处理器：

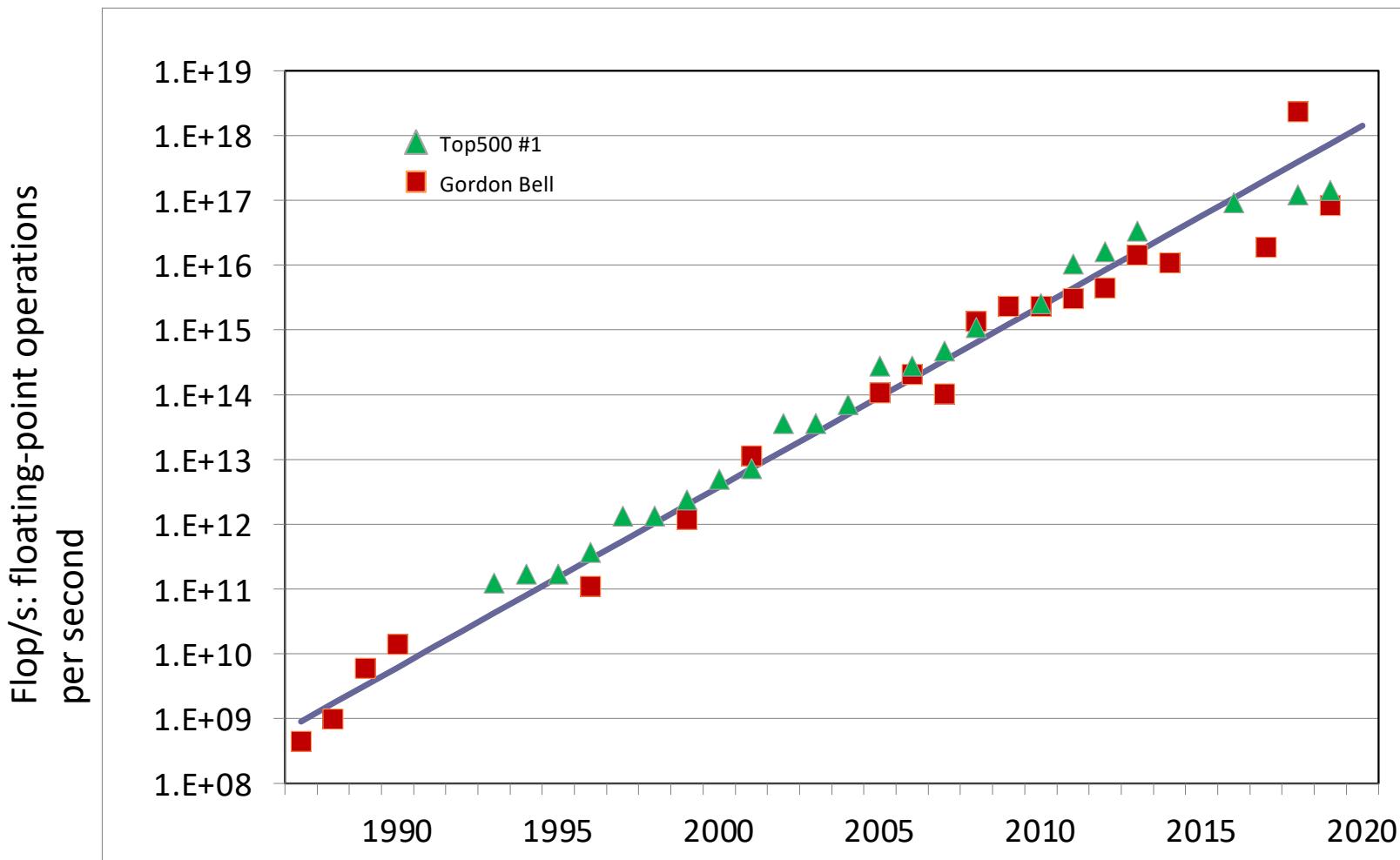
包括四个核组；
每个核组包括1个主核和
64个从核；
每个核组8GB内存。

40960个众核处理器。

You need to know (1)
how to program on a
processor board; (2) how
to coordinate 40960
processor boards.

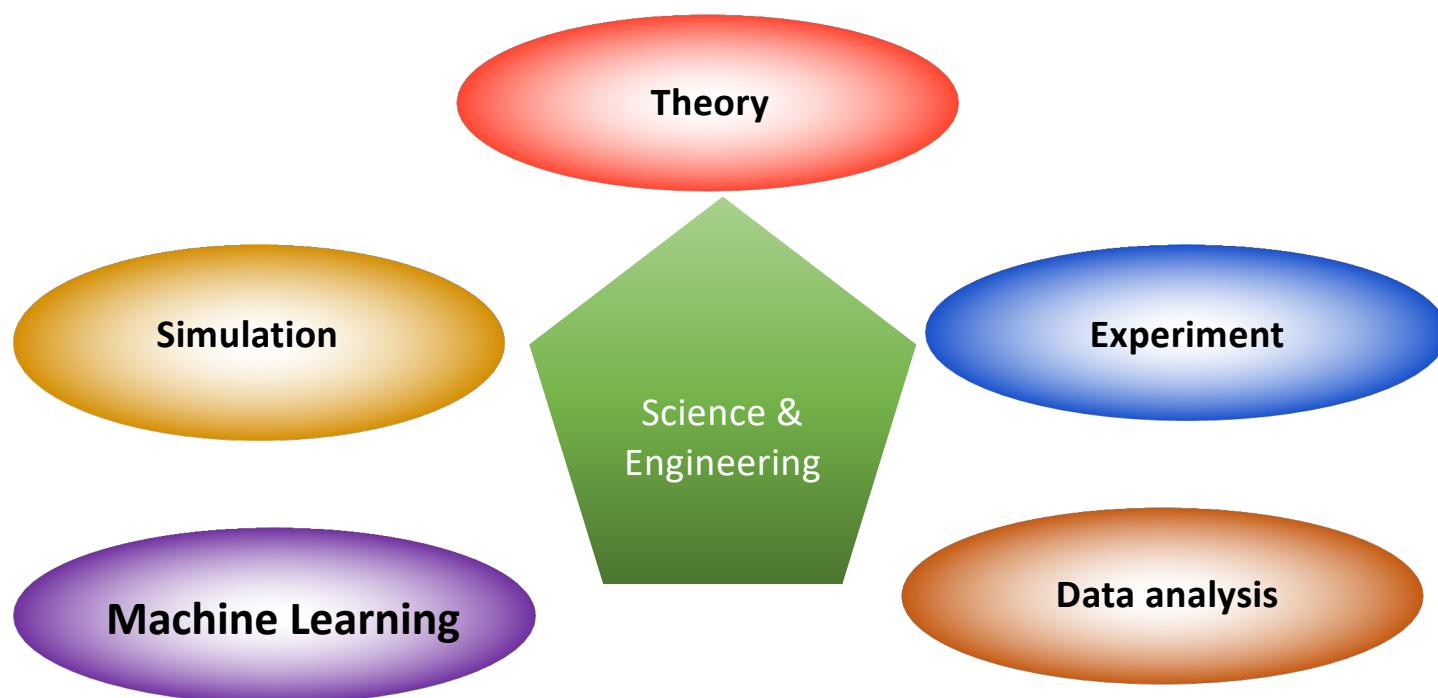


Gordon Bell Prizes



Science and Engineering Problems using HPC

The five paradigms of science & engineering



The third pillar

- In science, we use mathematics to understand physical systems
- Different fields of science explore different domains of the universe, and have their own sets of equations, encapsulated in theories
- Determining the theories, an governing equation requires observation or experimentation and testing hypotheses

The third pillar

- Why should we care about scientific computing?
 - Computational research has emerged to complement experimental methods in basic research, design, optimization, and discovery in all facets of engineering and science.
 - In certain areas, computational simulations are the only possible approach to analyze a problem:
experiments may be cost prohibitive;
experiments may be impossible.
- Simulation capabilities rely heavily on the underlying computer power (e.g. the amount of memory, total compute processors, and processor performance).

The third pillar

THE GRAND CHALLENGE EQUATIONS

$$B_i A_i = E_i A_i + \rho_i \sum_j B_j A_j F_{ji} \quad \nabla \times \vec{E} = - \frac{\partial \vec{B}}{\partial t} \quad \vec{F} = m \vec{a} + \frac{dm}{dt} \vec{v}$$

$$dU = \left(\frac{\partial U}{\partial S} \right)_V dS + \left(\frac{\partial U}{\partial V} \right)_S dV \quad \nabla \cdot \vec{D} = \rho \quad Z = \sum_j g_j e^{-E_j/kT}$$

$$F_j = \sum_{k=0}^{N-1} f_k e^{2\pi i j k N} \quad \nabla^2 u = \frac{\partial u}{\partial t} \quad \nabla \times \vec{H} = \frac{\partial \vec{D}}{\partial t} + \vec{J}$$
$$p_{n+1} = r p_n (1 - p_n) \quad \nabla \cdot \vec{B} = 0 \quad P(t) = \frac{\sum_i W_i B_i(t) P_i}{\sum_i W_i B_i(t)}$$

$$-\frac{\hbar^2}{8\pi^2 m} \nabla^2 \Psi(r,t) + V \Psi(r,t) = -\frac{\hbar}{2\pi i} \frac{\partial \Psi(r,t)}{\partial t} \quad -\nabla^2 u + \lambda u = f$$

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla) \vec{u} = -\frac{1}{\rho} \nabla p + \gamma \nabla^2 \vec{u} + \frac{1}{\rho} \vec{F} \quad \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f$$

- NEWTON'S EQUATIONS • SCHROEDINGER EQUATION (TIME DEPENDENT) • NAVIER-STOKES EQUATION •
- POISSON EQUATION • HEAT EQUATION • HELMHOLTZ EQUATION • DISCRETE FOURIER TRANSFORM •
- MAXWELL'S EQUATIONS • PARTITION FUNCTION • POPULATION DYNAMICS •
- COMBINED 1ST AND 2ND LAWS OF THERMODYNAMICS • RADIOSITY • RATIONAL B-SPLINE •

Simulation in Science and Engineering

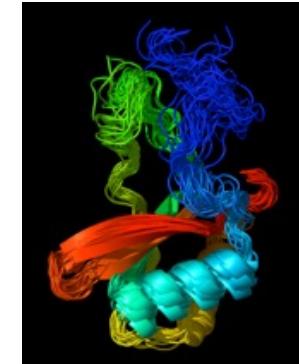
High performance computing (HPC) simulation to understand things that are:

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

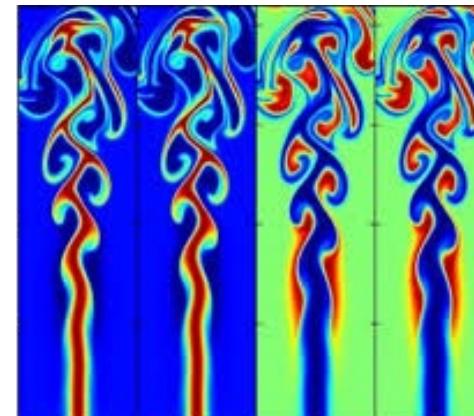
for experiments



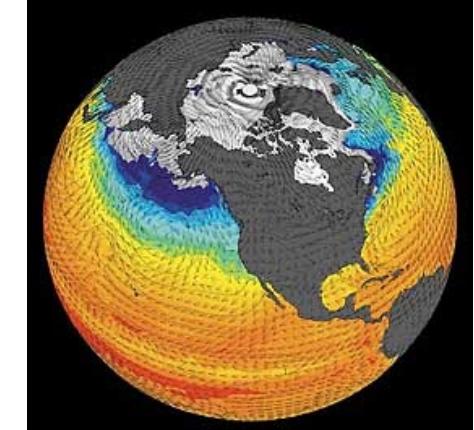
Understanding the universe



Proteins and diseases



Energy-efficient jet engines

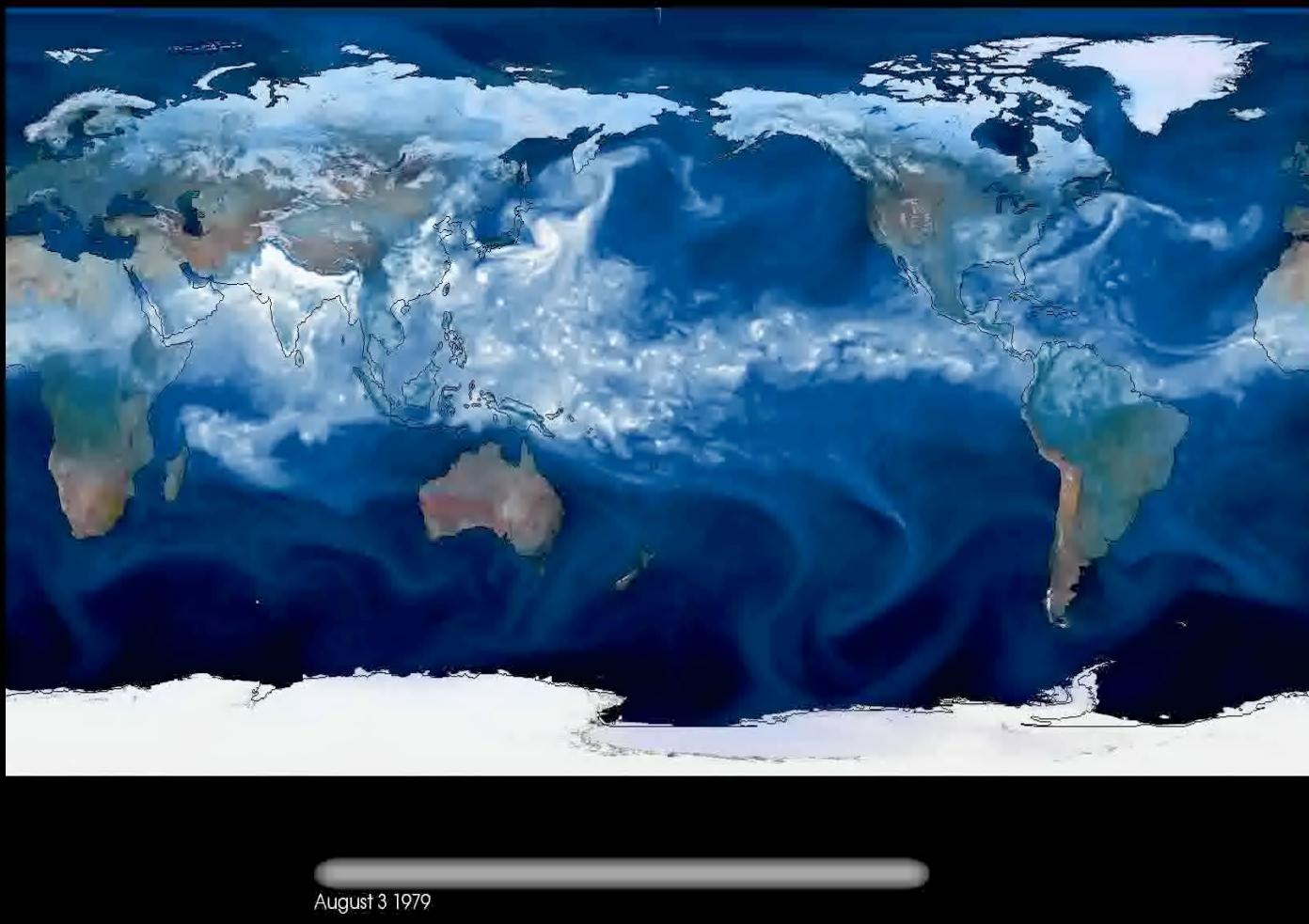


Climate change

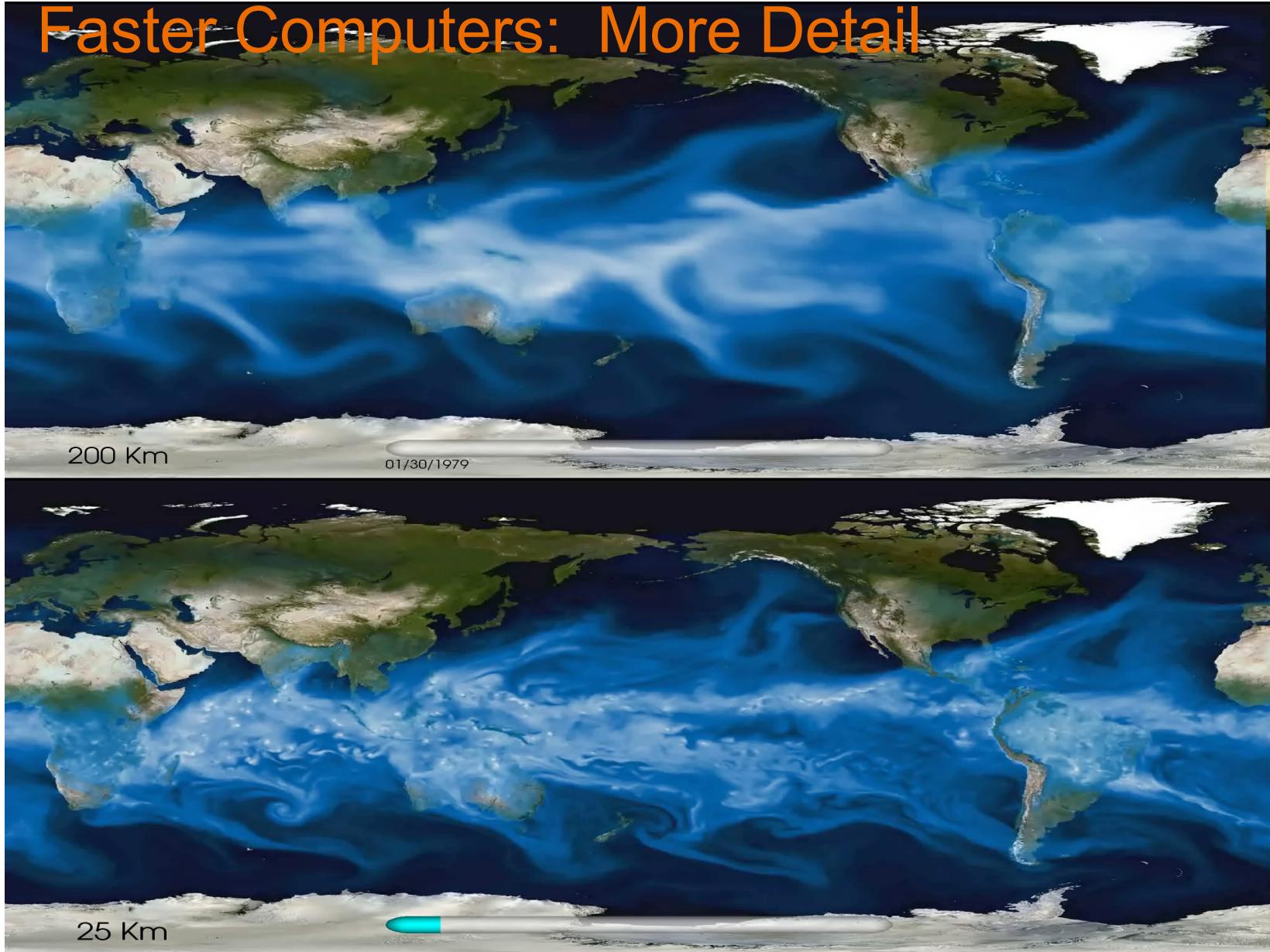
Weather Modeling

- For modeling a hurricane region:
 - Assume region of interest is 1000 X 1000 miles, with height of 10 miles.
 - Partition into segments of 0.1 x 0.1 x 0.1 miles: 10^{10} grid points
 - Simulate 2 days, with 30-minute timesteps: 100 total timesteps
- Assume the computations at each grid point require 100 instructions.
 - A single timestep then requires 10^{12} instructions.
 - For two days we need 10^{14} instructions
 - For serial computer with 10^8 instructions/sec, this takes 10^6 seconds (10 days!) to predict next 2 days!!
- **THIS REQUIRES PARALLELISM FOR PERFORMANCE TO PREDICT**
 - Also requires lots of memory which implies parallelism
- All major weather forecast centers (US, Europe, Asia) have supercomputers with 1000s of processors.

Simulations Show the Effects of Climate Change in Hurricanes



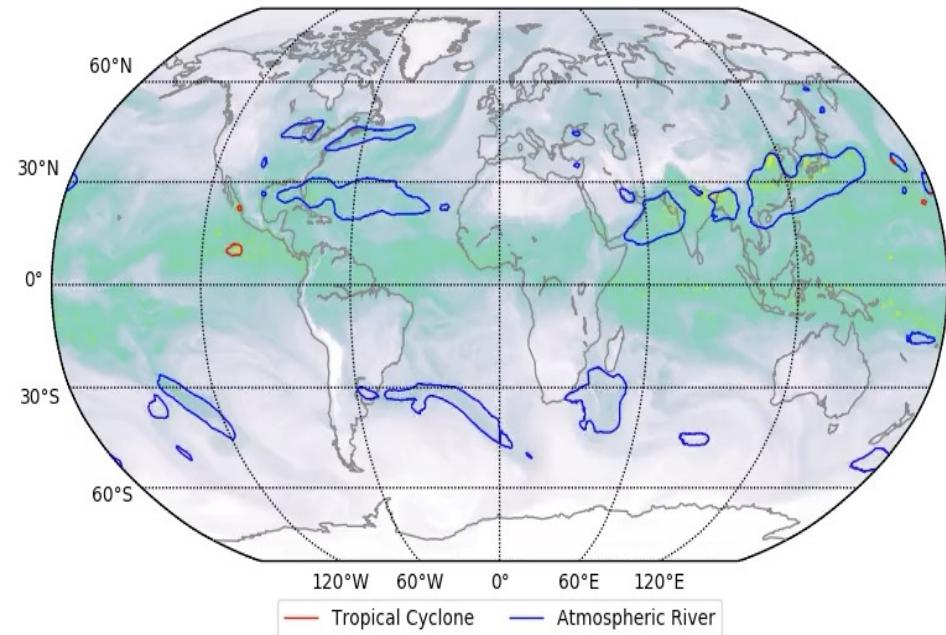
Faster Computers: More Detail



Big Data and Machine Learning

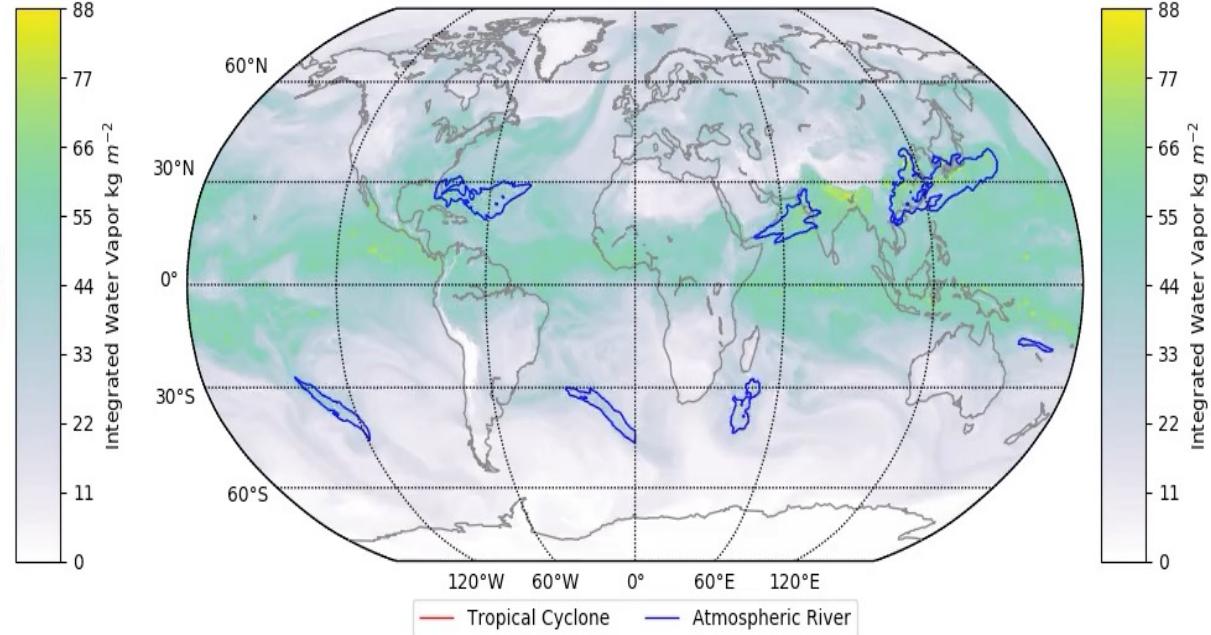
Predicted Extreme Weather

Predicted Extreme Weather Patterns 2107-07-07



Ground Truth Extreme Weather

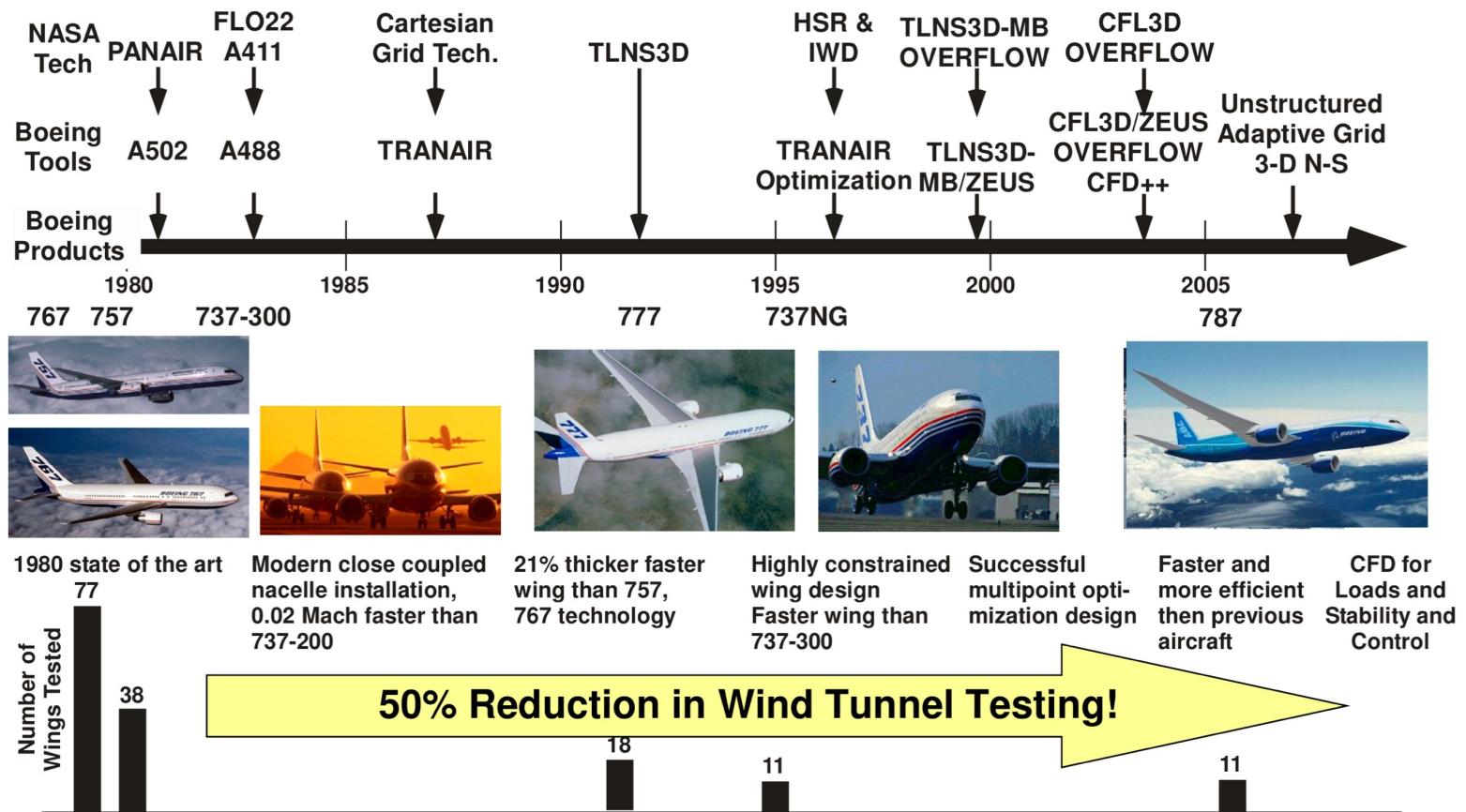
Ground Truth Extreme Weather Patterns 2107-07-07



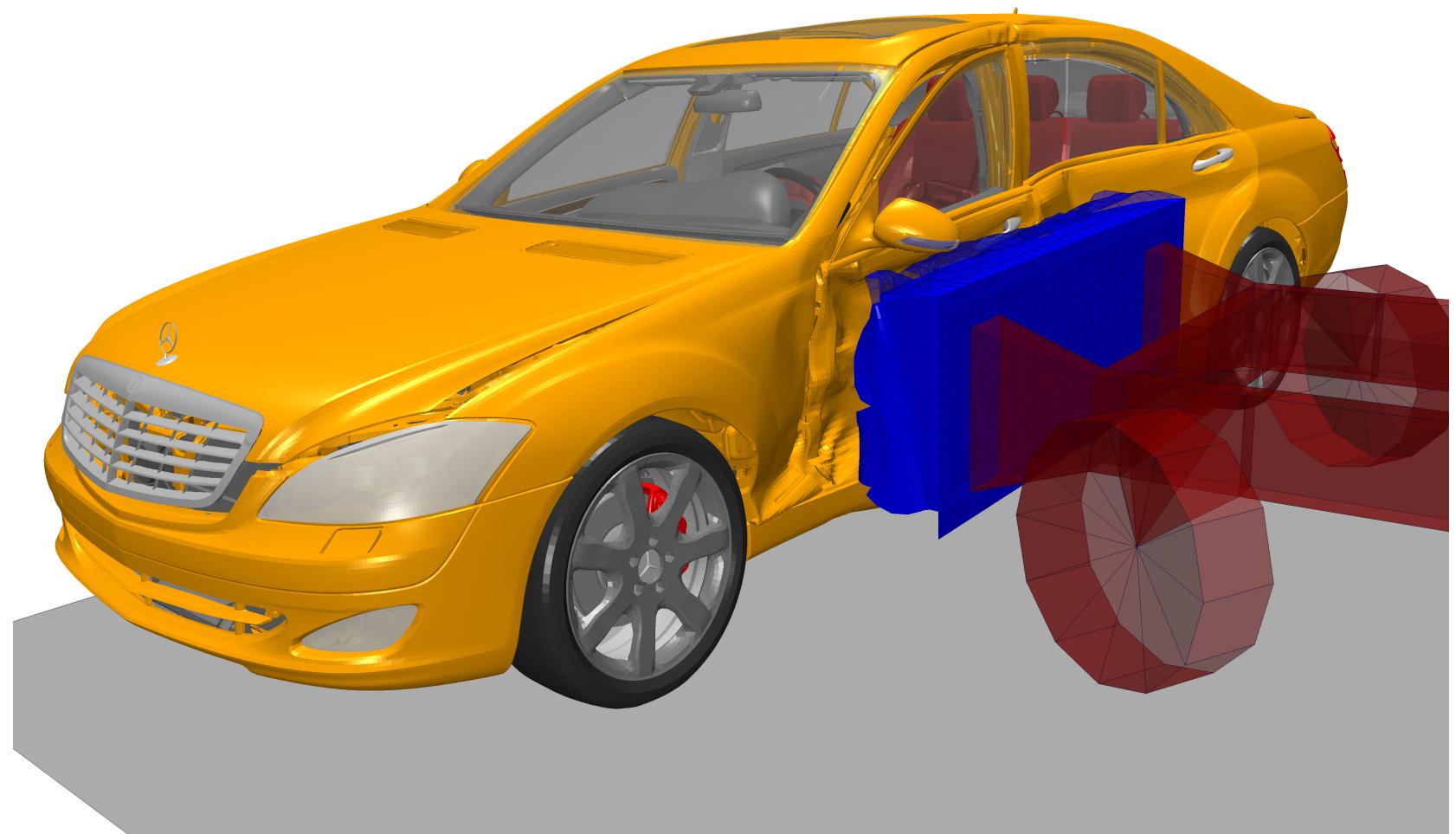
- Deep learning results are smoother than heuristic labels
- Achieved over 1 EF peak on OLCF Summit: Gordon Bell Prize in 2018

Thorsten Kurth, Sean Treichler, Joshua Romero, Mayur Mudigonda, Nathan Luehr, Everett Phillips, Ankur Mahesh, Michael Matheson, Jack Deslippe, Massimiliano Fatica, Prabhat, Michael Houston

Some applications



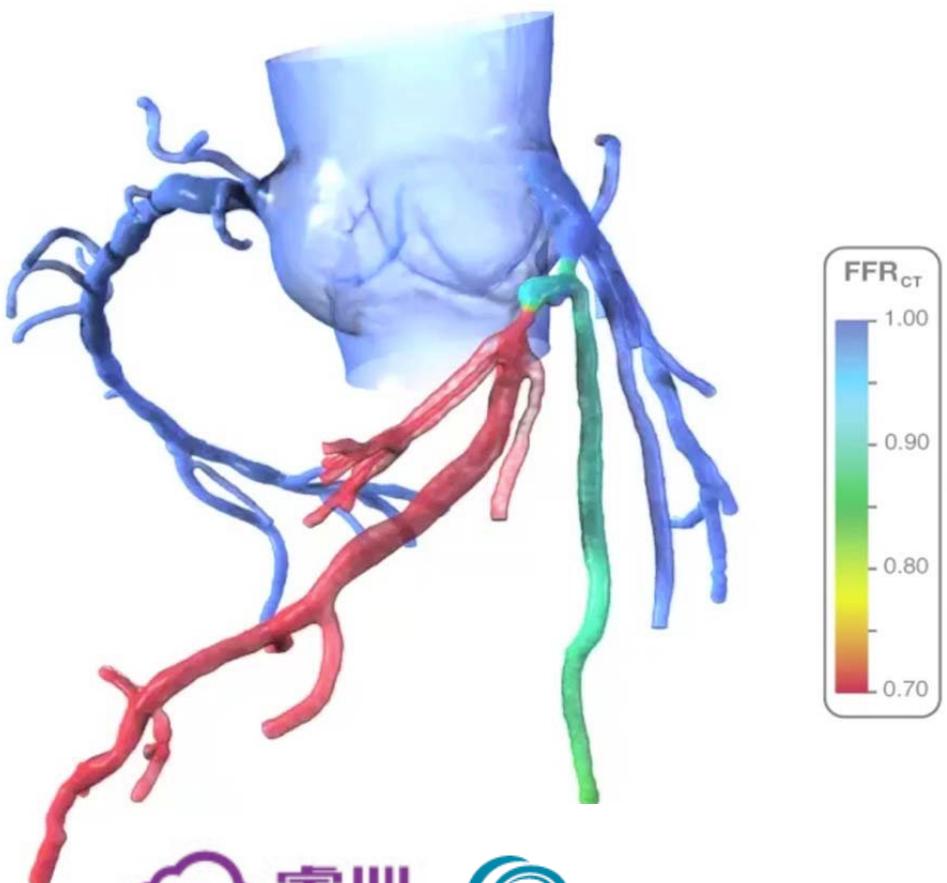
Some applications



Some applications

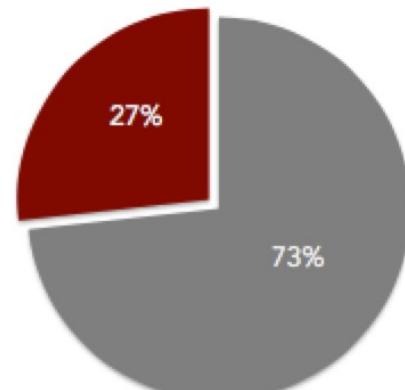


Some applications

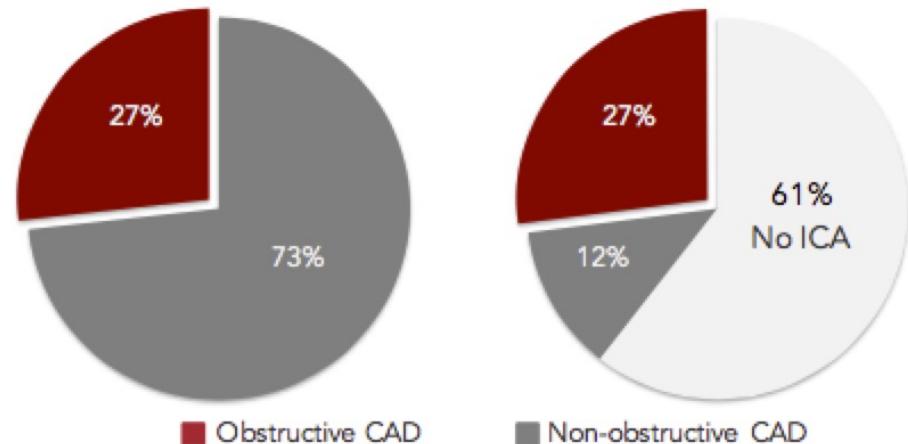


Invasive Catheterization (ICA) with No Obstructive Disease

Usual Care



CTA/FFR_{CT} Guided



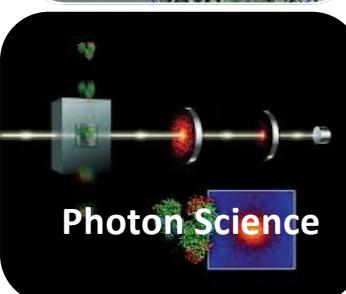
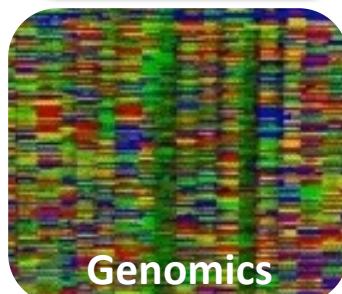
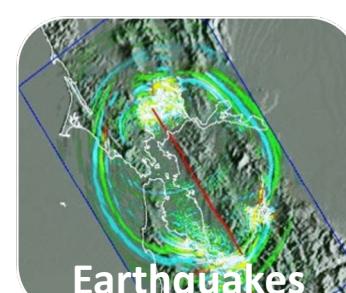
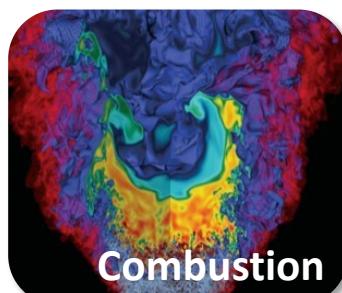
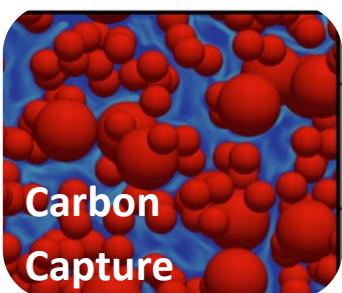
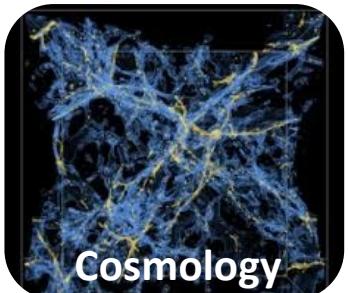
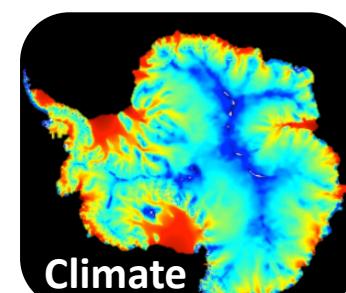
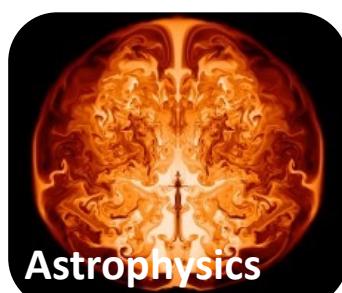
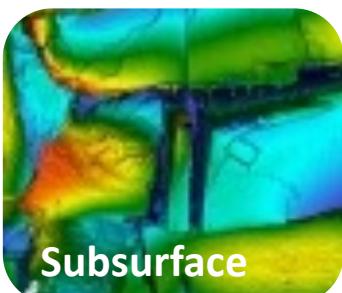
83% reduction of ICAs that found no obstructive CAD

No adverse clinical events in patients in whom ICA was cancelled.



P.S. Douglas, et al. European Heart Journal 2015.

Some applications



What is in this course?

Course goals

- UNIX/Linux exposure
 - Command line
 - Compilers
 - Libraries
- Some numerical background
 - Linear algebra
 - Numerical PDE
- Good practices of software development
 - Version control
 - Build systems
 - Debugging skills
- Parallel computing
 - MPI
 - PETSc

What you should get out of the course

Understanding of:

Part – 1

- Serial computer architecture
- Linux basics
- Version control of software (git)
- Compilers and cross-platform compiling tools (CMake)
- Debugging and code profiling

Part – 2

- Numerical analysis basics (linear algebra, PDEs, etc.)
- Machine learning basics

What you should get out of the course

Part – 3

- Engineering computing libraries (PETSc)
- Visualization of data (VTK & Paraview)
- I/O of data (HDF5)

Part – 4

- Parallel computer architecture
- Parallel computing basics
- MPI programming

Part – 5

- GPU programming (if time permits)



We will schedule a tour to the computing center

Methods and Practice

Practice is important.

You are encouraged to bring your laptop to class (not for video games).

There will be some theory, which is not too hard.

Slides will be shared with you on blackboard.

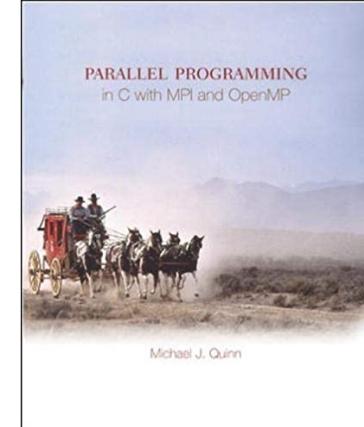
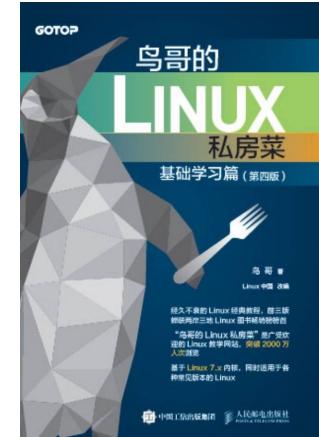
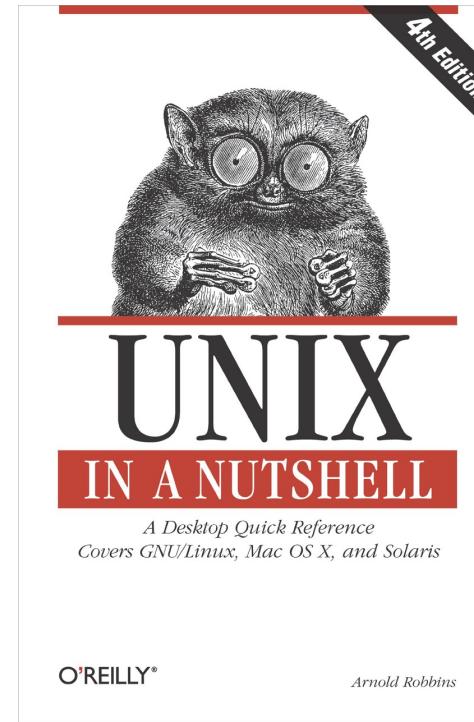


References

We recommend a few books for general reading.

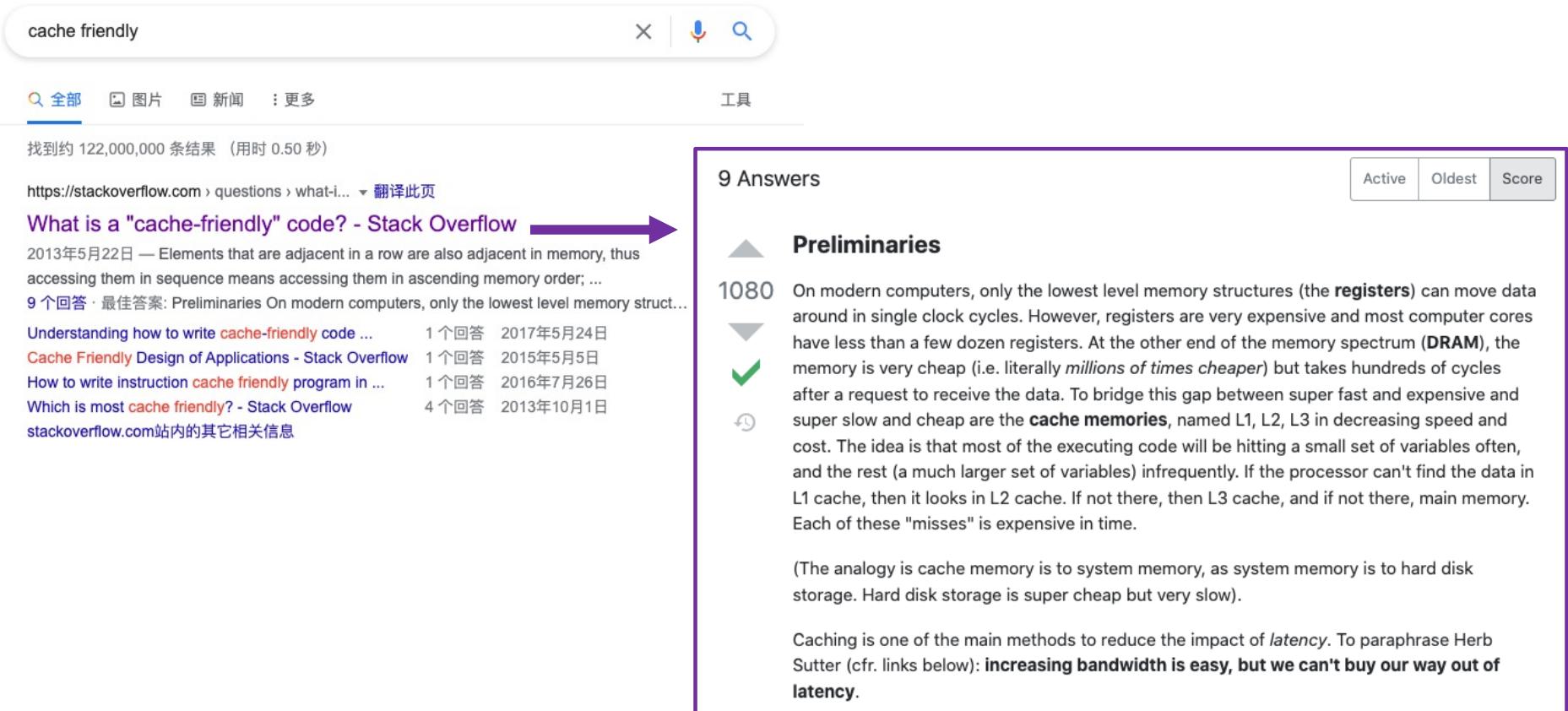
Tutorials may be distributed in class on specific topics
(Makefile, VTK, PETSc, etc.)

Slides will become available.



Some recommendations

- Use a good search engine and Stack Overflow to learn things



cache friendly

找到约 122,000,000 条结果 (用时 0.50 秒)

https://stackoverflow.com › questions › what-i... 翻译此页

What is a "cache-friendly" code? - Stack Overflow →

2013年5月22日 — Elements that are adjacent in a row are also adjacent in memory, thus accessing them in sequence means accessing them in ascending memory order; ...

9 个回答 · 最佳答案: Preliminaries On modern computers, only the lowest level memory struct...

Understanding how to write cache-friendly code ... 1 个回答 2017年5月24日

Cache Friendly Design of Applications - Stack Overflow 1 个回答 2015年5月5日

How to write instruction cache friendly program in ... 1 个回答 2016年7月26日

Which is most cache friendly? - Stack Overflow 4 个回答 2013年10月1日

stackoverflow.com站内的其它相关信息

9 Answers

1080 Preliminaries

On modern computers, only the lowest level memory structures (the **registers**) can move data around in single clock cycles. However, registers are very expensive and most computer cores have less than a few dozen registers. At the other end of the memory spectrum (**DRAM**), the memory is very cheap (i.e. literally *millions of times cheaper*) but takes hundreds of cycles after a request to receive the data. To bridge this gap between super fast and expensive and super slow and cheap are the **cache memories**, named L1, L2, L3 in decreasing speed and cost. The idea is that most of the executing code will be hitting a small set of variables often, and the rest (a much larger set of variables) infrequently. If the processor can't find the data in L1 cache, then it looks in L2 cache. If not there, then L3 cache, and if not there, main memory. Each of these "misses" is expensive in time.

(The analogy is cache memory is to system memory, as system memory is to hard disk storage. Hard disk storage is super cheap but very slow).

Caching is one of the main methods to reduce the impact of *latency*. To paraphrase Herb Sutter (cfr. links below): **increasing bandwidth is easy, but we can't buy our way out of latency.**

Some recommendations

- Use AI tools (e.g. deepseek)

what is a cache-friendly code



Cache-friendly code is code that is designed to optimize the use of the CPU cache, a small but very fast memory located close to the CPU. The goal of cache-friendly code is to minimize cache misses, which occur when the data needed by the CPU is not found in the cache and must be fetched from slower main memory (RAM). Cache misses can significantly degrade performance, so writing cache-friendly code is crucial for high-performance applications.

Key Principles of Cache-Friendly Code

1. Temporal Locality:

- **Principle:** Reuse data that has been recently accessed.
- **Implementation:** Access the same data multiple times in a short period to keep it in the cache.

Grades

- The final grade will be based on
 - Homework (40%) (Around 5 assignments)
 - Course participation (10%)
 - Projects (50%)

We have ‘practice’ in the course name. There will be assignments on practical computing and programming.

You need to know some basics of C/C++.

If you use Fortran, come and discuss with me (I am not a Fortran programmer ☺).

Matlab/Python is **not** OK.

Some policies

- All codes and written work should be your own
 - Homework will be assigned on a regular basis (approximately once every two weeks).
 - Make your own code repositories private and do not share after the semester ends (will talk about git repository later).
 - We do **NOT** take late submission of HW.
 - It is allowed to revise your HW and get points back. (Do NOT abuse this.)
- The final project will be done individually
 - Your contribution will be reflected on the Git commit history.
 - You will have to submit a written report

Some policies

- Plagiarism (作弊/抄袭)
 - Copy other's results/codes
 - ZERO for the course if detected (本门课程记零分) .

Homework 0

- Install Ubuntu 18 on your machine.
- Go to ubuntu.com/tutorials/install-ubuntu-desktop-1804
- Teaching assistants can help you get a USB install drive
- Bring your laptop and show me your Linux system

The screenshot shows a web browser displaying the Canonical Ubuntu website at ubuntu.com/tutorials/install-ubuntu-desktop-1804#1-overview. The page title is "Install Ubuntu desktop 18.04". The main content area is titled "1. Overview" and contains the following text:

This tutorial covers the installation of a previous *Long Term Support* release (Ubuntu 18.04 LTS). Jump to [Install Ubuntu desktop](#) if you wish to install the latest version.

The Ubuntu desktop is easy to use, easy to install and includes everything you need to run your organisation, school, home or enterprise. It's also open source, secure, accessible and free to download.

The page also shows a screenshot of the Ubuntu desktop environment with the Firefox browser open, displaying the same tutorial page. The desktop has a purple theme with icons for various applications like Activities, Dash, and the Dash menu.

At the bottom of the page, there is a note: "In this tutorial, we're going to install Ubuntu desktop onto your computer, using either your computer's DVD drive or a USB flash drive."

Navigation links at the bottom include "Suggest changes >" and "about 0 minutes to go < >".

Homework 0

- You do not have to delete your Windows system

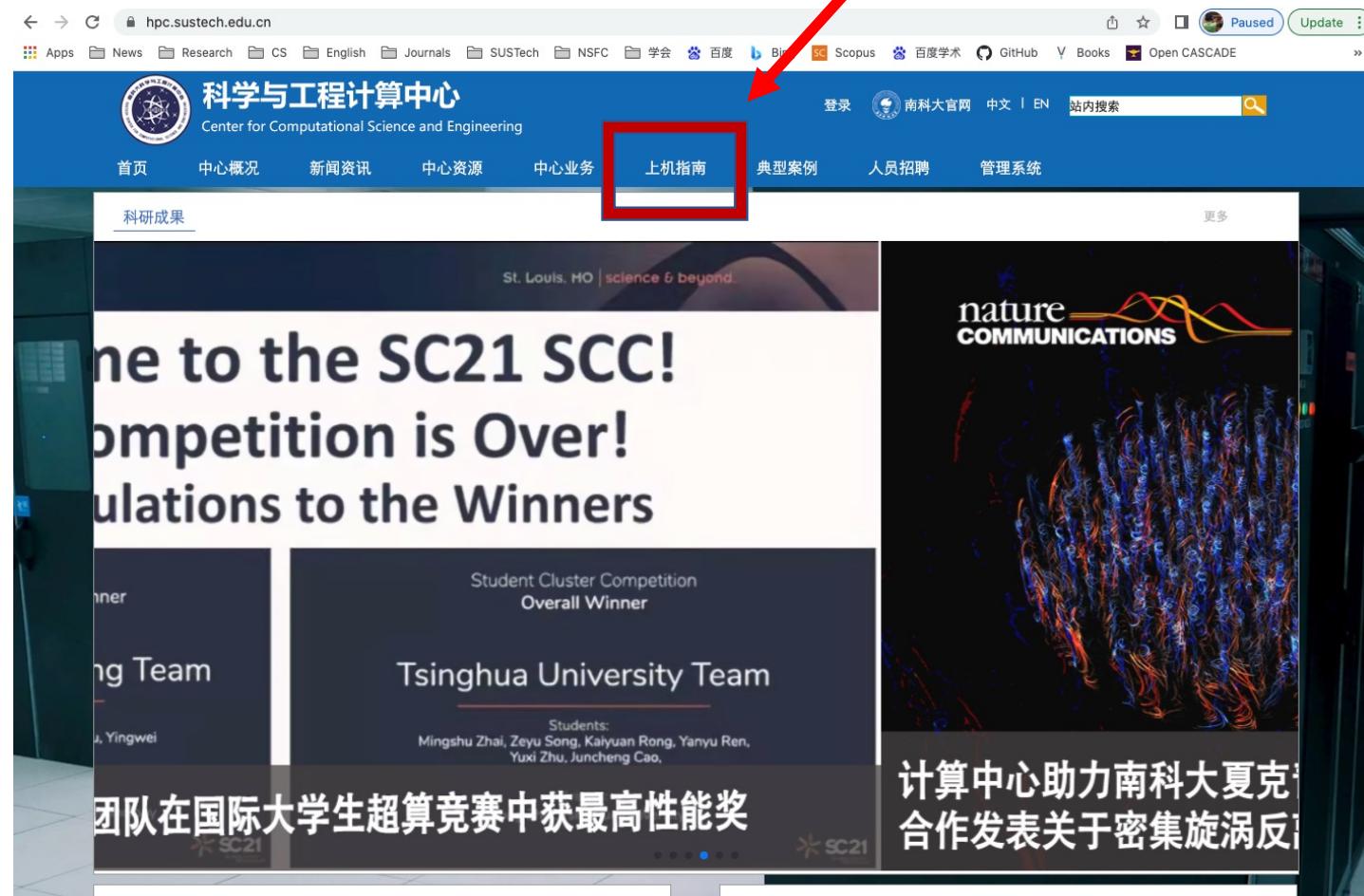
<https://www.itzgeek.com/how-tos/linux/ubuntu-how-tos/how-to-install-ubuntu-18-04-alongside-with-windows-10-or-8-in-dual-boot.html>

- You need to prepare Windows for dual-boot

- Extra partitions to allocate Linux ubuntu

Homework 0

- Get your Tai-Yi account



Homework 0

- Get your Tai-Yi account

- Remote login by SSH

ssh username@172.18.6.175

at your command line. username is your Tai-Yi account.

```
juliu::Kolmogorov { ~ }  
-> ssh mae-liuj@172.18.6.175  
mae-liuj@172.18.6.175's password: ?
```

- Due in two weeks (**we will ask for your user id**).