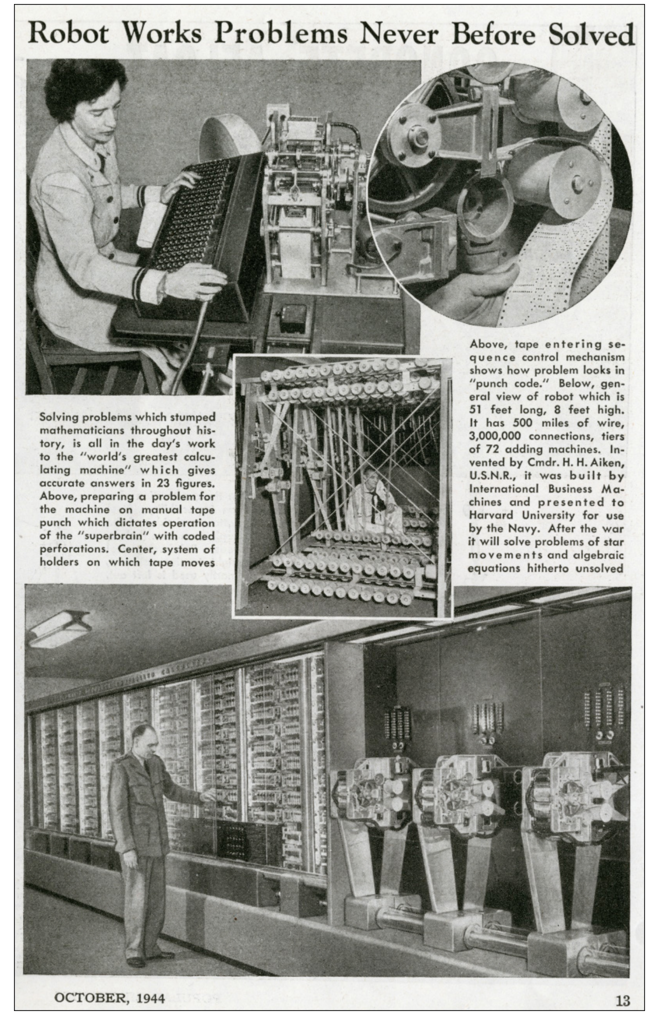
Revolution Started HERE!

"At the present time there exist problems beyond our ability to solve, not because of theoretical difficulties, but because of insufficient means of mechanical computation"

Howard Aiken, Harvard Mark I Designer, 1937



Robot Works Problems Never Before Solved

Solving problems which stumped mathematicians throughout history, is all in the day's work to the "world's greatest calculating machine" which gives accurate answers in 23 figures. Above, preparing a problem for the machine on manual tape punch which dictates operation of the "superbrain" with coded perforations. Center, system of holders on which tape moves

Above, tape entering sequence control mechanism shows how problem looks in "punch code." Below, general view of robot which is 51 feet long, 8 feet high. It has 500 miles of wire, 3,000,000 connections, tiers of 72 adding machines. Invented by Cmdr. H. H. Aiken, U.S.N.R., it was built by International Business Machines and presented to Harvard University for use by the Navy. After the war it will solve problems of star movements and algebraic equations hitherto unsolved

OCTOBER, 1944 13

Mark I was designed in 1937 by a Harvard graduate student, Howard H. Aiken to simulate physics problems encountered in his research

Introduction:
Large-Scale Computational and Data Science

CS205: Computing Foundations for Computational Science
Dr. David Sondak
Spring Term 2021

HARVARD
School of Engineering and Applied Sciences

IACS
INSTITUTE FOR APPLIED COMPUTATIONAL SCIENCE
AT HARVARD UNIVERSITY

Lectures developed by Dr. Ignacio M. Illorente

Roadmap

Large-Scale Computational and Data Science

Computational Science

Data Science

The Need for Parallel Processing

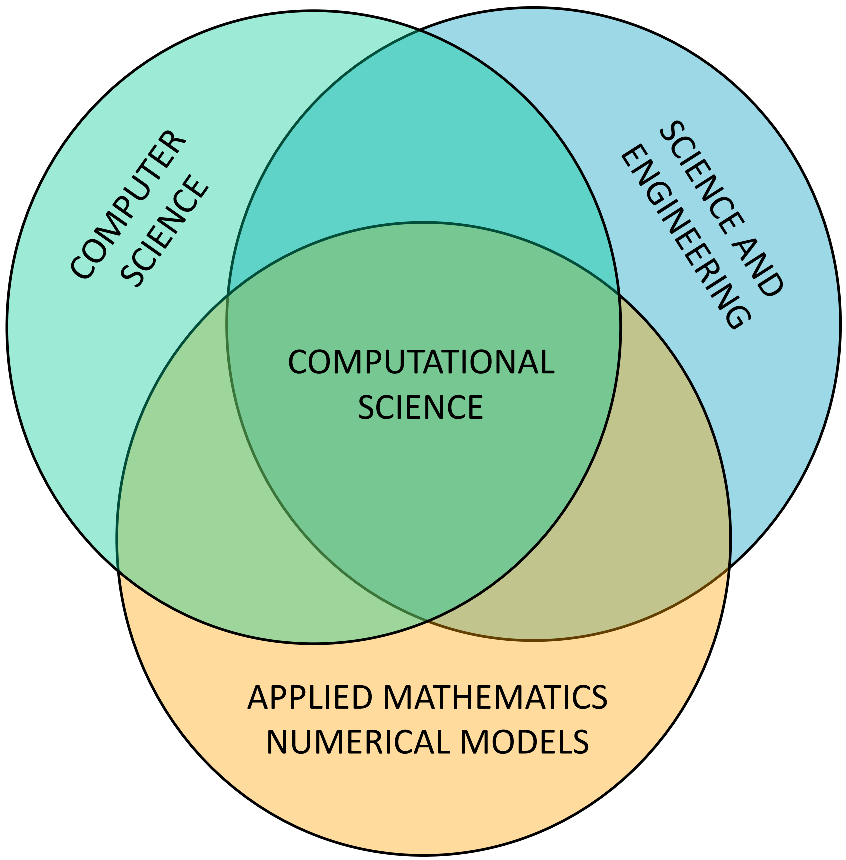
The Challenges for Parallel Processing

Description of the Course

Computational Science

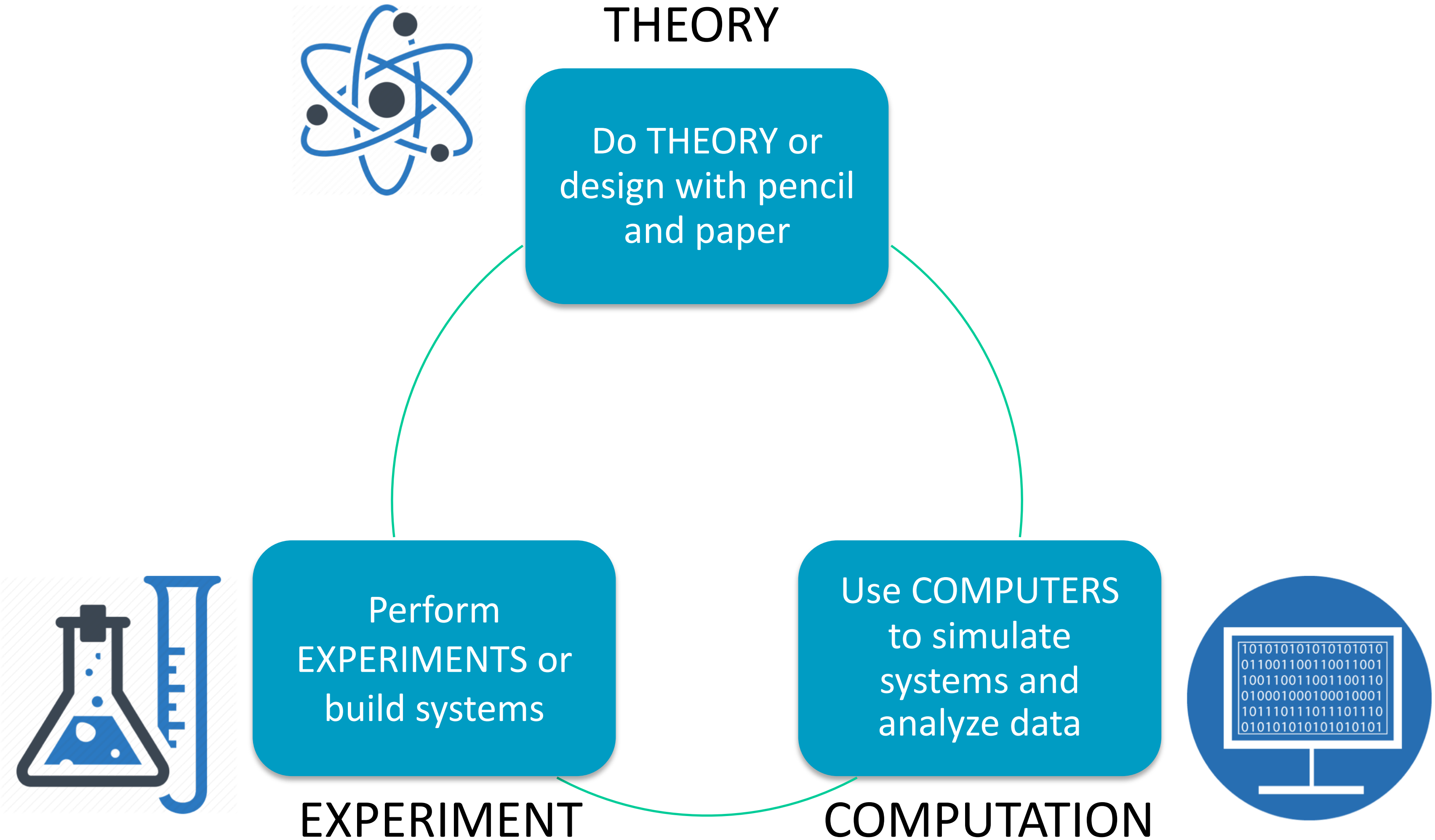
What is it?

Computational Science is an interdisciplinary field concerned with the use of applied mathematics, computational methods and computing platforms to perform numerical simulations in science and engineering



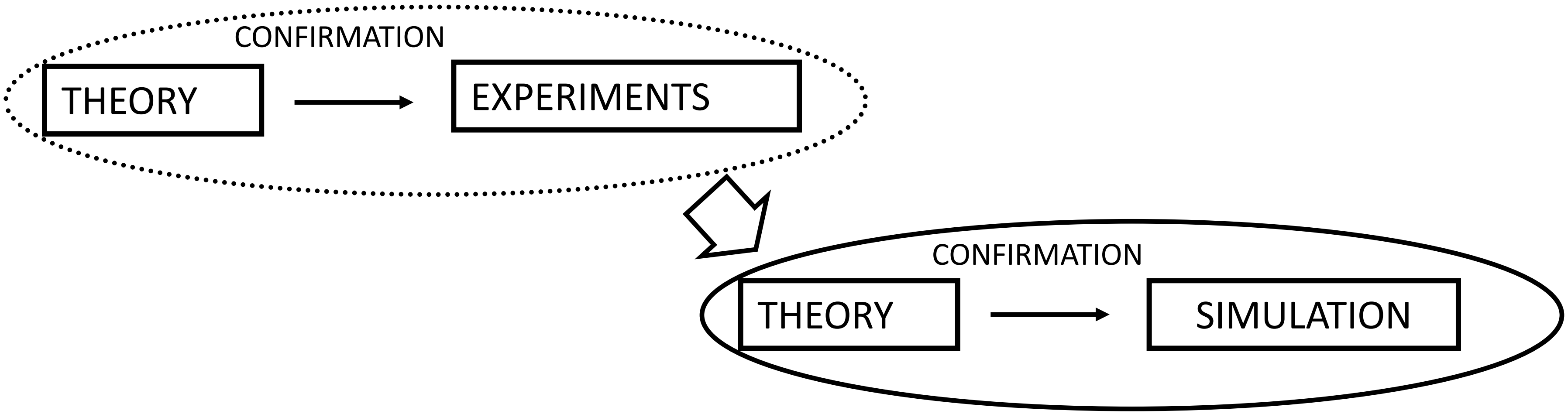
Computational Science

The Third Pillar in Scientific Discovery



Computational Science

Advance Scientific Discovery, Knowledge, and Practice



Too slow

- Wait for climate or galactic evolution

Too dangerous

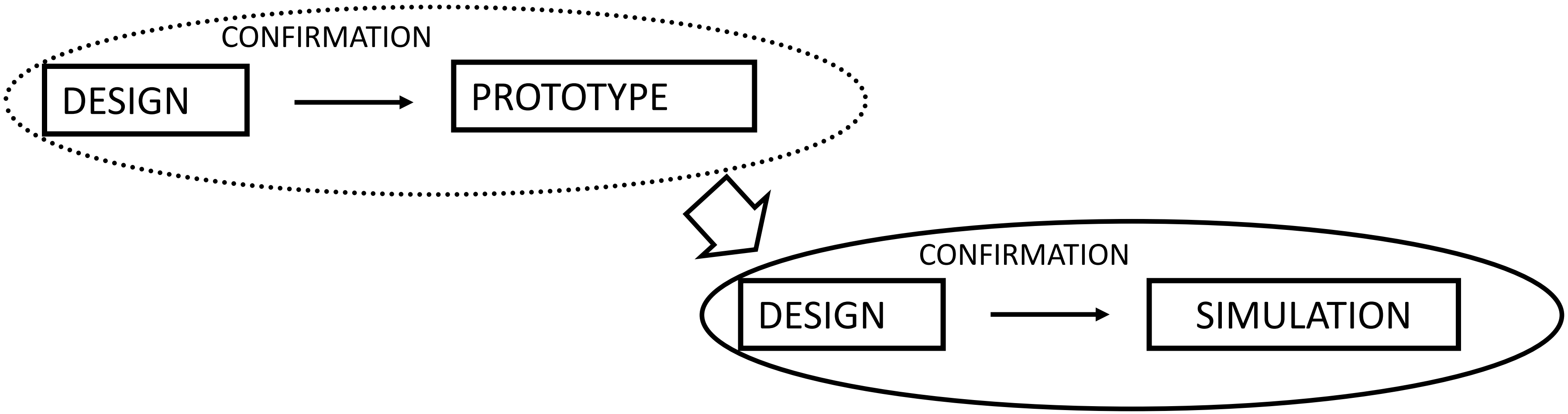
- Drug design or climate experimentation

Too complex

- Model processes across interdisciplinary boundaries

Computational Science

Product and Process Development and Manufacturing



Too difficult

- Build very large wind tunnels

Too expensive

- Build a throw-away passenger jet

Too dangerous

- Weapons

Computational Science

The Process to Perform Numerical Experiments in a Virtual Lab

Real System
- Science or engineering problem statement

Mathematical Modeling
- Complex system of equations

Numerical Approximation
- Applied mathematics methods

Computer Simulation
- Programming, Computing Tools and Platforms

Analysis and Verification
- Post processing utilities

AERODYNAMICS

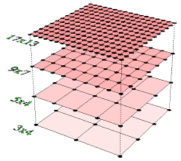
NAVIER-STOKES PDE

$$\frac{\partial u}{\partial t}+\frac{1}{r^\tau}\frac{\partial(r^\tau uu)}{\partial r}+\frac{\partial(vu)}{\partial z}=-\frac{\partial p}{\partial r}+\frac{1}{Re}\frac{\partial}{\partial z}\left(\frac{\partial u}{\partial z}-\frac{\partial v}{\partial r}\right)+\frac{1}{Fr^2}g_r,$$

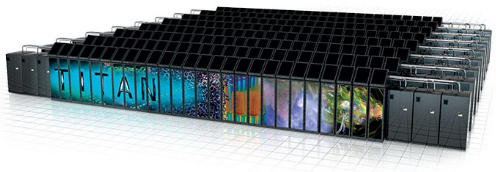
$$\frac{\partial v}{\partial t}+\frac{1}{r^\tau}\frac{\partial(r^\tau uv)}{\partial r}+\frac{\partial(vv)}{\partial z}=-\frac{\partial p}{\partial z}+\frac{1}{Re\,r^\tau}\frac{\partial}{\partial r}\left(r^\tau\left(\frac{\partial u}{\partial z}-\frac{\partial v}{\partial r}\right)\right)+\frac{1}{Fr^2}g_z,$$

$$\frac{1}{r^\tau}\frac{\partial(r^\tau u)}{\partial r}+\frac{\partial v}{\partial z}=0,$$

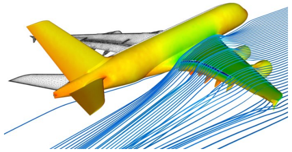
SYSTEMS OF EQUATIONS



PARALLEL COMPUTING



VISUALIZATION



Computational Science

The Need for Performance

- Lower Execution Time. More performance to reduce the execution time of the simulations
- Higher Accuracy. More performance to simulate the system with a more accurate numerical approximation
- More Complex Models. More performance to simulate the system with a more complex mathematical model

COMPUTATIONAL SCIENCE
Numerical simulation of systems in science and engineering

BIG COMPUTE (HPC)
Large-scale parallel processing of compute-intensive applications

What is ”low execution time”?

Computational Science

The Need for Performance

Climate Simulation

- Mathematical Model: PDEs
- Spatial Discretization: Latitude, Longitude, Altitude
- Climate Functions: Pressure, Temperature, Humidity, WindSpeed (vector)
- Time Discretization: Explicit leap-frog method

$$\text{climate}(x, y, z, t) \Rightarrow \text{climate}(x, y, z, t + \Delta t)$$

Computational Science

The Need for Performance

Base System

- Processor: Intel(R) Core(TM) i9-7900X CPU @ 3.30GHz (10/20 cores/threads) => 100 Gflops

Base Simulation

- Geographic Area: 10^4 km x 10^4 km x 10 km
- Grid Cell: 1km x 1km x 1km => 10^9 cells, and Δt : 1 minute
- Storage: 50 bytes/cell x 10^9 => 50 Gbytes
- Computing per Interval: 10^3 flop/cell x 10^9 => 10^{12} flop (1 Tflop)
- 1 day prediction: (1 Tflop x 1,440)/100 Gflops = 4 hours

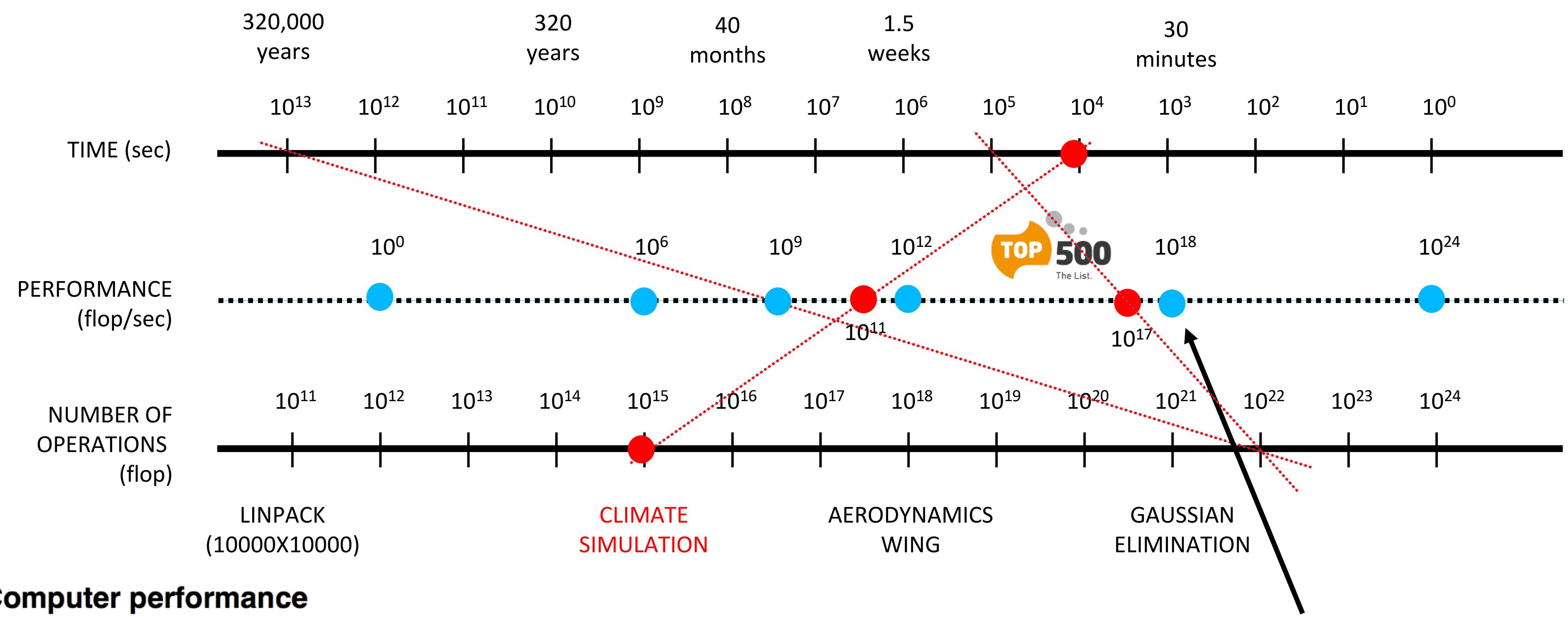
Time Critical Scenario
Simulation time to 1 hour

Complexity Critical Scenario
Computing per cell to 10^4 flops

Accuracy Critical Scenario
Grid to 10^{10} cells

Computational Science

The Need for Performance



Exascale Computing, when?

Computer performance

Name	Unit	Value
kiloFLOPS	kFLOPS	10^3
megaFLOPS	MFLOPS	10^6
gigaFLOPS	GFLOPS	10^9
teraFLOPS	TFLOPS	10^{12}
petaFLOPS	PFLOPS	10^{15}
exaFLOPS	EFLOPS	10^{18}
zettaFLOPS	ZFLOPS	10^{21}
yottaFLOPS	YFLOPS	10^{24}

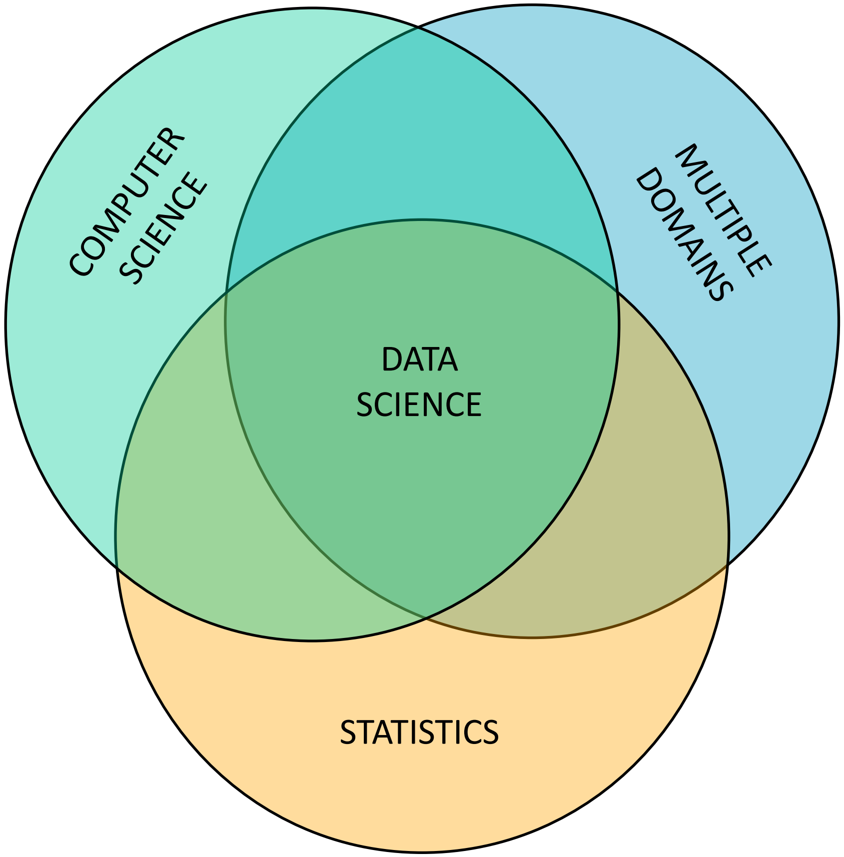
NUMBER OF OPERATIONS = PERFORMANCE X TIME

100 flop = 100 flop/sec X 1 sec

Data Science

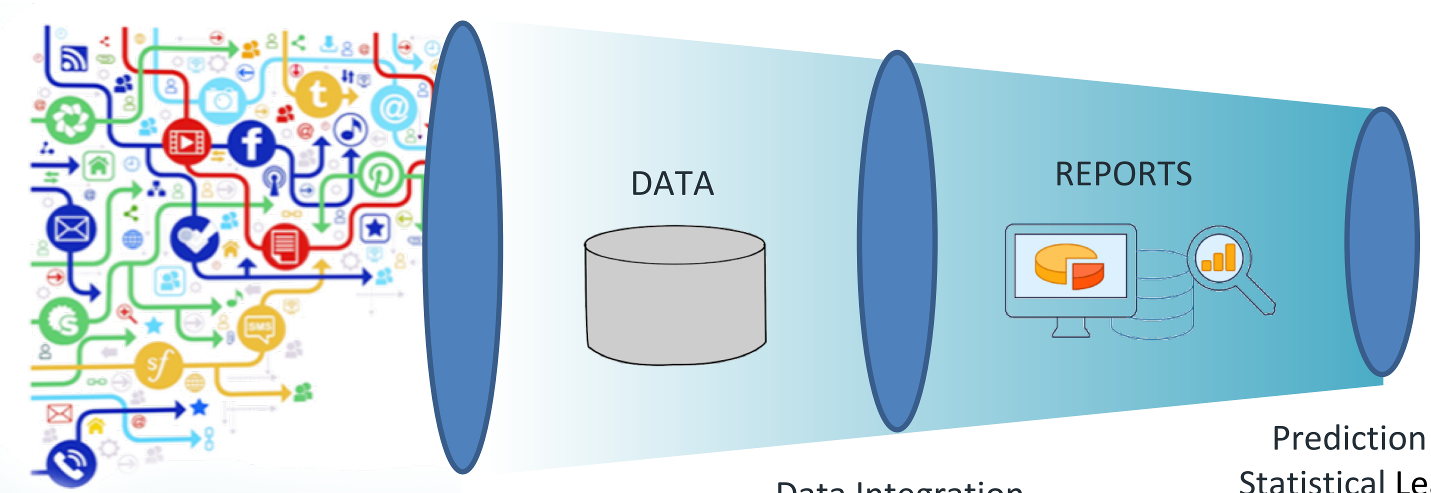
What is it?

Data Science is an interdisciplinary field concerned with the use of statistics, computational methods and computing platforms to extract knowledge or insights from data (beyond science and engineering)



Data Science

The Process



Discovered patterns
Indicators for decisions
…

Prediction and Statistical Learning

Data Integration, Processing and Analysis

Data Collection and Cleaning

DATA-INTENSIVE COMPUTATION

Parallel processing of huge data sets distributed across a large number of computers

Data Science

What Is Big Data?

2013 - Google: 10 EB of storage

2013 - Twitter: 300+ PB

2013 - Facebook: 300+ PB

2014 - eBay: 150 PB

2014 - NOAA: 17 PB of climate data

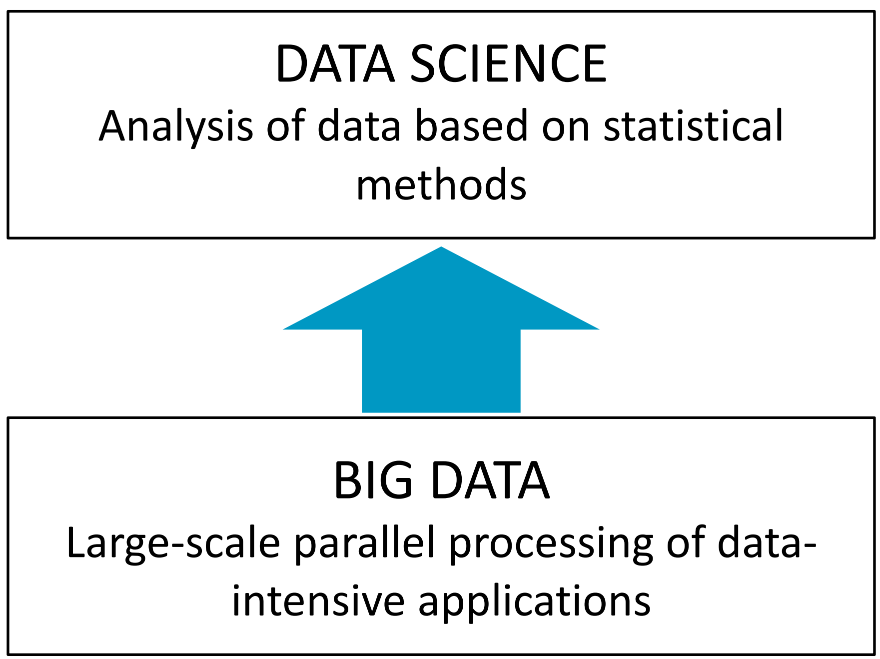
2015 - CERN's LHC:
- 600M events / sec, 1 MB per event
- Filtered down to ~ 5 GB / sec, 160 PB / year

2025 - IDC predicts that the world's data will reach 175 Zetabytes

Data Science

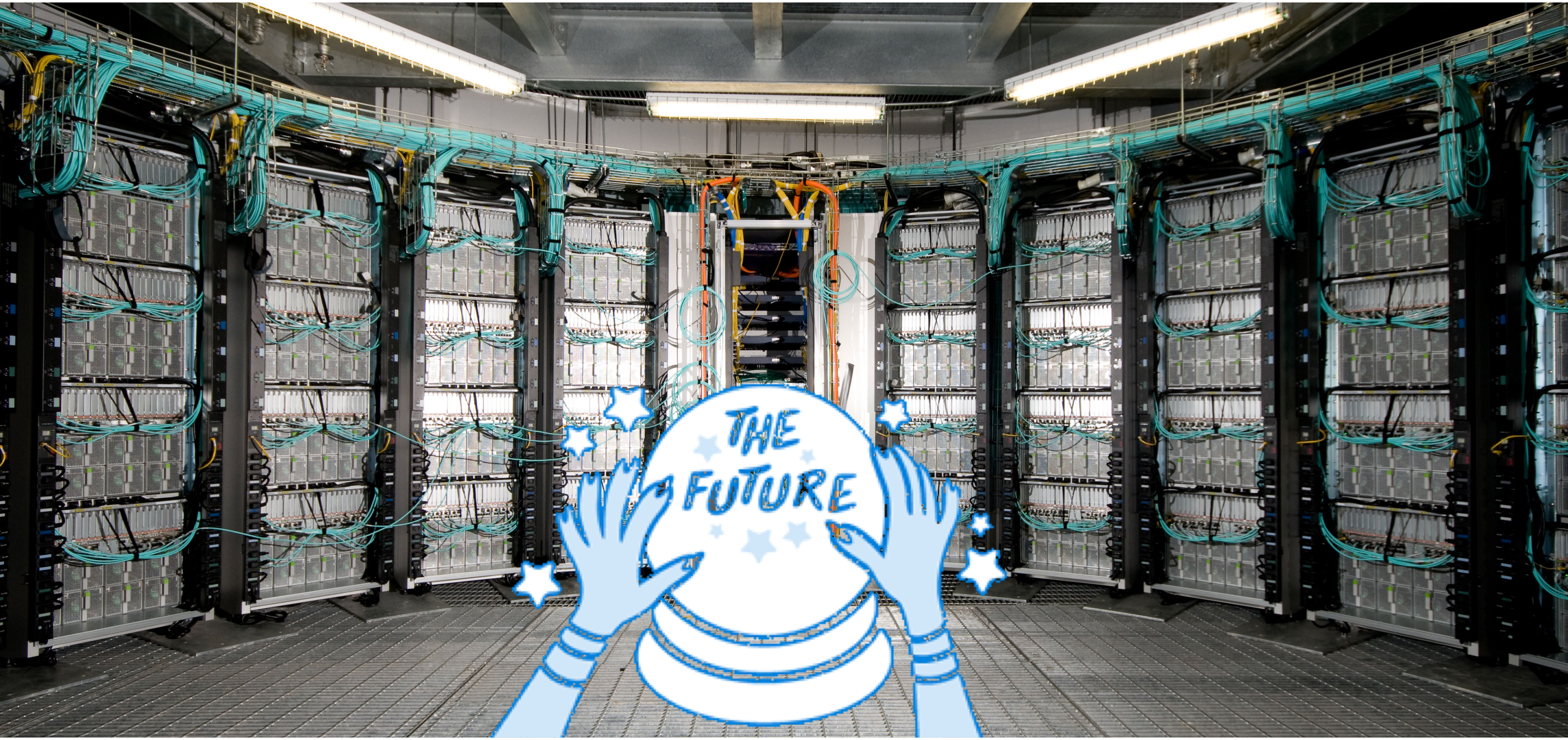
Need for Performance

- Lower Execution Time. More performance to reduce the execution time of the data analysis
- Higher Accuracy. More performance to analyze a larger data set
- More Complex Model. More performance to analyze data with a more complex statistical method



Data Science

Synergies with Computational Science



Data Science

Synergies with Computational Science

Scientific and Engineering Problem

Simulation-driven analysis and decision making

Data-driven analysis and decision making

Analysis of data generated by numerical simulation
Massive data are analyzed through computational models

Data Science

Synergies with Computational Science

HPC: Tightly Coupled

Fluid dynamics Weather forecasting Materials simulation Crash simulations	Seismic processing Metagenomics Astrophysics Deep learning
Risk simulations Molecular modeling Contextual search Logistics sim	Animation Semiconductor sim Image processing Genomics

HTC: Loosely Coupled

DATA LIGHT

Minimal requirements for data management

DATA HEAVY

Requires high performance storage and big data processing

Source: AWS

The Need for Parallel Processing

Moore's Law

Electronic- Abril1965

Cramming More Components onto Integrated Circuits

GORDON E. MOORE, LIFE FELLOW, IEEE

With unit cost falling as the number of components per circuit rises, by 1975 economics may dictate squeezing as many as 65 000 components on a single silicon chip.

The future of integrated electronics is the future of

Each approach evolved rapidly and converged so that each borrowed techniques from another. Many researchers believe the way of the future to be a combination of the various approaches.

Fig. 2.

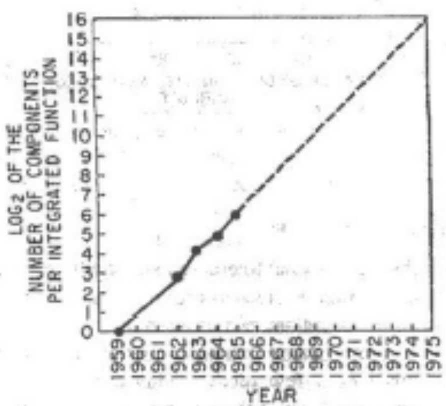


Fig. 3.

diagram to technological realization without any sp engineering.

It may prove to be more economical to build systems out of smaller functions, which are separately p aged and interconnected. The availability of large funct combined with functional design and construction, sh allow the manufacturer of large systems to design construct a considerable variety of equipment both ra and economically.

IX. Linear Circuitry

Integration will not change linear systems as radical digital systems. Still, a considerable degree of integr will be achieved with linear circuits. The lack of l value capacitors and inductors is the greatest fundam limitation to integrated electronics in the linear area.

The Need for Parallel Processing

Moore's Law

"the number of transistors in a dense integrated circuit doubles approximately every two years"

Annual 40% performance improvement or
30% cost drop

(1982-2000): Total 3,200 X

Cars:
320,000 Mph
64,000 miles/gal

Aeronautics:
L.A -> N.Y. in 5.5 seconds (Mach 3,200)

However, processing speed is limited by:

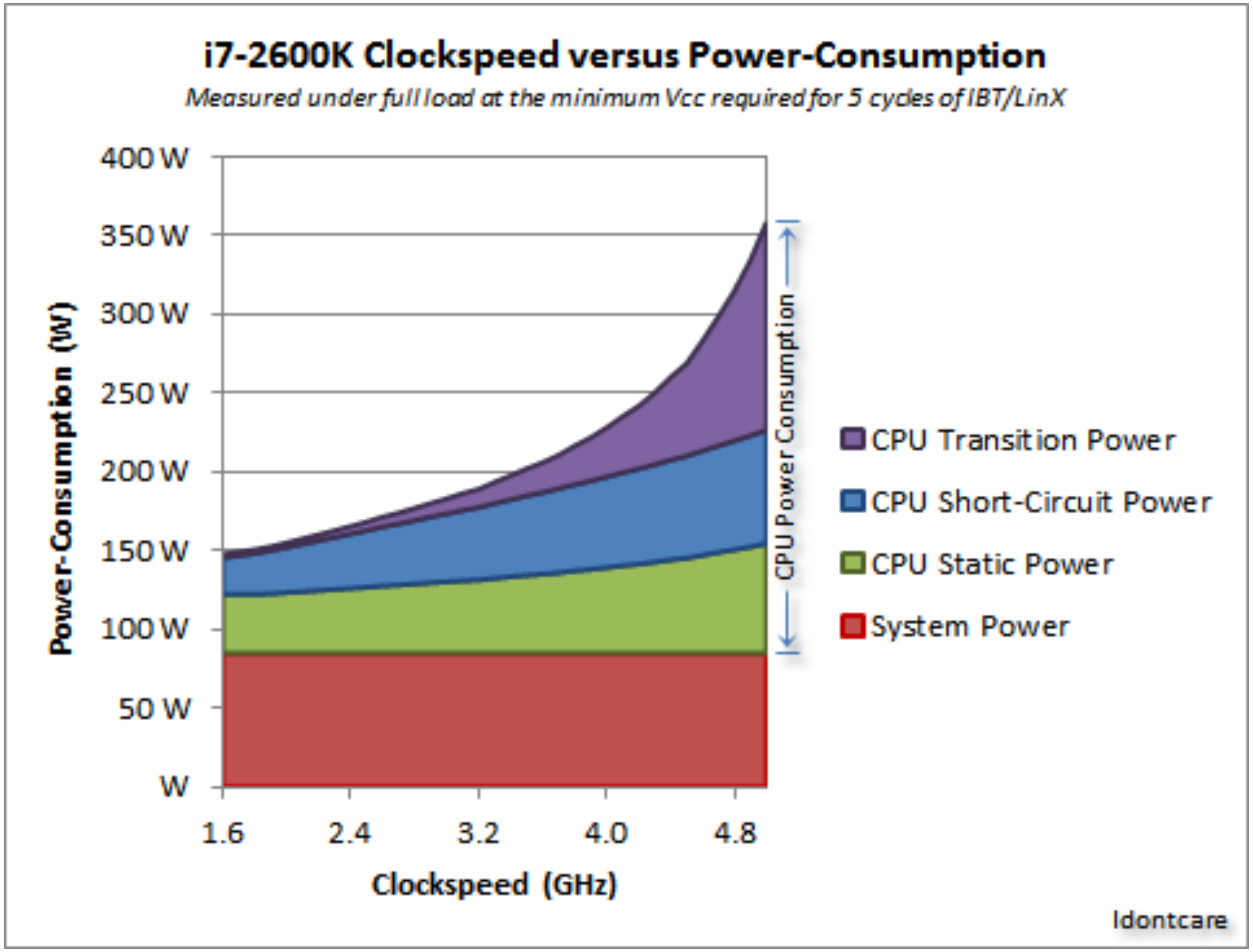
- Memory latency
- Instruction level parallelism (ILP)
- Overall temperature and power consumption
- …. Moore's law is achieving the limits of silicon technology

The Need for Parallel Processing

The Power Consumption Wall

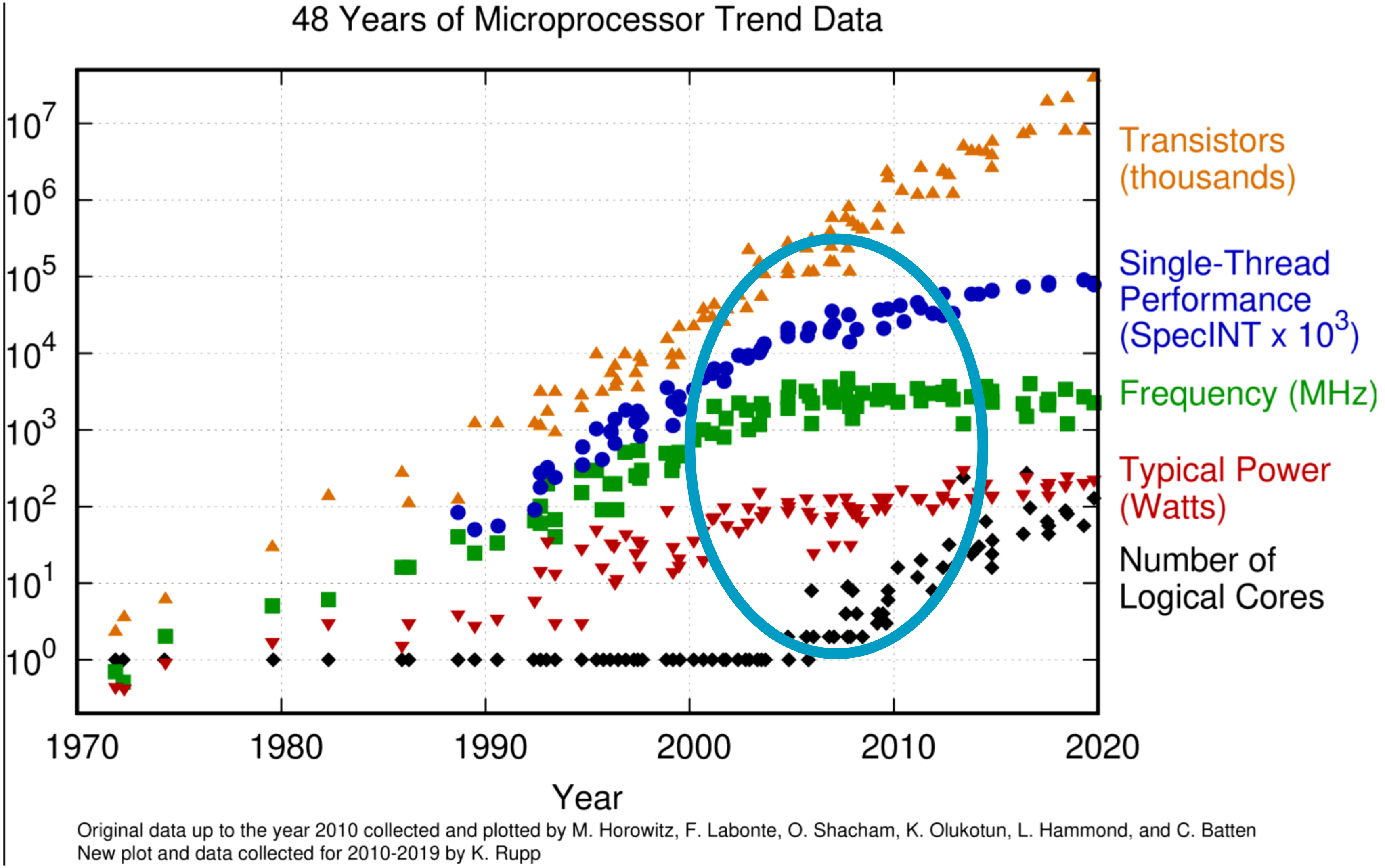
CPU transition power increases (exponentially!)

```
Core i7-860   (45 nm)        2.8 GHz     95 W
Core i7-965   (45 nm)        3.2 GHz    130 W
Core i7-3970X (32 nm)        3.5 GHz    150 W
```



The Need for Parallel Processing

The Power Consumption Wall



https://github.com/karlrupp/microprocessor-trend-data

The Challenges for Parallel Processing

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

For more on the validity of these statements see: [The '640K' quote won't go away -- but did Gates really say it?]

The Challenges for Parallel Processing

Parallel Architectures and Systems Require…

... rethinking software design

... learning new skills

... learning new programming models

... rethinking algorithms

... learning new strategies

... learning new tools

In a rapidly changing technology environment!

- Exponential growth of data
- New data processing platforms
- Open-source software
- Increase in multi-core parallelism and faster networks
- IaaS cloud providers
- Adoption of software containers
- …

The Challenges for Parallel Processing

Is This Course for You?

Conventional wisdom in scientific programming

- Old conventional wisdom: Programming is hard
- New conventional wisdom: Parallel programming is *really* hard
- 2 kinds of scientific programmers
    1. Those using single processors
    2. Those who can use 100s of processors ⟸ **CS205**
- Big steps for programmers
    - From 1 processor to 2 processors
    - From 100s of processors to 1000s of processors

Can Computer Architecture Affect Scientific Productivity?, David Patterson, 2005
http://www.lanl.gov/conferences/salishan/salishan2005/davidpatterson.pdf

CS205: Aim and Objectives

Learn Parallel Computational Thinking and Tools

Practical overview of:

- Foundations of “parallel thinking”
- Aspects to consider when designing large-scale applications
- Parallel programming models for compute- and data-intensive applications, and
- Existing platforms, open-source tools and cloud services to support their execution

After the course, you will be in a great position to:

- Make effective use of the diverse, and rapidly changing, landscape of programming models, platforms and computing architectures for high performance computing and big data
- Decide which kind of programming model and platform is appropriate to meet your scalability and performance
- Apply the enduring principles behind these rapid changes in technology that remain true, no matter which version of a particular platform you are using

CS205: Contents

A Practical View: From Design to Implementation

INTRO: LARGE-SCALE COMPUTATIONAL AND DATA SCIENCE

A. PARALLEL PROCESSING FUNDAMENTALS

A.1. Parallel Processing Architectures

A.2. Large-scale Processing on the Cloud

A.3. Practical Aspects of Cloud Computing

A.4. Application Parallelism

A.5. Designing Parallel Programs

B. PARALLEL COMPUTING

B.1. Foundations of Parallel Computing

B.2. Performance Optimization

B.3. Accelerated Computing

B.4. Shared-memory Parallel Processing

B.5. Distributed-memory Parallel Processing

C. PARALLEL DATA PROCESSING

C.1. Foundations of Data Processing

C.2. Batch Data Processing

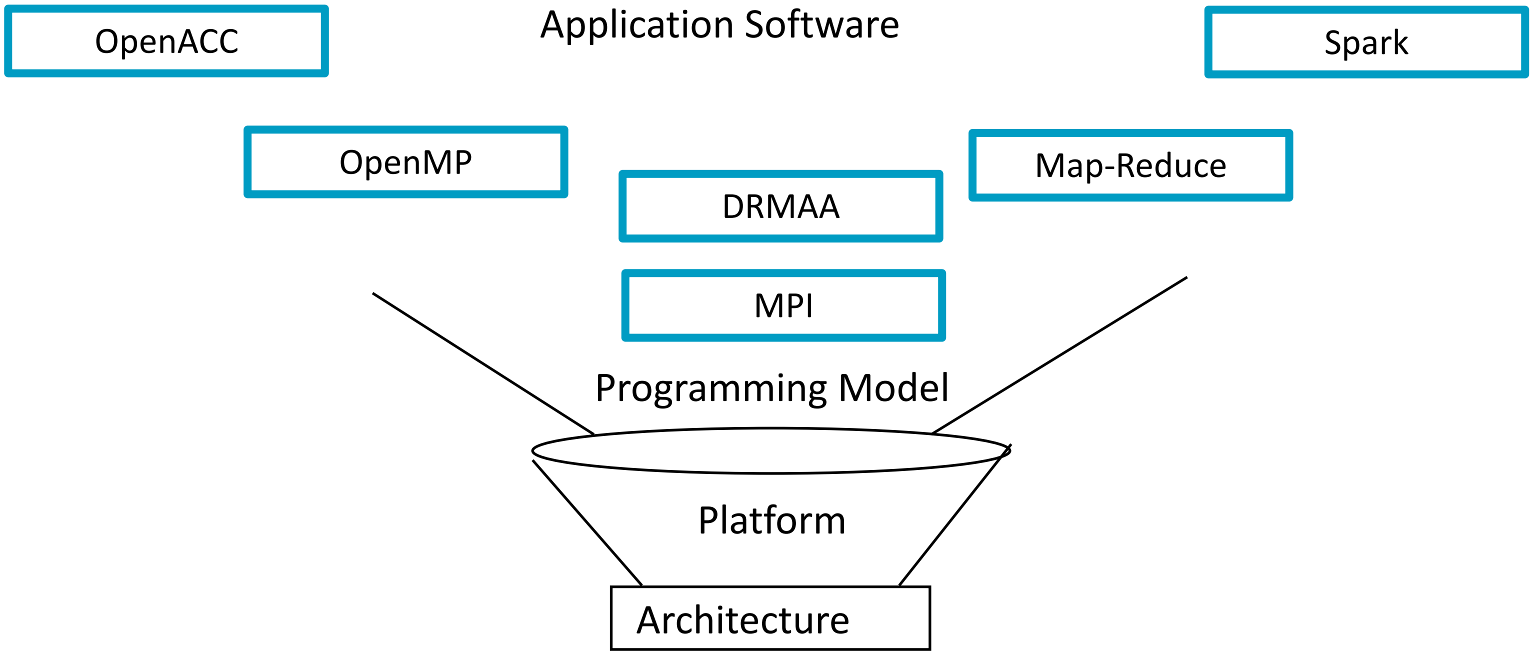
C.3. Dataflow Processing

C4. Stream Data Processing

WRAP-UP: ADVANCED TOPICS

CS205: Contents

Programming Models, Platforms and Infrastructures



amazon web services

CLOUD

FAS RC CANNON
HARVARD'S LARGEST CLUSTER

"ON-PREMISES"

CS205: Prerequisites

What You Need?

Basic programming experience in C and Python

Basic knowledge of Linux including using the command line

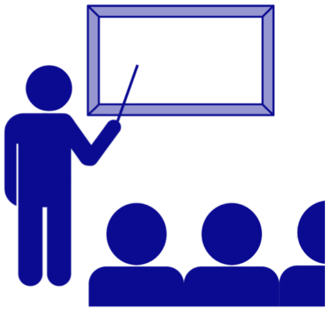
Basic understanding of algorithms (CS107/AC207 or CS50)

Willingness to learn new programming models and platforms

- Time consuming
- We will be on the bleeding edge
- We will try to make it fun!

CS205: Methodology

The “Threads” of Learning



Lectures with the principles, examples, reading assignments, and open discussions

Guest lectures from experts in the field to expose students to real-world life experiences

Hands-on sessions to practice examples, programming assignments (homework), and lab sessions with infrastructure guides

Course final project to apply the concepts in real-life

CS205: Methodology

Lecture, Hands-on and Lab Sessions

Lecture Sessions (TUE/THU 1:30PM-2:45PM on Zoom)

- Theoretical concepts to build a conceptual framework
- Simple examples and **case studies** to illustrate the theory
- Reading assignments for open discussion in class to develop critical thinking and problem-solving strategies
- Quizzes to assess your understanding of the material
- Guest lectures by experts

Hands-on Sessions (TUE/THU 1:30PM-2:45PM on Zoom)

- Learn and practice the main programming models

Lab Sessions (Variety of times on Zoom)

- Allow students to become familiar with the computing and data processing infrastructure on AWS by following the infrastructure guides
- Provide help with the programming assignments and final project

CS205: Methodology

Practical (Programming) Assignments

- Lecture, hands-on and lab sessions are complemented by 3 programming assignments, one for each main part of the course, to bridge the theory with the practice

- Mostly consist of programming assignments to exercise a technology or programming model

  - ✓ Compiler optimization
  - ✓ Computing acceleration with OpenACC
  - ✓ Shared-memory parallel programming with OpenMP
  - ✓ Distributed-memory parallel programming with MPI
  - ✓ Batch data processing with MapReduce
  - ✓ Dataflow processing with Spark

- Students are expected to have basic programming experience, familiarity with Python and C, basic knowledge of Linux including using the command line

CS205: Methodology

Infrastructure Guides

Programming assignments and final project can be developed on:

CLOUD

"ON-PREMISES"

We provide guides to illustrate how to deploy parallel computing and big data processing frameworks on AWS:

I1. First access to AWS
I2. OpenNebula Sandbox on AWS
I3. Docker on AWS
I4. OpenACC on AWS
I5. Performance optimization on AWS
I6. OpenMP on AWS
I7. MPI cluster on AWS
I8. Hadoop cluster on AWS
I9. Install Spark in local mode
I10. Spark cluster on AWS

CS205: Methodology

Final Project: Peer-based Learning

- A major component of the course is a final programming project

- Solve a compute or data intensive scientific problem using the platforms, tools and systems introduced in the course. Collect the data, implement the tool, and analyze the performance of an end to end application

- Six milestones

  - ✓ Team formation
  - ✓ Project proposal
  - ✓ In-class project proposal presentation
  - ✓ In-class progress presentation with design
  - ✓ Final project submission
  - ✓ Final presentation to teaching staff

- You are required to form teams and to partition the work among the team members.

CS205: Methodology

Grading

- **Homework (40%):** Programming assignments
- **Final Project (40%):** Final project
- **Quizzes (10%):** Assessments of your understanding of the material
- **Participation (10%):** Online forum posts, reading assignments, in-class participation and lecture attendance

Lectures are optional, but **strongly** encouraged. Good reasons for missing lecture include time zone conflicts.

Lab attendance is mandatory.

All students are expected to contribute online on Piazza, and participation on Piazza will contribute to the final grade.

Course Website

https://harvard-iacs.github.io/2021-CS205/

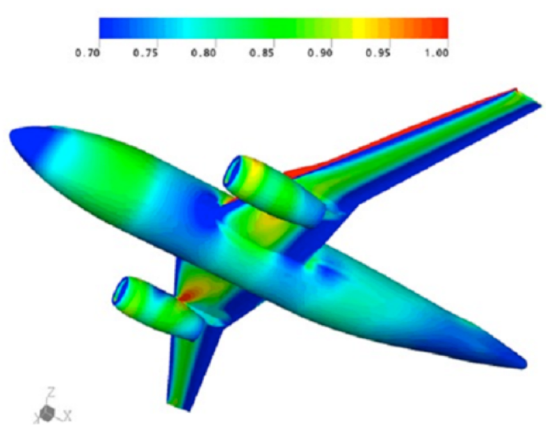
IACS 2021-CS205 | Syllabus | Schedule | Course Flow | Resources | Materials | Project | Search Topic

CS205: Computing Foundations for Computational Science

CS 205 Computing Foundations for Computational Science

IACS

"Big" Compute



Big Data

CS205: Staff

Lead Instructor: Dr. David Sondak

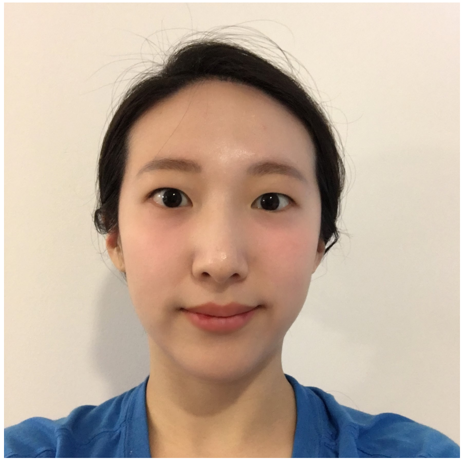
- Lecturer on Computational Science
- Research on fluid mechanics, machine learning, and physics-aware machine learning
- Teach CS107/AC207 in fall semesters
- Hobbies: Dogs, soccer, reading, outdoors

Email: dsondak@seas.harvard.edu

Room: https://harvard.zoom.us/my/dsondak

CS205: Staff

Teaching Fellows



Hayoun Oh
hayounoh@g.harvard.edu

Simon Warchol
simonwarchol@g.harvard.edu

Oluwatosin (Tosin) Alliyu
oalliyu@mde.harvard.edu

Haipeng Lin
hplin@seas.harvard.edu

Dovran Amanov
damanov@g.harvard.edu

Next Steps

- Complete mandatory course survey
  https://forms.gle/Hj5sqhbzAckP7v9L8

- Get signed up for **Piazza**
  http://piazza.com/harvard/spring2021/cs205

- Read https://harvard-iacs.github.io/2021-CS205/ (contents, syllabus…) carefully

- Get ready for **next lecture**:
  A.1.Parallel processing architectures

Next Steps

SIGN UP FOR A REGULAR AWS ACCOUNT

1. First apply for a regular account.

2. We will give you AWS credits soon.

DO NOT apply for an AWS Educate Starter Account

Questions

Large-Scale Computational and Data Science