Appetizer for course
“Parallel Algorithms and Programming”

Martin Schreiber

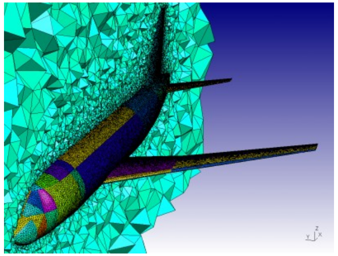
Université Grenoble Alpes

Inria

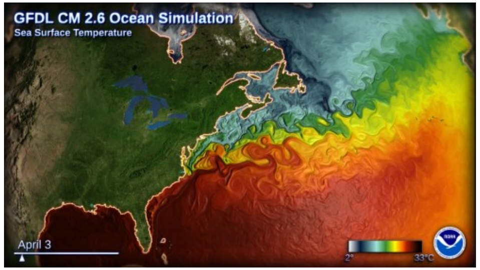
Laboratoire Jean Kuntzmann
Mathématiques Appliquées - Informatique

Demand for large compute power

- **Numerical simulation**
  - Weather / climate prediction
  - Design of (combustion) engines
  - Design of aircraft
  - Drug design
  - Flood predictions
  - Astronomy
  - Etc.



- **Other applications**
  (partly beyond this course)
  - Artificial Intelligence
  - Entertainment
  - Big Data
  - Etc.



Increasing computational demands

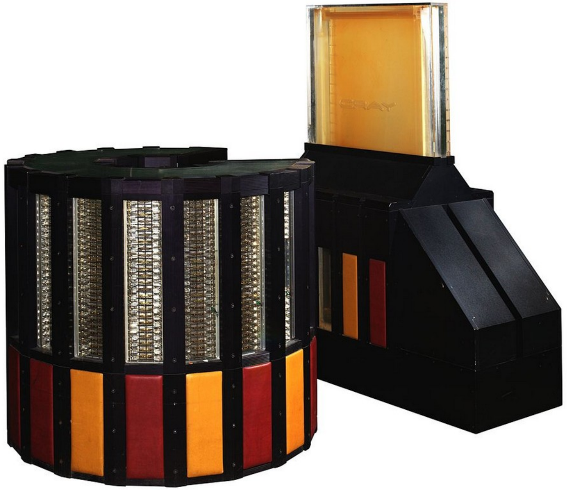
- **Faster**: Solve problems more rapidly
  - Climate simulations
  - Execute more requests per second (example Google search)
  - Enhance the response time of interactive applications (online games)
  - Etc.

- **More accurate**: Obtain better results for the same execution time
  - Weather/climate simulations (e.g., Météo France)
  - Use more complex models (multi-physics)
  - Etc.

How to cope with these demands? In the past...

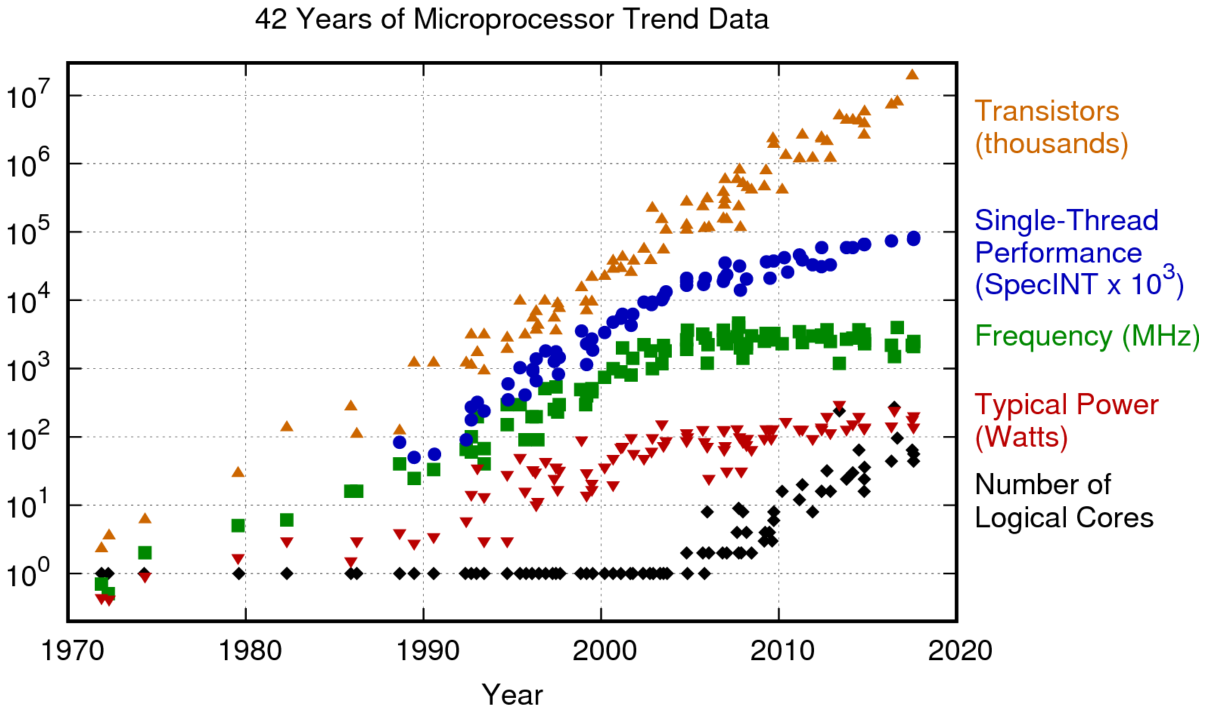
E.g. Cray 1 (1975)

Cray 2 (1985)



- Vector architectures
- No (significant) algorithmic changes required to use algorithm on next generation

How to cope with these demands? And now...

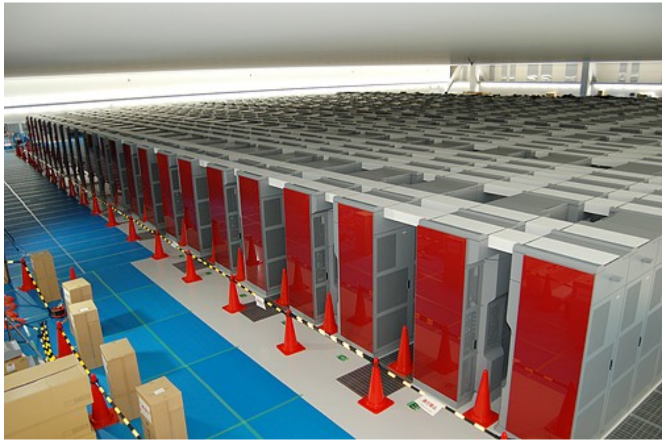


Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2017 by K. Rupp

How many cores are on your smartphone?

Not only for super computers...

...because it’s everywhere



Supercomputers

mobile devices (e.g., iPad)

Green computing

Goal of this course

Theory:
We will learn about, e.g.,

- Different **parallelization levels of architectures** (ILP, shared, distributed memory)
- **parallelization patterns**
- **parallel performance** => modeling (find the best computation / communication ratio / architecture)

Practical:
We will learn about, e.g.,

- … different parallel programming models (OpenMP, MPI)
- … performance measurements
- … parallel debugging

Summary

- **Parallel** architectures
- **Parallel** algorithms
- **Parallel** programming models

Prerequisites & schedule

Prerequisites

- Computer architectures
- Operating systems
- C programming skills

Schedule

- 15 hours of **lectures**
- Used for the theory

Prepares you for the **tutorials & lab**

- 18 hours of **tutorials & lab work**
- Tutorials: Bridge the theory from the lectures to the lab
- **Labs: Hands-on work with OpenMP & MPI programming**

Grading

- One grade from the **labs** (2 submissions)

- One **final exam** (covering lectures, tutorials & labs)

- Planned **final grade**
  = 0.3x lab + 0.7x exam

Martin Schreiber, Université Grenoble Alpes / Laboratoire Jean-Kuntzmann

Team

- **Lectures**
  - **Martin** SCHREIBER
  - Member of UGA, also LJK & Inria
    https://www.martin-schreiber.info/
  - Member of MPI Forum and (soon) OpenMP ARB

- **Tutorials/labs**

Guillaume
RAFFIN

Benjamin
ZANGER

Any questions?

General questions

Martin Schreiber: martin.schreiber@univ-grenoble-alpes.fr

Lab/tutorial questions

Guillaume Raffin: Guillaume.Raffin@univ-grenoble-alpes.fr

Benjamin Zanger: Benjamin.Zanger@univ-grenoble-alpes.fr