

GP-GPU and High Performances Computing

Lecture 1

Introduction

What day are you born ?

(keep the day in your head)

Add the digits

If number is greater than 10, keep the last digit

If number is less than 10, keep the digit

Say the number aloud

Find the smallest number in your row - in the room

What Did You Just Do?

- Each of you computed a **local, independent result** (a checksum, then a row minimum)
- You combined these partial results **step by step** – row → room – until a single answer remained
- This "compute locally, then combine progressively" pattern is a **parallel reduction**

You just ran, by hand, the exact algorithm GPUs use to sum or find the minimum of millions of values – we'll implement it in CUDA in Lecture 5, and optimize it further in Lecture 11.

TOP 500

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
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
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
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
4	JUPITER Booster - BullSequana XH3000, GH Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, RedHat Enterprise Linux, EVIDEN EuroHPC/FZJ Germany	4,801,344	793.40	930.00	13,088
5	Eagle - Microsoft NDv5, Xeon Platinum 8480C 48C 2GHz, NVIDIA H100, NVIDIA Infiniband NDR, Microsoft Azure United States	2,073,600	561.20	846.84	

HPCG 500

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	HPCG (TFlop/s)
1	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	17406.90
2	7	Supercomputer Fugaku - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	16004.50
3	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	14054.00
4	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	5612.60
5	9	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	4586.95

Green 500

Rank	TOP500 Rank	System	Cores	Rmax (PFlop/s)	Power (kW)	Energy Efficiency (GFlops/watts)
1	259	JEDI - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, ParTec/EVIDEN EuroHPC/FZJ Germany	19,584	4.50	67	72.733
2	148	ROMEO-2025 - BullSequana XH3000, Grace Hopper Superchip 72C 3GHz, NVIDIA GH200 Superchip, Quad-Rail NVIDIA InfiniBand NDR200, Red Hat Enterprise Linux, EVIDEN ROMEO HPC Center - Champagne-Ardenne France	47,328	9.86	160	70.912
3	484	Adastra 2 - HPE Cray EX255a, AMD 4th Gen EPYC 24C 1.8GHz, AMD Instinct MI300A, Slingshot-11, RHEL, HPE Grand Equipement National de Calcul Intensif - Centre Informatique National de l'Enseignement Suprieur (GENCI-CINES) France	16,128	2.53	37	69.098
4	183	Isambard-AI phase 1 - HPE Cray EX254n, NVIDIA Grace 72C 3.1GHz, NVIDIA GH200 Superchip, Slingshot-11, HPE University of Bristol United Kingdom	34,272	7.42	117	68.835

General information

General course information

- **Contact:** Christophe Picard
 - christophe.picard@univ-grenoble-alpes.fr
- **Offices:**
 - 174 - IMAG Building - 1st floor
 - C102 - Ensimag Building - 1st floor
- **Email headers:** [CGPU]
- Course notes available on my website
- Class organization
 - Lectures and labs
 - One written exam and a project

About the project

- It is your project. You choose the subject.
- It can be a group project (2 or 3).
- It should be related to your major (MMIS, MSIAM, MoSiG).
- You may reused project from other courses, but you should propose an improvement.
- It must involve some programming (Python, C, C++, Julia, Fortran).
- It must involve some design. You are not allowed to just use a library for parallelism.
- You must submit your proposal for project by **October 13th 2026**.

Grading the project

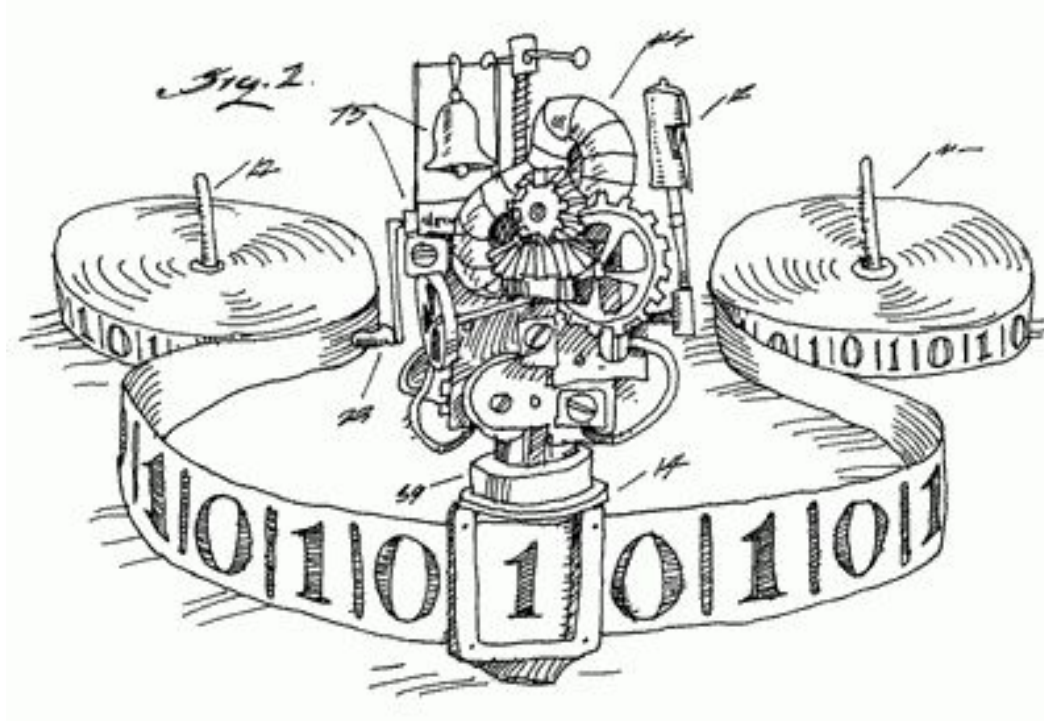
The grade will depend on:

- The quality of the code.
- The quality of the report.
- The design of the parallelism.
- The performances.
- The number of students in the project.
- The difficulty of the subject.

Introduction

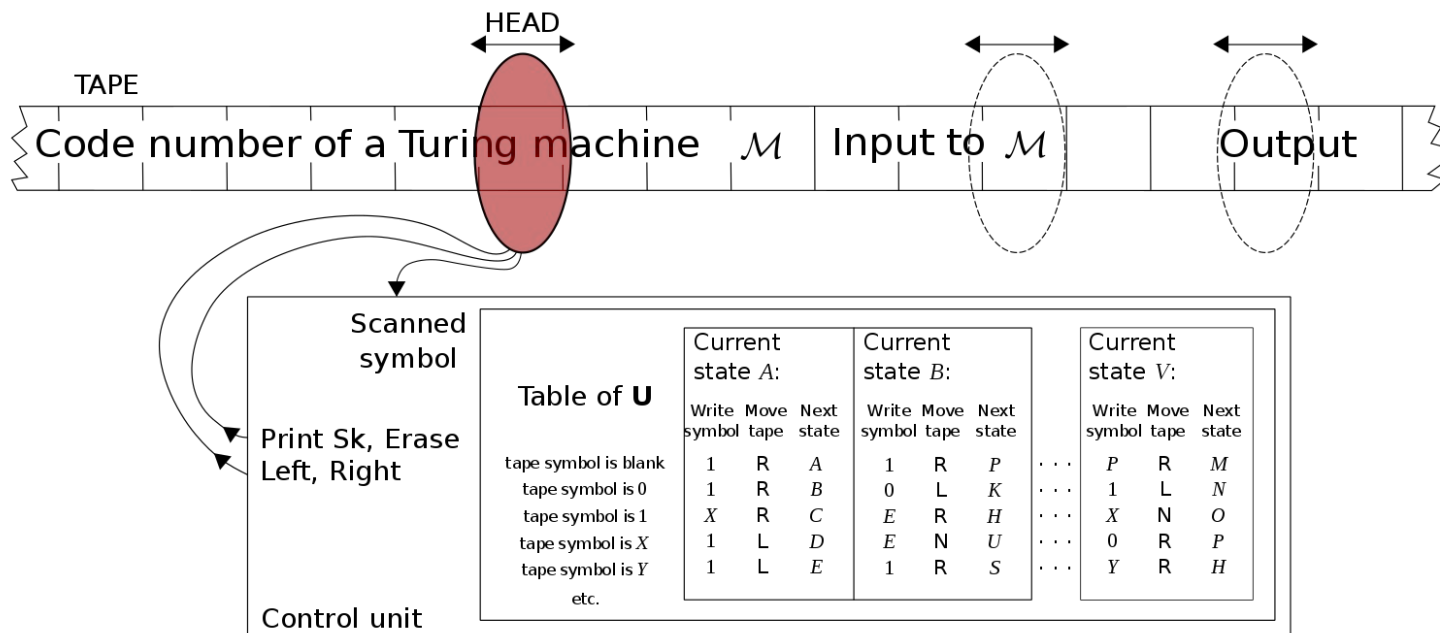
What's a computer look like?

Turing machine



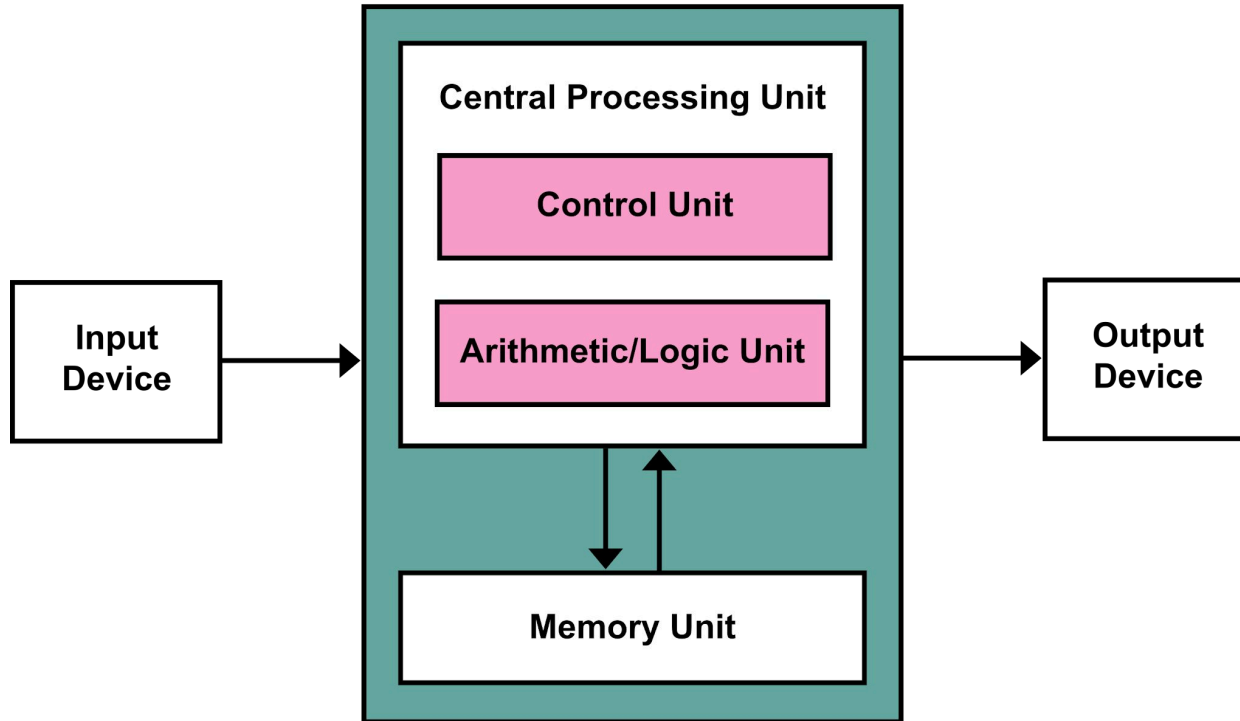
What's a computer look like?

A universal Turing Machine



What's a computer look like?

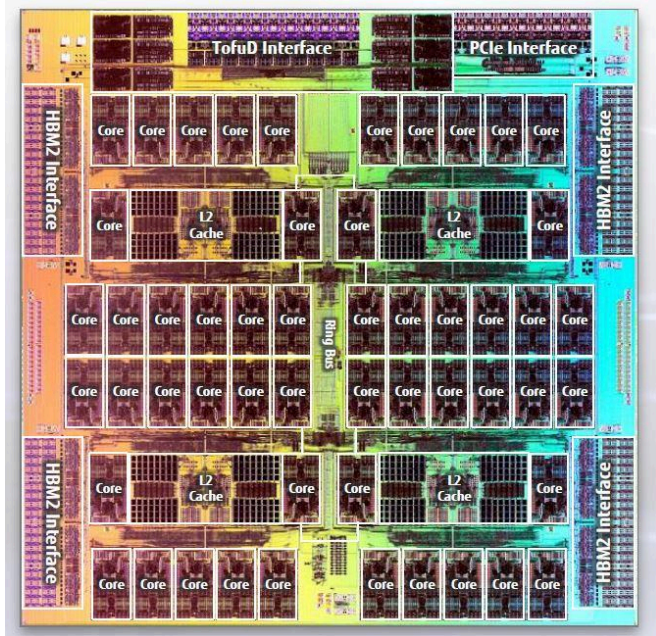
Von Neumann Architecture



The Von Neumann Architecture

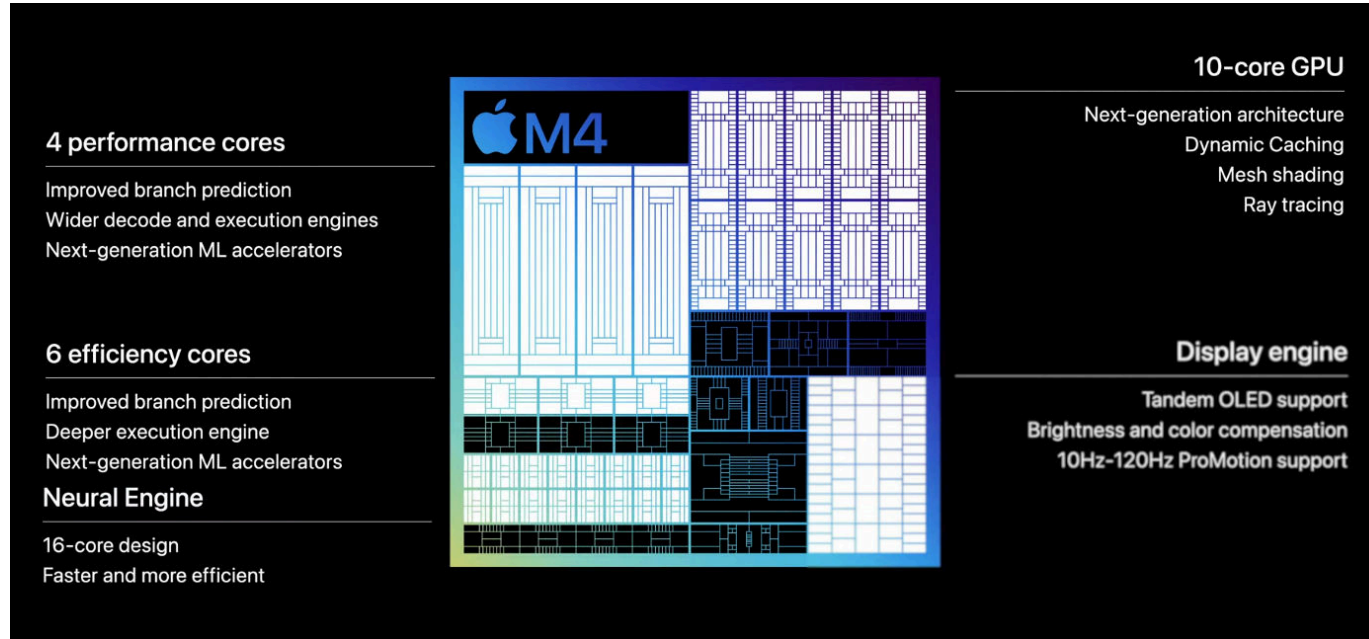
- CPU and Memory Unit.
- Instructions are executed sequentially.

A64FX CPU



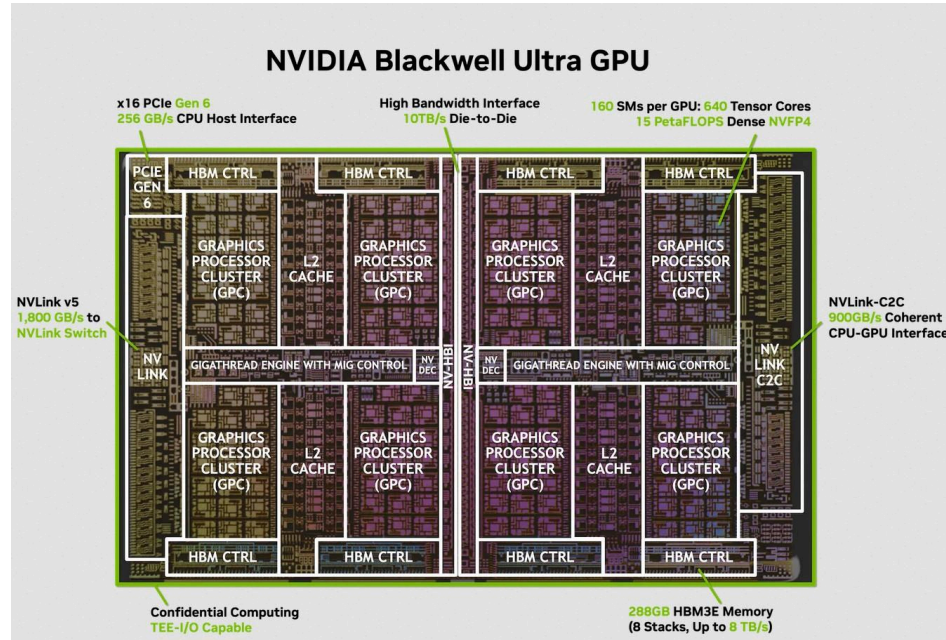
- 4 NUMA nodes
- 12 compute cores each
- Memory : 32 GB
- FLOPS : 13.5×10^{12}

Apple M4



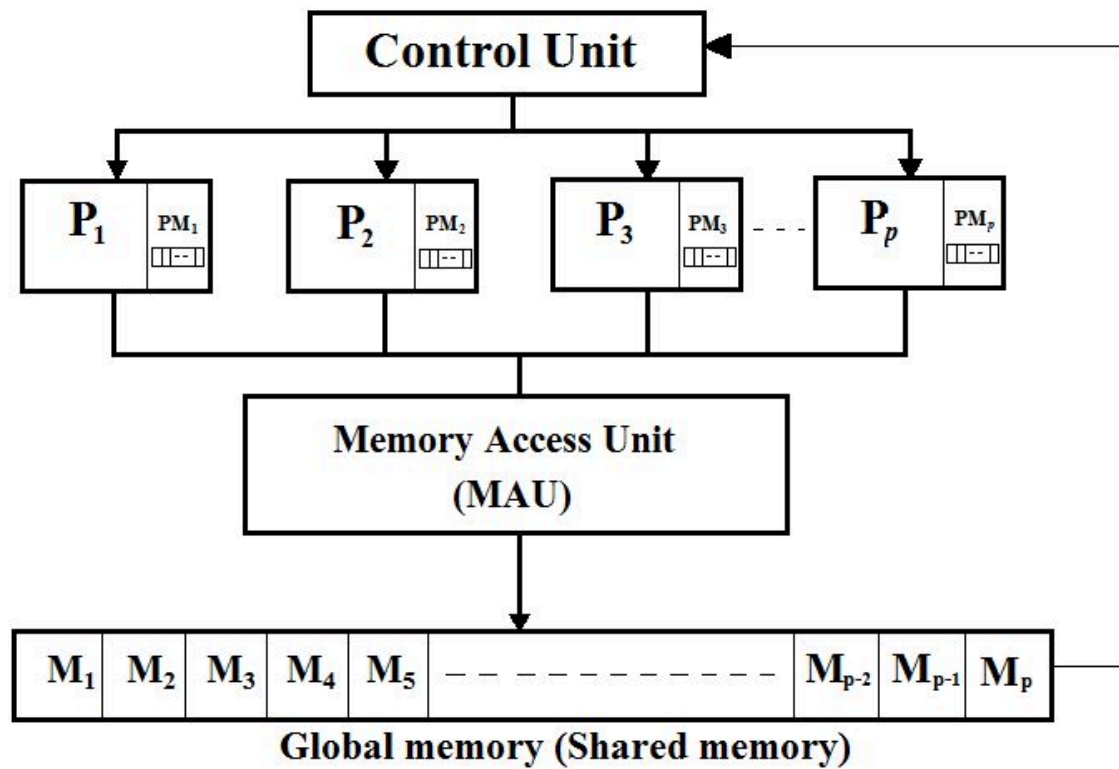
FP16 : 8.52 TFLOPS FP32 : 4.26 TFLOPS Execution units : 160 Max. Memory : 20GB

Nvidia Blackwell



FP16 : 2.25 PFLOPS FP32 : 1.12 PFLOPS Execution units : 192 Max. Memory : 48GB

PRAM Model



The Von Neumann Architecture (Cont.)

- Algorithmic cost measured using **Big-O notation**.
- This model is incomplete for predicting real performance.

The Sequential Model is Wrong

- Modern computers are **parallel entities**.
- The **Frequency Wall** led to the focus on parallelism.

Parallelism is

- Parallelism must be exploited for **time efficiency** and **power savings**.

Goals of Parallel Programming (I/II)

- Solve the **same problem faster** (Strong Scaling).

Goals of Parallel Programming (II/II)

- Solve a **larger problem** in **constant time** (Weak Scaling).

Reasons NOT to Write Parallel Programs

- Tools are difficult to use: expect 10x programming effort
- for many problem performance does not improve
- This course will help to decide when to develop a parallel approach and how to write it.

Context

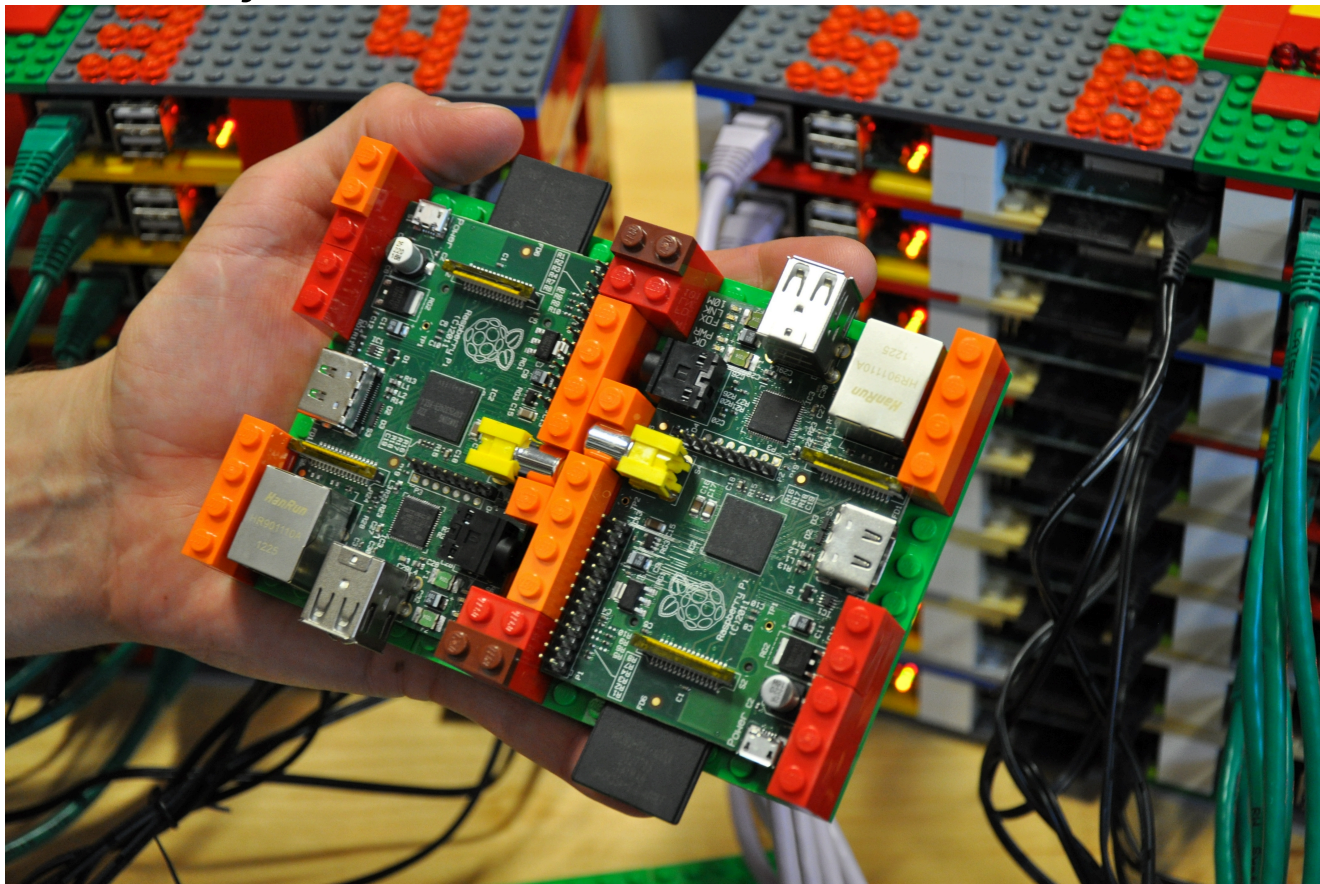
- Decrease time to solution.
- Solve larger problem.
- Combine resources of several processing units: gain access to more memory and more processing power.
- Harness the processing power of modern architectures.
- Use idle computer to perform embarrassing parallelism computation (SETI@home).
- Improve the precision of computations in a limited time (weather forecast).

How to achieve efficient parallelism?

- **Processors:** multicore, memory, network, accelerators, instructions.
- **Compilers:** dedicated library, automatic parallelism.
- **Algorithms:** tailored algorithms.
- **Mathematics:** adapted numerical methods, evolutionary methods.

Few words on hardware

A student HPC system



A “real” HPC system



What you will be working with !

- 33 RTX A6000 are available.
- Ressources are on virtual machine (up to 3 students on each card)
- **Technical details :**
 - CUDA Cores 10,752
 - NVIDIA Tensor Cores 336
 - NVIDIA RT Cores 84
 - GPU 46 GB GDDR6
 - Performances FP32 - 38,7 TFLOPS



Conclusions

- Every computer is a parallel computer
- Parallel computers need independent work to run their many cores (or other resources) efficiently

Themes of this class

- Identify available parallelism in application
- Design parallel approaches
- Understand parallel hardware and how to optimize parallel performance

Roadmap

Lecture	Topic
2	GPU Architecture
3 - 4	GPU Programming: threads, memory hierarchy, warps, occupancy
5 - 6	Parallel Patterns: Reduction, Scan
7	Matrix Multiplication & Tiling
8	Design of Parallel Programs (Foster's Methodology)
9	Performance Metrics: Amdahl's Law, Roofline, Profiling
10	Sparse Matrix Methods
11	Histograms & Atomics
12	Graph Algorithms: BFS, Sorting
13	Course Synthesis