Smilei)

4th Workshop

Summary for GDR - APPEL

November 2023

Arnaud Beck – Laboratoire Leprince - Ringuet

eli beamlines

Outline

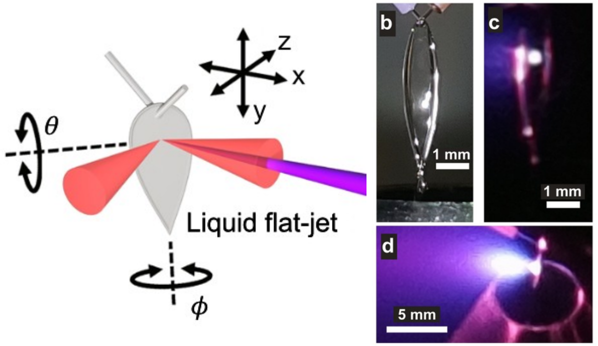
- Smilei status and review
- PIC basics
- Super computing landscape

Project Review

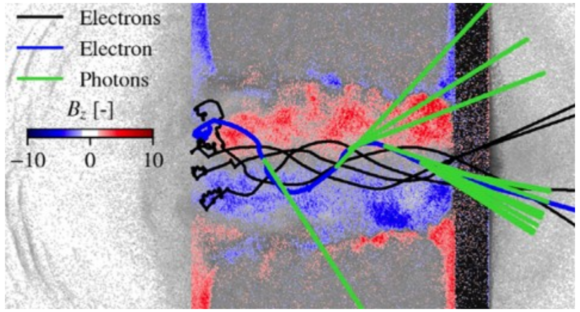
The Particle-In-Cell (PIC) simulation of plasmas

from Laboratory Plasmas ...

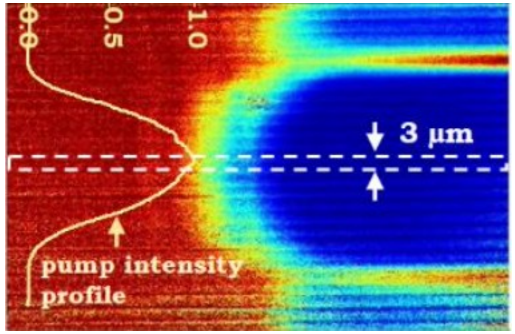
High-Harmonic Generation



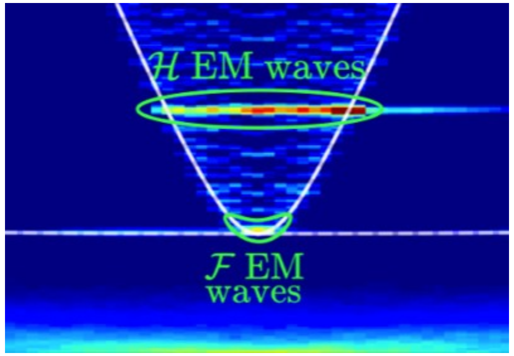
Radiation sources from DLT



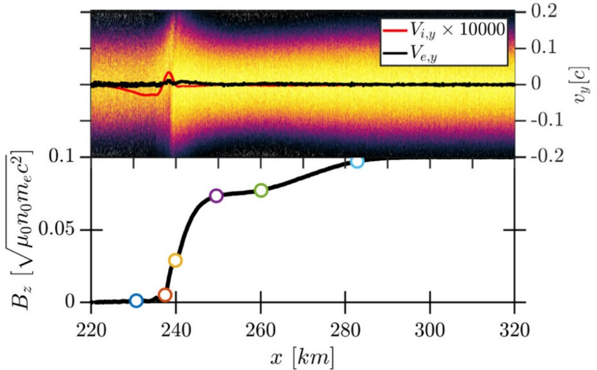
Solid to plasma transition



Solar radio-burst



Comet environment



Collisionless shocks & Dark Matter

... to Space & Astrophysical Plasmas

Smilei in a nutshell

2013
Start of the project*

2014
Gitlab
release to co-dev

2016
1st physics studies &
large scale simulations
Github

2018
Reference paper

*objective: develop the first open-source PIC code harnessing new paradigms of high-performance computing

Open-source & Community-Oriented
documentation • chat • online tutorials • post processing & visualization
training workshops • summer school & master trainings • issue reporting

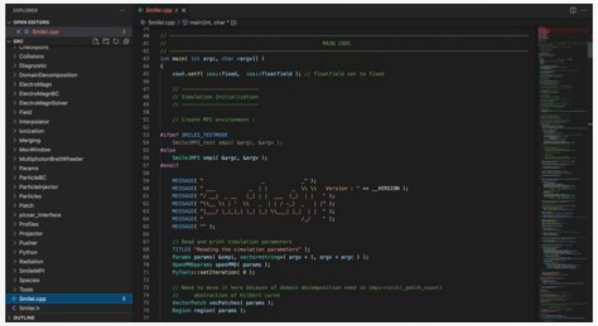
Multi-Physics & Multi-Purpose
advanced physics modules: geometries, collisions, ionization, QED
broad range of applications: from laser-plasma interaction to astrophysics

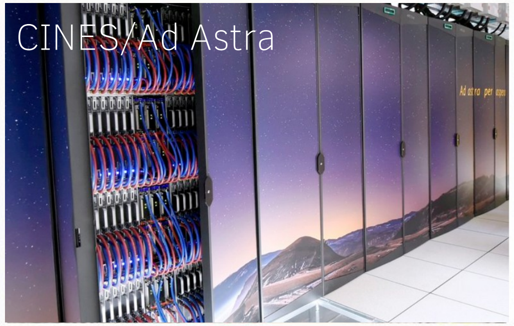
High-performance
C++/Python • MPI/OpenMP/OpenACC/CUDA/HIP • SIMD • HDF5
designed for the latest architectures

What you get with Smilei

A high-performance PIC code

running on various supercomputers worldwide

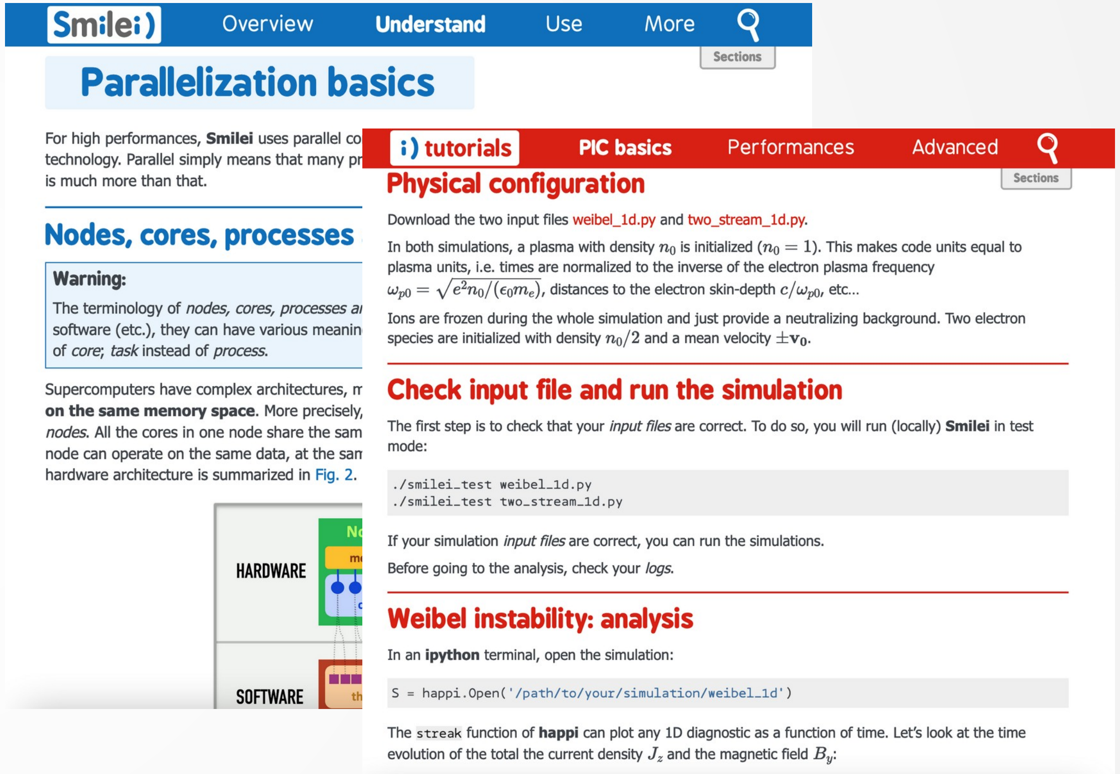




with dedicated **post-processing tools** (Happi) and an ensemble of **benchmarks** (Easi, for continuous integration)

An extensive documentation

with online tutorials



Parallelization basics

For high performances, **Smilei** uses parallel co
technology. Parallel simply means that many pr
is much more than that.

Nodes, cores, processes

Warning:

The terminology of *nodes, cores, processes* ar
software (etc.), they can have various meanin
of *core; task* instead of *process.*

Supercomputers have complex architectures, m
on the same memory space. More precisely,
nodes. All the cores in one node share the sam
node can operate on the same data, at the sam
hardware architecture is summarized in Fig. 2.

Physical configuration

Download the two input files weibel_1d.py and two_stream_1d.py.

In both simulations, a plasma with density n_0 is initialized ($n_0 = 1$). This makes code units equal to plasma units, i.e. times are normalized to the inverse of the electron plasma frequency $\omega_{p0} = \sqrt{e^2 n_0/(\epsilon_0 m_e)}$, distances to the electron skin-depth c/ω_{p0}, etc…

Ions are frozen during the whole simulation and just provide a neutralizing background. Two electron species are initialized with density $n_0/2$ and a mean velocity $\pm\mathbf{v_0}$.

Check input file and run the simulation

The first step is to check that your *input files* are correct. To do so, you will run (locally) **Smilei** in test mode:

```
./smilei_test weibel_1d.py
./smilei_test two_stream_1d.py
```

If your simulation *input files* are correct, you can run the simulations.

Before going to the analysis, check your *logs*.

Weibel instability: analysis

In an **ipython** terminal, open the simulation:

```
S = happi.Open('/path/to/your/simulation/weibel_1d')
```

The `streak` function of **happi** can plot any 1D diagnostic as a function of time. Let's look at the time evolution of the total the current density J_z and the magnetic field B_y:

and a collaborative community

Github
sources, issues

Element
chat

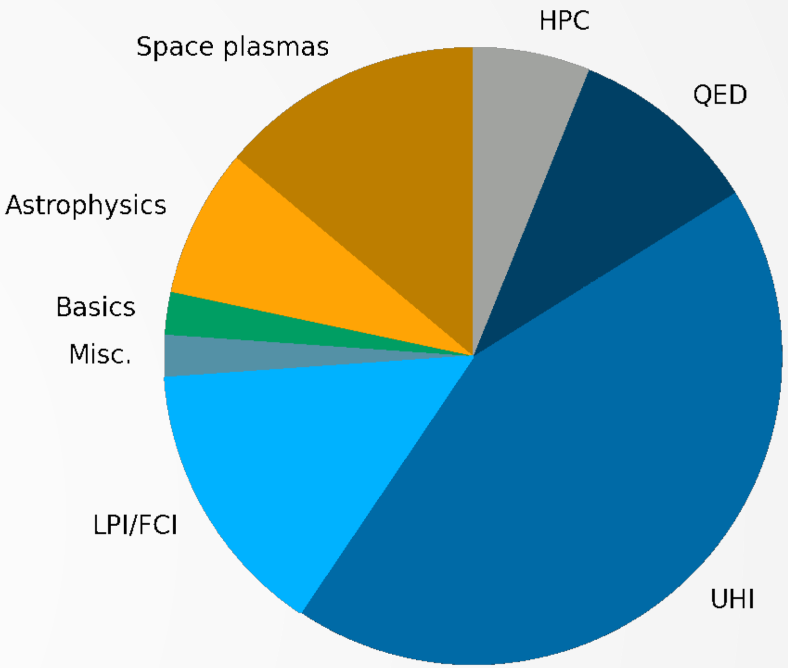
Smilei is a research & teaching platform

Scientific production is rich ...

130+ peer-reviewed papers have been published using Smilei
10+ PhD theses have already been defended

... and focuses on various applications

LPI/IFE : laser-plasma interaction / inertial fusion for energy
UHI : Ultra-high intensity
QED : Quantum electrodynamics (extreme light)
HPC : high-performance computing
Space plasmas & astrophysics

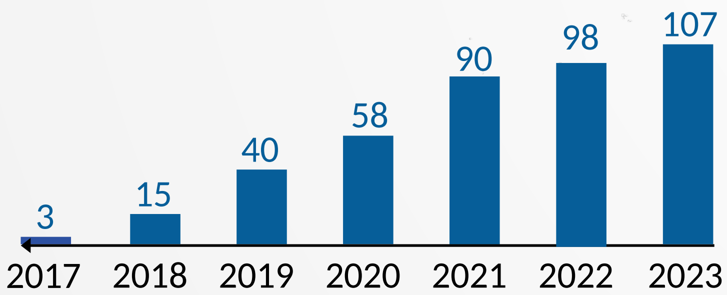


Teaching plasma physics

at the Master/doctoral levels in Europe
in various winter/summer schools
in user & training workshops
via online tutorials

Smilei's user community is international & steadily growing

400+ citations for Smilei reference paper*



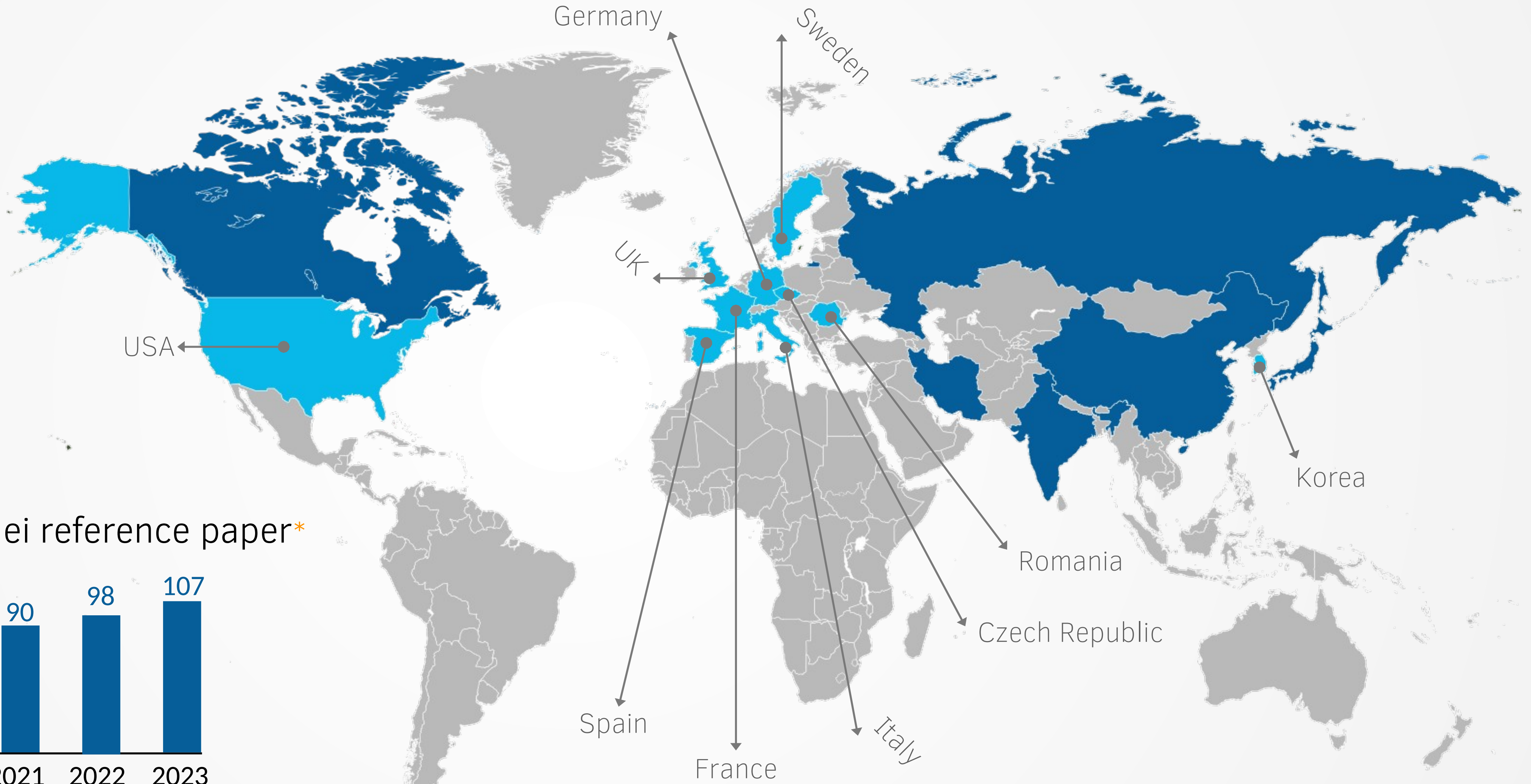
Citations per year / Google Scholar

countries of 1st authors affiliation

---

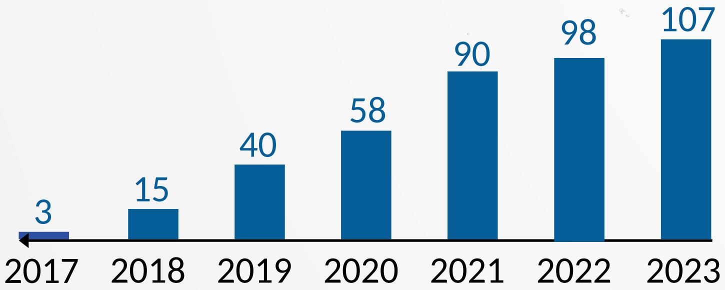
*Dérouillat et al., Comp. Phys. Comm. 222, 351 (2018)

Smilei's user community is international & steadily growing



This year workshop welcomes participants from 10 countries!

400+ citations for Smilei reference paper*



Citations per year / Google Scholar

*Dérouillat et al., Comp. Phys. Comm. 222, 351 (2018)

A project anchored in the French & European HPC landscape

Integration in the French & European HPC landscapes

GENCI
Le calcul intensif au service de la connaissance

- running on all super-computers in France and many in Europe
- 10s millions computing hours every year via GENCI & PRACE/EuroHPC
- GENCI technological survey
- French Project NumPEX, Exascale project

Special/early access to various machines

2015	IDRIS/Turing BlueGene-Q
2016	CINES/Occigen
2018	TGCC/Irene-Joliot-Curie
2019	IDRIS/Jean Zay
2021	RIKEN/Fugaku
2022	CINES/Adastra (GPU)

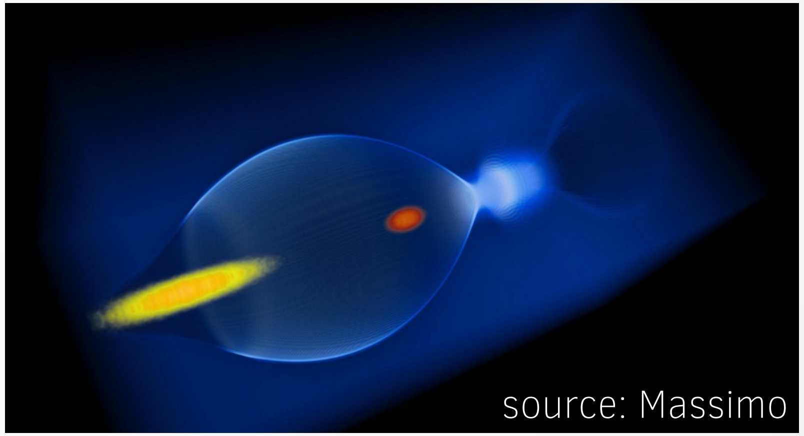
A few recent highlights ...

Code & HPC aspects

- optimization on ARM/RISC architectures*
- parallelization by task**

*Lobet et al., HPCAsia (2022) **Massimo et al., PASC (2022)

Additional physics modules (v4.7 and 4.8)



source: Massimo

- upgrade of the Happi post-processing toolbox & diagnostic suit
- upgrade of the binary collision approach (nuclear reactions)
- Advanced boundary conditions :
  - LaserOffset
  - Perfectly Matched Layers (also work with envelope)
- advanced solvers for laser/particle-driven wakefield acceleration:
  - envelope models in various geometries & accounting for ionization
  - B-TIS3* interpolation scheme to mitigate numerical Cherenkov

More also from our user community

- 30+ new articles published in peer-reviewed journal since our last workshop in March 2022!
- coupling with various codes / experimental data / Machine learning

*Bourgeois & Davoine, J. Plasma Phys. (2023)

… and a very big one!

Smilei 5.0 has just been released and it runs on NVIDIA & AMD GPUs !

NVIDIA

OpenACC, CUDA

AMD

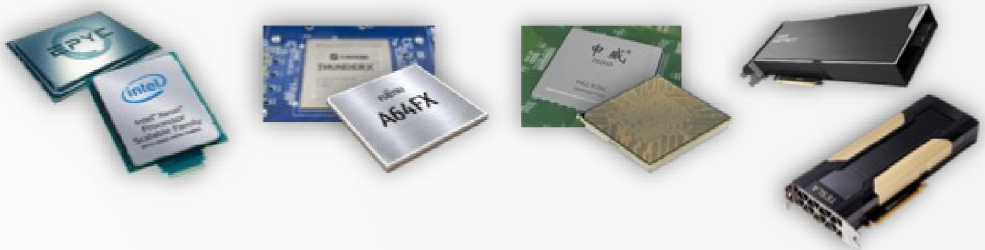
OpenMP, HIP

Standard 2D and 3D simulations are supported

- extensive rewriting to run of both architectures & to insure performance!
- 2D and 3D cartesian geometries with various boundary conditions
- implementation is almost transparent to the user: `Main(..., gpu_computing=True)`
- porting of additional physics modules & advanced solvers is still work in progress
- additional releases will come regularly this year … but there's already plenty you can do!

Perspectives

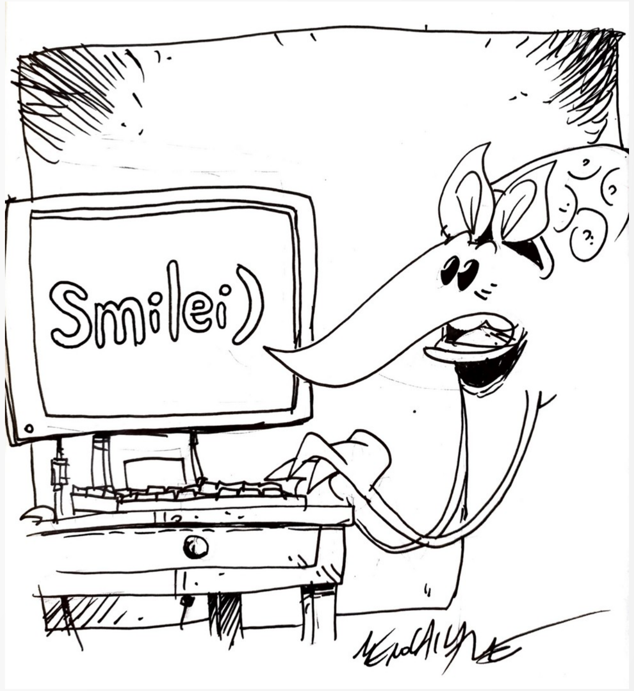
Code & HPC aspects



- GPU porting: AM geometry, adv. phys. modules, load-balancing
- parallelization by task, asynchronism
- advanced IO management (AI approach)
- Boosted frame

Additional physics modules

- coupling with the strong-field QED ToolKit (collab. with MPIK, Heidelberg)
- additional atomic physics processes (Bremsstrahlung & Bethe-Heitler)
- advanced laser field injectors (collab. with ELI Beamlines & CEA/DAM)
- additional nuclear fusion processes (collab. with CELIA)



Keep on building & animating the user community

- encouraging new developers to join in
- developing an online teaching platform (beyond the tutorial approach)
- preparing next user & training workshop !

PIC Basics

What is a PIC code supposed to do?

- Simulate a plasma with kinetic effects (not hydrodynamics)
- Neglect particle-particle interactions (collisions)
- Electromagnetic effects (not electrostatic)

Distribution function

$$f_s(t, \mathbf{x}, \mathbf{p})$$

Vlasov equation

$$\partial_t f_s + \mathbf{v} \cdot \nabla f_s + \overset{\text{Mean force}}{\mathbf{F}} \cdot \nabla_p \overset{\text{Mean distribution}}{f_s} = \cancel{(\partial_t f_s)_{\text{collisions}}}$$

Maxwell equations

$$\nabla \cdot \mathbf{E} = \frac{\rho}{\epsilon_0} \qquad \partial_t \mathbf{E} = -\frac{1}{\epsilon_0}\mathbf{J} + c^2 \nabla \times \mathbf{B}$$

$$\nabla \cdot \mathbf{B} = 0 \qquad \partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

The fields are defined on a grid

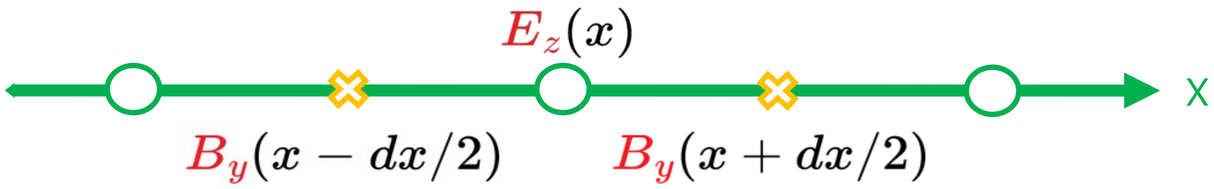
There are several ways to solve Maxwell on a grid.
Let us illustrate with the most common technique
"*Finite Difference Time Domain*" (FDTD)

Maxwell-Ampere in 1D:

$$\partial_t E_z = \partial_x B_y - J_z \quad \Rightarrow \quad (\partial_t E_z)(x) = (\partial_x B_y)(x) - J_z(x)$$

$$\Rightarrow \quad \partial_t \, E_z(x) = \frac{B_y(x+\Delta x/2) - B_y(x-\Delta x/2)}{\Delta x} - J_z(x)$$

offset in
space



A simplified distribution function

Vlasov = partial differential equation in a 6D space.

$$\partial_t f_s + \mathbf{v} \cdot \nabla f_s + \mathbf{F} \cdot \nabla_p f_s = 0$$

Direct integration (*Vlasov codes*) has a tremendous computational cost.

In a PIC code, the distribution function is approximated as a sum over macro-particles

$$f_s(t, \mathbf{x}, \mathbf{p}) = \sum_{p=1}^{N} w_p \, S(\mathbf{x} - \mathbf{x}_p(t)) \, \delta(\mathbf{p} - \mathbf{p}_p(t))$$

Statistical weight

Shape function

From Vlasov to the macro-particle motion

$$f_s(t,\mathbf{x},\mathbf{p}) = \sum_{p=1}^{N} w_p\, S(\mathbf{x} - \mathbf{x}_p(t))\, \delta(\mathbf{p} - \mathbf{p}_p(t))$$

&

$$\partial_t f_s + \mathbf{v}\cdot\nabla f_s + \mathbf{F}\cdot\nabla_p f_s = 0$$

Integrate over $\boldsymbol{p}$ $\qquad \partial_t \mathbf{x}_p = \mathbf{v}_p$

Multiply by $\boldsymbol{p}$ then integrate over $\boldsymbol{p}$ and $\boldsymbol{x}$ → $\qquad \partial_t \mathbf{p}_p = q_s\,\mathbf{E}_p + q_s\,\mathbf{v}_p \times \mathbf{B}_p$

The movement of macro-particles is essentially that of real particles

But …

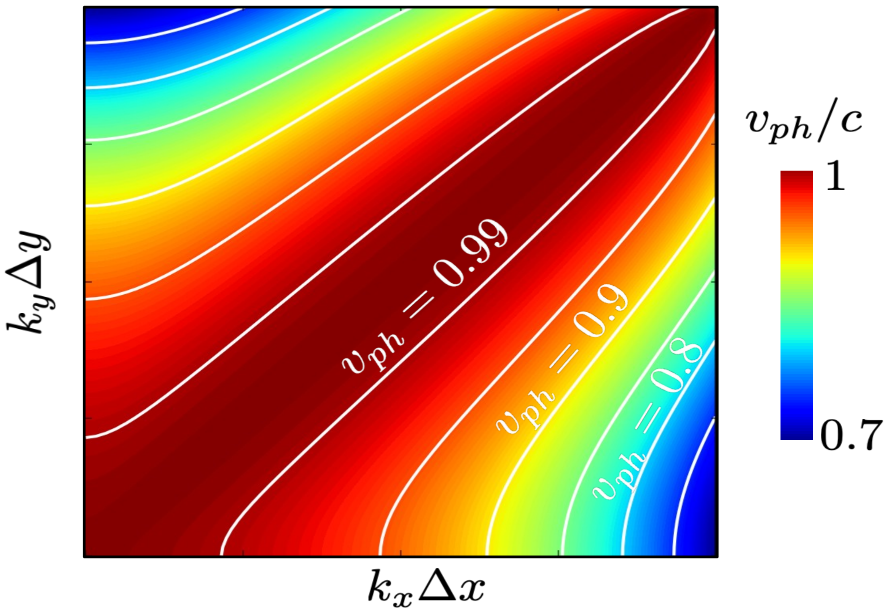
$$\begin{cases} \mathbf{E}_p = \int \mathbf{E}(\mathbf{x})\, S(\mathbf{x} - \mathbf{x}_p)\, d^3\mathbf{x} \\ \mathbf{B}_p = \int \mathbf{B}(\mathbf{x})\, S(\mathbf{x} - \mathbf{x}_p)\, d^3\mathbf{x} \end{cases}$$

The fields are “averaged” around the particle position.

The numerical vacuum is dispersive and anisotropic !

FDTD equations + search for wave-like solutions

➡ **Dispersion relation** $$\Delta t^{-2}\ \sin^2(\omega\Delta t/2) = \sum_{a=x,y,z} \Delta a^{-2}\ \sin^2(k_a\Delta a/2)$$



Dispersive

⬇

Numerical Cherenkov radiation

The timestep cannot be too large

From the dispersion relation, one can show that **stability requires**:

$$\Delta t^{-2} > \sum_{a=x,y,z} \Delta a^{-2}$$

$$\Delta t < \left(\sum_{a=x,y,z} \Delta a^{-2} \right)^{-1/2}$$

Courant-Friedrich-Levy (CFL) condition

The cell size cannot be too large either

Depending on the situation you may need to resolve:

- ✓ The **Debye length** (or the simulation will have numerical heating)
- ✓ The **laser wavelength** (or it won’t propagate)
- ✓ The **skin depth**

⚠ Often, a PIC simulation won’t crash when the results are meaningless.
Users must understand the limitations and test.

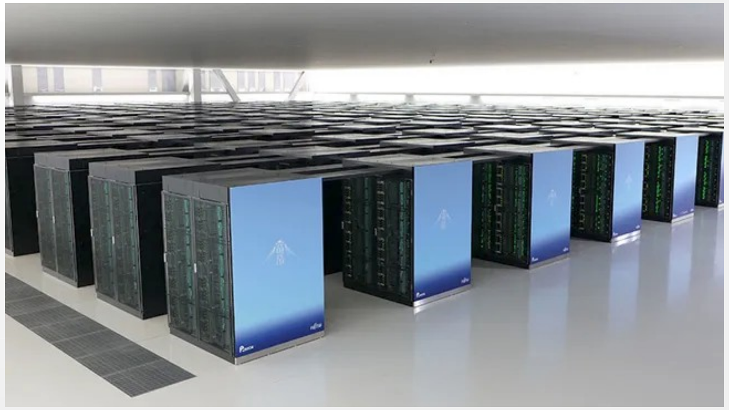
Super Computing Landscape

There are limitations to what personal computers can do

- 3D simulation
- 1000^3 cells
- 8 particles per cell
- 10^6 iterations
- 25 ns per particle per iteration on a single modern processor

More than 6 years to make this simulation on a good desktop computer !

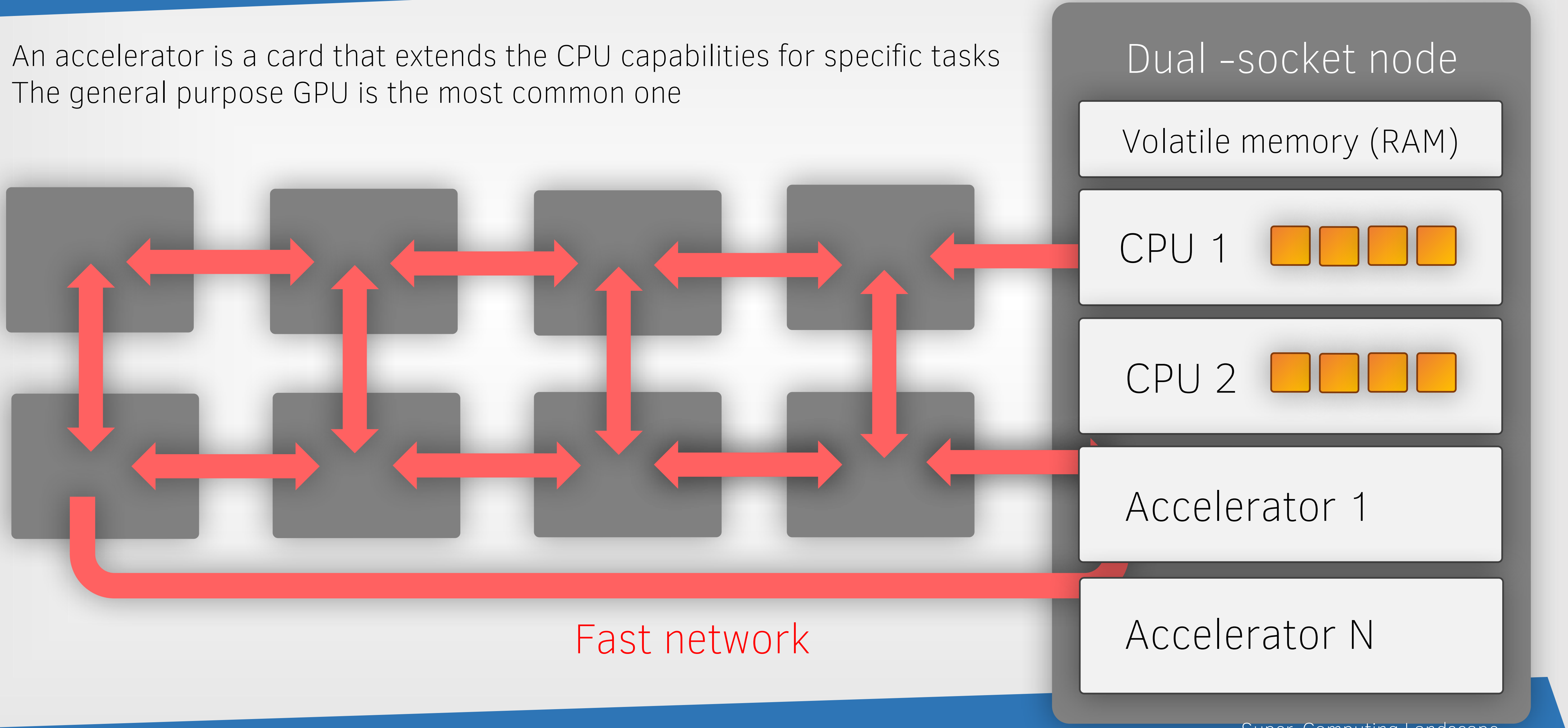
(Assuming you have a few TB of RAM and another few in disk space to store the results)



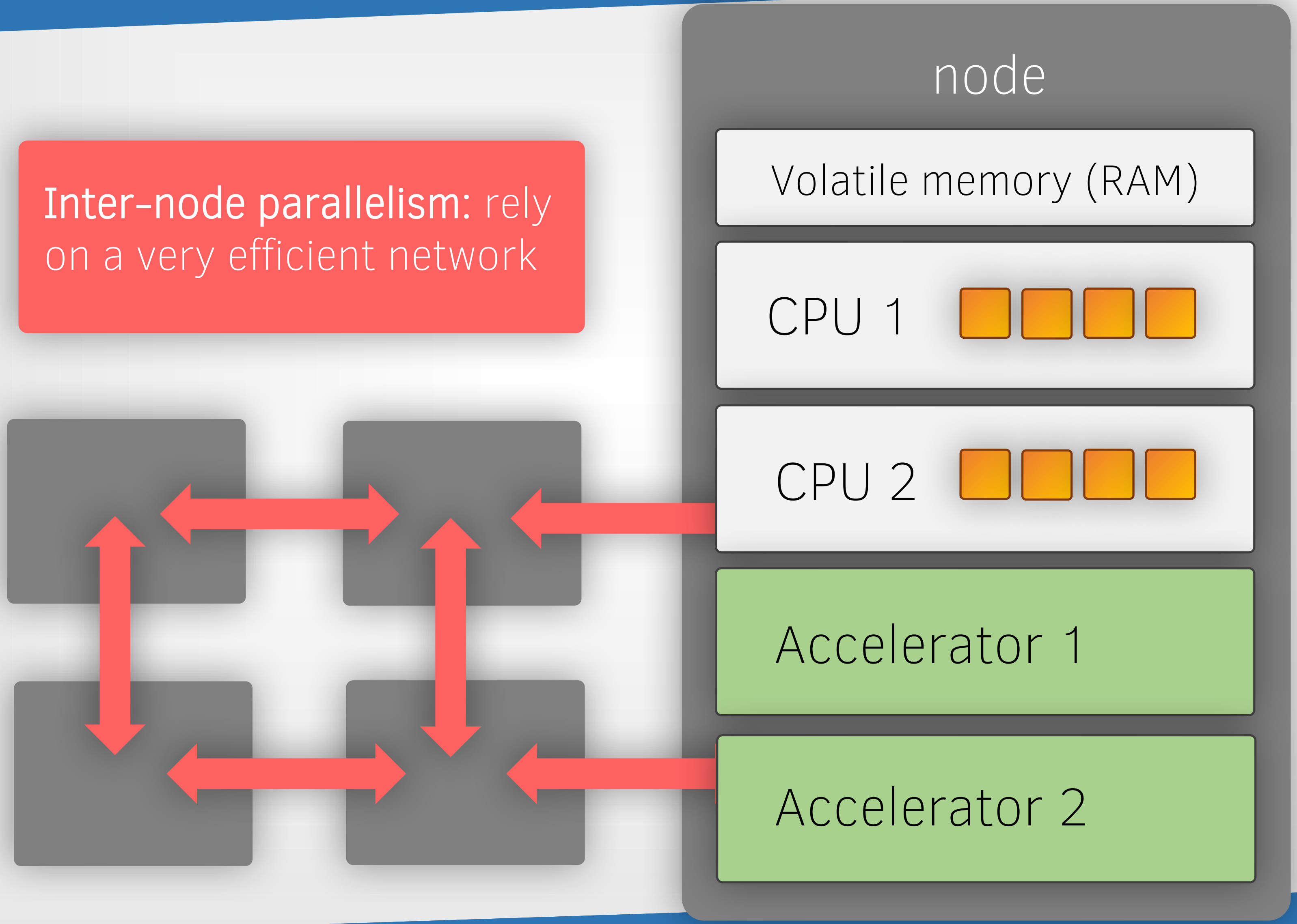
If we really want to do this, we need to use a « Super » computer.

Super computing in a nutshell

- An accelerator is a card that extends the CPU capabilities for specific tasks
- The general purpose GPU is the most common one



Different parallelism levels to handle

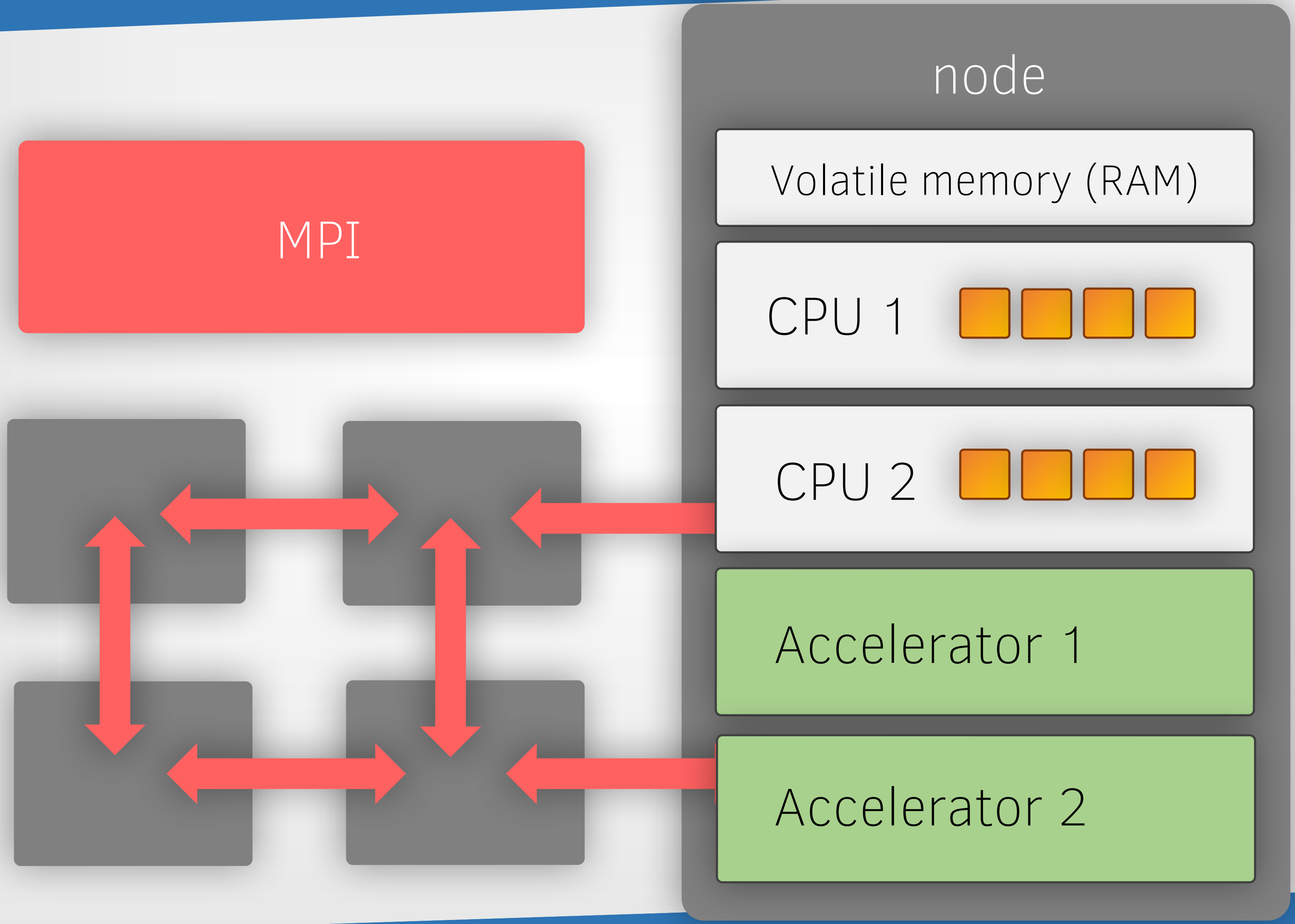


Inter-node parallelism: rely on a very efficient network

Intra-node parallelism: how to deal with all the cores efficiently

Heterogeneity: nodes with different types of computing units

Many software technologies adapted to each level



Smilers choices

OpenMP

OpenMP, OpenACC, CUDA, HIP

Programming challenges for HPC applications

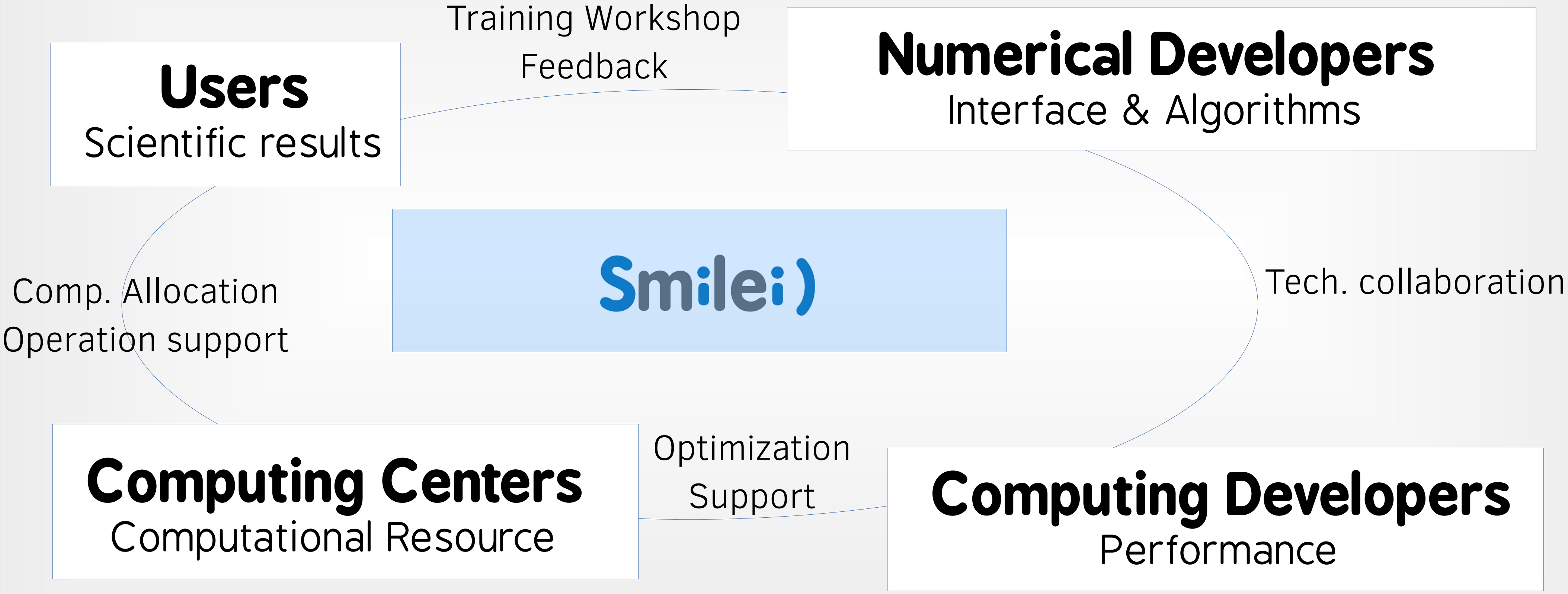
- Developing efficiently for a super-computer is more difficult than for a simple desktop computer:

  - Communication and synchronization through the network between the nodes
  - Load balancing between the nodes
  - Work share within the node (between cores and/or accelerators)
  - Node heterogeneity (CPU/GPU)
  - Memory usage
  - Architecture-specific optimizations (Memory affinity/hierarchy, Vectorization...)
  - Etc

- Typical HPC applications use only a fraction of the total theoretical peak computational power.

- Efficiency on a given hardware also strongly depends on the type of algorithm and the physical case.

The Ecosystem of an HPC application



Users
Scientific results

Training Workshop
Feedback

Numerical Developers
Interface & Algorithms

Comp. Allocation
Operation support

Smilei)

Tech. collaboration

Computing Centers
Computational Resource

Optimization
Support

Computing Developers
Performance

What are the limiting factors of super-computers ?

- Energy consumption => environmental impacts and financial cost
- Memory capability
- Network performance
- Core performance
- File system performance
- Building and maintenance cost
- And more

Computing environmental and financial limitations


Cobalt mine in Congo (2016)


Toxic lake in China

▶ A super-computer has significant environmental impacts: greenhouse gas, pollution, human rights, water, etc.

▶ Powering a super computer is not cheap.

- Today an exascale 21MW system power bill is ~ 30 Million €/year.
- Electricity price is very volatile.

▶ Today it already happens that computing centers are asked to shut down for periods of time for energy savings.

Memory capability is another limitation to the node-level parallelism

- Computational performance has increased faster than the memory capability in both **size** and **access speed**
- Available memory per core has even slightly decreased
- Many algorithm implementations limited by the memory bandwidth and/or the memory size and not the computation power (memory bound algorithms)

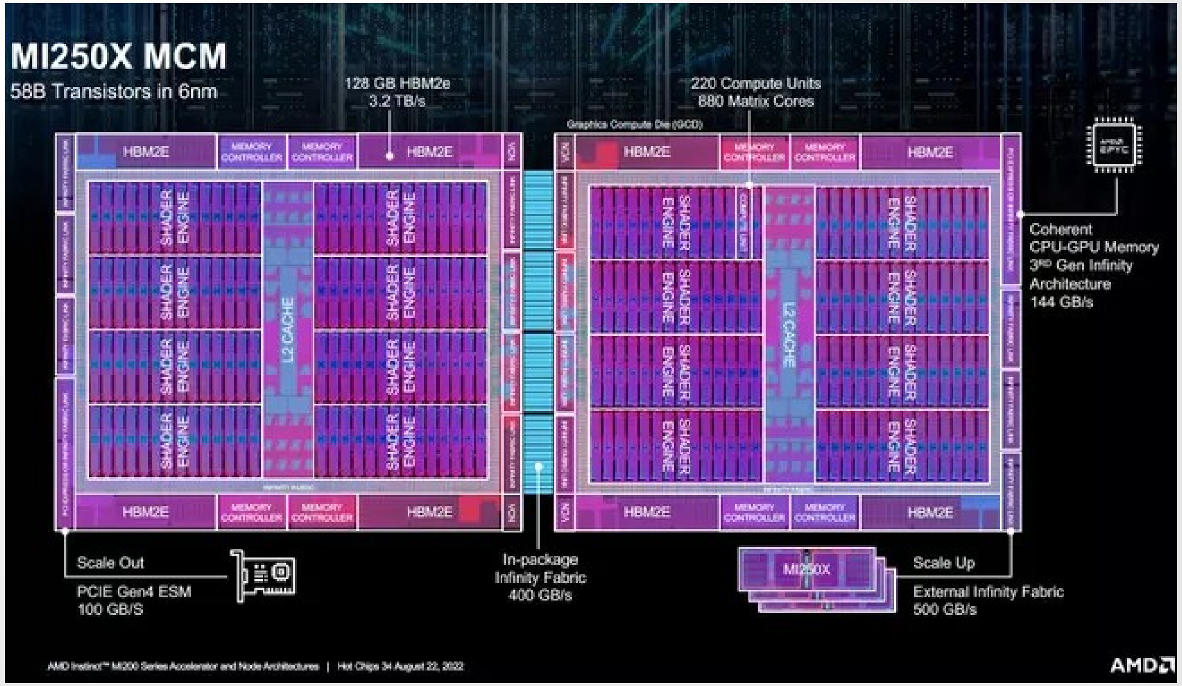
Network performance is another strong issue to unlimited parallelism

- Network technologies have also evolved less rapidly than the computational power
- Nodes have to communicate and synchronize through the network
- More cores and more nodes lead to more network usage and pressure

- Same problem as the road interconnection network

Trying to overcome the limitations with GPUs

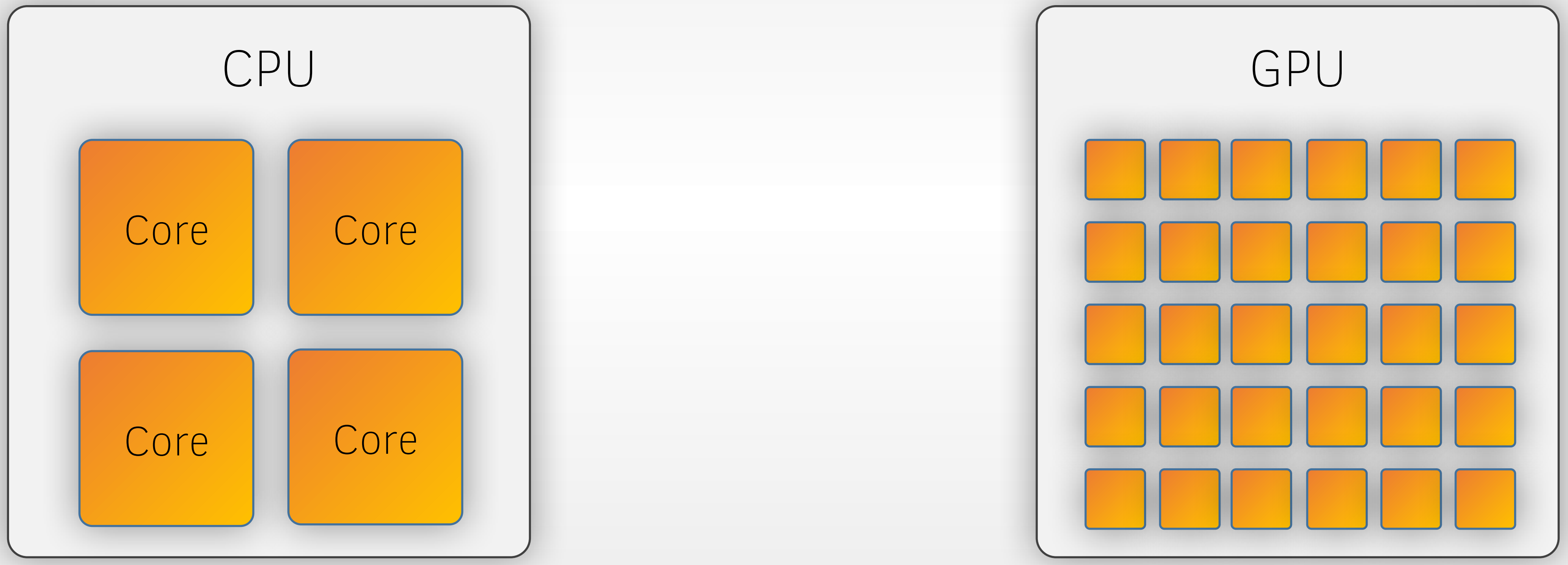
- Hardware dedicated to computing to make it more energy efficient.
- More compact computing units for less network communication.
- High memory bandwidth.



AMD Instinct™ MI200 Series Accelerator and Node Architectures | Hot Chips 34 August 22, 2022

GPU basic description

Both share the same DNA but:



CPU have few complex cores

GPU have a lot of simple cores

Today’s landscape: The TOP500 list

TOP 500
The List.

- A global ranking of the 500 most powerful super-computers
- Databases available for statistics
- Updated in November (SC event) and June (ISC event) every year
- https://www.top500.org/

How the performance is measured

- A performance metric is the number of operations per second called “flops”.
- TOP500 ranking is based on the LINPACK benchmark that consists on solving a dense system of linear equations.
- The LINPACK measured performance is always less than the theoretical peak performance.
- The LINPACK measured performance is always much higher than the performance measured with “real” applications.

Most-powerful HPC systems today

Summit (rank 5)

- USA
- IBM CPUs + NVIDIA V100 GPUs
- 149 Pflops
- 10 MW

Fugaku SC (rank 2)

- Japan
- ARM CPUs
- 442 Pflops
- 29,8 MW

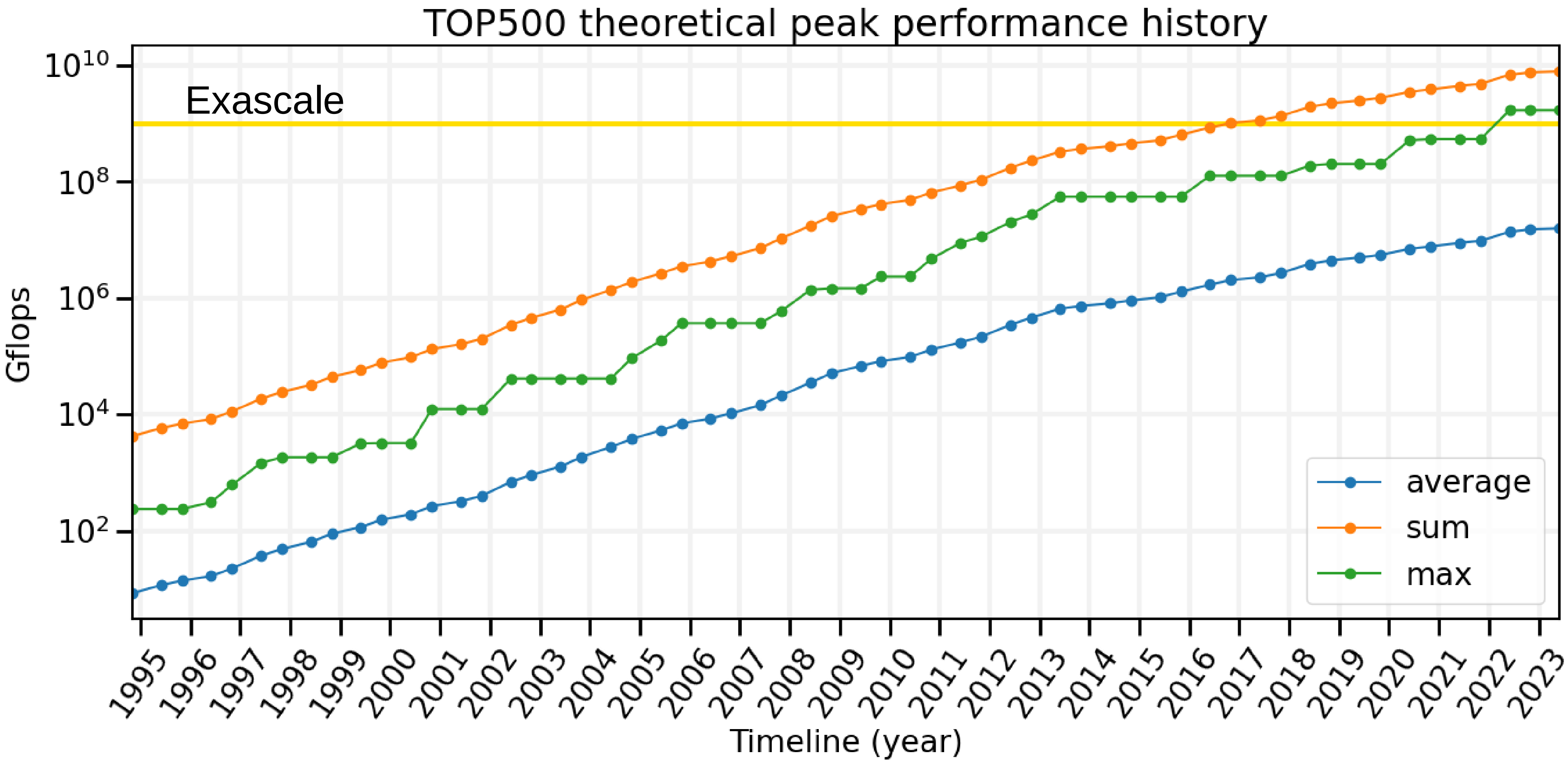
Frontier (rank 1)

- USA
- AMD CPU+GPU
- 1194 Pflops
- 22,7 MW

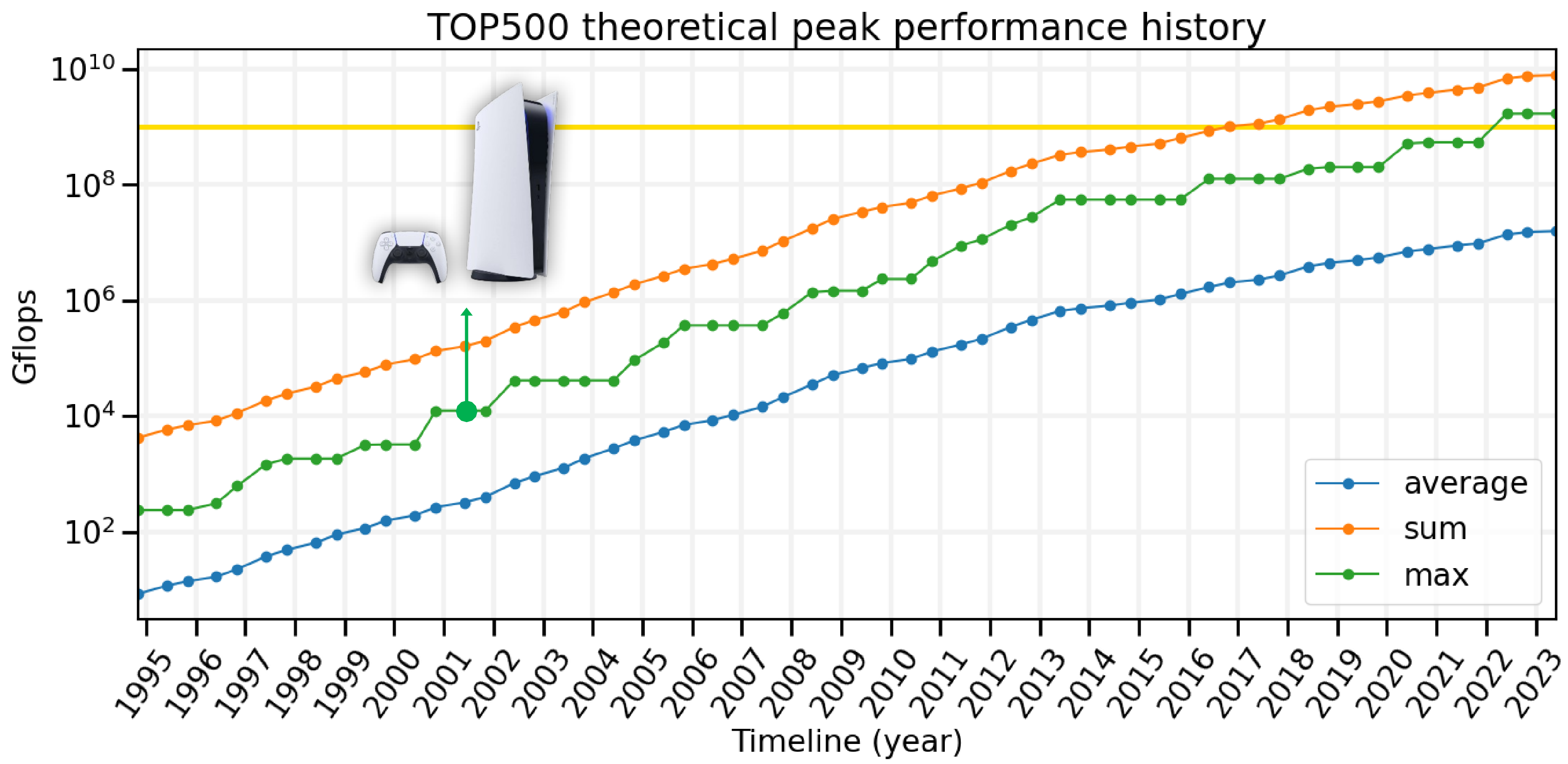
Adastra (rank 12)

- France
- AMD CPUs + GPU
- 46 Pflops
- 0,9 MW

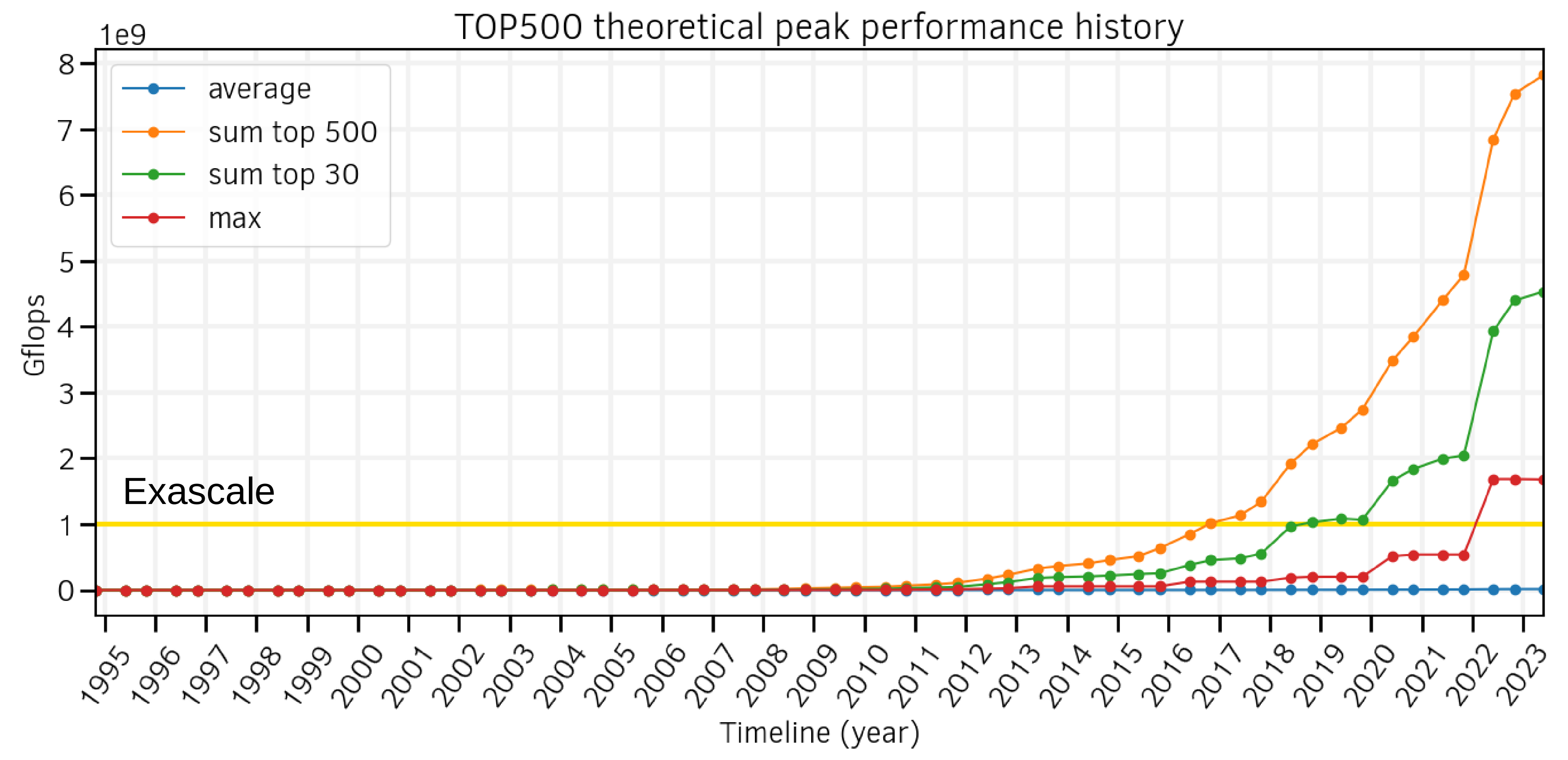
Evolution of computing power



Evolution of computing power

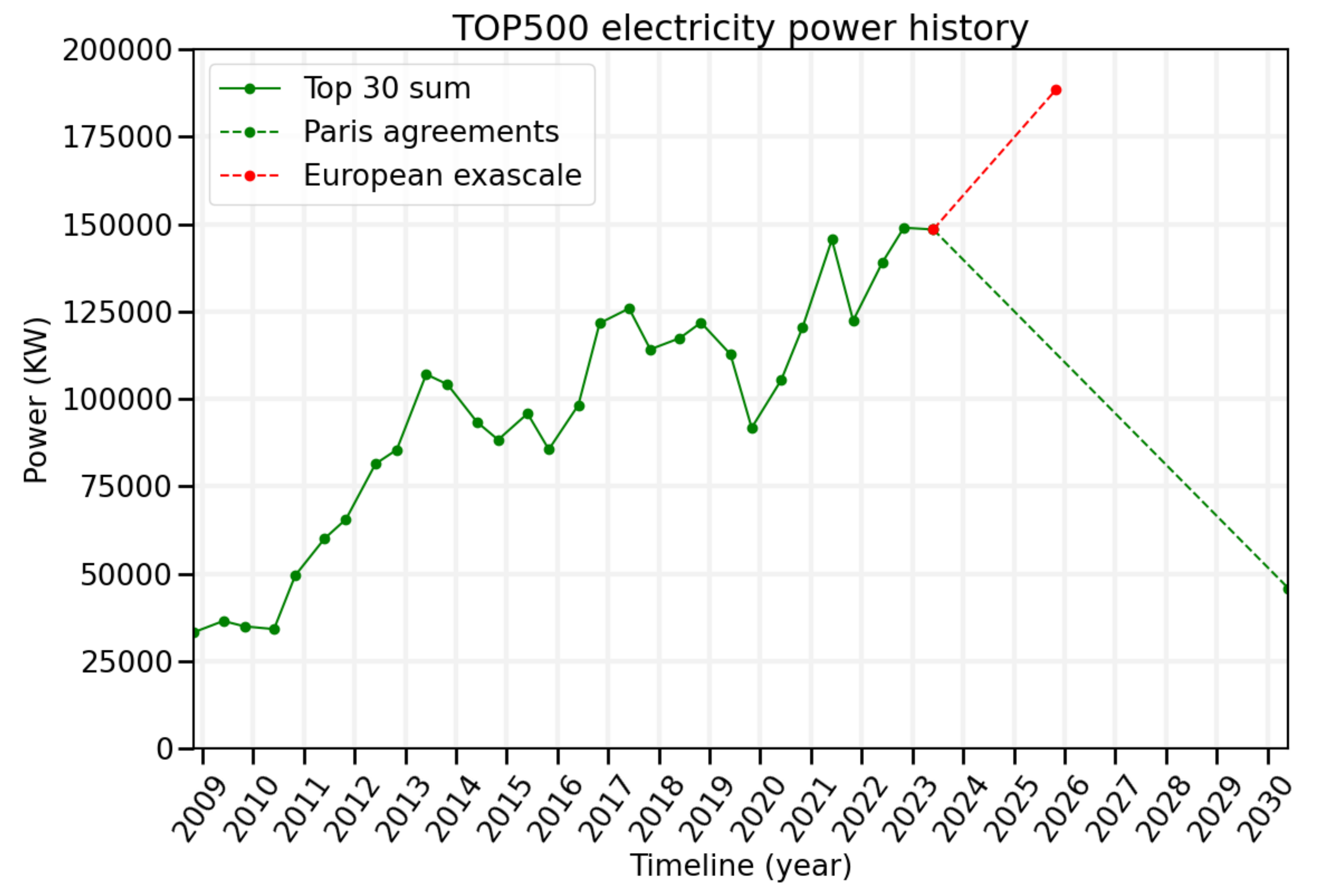


Evolution of computing power



Focus on the TOP500 electricity consumption

- Power increase is slower than performance.
- Important gain of efficiency.
- Europe has announced 2 exascale systems before 2026.
- This does not account for the increase of the number of HPC systems.



Technological focus on energy efficiency

Energy efficiency is a major driver of technological development

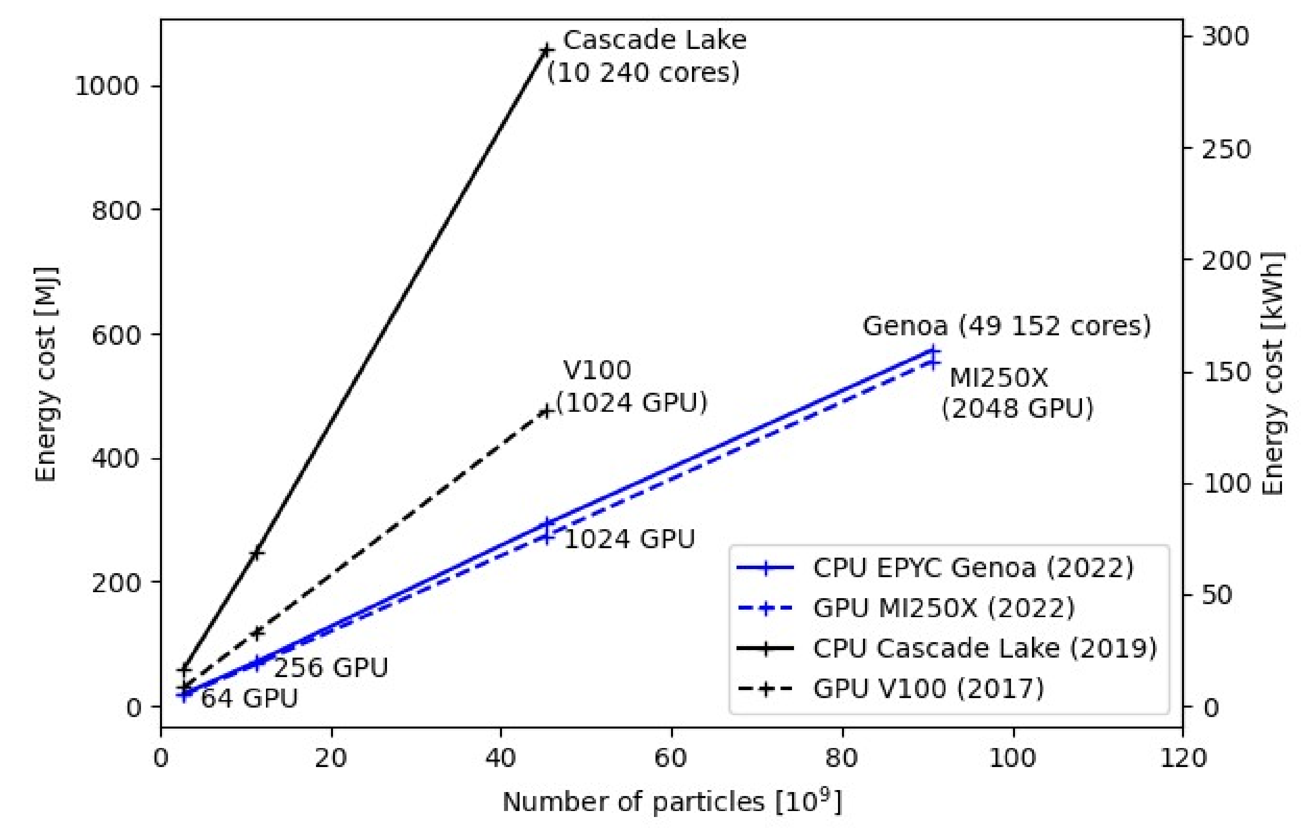
With 10 year old technologies,
an exascale system would
need a power of 1 GW

With 5 year old technologies,
an exascale system would
need a power of 120 MW

At present technologies, an
exascale system consumes 20 MW

(And yet energy consumption
keeps increasing !!)

Energy: the proper metric for software performance too



- Weak scaling: the resources scale with the problem size.
- The configuration is optimized for each system.
- Results may differ with another physical case.
- The energy cost depends linearly on the size.
- Be aware of the “Rebound effect”.

Software efficiency for LWFA

Reducing simulation impact for LWFA

- ▶ AM Geometry
- ▶ Envelope model
- ▶ PML
- ▶ Boosted frame

Thanks & Keep Smileing !

Thanks for supporting this event

eli beamlines

PLAS@PAR
Plasmas à Paris

GdR APPEL
Groupement de Recherche Accélérateurs Plasma PompÉs par Lasers

Contributing labs, institutions & funding agencies

MAISON DE LA SIMULATION

PLAS@PAR
Plasmas à Paris

P2IO
Physique des 2 Infinis et des Origines

PALM
Laboratoire d'Excellence
Physique : Atomes Lumière Matière

LPP

irap
astrophysique & planétologie

iris

intel

CELIA

GdR APPEL
Groupement de Recherche Accélérateurs Plasma PompÉs par Lasers

Physique des Infinis
Une Initiative Sorbonne Université

AIDAS
AI Data Analytics and Scalable Simulations

cnrs

cea

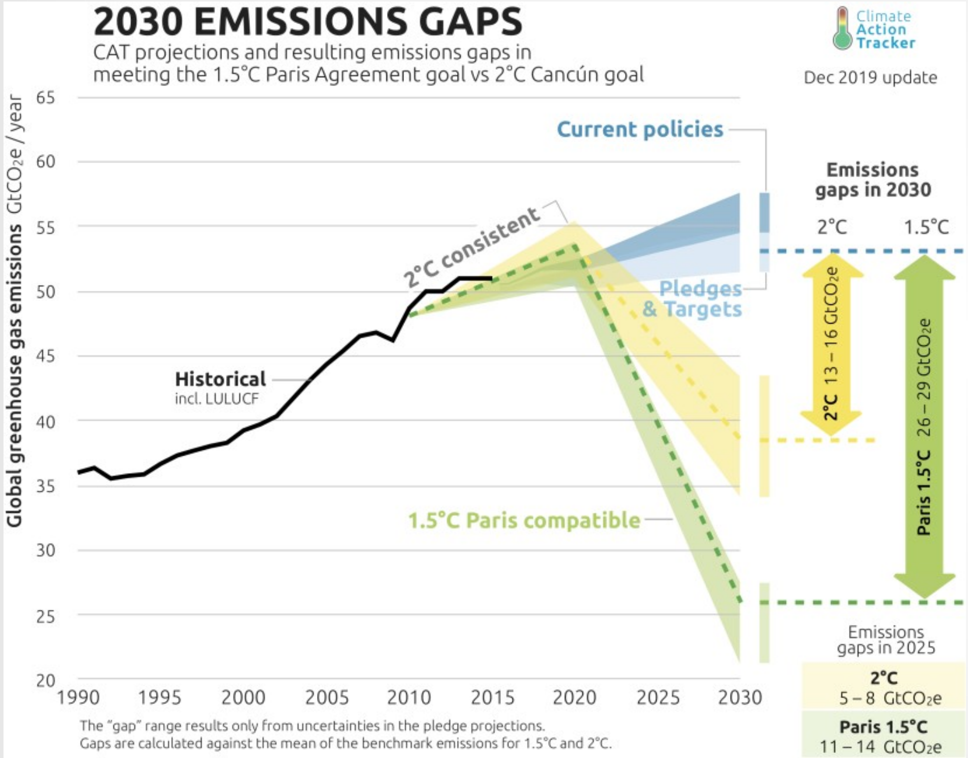
SORBONNE
UNIVERSITÉ

université
PARIS-SACLAY

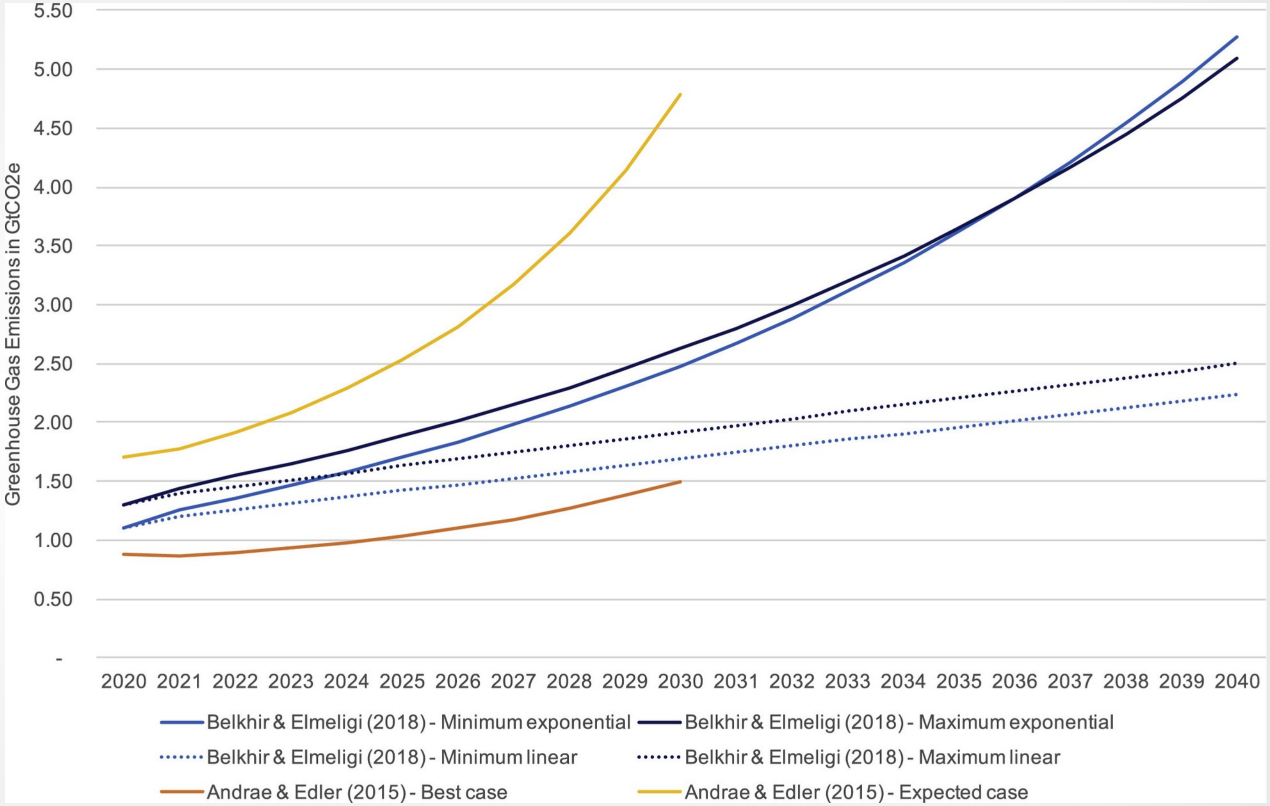
l'X
ÉCOLE
POLYTECHNIQUE

INSTITUT
POLYTECHNIQUE
DE PARIS

Environmental limits



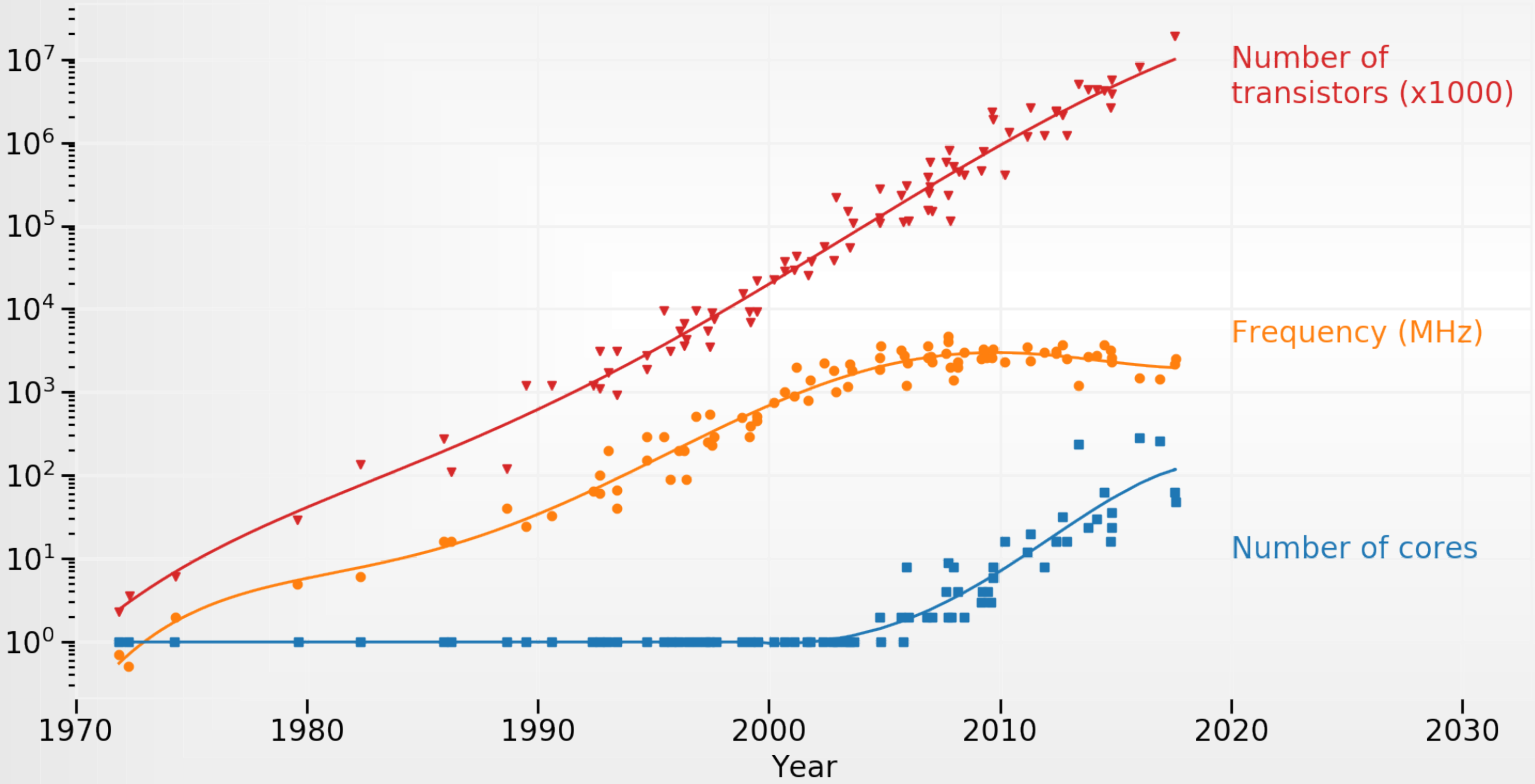
Paris agreement in 2015 (196 countries)



Freitag et al, 2021, «The climate impact of ICT: A review of estimates, trends and regulations» (open access).

ICT is the infrastructure and components that enable modern computing.

CPU micro-architecture trend



Computational power increase:

- Decrease of the manufacturing process
- Increase of the number of transistors per socket
- Increase of the number of cores per socket
- Similar frequency / larger vector size
- Share parallelism more and more important