Smilei)

Scalability of free softwares

Café Calcul
Avril 2024

Arnaud Beck
Laboratoire Leprince-Ringuet

eli | beamlines

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR
ET DE LA RECHERCHE
Liberté
Égalité
Fraternité

Ouvrir
la science !

Dedicated to Open-Science !)

PRIX
SCIENCE OUVERTE
DU LOGICIEL
LIBRE
DE LA RECHERCHE
2023

CATÉGORIE
SCIENTIFIQUE ET
TECHNIQUE

Smilei

ouvrirlascience.fr

Prix « Science Ouverte » du logiciel libre de la recherche

MINISTÈRE
DE L'ENSEIGNEMENT
SUPÉRIEUR,
DE LA RECHERCHE
ET DE L'INNOVATION
Liberté
Égalité
Fraternité

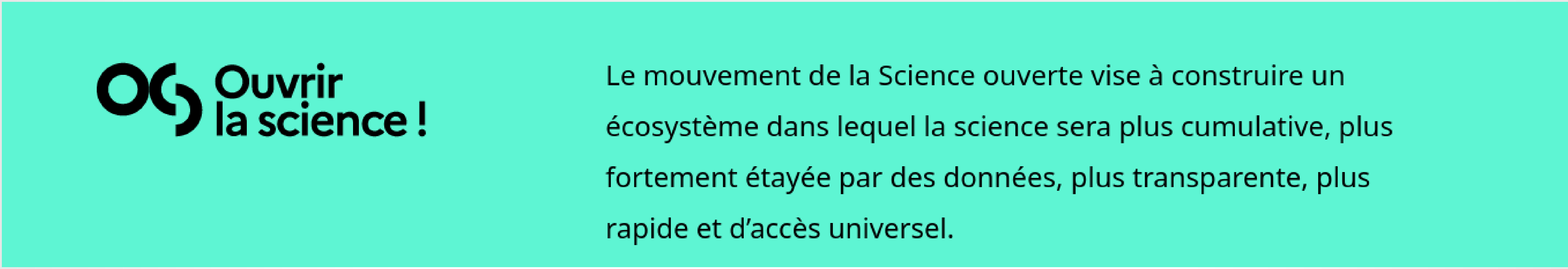
4 catégories
Lauréat et Espoir pour chacune

- Documentation
- Science et Technique
- Communauté
- Coup de cœur du jury

L'objectif avoué est de faire une vitrine pour les projets publiques libres.
Il y a un prix pour les données également.

F.A.Q :

- Sur candidature dossier
- Pas de récompense pécuniaire
- Concerne tous les domaines de recherche
- ~ 70 candidats en 2023



Ouvrir la science !

Le mouvement de la Science ouverte vise à construire un écosystème dans lequel la science sera plus cumulative, plus fortement étayée par des données, plus transparente, plus rapide et d’accès universel.

Workshop : “Software, Pillar of Open Science”

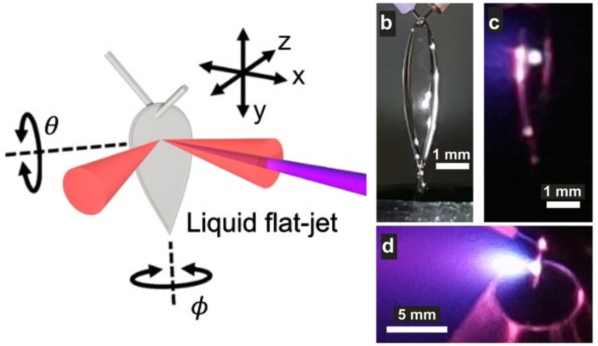
- **Recognition** of research software contributions and **visibility** of research software
- Software contribution to research **reproducibility**
- Social impact and **sustainability** of **publicly funded** research software

Overview

The Particle-In-Cell (PIC) simulation of « extreme » plasmas

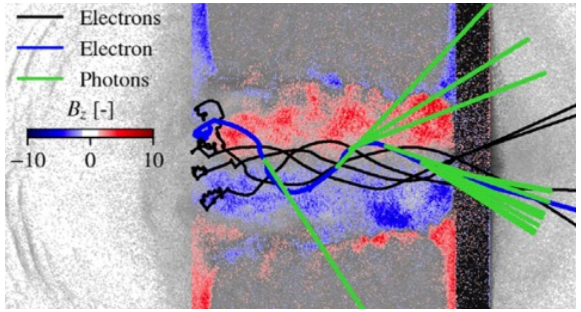
Institute for Basic Science, Gwangju

High-Harmonic Generation



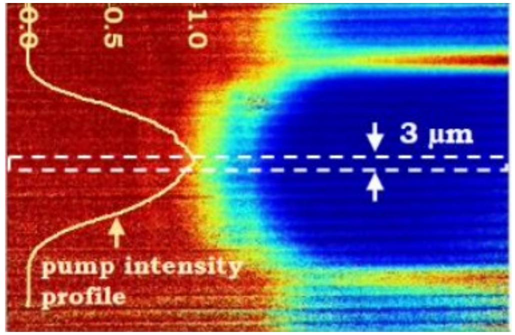
Politecnico, Milano

Radiation sources from DLT

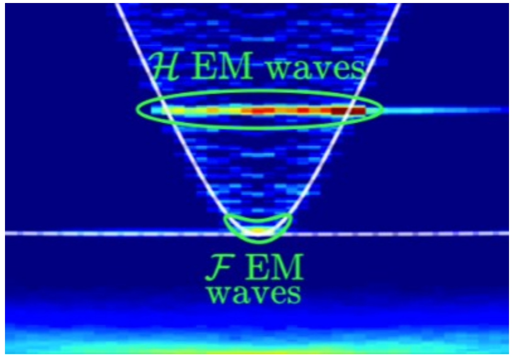


Helmoltz Institute, Jena

Solid to plasma transition

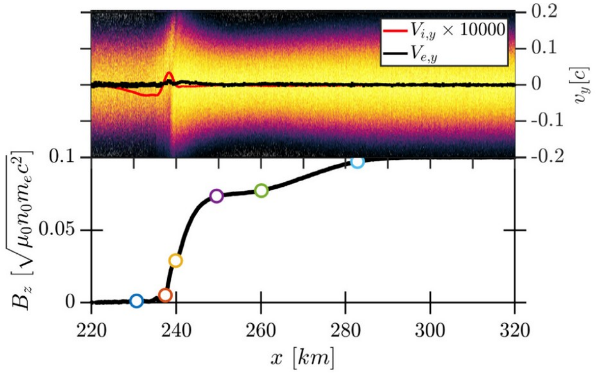


Solar radio-burst



LPP, Ecole polytechnique

Comet environment

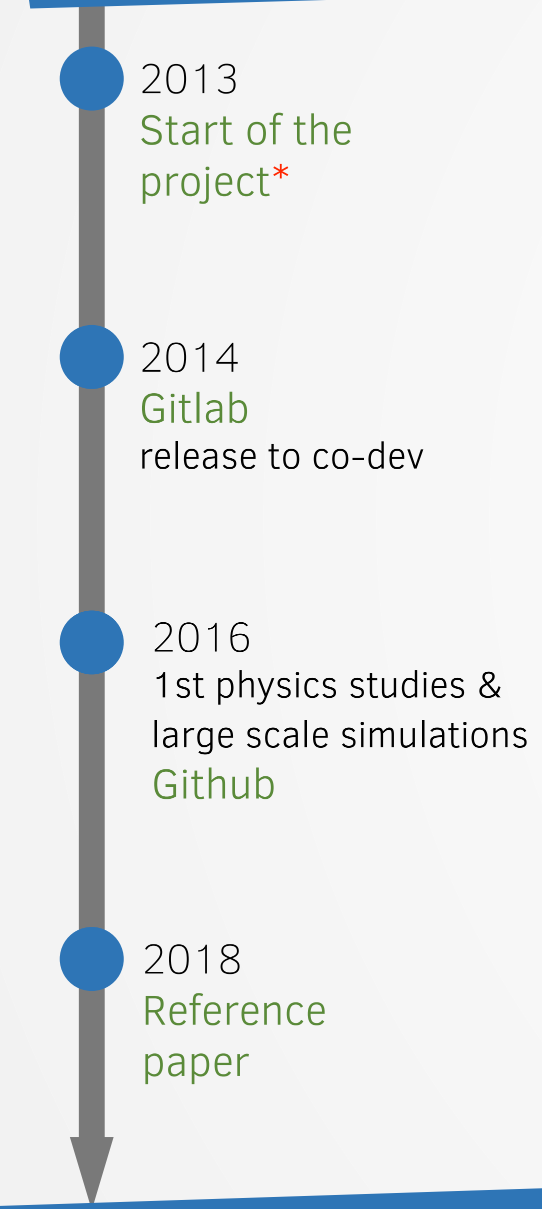


Imperial College London

Collisionless shocks & Dark Matter

University of California

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

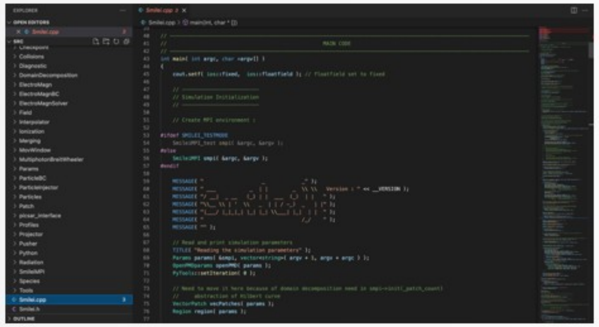
High-performance

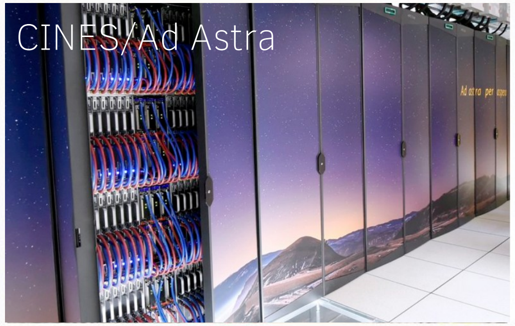
C++/Python • MPI/OpenMP/OpenACC/CUDA/HIP • SIMD • HDF5
designed for the latest architectures

What you get with Smilei

A high-performance PIC code

running on various supercomputers worldwide

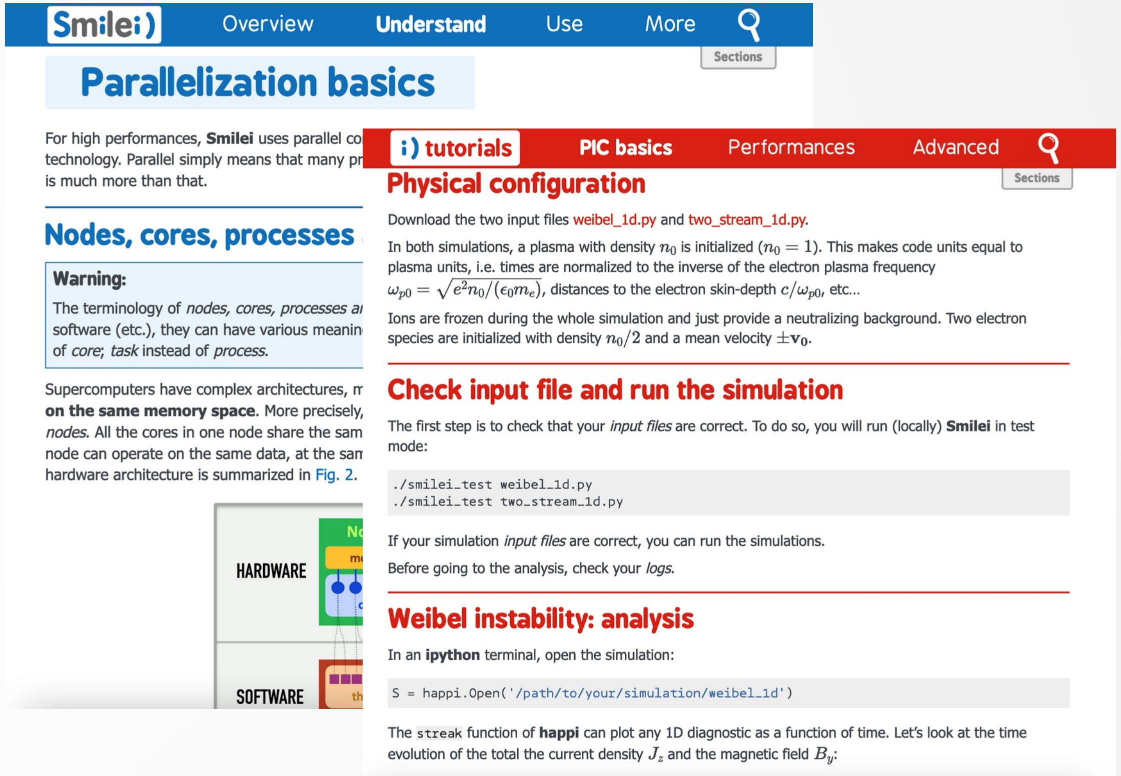




with dedicated **post-processing tools** (Happi) and an ensemble of **benchmarks** (Easi, for continuous integration)

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* a
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')
```

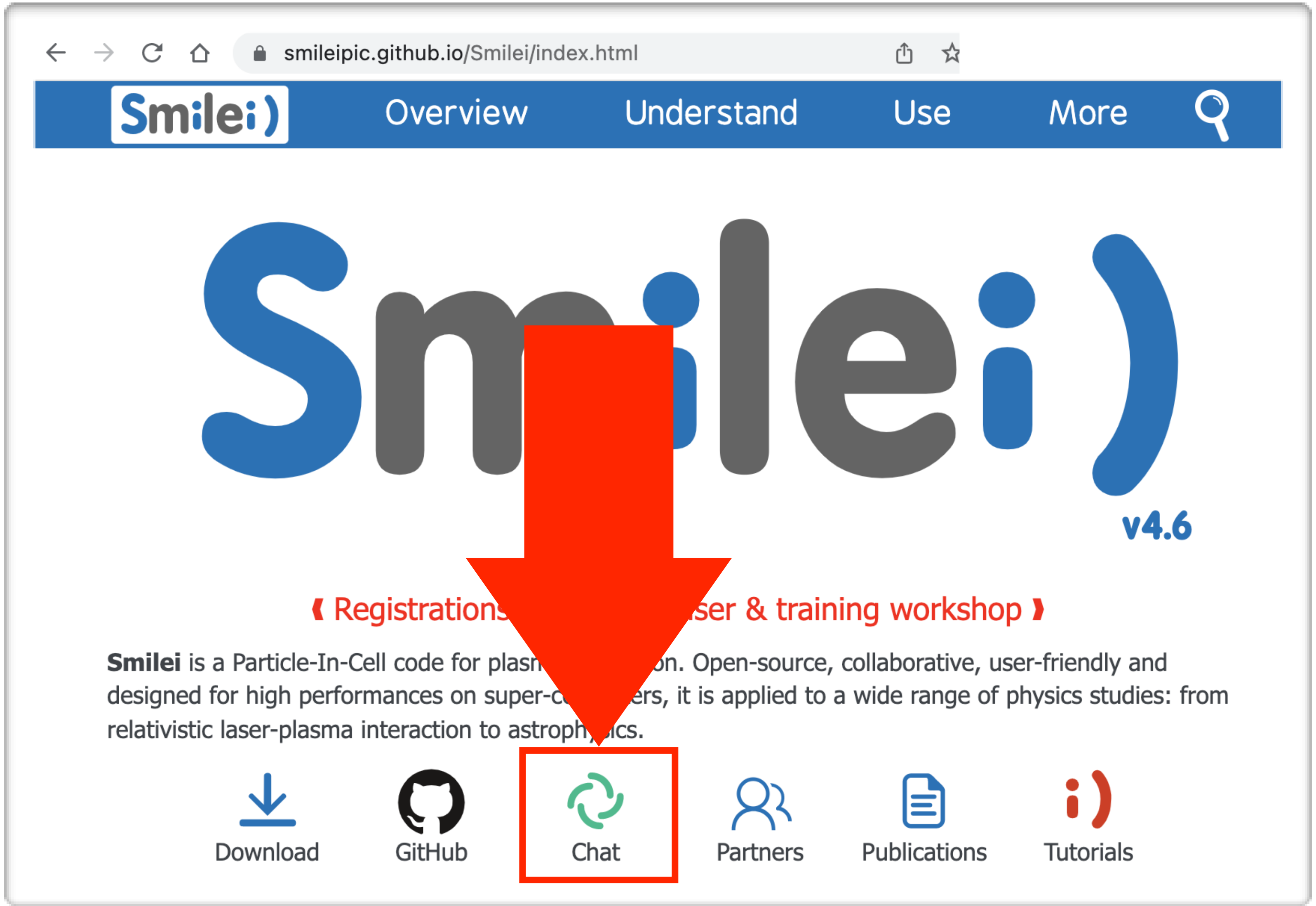
The `streak` function of **happi** can plot any 1D diagnostic as a function of time. Let's look at the time evolution of the total the current density J_z and the magnetic field B_y:

and a collaborative community

Github
sources, issues, discussions

Element
chat

A global Smilei chatroom

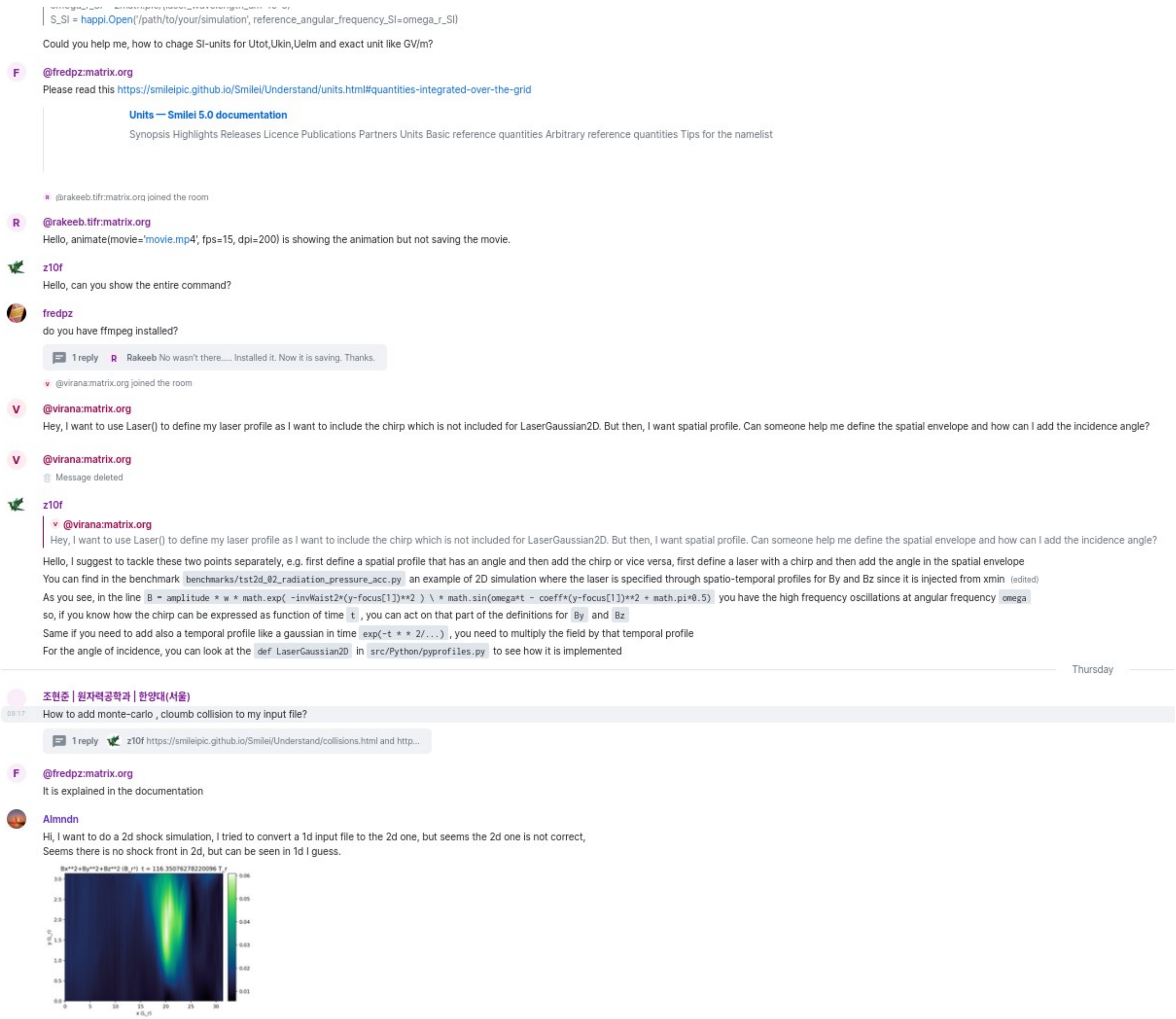


smileipic.github.io/Smilei/index.html

❰ Registrations ser & training workshop ❱

Smilei is a Particle-In-Cell code for plasm on. Open-source, collaborative, user-friendly and designed for high performances on super-c ers, it is applied to a wide range of physics studies: from relativistic laser-plasma interaction to astroph ics.

A global Smilei chatroom



S_SI = happi.Open('/path/to/your/simulation', reference_angular_frequency_SI=omega_r_SI)

Could you help me, how to chage SI-units for Utot,Ukin,Uelm and exact unit like GV/m?

@fredpz:matrix.org
Please read this https://smileipic.github.io/Smilei/Understand/units.html#quantities-integrated-over-the-grid

Units — Smilei 5.0 documentation
Synopsis Highlights Releases Licence Publications Partners Units Basic reference quantities Arbitrary reference quantities Tips for the namelist

@rakeeb.tifr:matrix.org joined the room

@rakeeb.tifr:matrix.org
Hello, animate(movie='movie.mp4', fps=15, dpi=200) is showing the animation but not saving the movie.

z10f
Hello, can you show the entire command?

fredpz
do you have ffmpeg installed?

1 reply R Rakeeb No wasn't there..... Installed it. Now it is saving. Thanks.

@virana:matrix.org joined the room

@virana:matrix.org
Hey, I want to use Laser() to define my laser profile as I want to include the chirp which is not included for LaserGaussian2D. But then, I want spatial profile. Can someone help me define the spatial envelope and how can I add the incidence angle?

@virana:matrix.org
Message deleted

z10f

> @virana:matrix.org
> Hey, I want to use Laser() to define my laser profile as I want to include the chirp which is not included for LaserGaussian2D. But then, I want spatial profile. Can someone help me define the spatial envelope and how can I add the incidence angle?

Hello, I suggest to tackle these two points separately, e.g. first define a spatial profile that has an angle and then add the chirp or vice versa, first define a laser with a chirp and then add the angle in the spatial envelope
You can find in the benchmark `benchmarks/tst2d_02_radiation_pressure_acc.py` an example of 2D simulation where the laser is specified through spatio-temporal profiles for By and Bz since it is injected from xmin (edited)
As you see, in the line `B = amplitude * w * math.exp( -invWaist2*(y-focus[1])**2 ) \ * math.sin(omega*t - coeff*(y-focus[1])**2 + math.pi*0.5)` you have the high frequency oscillations at angular frequency `omega`
so, if you know how the chirp can be expressed as function of time `t` , you can act on that part of the definitions for `By` and `Bz`
Same if you need to add also a temporal profile like a gaussian in time `exp(-t * * 2/...)` , you need to multiply the field by that temporal profile
For the angle of incidence, you can look at the `def LaserGaussian2D` in `src/Python/pyprofiles.py` to see how it is implemented

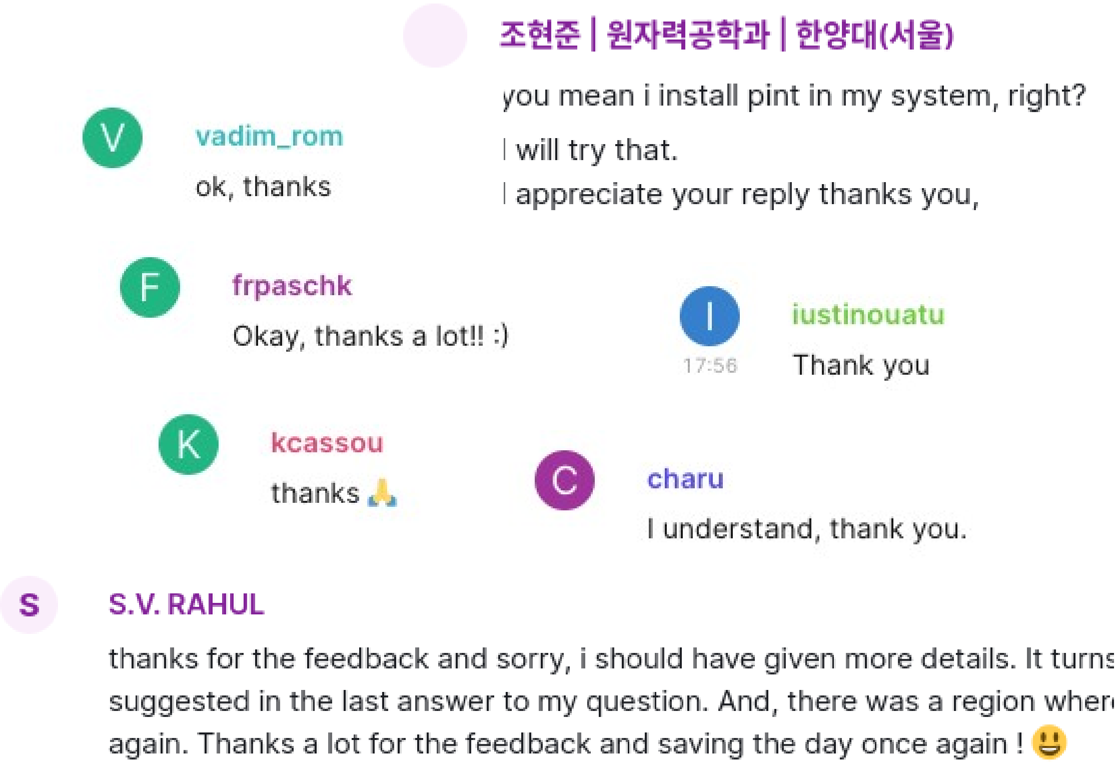
Thursday

조현준 | 원자력공학과 | 한양대(서울)
09:17 How to add monte-carlo , cloumb collision to my input file?

1 reply z10f https://smileipic.github.io/Smilei/Understand/collisions.html and http...

@fredpz:matrix.org
It is explained in the documentation

Almndn
Hi, I want to do a 2d shock simulation, I tried to convert a 1d input file to the 2d one, but seems the 2d one is not correct,
Seems there is no shock front in 2d, but can be seen in 1d I guess.



조현준 | 원자력공학과 | 한양대(서울)
you mean i install pint in my system, right?
I will try that.
I appreciate your reply thanks you,

vadim_rom
ok, thanks

frpaschk
Okay, thanks a lot!! :)

iustinouatu
17:56 Thank you

kcassou
thanks 🙏

charu
I understand, thank you.

S.V. RAHUL
thanks for the feedback and sorry, i should have given more details. It turns
suggested in the last answer to my question. And, there was a region wher
again. Thanks a lot for the feedback and saving the day once again ! 😃

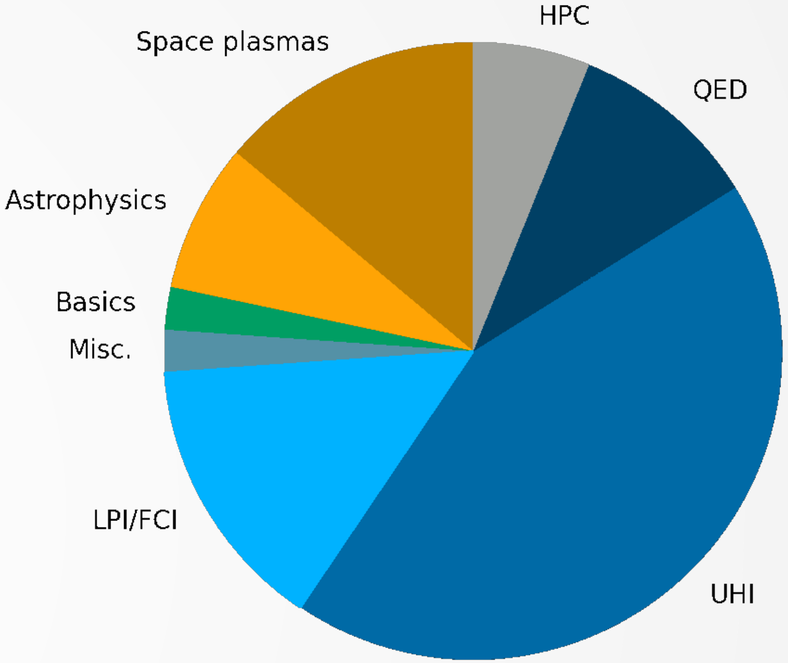
Smilei is a research & teaching platform

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
EUR Plasma – Ecole polytechnique

Smilei Workshops

Smilei)
3rd Smilei user & training workshop
9-11 March 2022

Numerical investigation of high-energy photon emission in double-layer targets with particle-in-cell codes

Marta Galbiati

POLITECNICO MILANO 1863

NanoLab

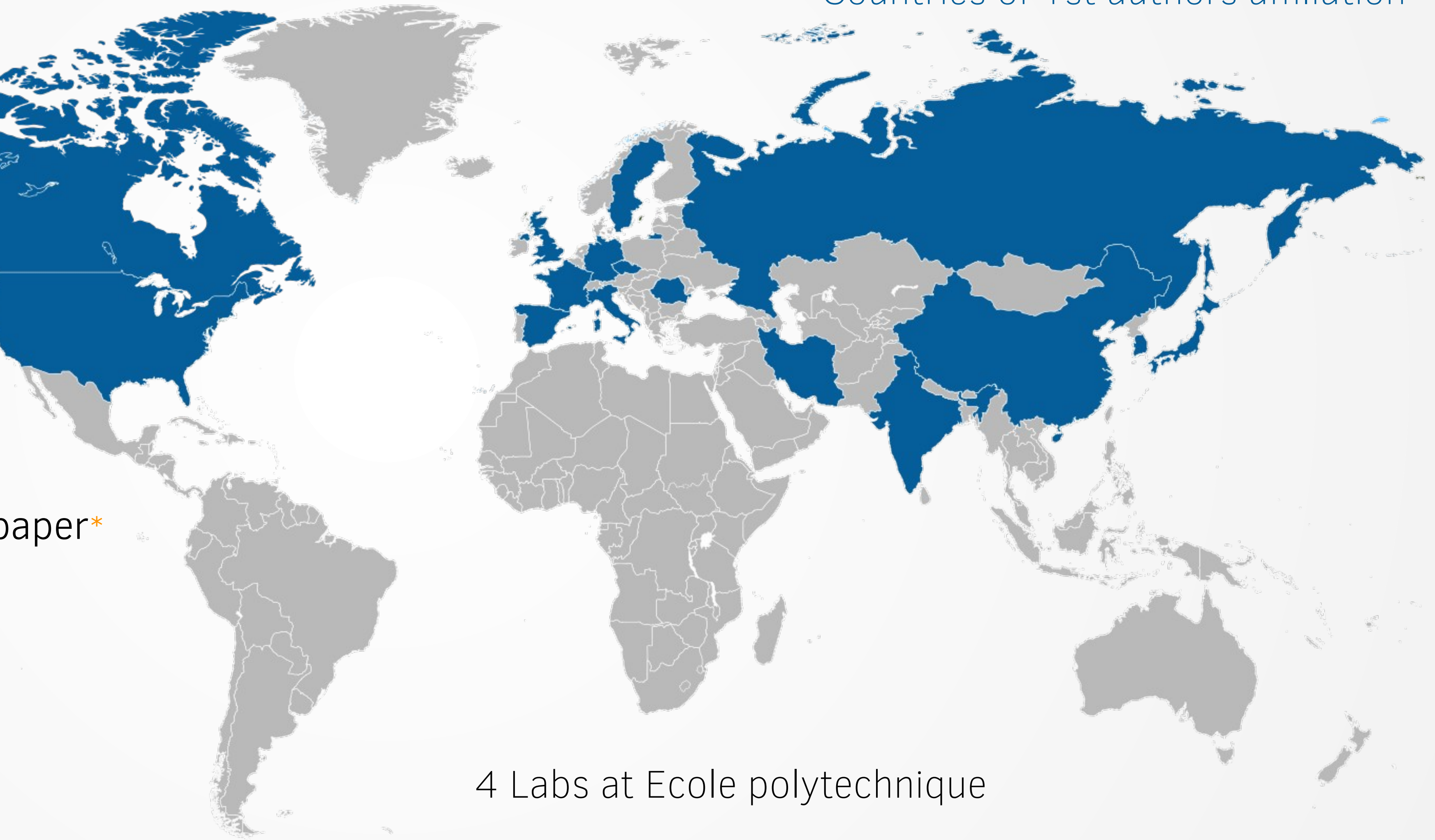
Smilei) 3rd Smilei user & training workshop 9 - 11 March 2022

GdR APPEL

Groupement de Recherche Accélérateurs Plasma PompÉs par Lasers

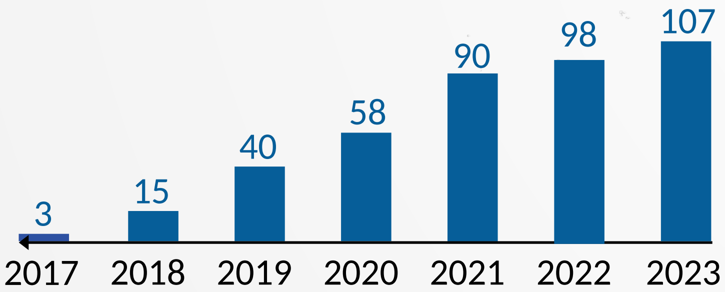
Smilei's users community is international & steadily growing

Countries of 1st authors affiliation



4 Labs at Ecole polytechnique

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

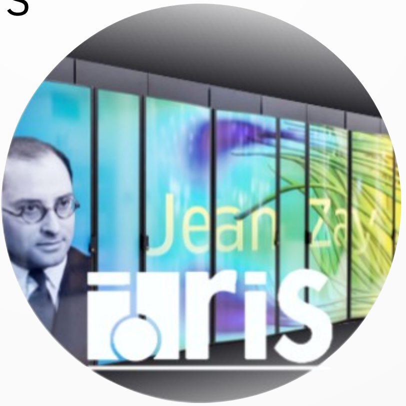
Integration in the French & European HPC landscapes

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)



Recent highlight

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

OpenACC, CUDA

OpenMP, HIP

Standard 2D and 3D simulations are supported

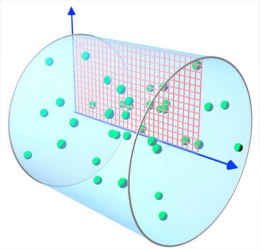
- extensive rewriting to run of both architectures & to insure performance!
- 2D and 3D cartesian geometries with various boundary conditions
- implementation is almost transparent to the user: `Main(..., gpu_computing=True)`
- porting of additional physics modules & advanced solvers is still work in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

=> Source of a new wave of users and related issues

Example of contributions

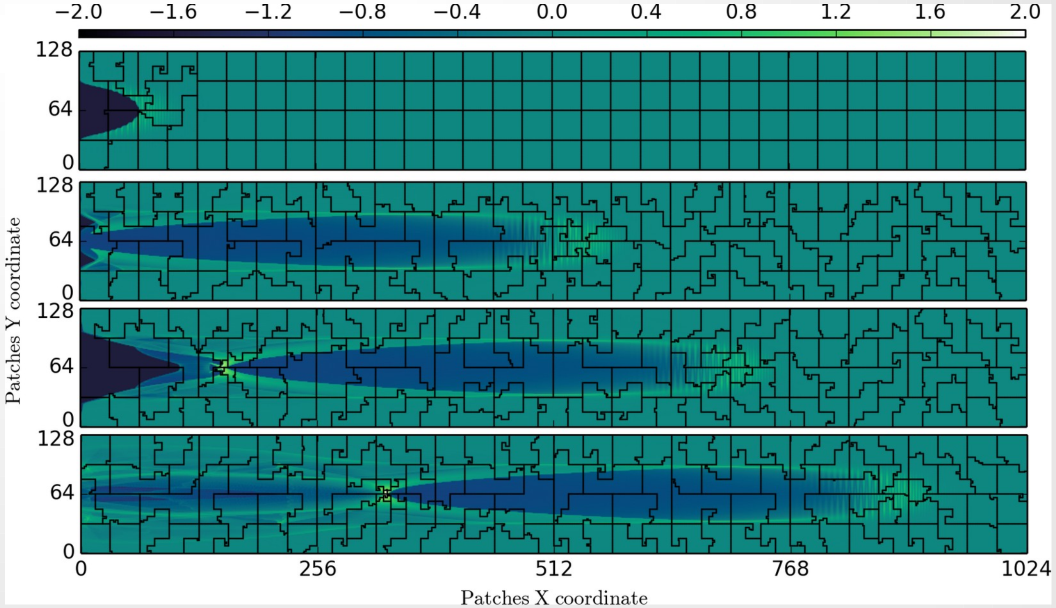
Azimuthal modes geometry (I. Zemzemi)

Zemzemi et. al., J. Phys.: Conf. Ser. 1596, 012055 (2020)



Dynamic load balancing and SIMD (A. Beck)

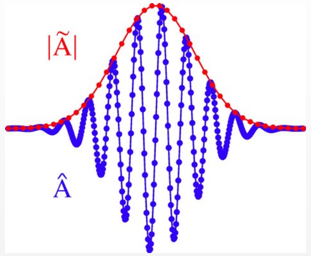
Beck et. al. Computer Physics Communications 244, (2019)



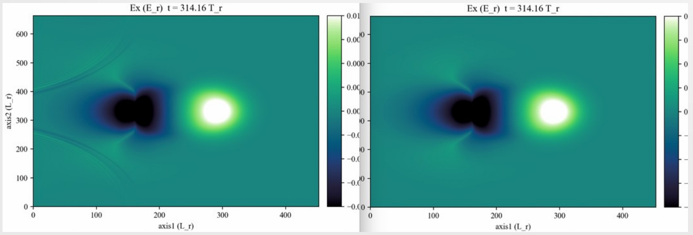
Envelope model (F. Massimo)

Massimo et. al. Plasma Phys. Control. Fusion 61, 124001 (2019)

Massimo et. al. Phys. Rev. E 102, 033204 (2020)



Perfectly Matched Layers (G. Bouchard)



The Smilei dev-team

Co-development between HPC specialists & physicists

MAISON DE LA SIMULATION

Charles Prouveur***
Mathieu Lobet*
Julien Derouillat

Haïthem Kallala, Juan Jose Silva Cuevas

Arnaud Beck*
Guillaume Bouchard (now at CEA)
Imène Zemzemi

Francesco Massimo*

Mickael Grech*
Frederic Perez*
Tommaso Vinci*

Marco Chiaramello, Anna Grassi

*permanent staff **Code architect
(CNRS DDOR, w.s.f. INP, INSU, IN2P3)

PIC (very) 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

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

Mean force Mean distribution

Vlasov equation

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

Maxwell equations

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

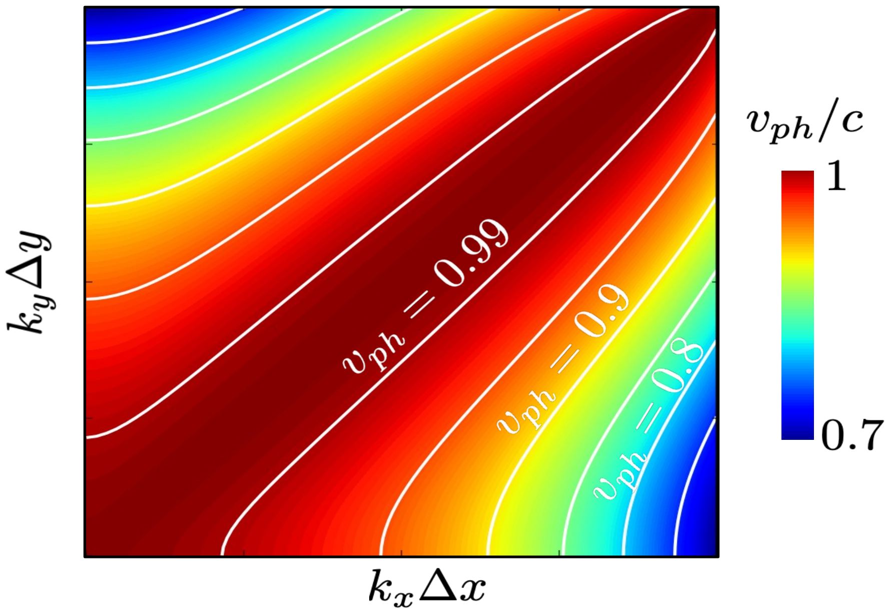
$$\nabla \cdot \mathbf{B} = 0 \qquad \partial_t \mathbf{B} = -\nabla \times \mathbf{E}$$

The 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

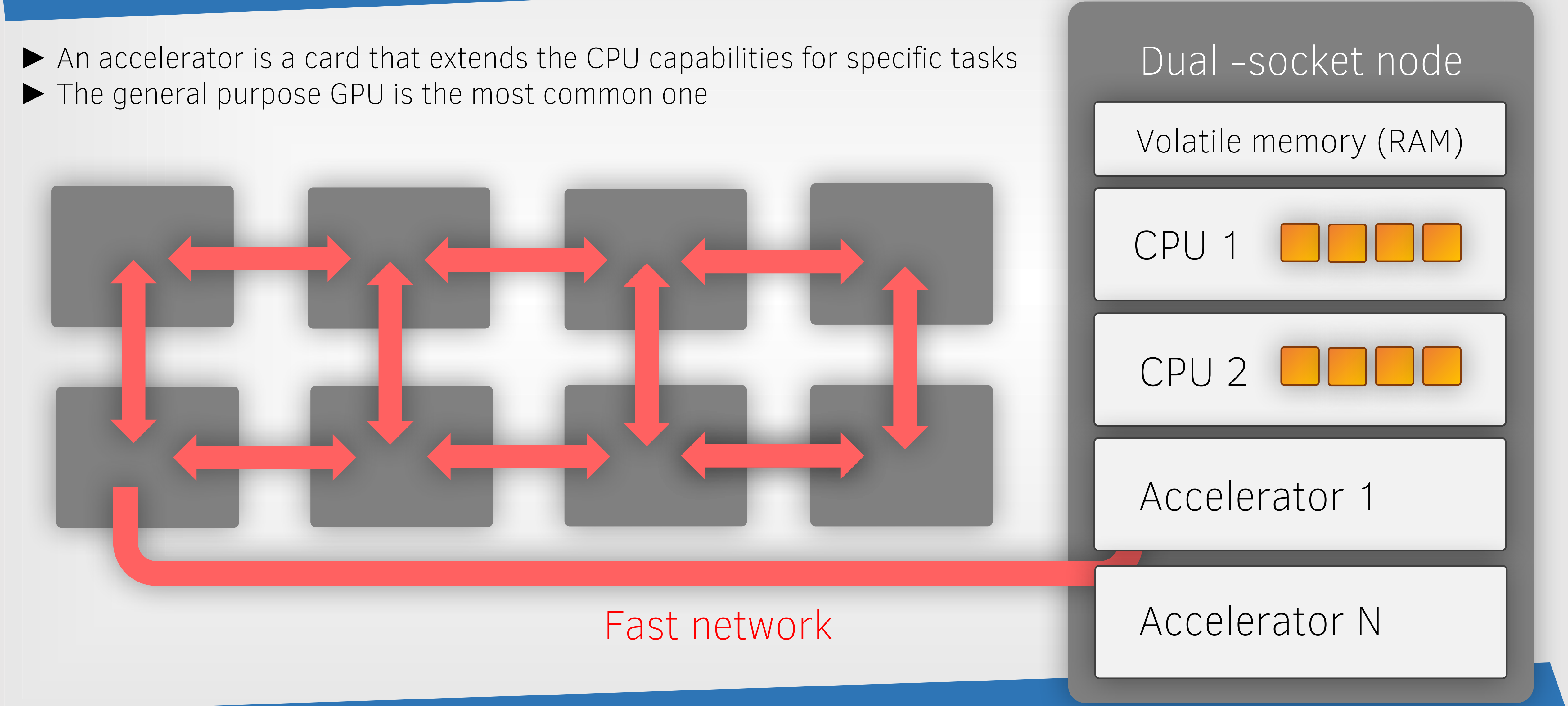
↓

Numerical Cherenkov radiation

High Performance Computing

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



Dual -socket node

Volatile memory (RAM)

CPU 1

CPU 2

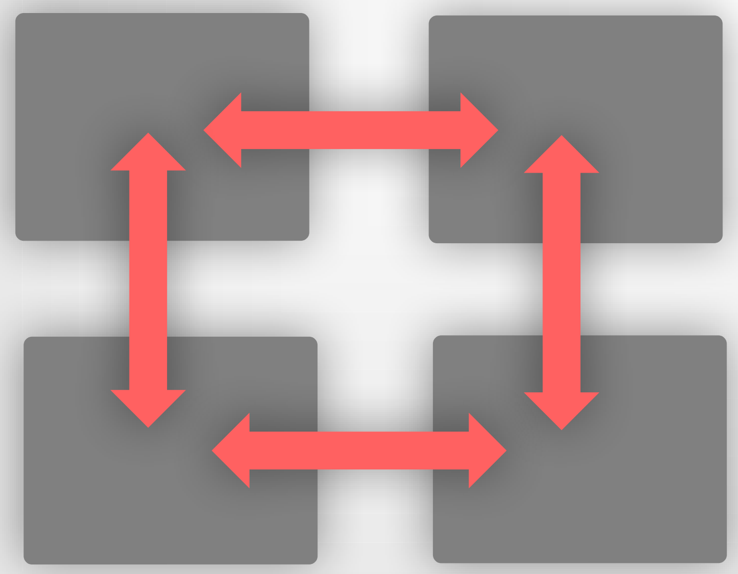
Accelerator 1

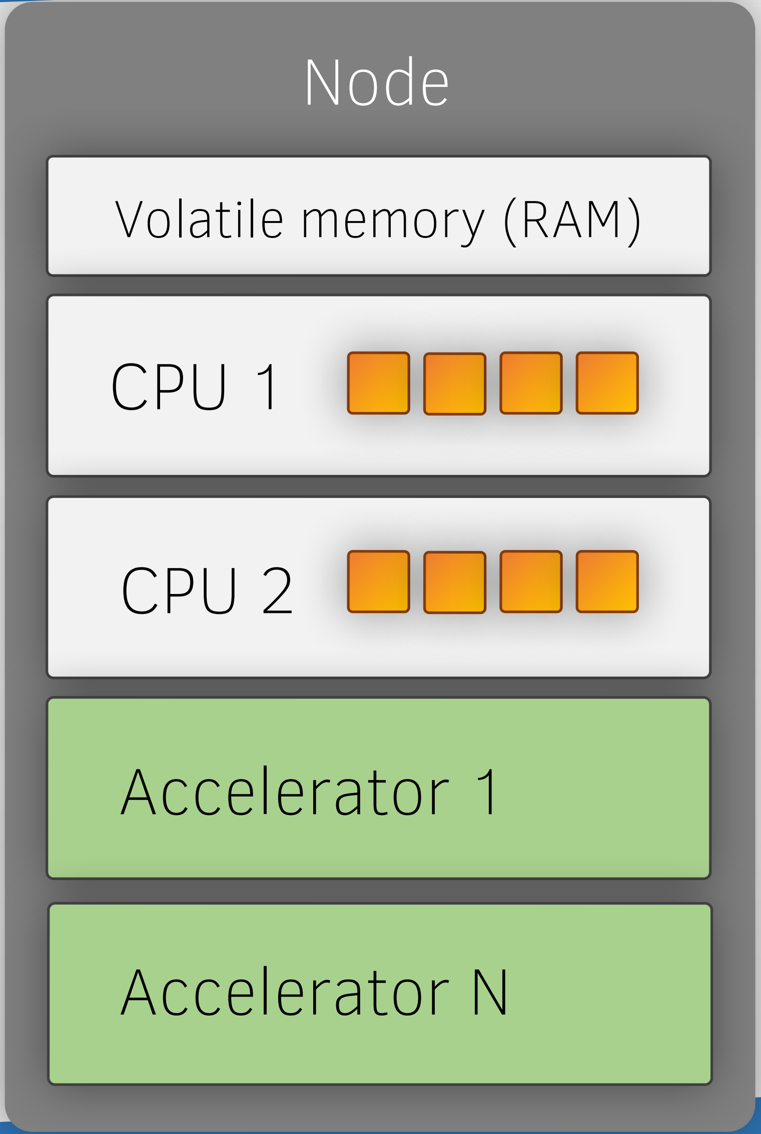
Accelerator N

Fast network

Different parallelism levels to handle

Inter-node parallelism: rely on a very efficient network





Node

Volatile memory (RAM)

CPU 1

CPU 2

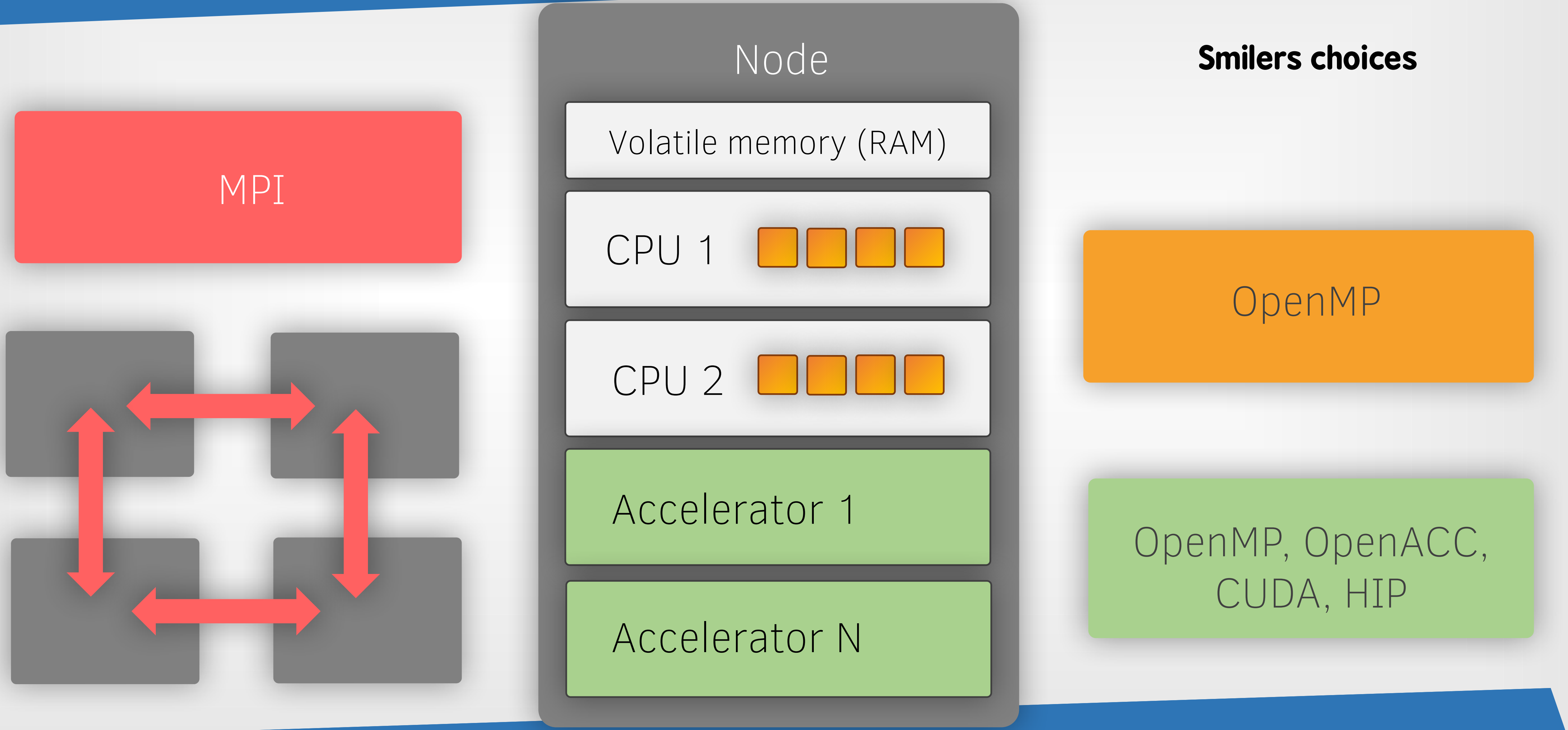
Accelerator 1

Accelerator N

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



MPI

Node

Volatile memory (RAM)

CPU 1

CPU 2

Accelerator 1

Accelerator N

Smilers choices

OpenMP

OpenMP, OpenACC, CUDA, HIP

Programming challenges for HPC applications

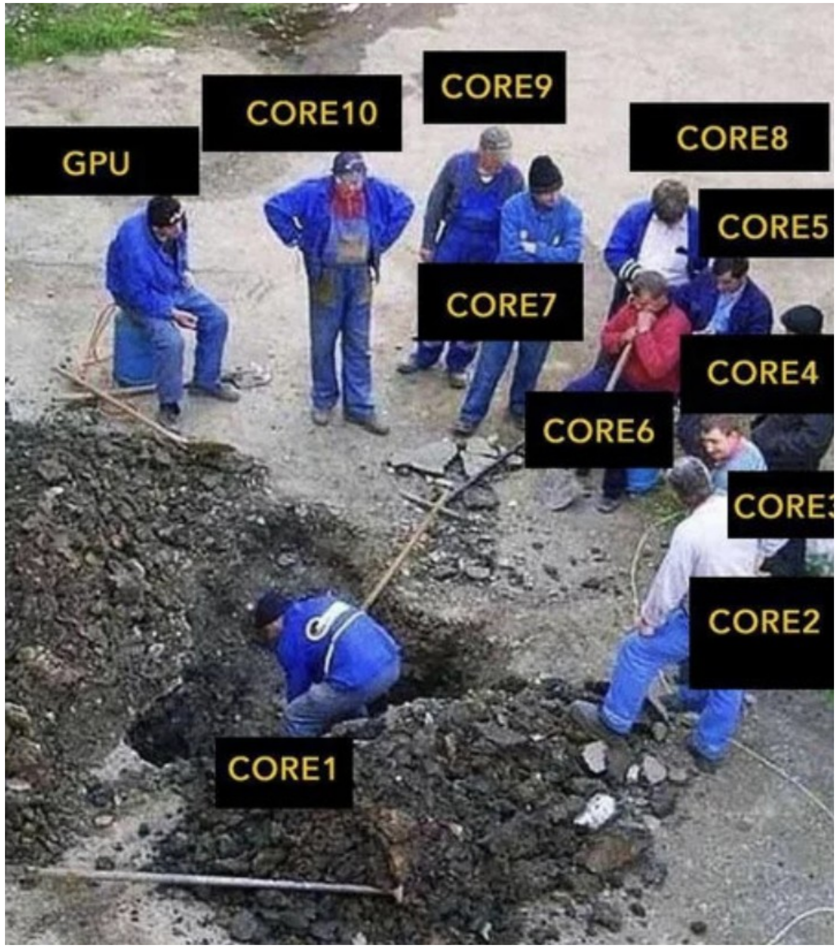
- Developing efficiently for a super-computer is more difficult than for a simple desktop computer:

  - Communications and synchronizations through the network between the nodes
  - Load balancing between the nodes
  - Work share within the node (between cores and/or accelerators)
  - Node heterogeneity
  - Memory usage (CPU/GPU)
  - 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.

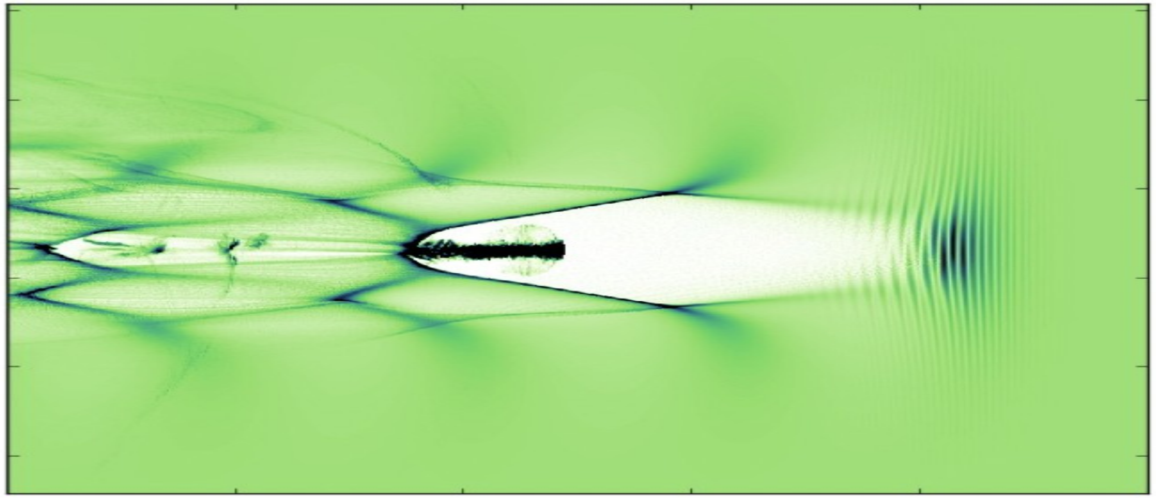
Keep load balance between computing units



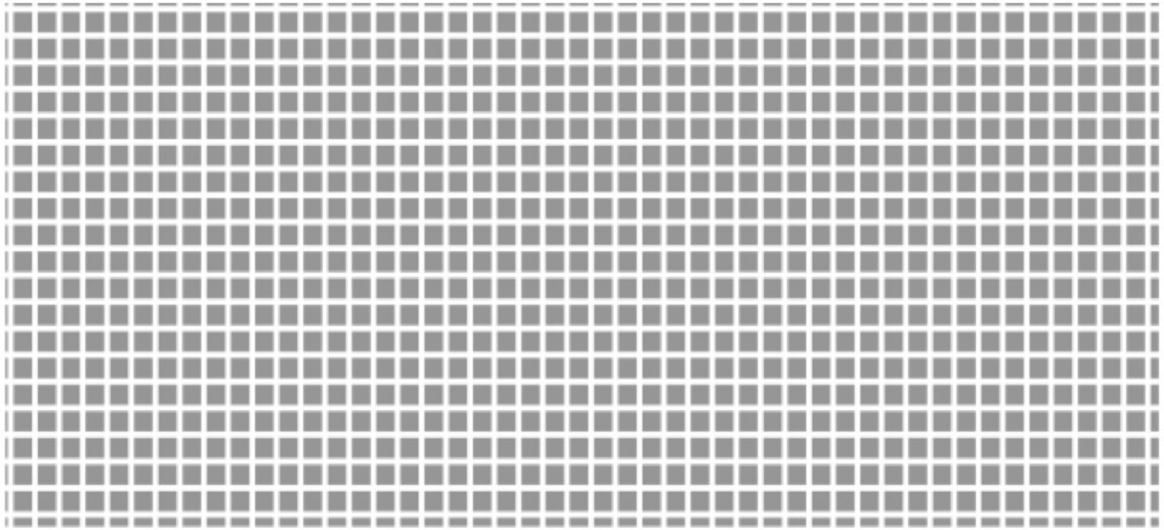
It's totally useless
to use more parallel computing units
if only one or a few
are doing all the work

Decompose your program in small
parallelizable units and
distribute them evenly
between computing units
working in parallel

A PIC code discretizes space with cells

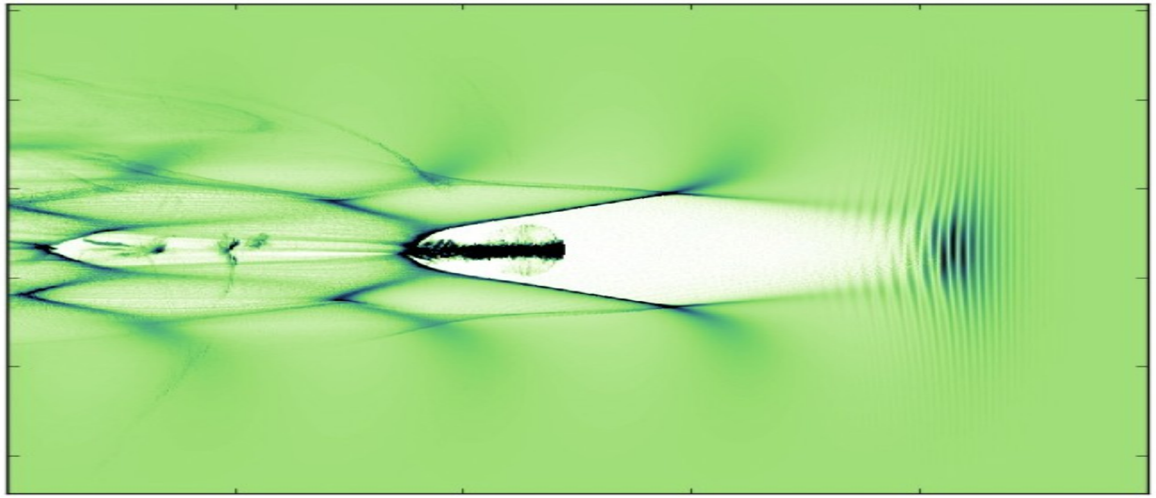


What you see:
Laser Wakefield Acceleration

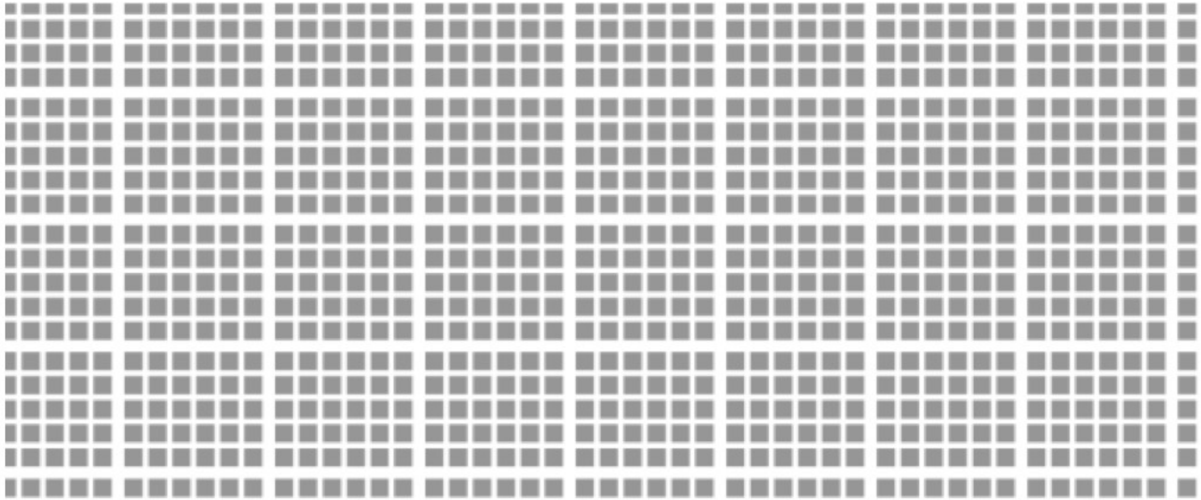


What the computer sees:
a collection of **cells** (figure not in scale)
populated by fields and macro-particles

In Smilei, cells are grouped in patches



What you see:
Laser Wakefield Acceleration

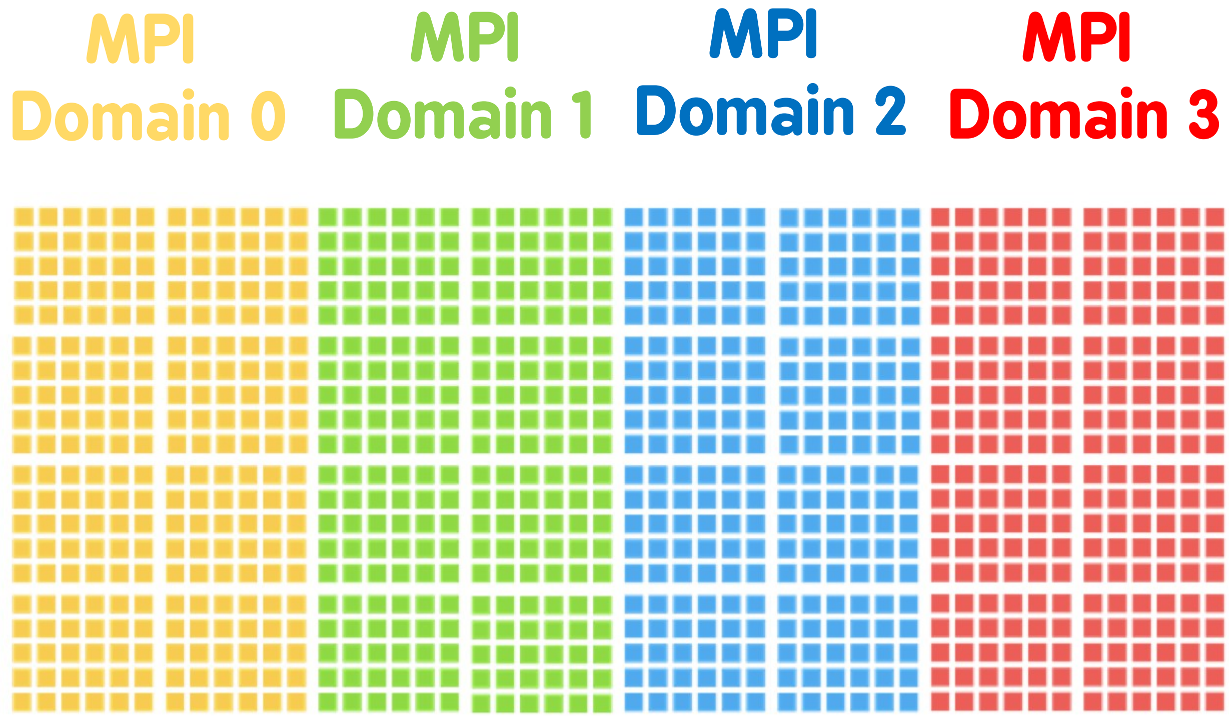


What the computer sees:
a collection **patches** made of cells
(figure not in scale)
populated by fields and macro-particles

In Smilei, patches are grouped in different memory locations = MPI domains

MPI domains are made of contiguous patches

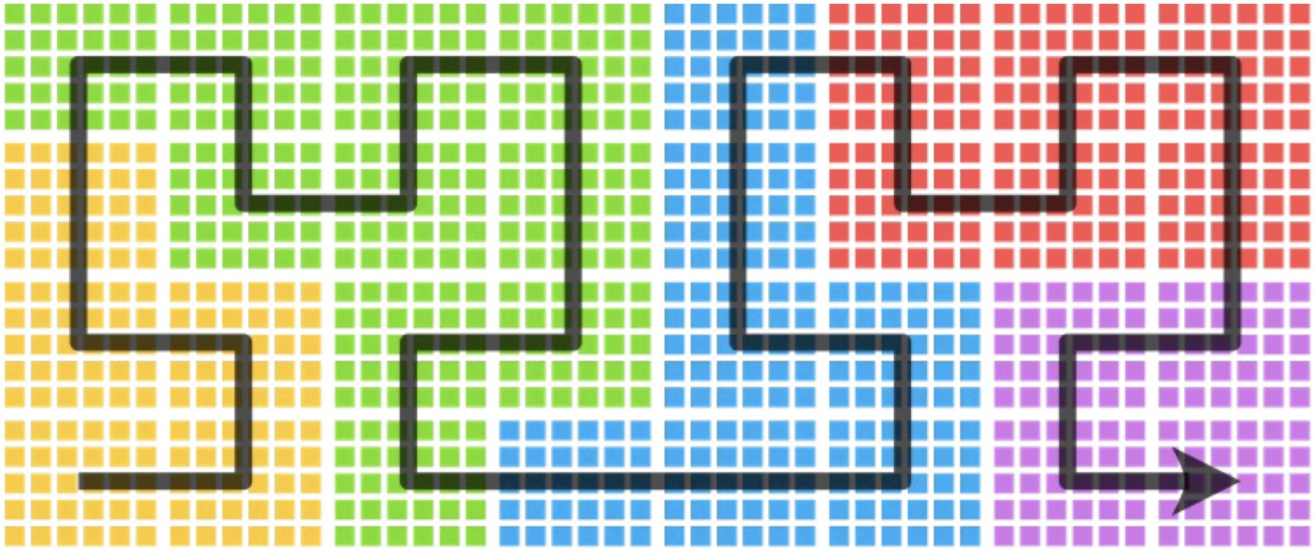
Example of Cartesian MPI decomposition (possible in Smilei but not default)



More common MPI decomposition in Smilei

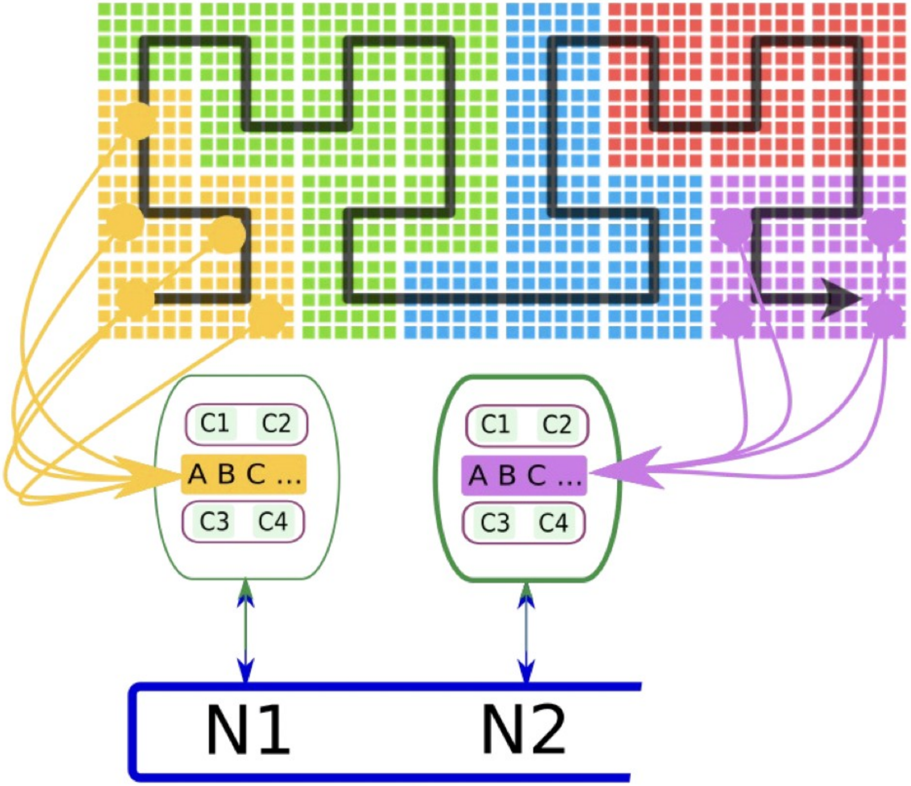
Hilbertian MPI decomposition

(good for patch exchange)



MPI Domain 0 **MPI Domain 1** **MPI Domain 2** **MPI Domain 3** **MPI Domain 4**

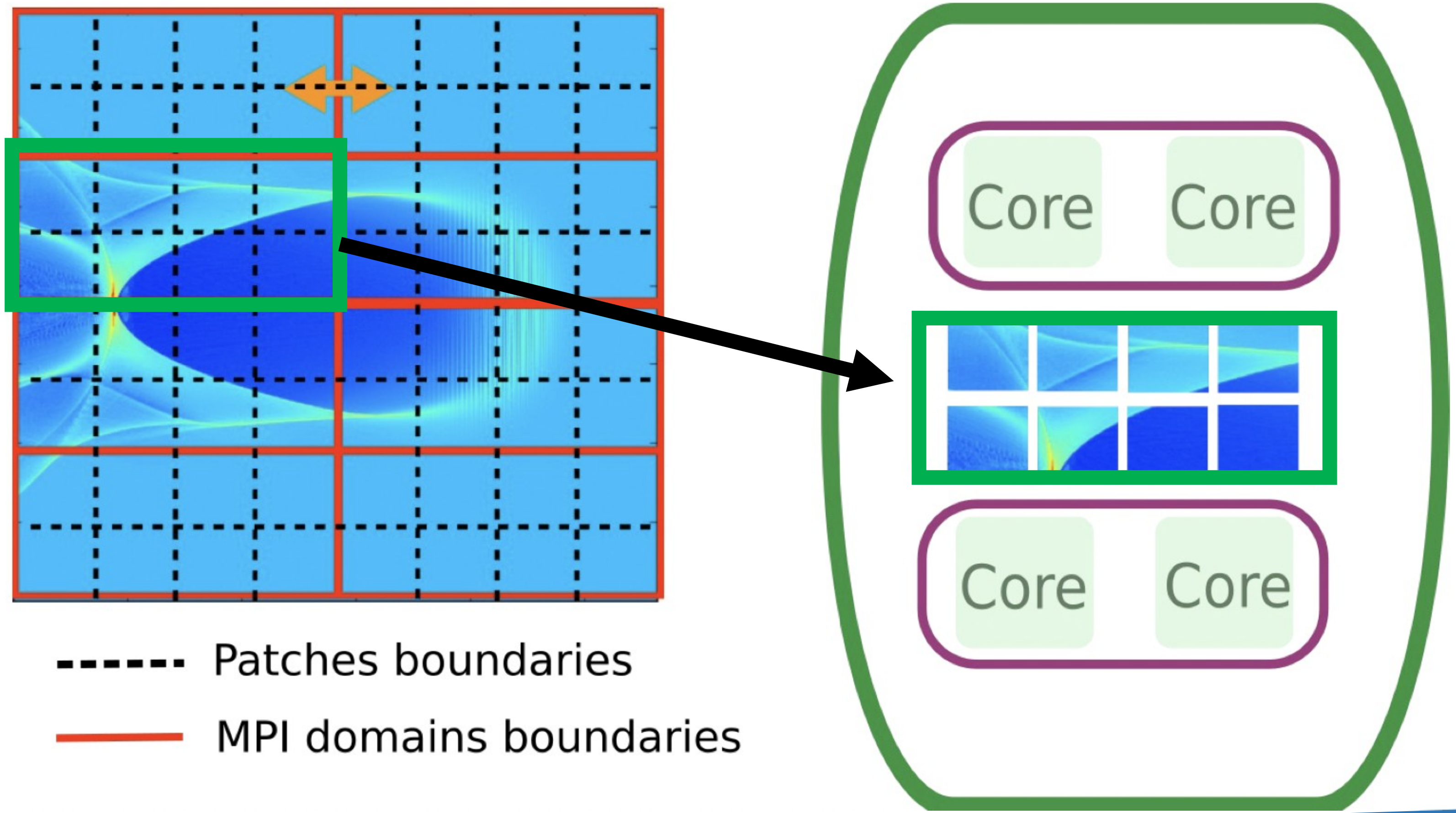
MPI domains are assigned to computing nodes



MPI Domain 1 MPI Domain 2 MPI Domain 3 MPI Domain 4 MPI Domain 5

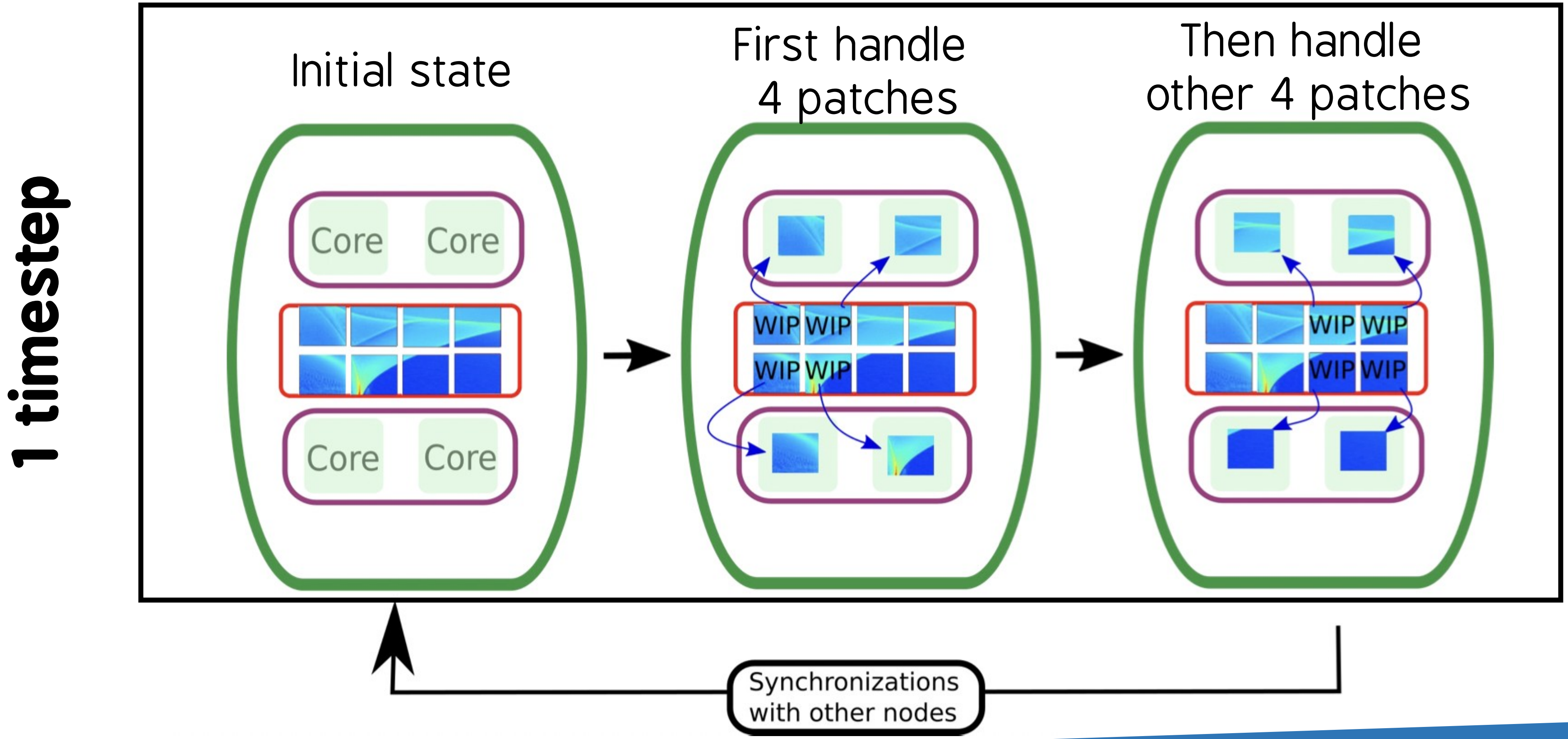
Ok, but where are the patches in the supercomputer?

All patches of the MPI domain owned by the local node
are stored in the memory



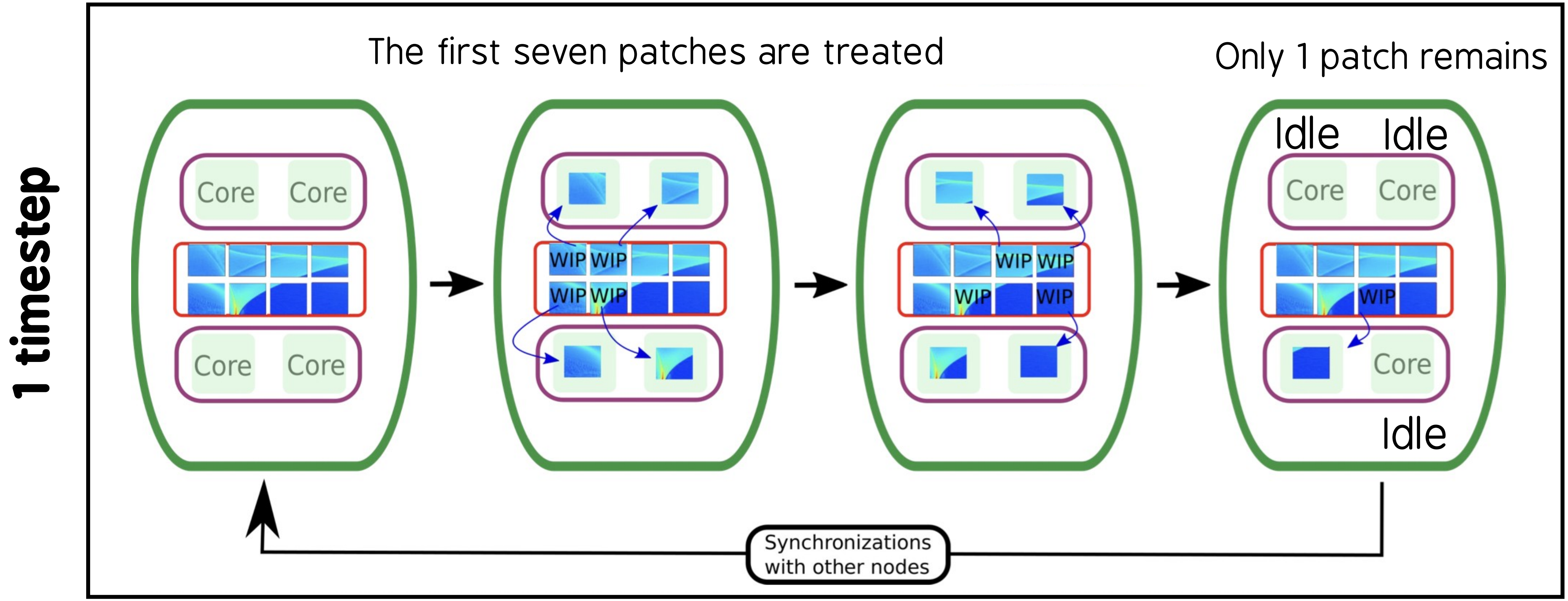
The OpenMP scheduler distributes patches to threads

The OpenMP scheduler assigns cores to patches via the **openMP threads**.
The number of OpenMP threads is fixed by the user and should be **one per core**.



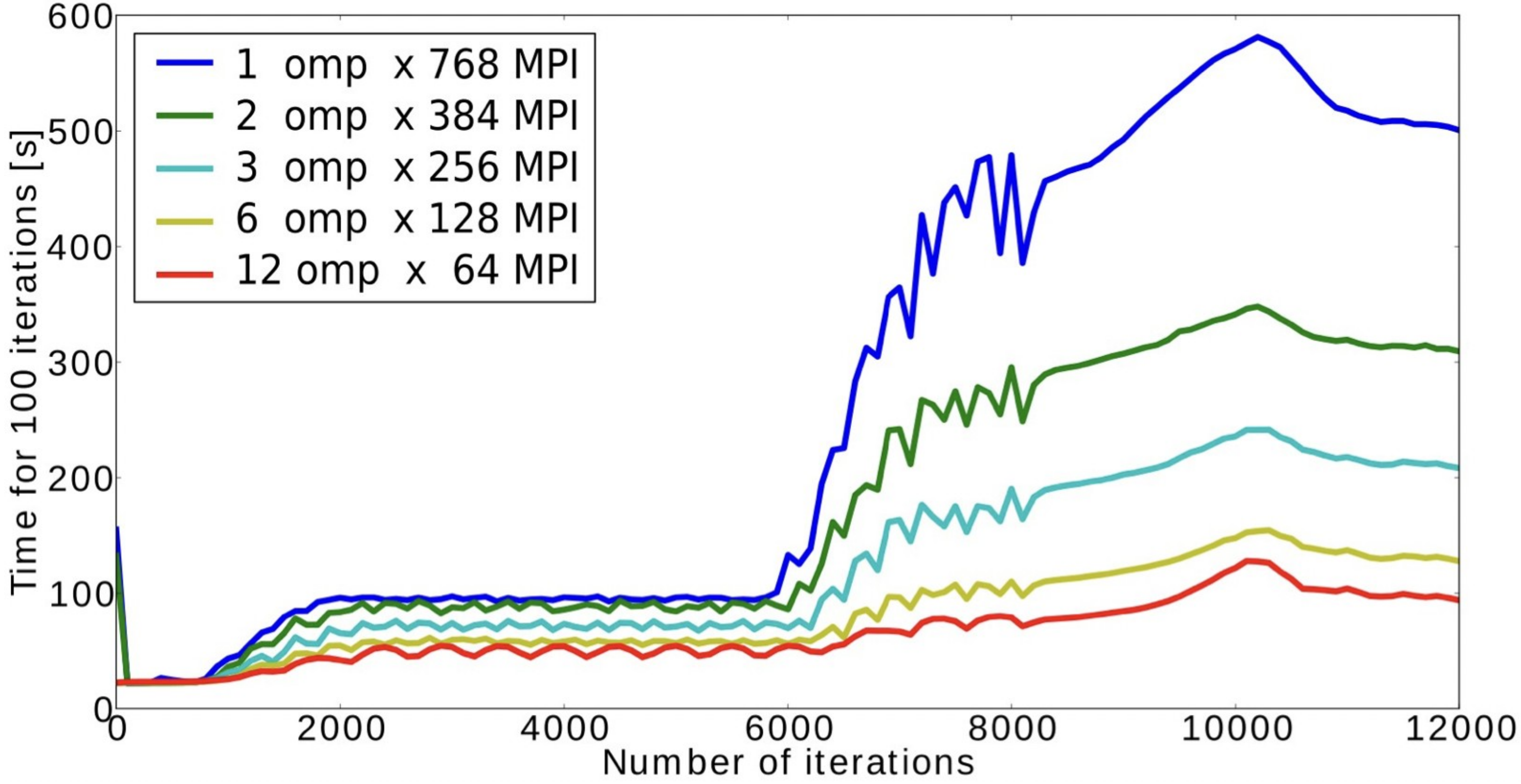
OpenMP threads and load imbalance

Imbalance of patch loads induces idle time



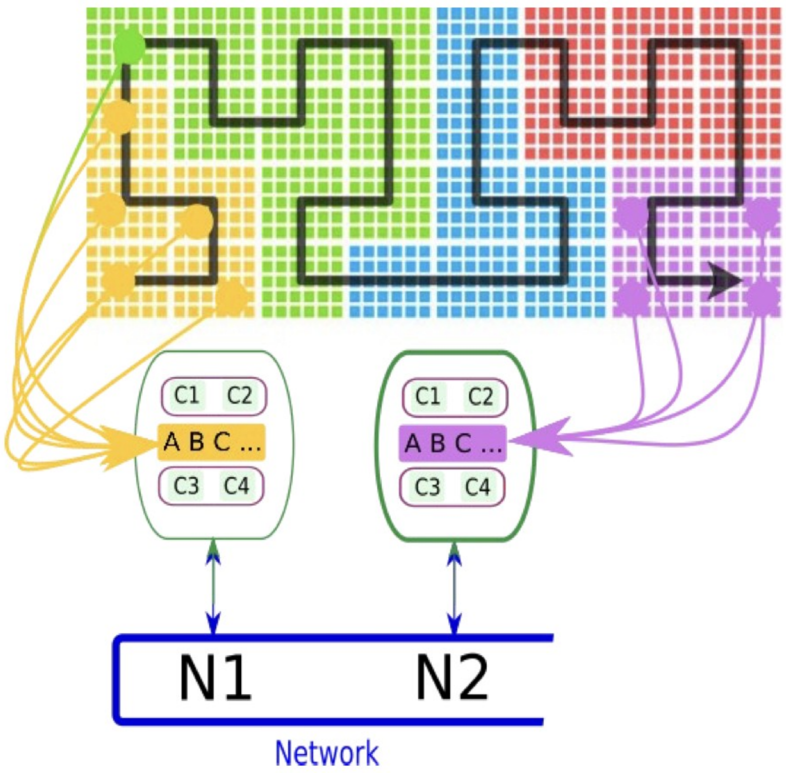
The OpenMP dynamic scheduler balances the load

Example with increasing load imbalance:
Laser Wakefield Acceleration

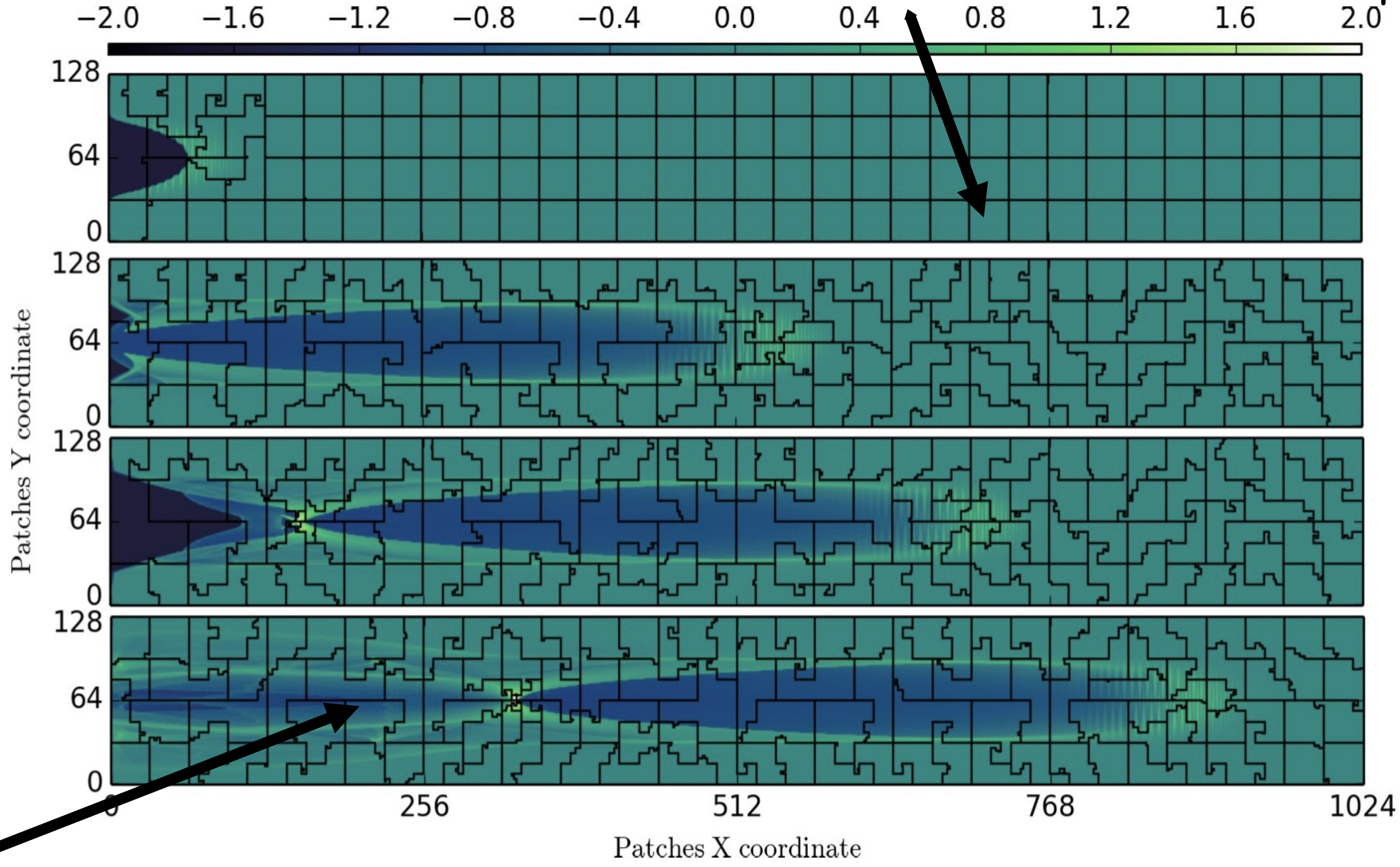


Note: no MPI load balancing
is used for this figure!
(See following slides)

Patches are exchanged between MPI domains through dynamic load balancing between MPI processes



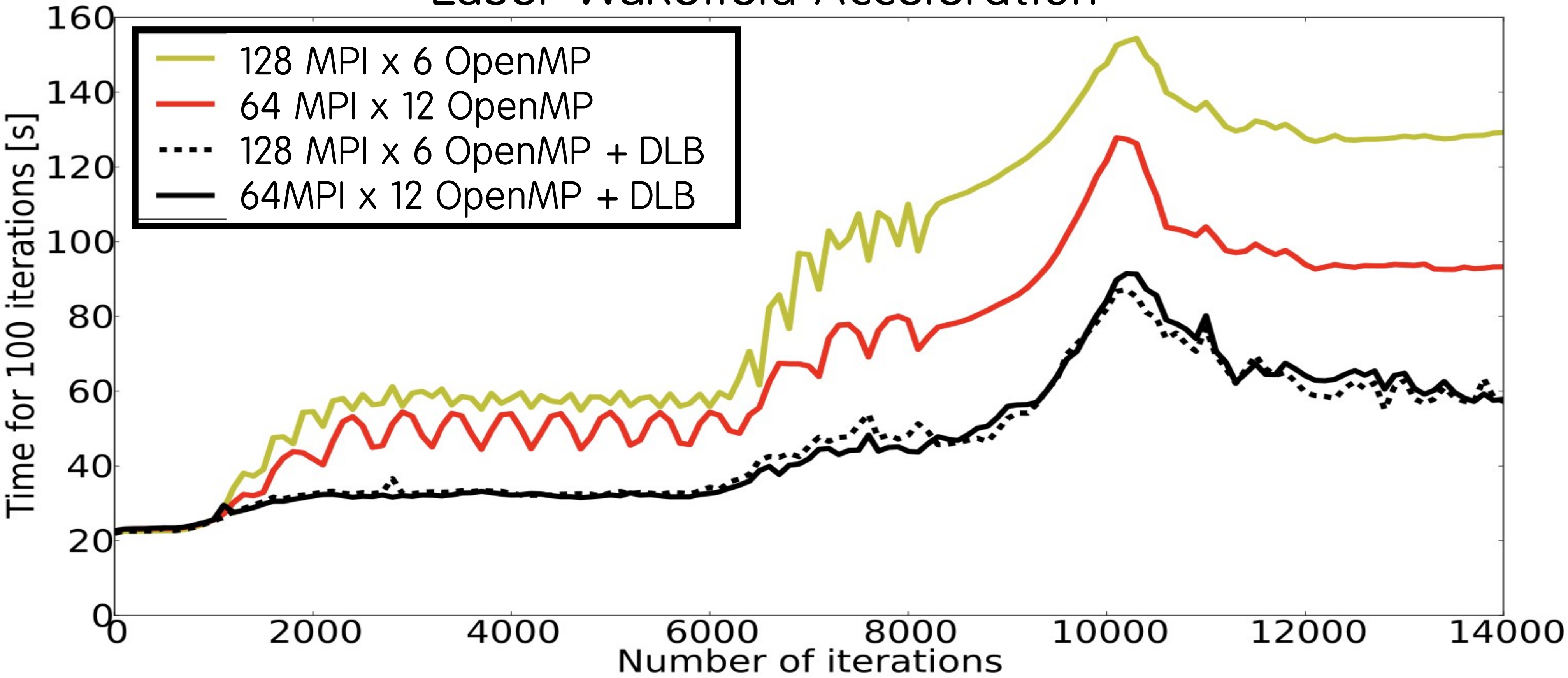
MPI domains still with their initial shape



MPI domains with Irregular shape after patch exchanges

Effects of Dynamic Load Balancing (DLB) between MPI

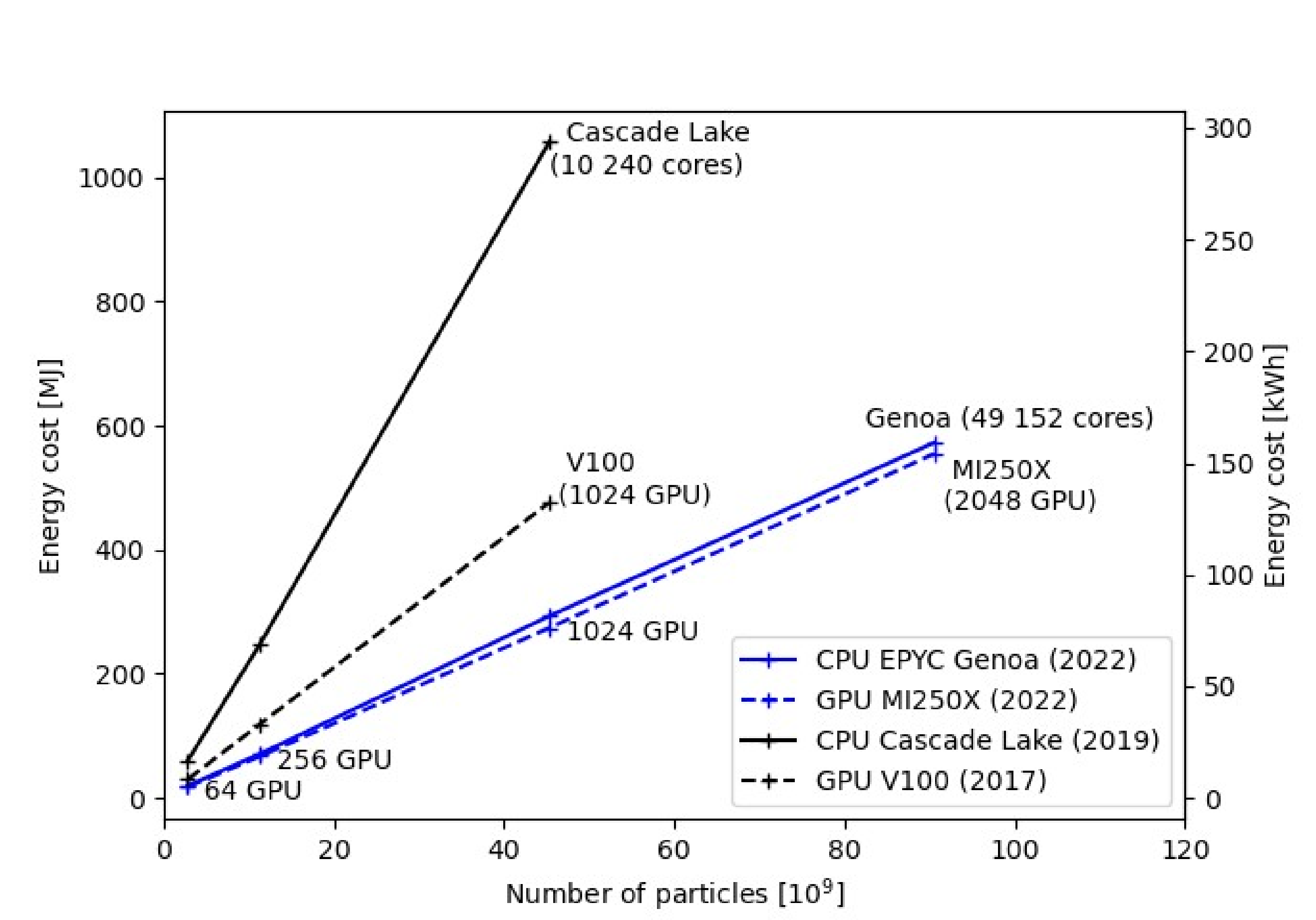
Example with increasing load imbalance:
Laser Wakefield Acceleration



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

Energy: the real metric for software performance



- ▶ 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”.

Scaling Free Softwares

Definition of free softwares

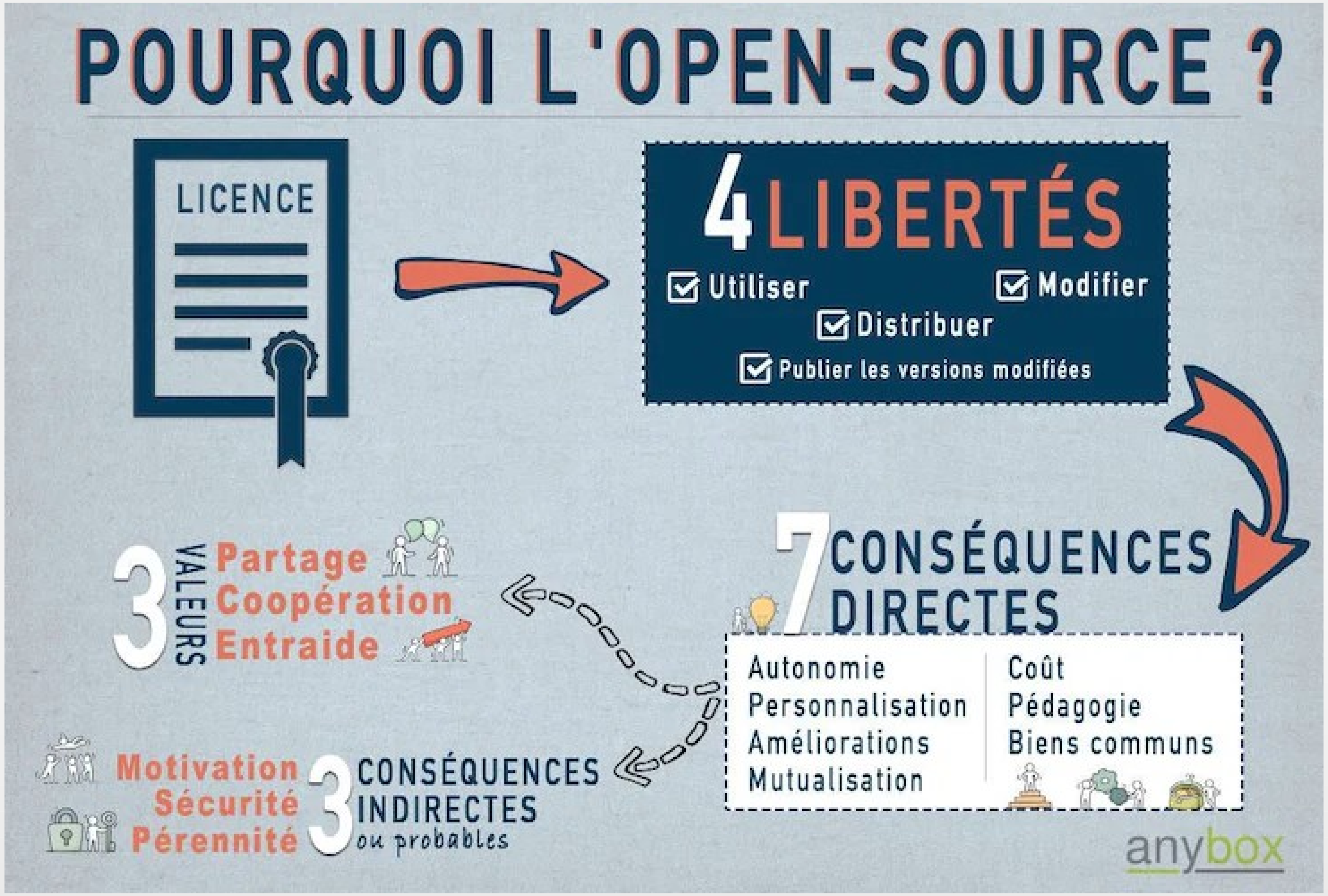
Free as in « free beer » : available at no cost.

Free as in « free speech » : open source, reusable.

« Broad set of working processes, social movements, and organizations that have formed around the production and distribution of software. »

The Labor of Maintaining and Scaling Free and Open-Source Software Projects
Geiger et. al. Proc. ACM Hum.-Comput. Interact., Vol 5, No. CSCW 1, April 2021

Why free softwares ?



POURQUOI L'OPEN-SOURCE ?

LICENCE

4 LIBERTÉS

☑ Utiliser
☑ Modifier
☑ Distribuer
☑ Publier les versions modifiées

7 CONSÉQUENCES DIRECTES

Autonomie
Personnalisation
Améliorations
Mutualisation
Coût
Pédagogie
Biens communs

3 VALEURS

Partage
Coopération
Entraide

3 CONSÉQUENCES INDIRECTES ou probables

Motivation
Sécurité
Pérennité

anybox

The scaling challenge

« As projects scale, work not only increases, but fundamentally changes. »

The Labor of Maintaining and Scaling Free and Open-Source Software Projects
Geiger et. al. Proc. ACM Hum.-Comput. Interact., Vol 5, No. CSCW 1, April 2021

Support requests skyrocket (especially when poorly documented).

Maintenance of documentation and collaborative tools.

Maintenance of interfaces with other softwares and/or standards.

Maintenance of performances in an evolving hardware environment.

Organizing and participating to meetings and events.

The software engineering challenge

Smilei CI: Approximately 160 tests are run automatically after each modification of the code.

That is still not enough to check all configurations and does not even account for the GPU implementation.

Today developing a new feature is faster than integrating it in the code.

There is a critical need for software engineers.

The recognition challenge: free = lousy ?

F/OSS are often seen as less valuable.

Impacts on recruitments of scarce qualified manpower, careers and sustainability of projects.

The community makes a project visible and helps enhance its value.

The recognition challenge in research

« Les codes doivent être gérés comme des infrastructures de recherche. »

Septembre 2023 : État des lieux de la production et de la valorisation
des logiciels issus de la recherche publique française (MESRI)

« Le code doit être considéré comme une plateforme expérimentale et le développement de nouveaux algorithmes/codes doit être vu comme celui d'une expérience »

Juillet 2023 : Nouveaux Enjeux du Numérique (Prospectives 2030 de CNRS Physique)

The funding challenge

4 CNRS institutes, Ecole polytechnique, U. P.-S. and CEA all contribute to Smilei in France

but

Smilei is not a project (ANR, master project, ERC, ...).

Smilei is not a laboratory or an official team.

Smilei is not a foundation or an association.

Smilei is a free licence.

« Free » is interpreted as having no substantial existence.

A large range of skills are needed but very difficult to get.

How can we fund public free softwares ?

The “catastrophic success” of a rapid growth

More maintenance (features, hardware, debuging...)

More communication (conference, workshop, documentation, promotion...)

More support (compilation, utilization, post-process ...)

More integration in house or from contributions (C.I., debuging pull requests)

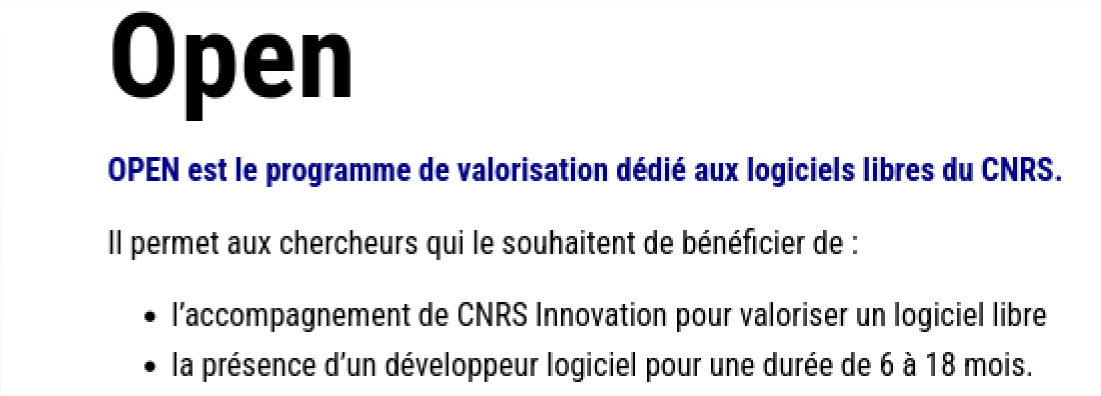
... leads to overworked maintainers.

« Scalar labor » is necessary to maintain a project at its new scale.

The “scalar debt” and inappropriate project support

Scalar debt => As a project advances, it will require more ressources.

A quick expansion will have to be « paid back » later.
Any help too limited in time might overwork the collaborators or bring the project to a stop.
Funding by project is inappropriate.



Open

OPEN est le programme de valorisation dédié aux logiciels libres du CNRS.

Il permet aux chercheurs qui le souhaitent de bénéficier de :

- l’accompagnement de CNRS Innovation pour valoriser un logiciel libre
- la présence d’un développeur logiciel pour une durée de 6 à 18 mois.

Beyond the single maintainer model

The situation remains under control as long as at least one person maintains the whole code.

As code and community grows, numerous tasks must be shared which implies synchronizations.

More difficult decision making.

Additional skills become critical (chat, git, PR, issues...)

Risks of exclusion of partners not familiar with collaborative tools.

Risk of having efficient developers turning into inefficient managers because the nature of the work changes.

Multiple maintainers need to scale trust

The single maintainer gives maintainer privileges to a trusted collaborator and so on ...

Linux has a pyramidal organization with Linus Torvalds on top,

In Smilei we have a board of trusted maintainers for different sections : GPU, input/output, post-process, vectorization, MPI synchronizations ...

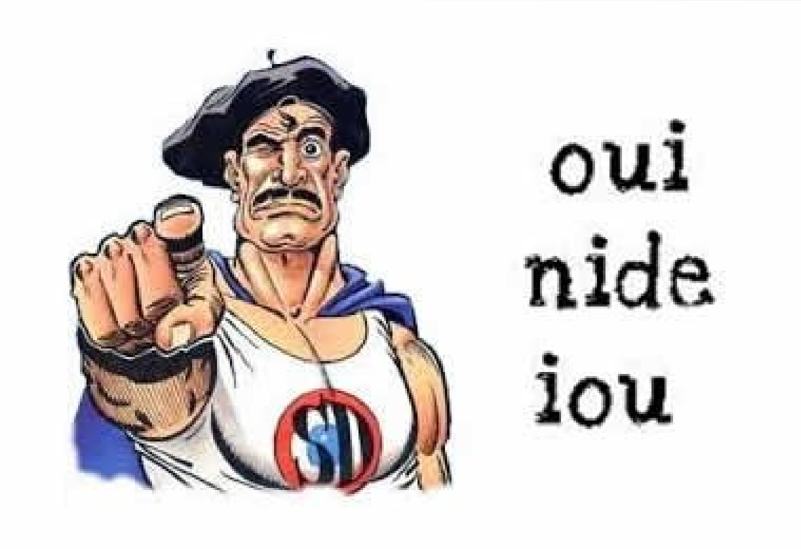
Requires collaborative work at interfaces.

No BDFL (Benevolent Dictator For Life).

Guido van Rossum
Python BDFL until 2018

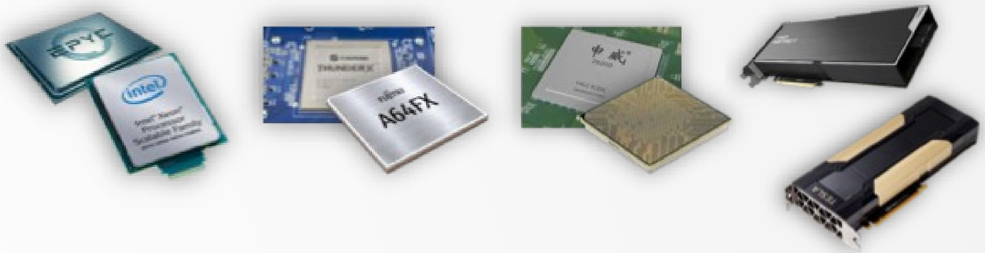
How to achieve a self sustaining community ?

- How to motivate, make possible and reward contributions to the community ?
- Online chat and Github are possible solutions but mostly remain a 1-way channel.
- 3 days workshops are great to initiate contacts but no effect on the long term.
- Does only size matter ?
- Wikipedia, Geant4, Linux ...



Perspectives

Code & HPC aspects



- GPU porting: AM geometry, adv. phys. modules, load-balancing
- GPU continuous integration
- Integration in a portable framework (Kokkos)
- 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 the next Smilei workshop !

Thanks & Keep Smileing !

Contributing labs, computing centers, institutions & funding agencies

MAISON DE LA SIMULATION

CINES Centre Informatique National de l'Enseignement Supérieur

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

idris

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