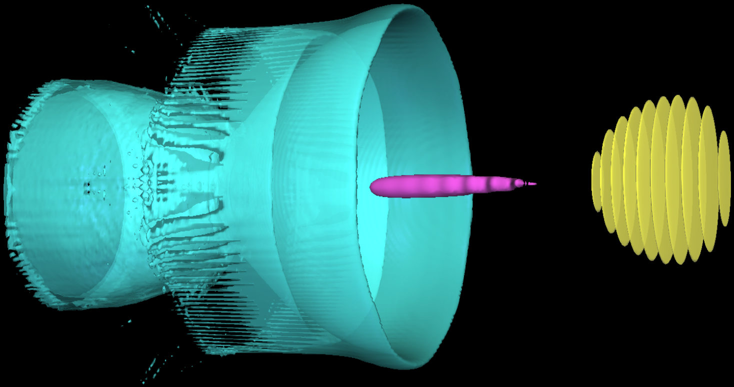
Laser Wakefield Acceleration Numerical Simulation

Arnaud Beck, Laboratoire Leprince-Ringuet, CNRS/IN2P3
Imen Zemzemi, Francesco Massimo, Martin Khojoyan, Arnd Specka



F. Massimo - LLR

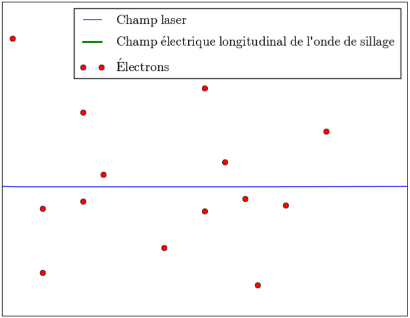
Structure

1. Laser Wakefield Acceleration
2. The HPC environment : Opportunities and Constraints
3. The PIC method and its parallelization
4. Accuracy issues, the example of numerical dispersion
5. Performances issues : the example of dynamic load balancing
6. Latest developments and results

Structure

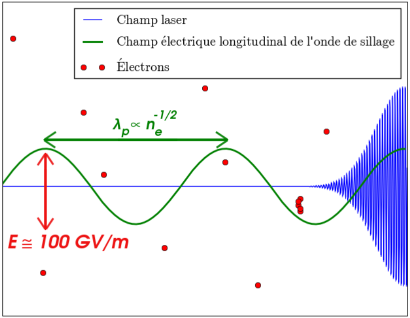
1. Laser Wakefield Acceleration
2. The HPC environment : Opportunities and Constraints
3. The PIC method and its parallelization
4. Accuracy issues, the example of numerical dispersion
5. Performances issues : the example of dynamic load balancing
6. Latest developments and results

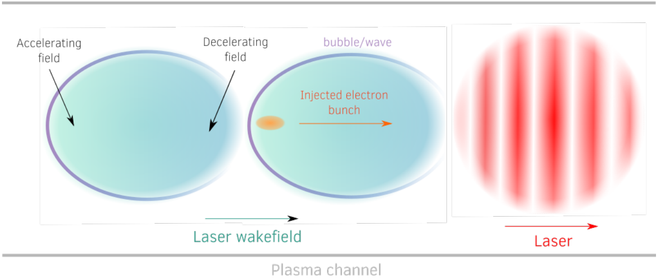
L'ACCÉLÉRATION PAR SILLAGE LASER
POUR
LES NULS



Champ laser
Champ électrique longitudinal de l'onde de sillage
Électrons

$c\tau_L < \lambda_p$

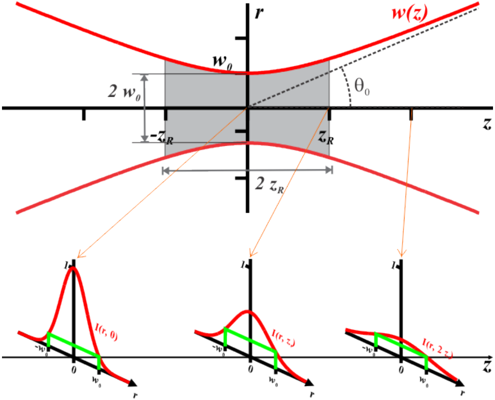



Accelerating field
Decelerating field
bubble/wave
Injected electron bunch
Laser wakefield
Laser
Plasma channel

Ponderomotive force

$$\boldsymbol{F_p} = \frac{e^2}{4\omega^2 m}\nabla \boldsymbol{E^2}$$

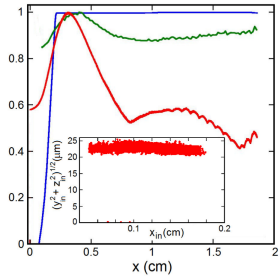
Diffraction in vacuum :



Optical indice in a plasma :

$$n = \sqrt{1 - \frac{\omega_p^2}{\omega^2}}$$

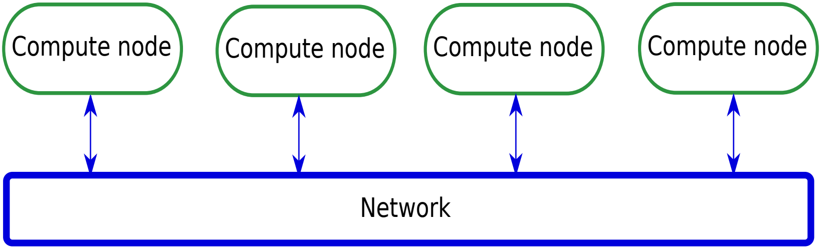
$$\omega_p^2 = \frac{n_e e^2}{\epsilon_0 m_e}$$



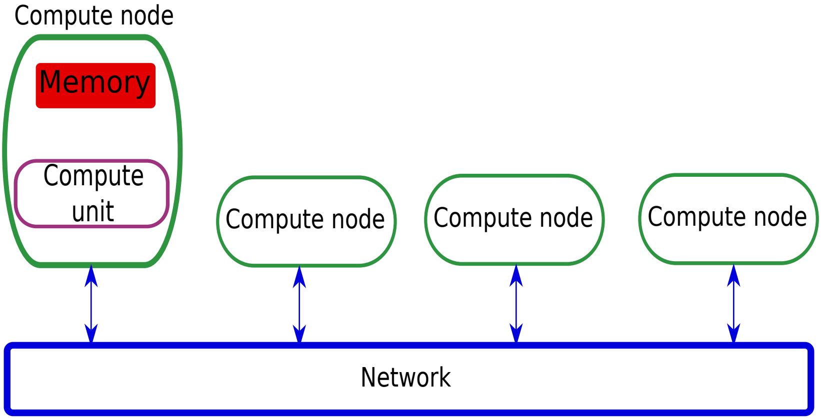
Physical processes at work in sub-30 fs, PW laser pulse-driven plasma accelerators, Beck et. al., NMIA, 2014.

- Describe **wave-particle interactions** accurately at **small time and space scales** over **a large number of oscillations**.
- Evaluate local **relativistic** optical index.
- Solve **Maxwell equations**.
- Account for the collective behaviour of the plasma at **large scales**.
- **3D** in space and in momentum.

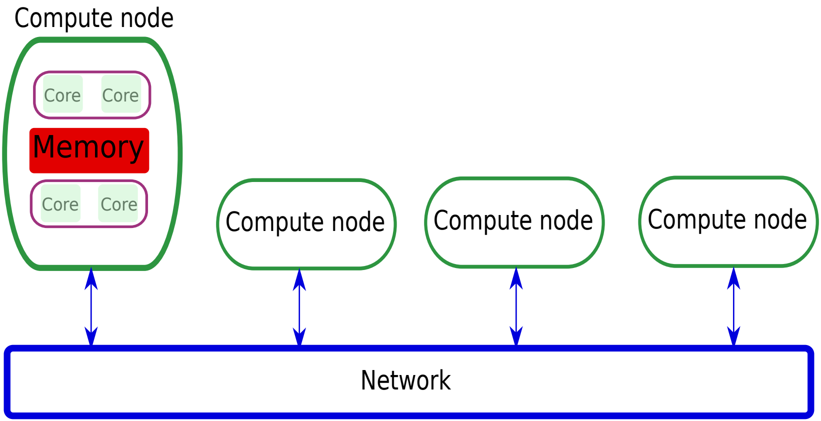
Structure



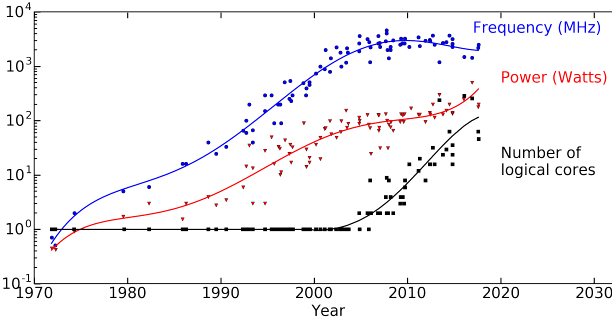
Distributed computing



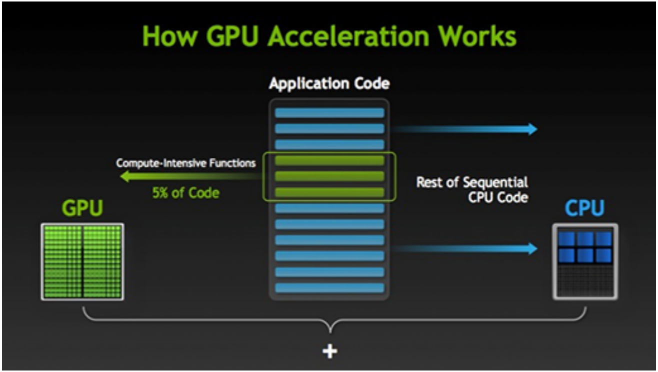
Distributed memory system



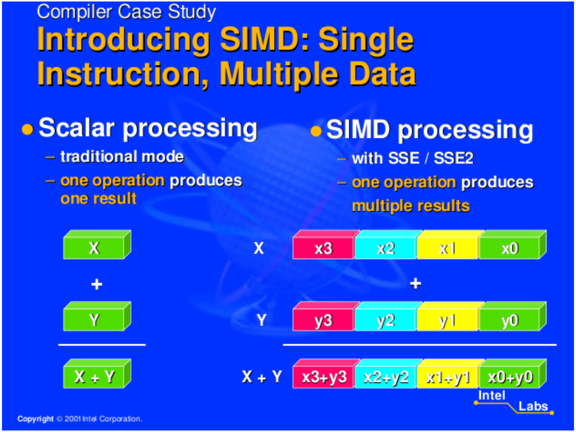
Distributed {shared memory} system



- Increased performances
- Reasonable energy budget



- Most energy efficient architecture today
- Difficult to adress :
  - Libraries : Cuda, OpenCl.
  - Directives programming : OpenMP 4 ou openACC.



- Excellent potential speed up, very good power budget.
- Heavy constraints on data structure and algorithm.
- Difficult to use at its full extent in a PIC code.

Why am I concerned ? What should I do ?

As a developer

1. Expose parallelism. Massive parallelization is key.
2. Focus on the algorithm and data structures. Not on architectures.
3. Reduce data movement : Computation is becoming cheaper, loads and stores not so much.
4. Be aware of the increasing gap between peak power and effective performances. The race to exascale is becoming a race to exaflops.

As a user

1. Disclaimer. Parallelization is performed by experts.

«A collaborative, open-source, multi-purpose PIC code for the next generation of super-computers»

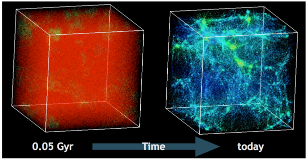
Smilei)

www.maisondelasimulation.fr/smilei

Structure

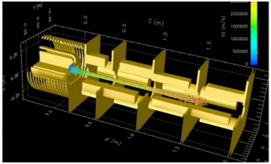
The Particle-In-Cell (PIC) method is a central tool for simulation over a wide range of physics studies

Cosmology



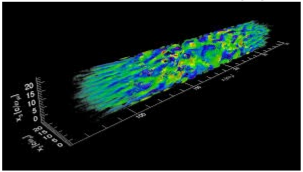
source: K. Heitmann, Argonne National Lab

Accelerator physics



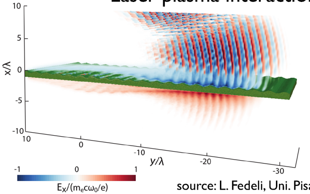
source: WARP , Berkeley Lab

Relativistic astrophysics



source: F. Fiuza, Livermore National Lab

Laser plasma interaction



source: L. Fedeli, Uni. Pisa

- Conceptually simple
- Efficiently implemented on (massively) parallel super-computers

A PIC code is a solver

$f_s(\mathbf{x},\mathbf{v})d\mathbf{x}d\mathbf{v}$ is the probability to find a particle of species s in the phase space point (**x**,**v**) around d**x**d**v**.

Vlasov equation

No collisions

$$\frac{\partial f_s}{\partial t} + \mathbf{v}\cdot\frac{\partial f_s}{\partial \mathbf{x}} + q_s/m_s(\mathbf{E} + \frac{\mathbf{v}\times\mathbf{B}}{c})\cdot\frac{\partial f_s}{\partial \mathbf{v}} = 0$$

Maxwell's equations

$$\begin{cases} \nabla\cdot\mathbf{E} = 4\pi\rho \\ \nabla\cdot\mathbf{B} = 0 \\ \nabla\times\mathbf{E} = -\frac{1}{c}\frac{\partial\mathbf{B}}{\partial t} \\ \nabla\times\mathbf{B} = \frac{1}{c}\frac{\partial\mathbf{E}}{\partial t} + \frac{4\pi}{c}\mathbf{J} \end{cases}$$

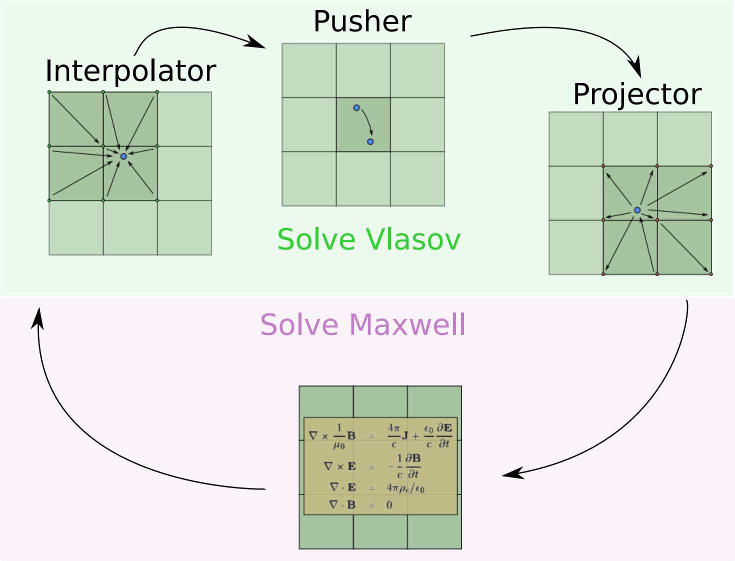
Moments equations

$$\rho = \sum_s^{n_s} q_s \int f_s d\mathbf{v}$$

$$\mathbf{J} = \sum_s^{n_s} q_s \int \mathbf{v} f_s d\mathbf{v}$$

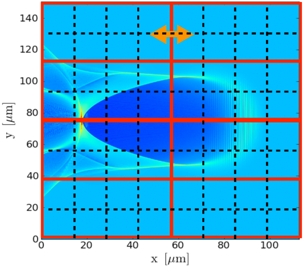
Difficulty : Equations are coupled !

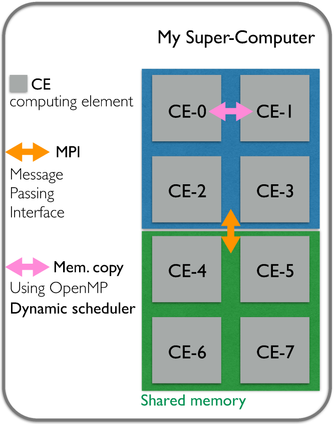
Explicit PIC code principle



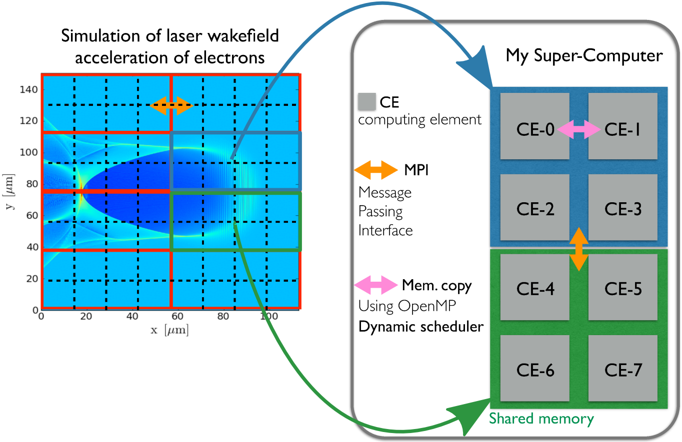
PIC code are ‘easily’ parallelized using domain decomposition

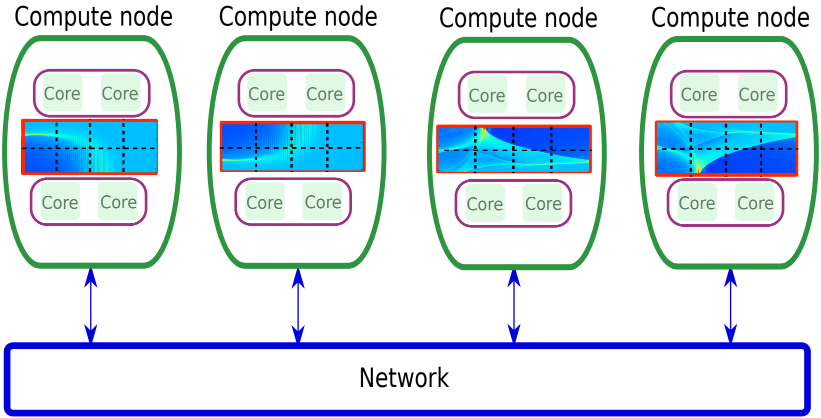
Simulation of laser wakefield acceleration of electrons



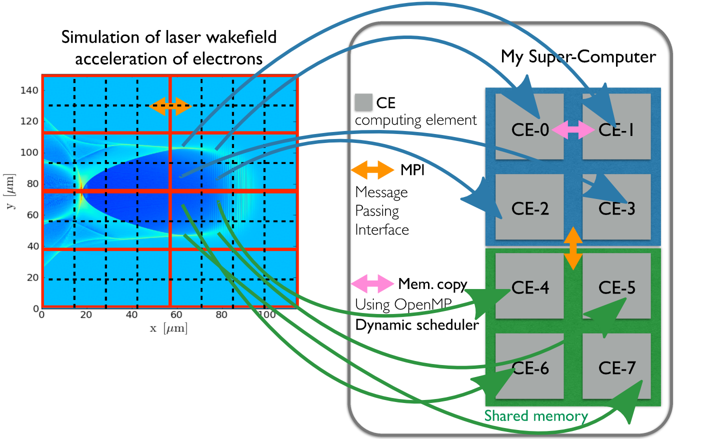


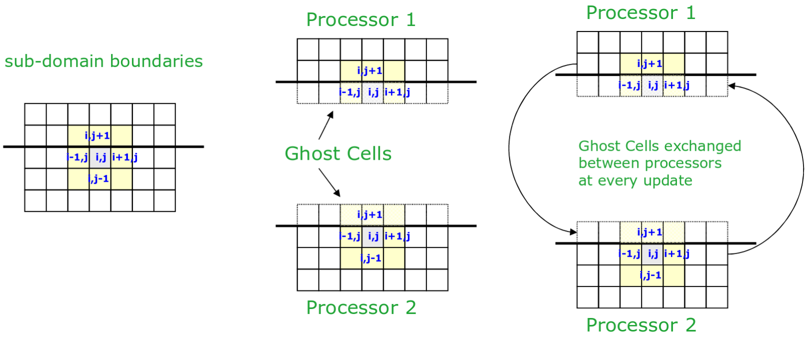
PIC code are ‘easily’ parallelized using domain decomposition




Compute node
Compute node
Compute node
Compute node
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Core
Network

PIC code are ‘easily’ parallelized using domain decomposition + Patch



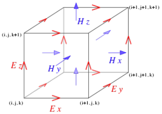


- If processors have a shared memory ==> OpenMP
- If processors have ditributed memory ==> MPI
- Same logic for particles

Structure

The Yee Lattice and discrete maxwell equations

Discrete Maxwell equations for finite difference time domain (FDTD) scheme :



$$\frac{(E_x)^{n+1}_{i+\frac{1}{2},j,k}-(E_x)^{n}_{i+\frac{1}{2},j,k}}{\Delta t} = -\frac{1}{\epsilon_0}(J_x)^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k} + c^2\left[\frac{(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k}-(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j-\frac{1}{2},k}}{\Delta y} - \frac{(B_y)^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}-(B_y)^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k-\frac{1}{2}}}{\Delta z}\right]$$

$$\frac{(E_y)^{n+1}_{i,j+\frac{1}{2},k}-(E_y)^{n}_{i,j+\frac{1}{2},k}}{\Delta t} = -\frac{1}{\epsilon_0}(J_y)^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k} + c^2\left[\frac{(B_x)^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}-(B_x)^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k-\frac{1}{2}}}{\Delta z} - \frac{(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k}-(B_z)^{n+\frac{1}{2}}_{i-\frac{1}{2},j+\frac{1}{2},k}}{\Delta x}\right]$$

$$\frac{(E_z)^{n+1}_{i,j,k+\frac{1}{2}}-(E_z)^{n}_{i,j,k+\frac{1}{2}}}{\Delta t} = -\frac{1}{\epsilon_0}(J_z)^{n+\frac{1}{2}}_{i,j,k+\frac{1}{2}} + c^2\left[\frac{(B_y)^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}-(B_y)^{n+\frac{1}{2}}_{i-\frac{1}{2},j,k+\frac{1}{2}}}{\Delta x} - \frac{(B_x)^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}-(B_x)^{n+\frac{1}{2}}_{i,j-\frac{1}{2},k+\frac{1}{2}}}{\Delta y}\right]$$

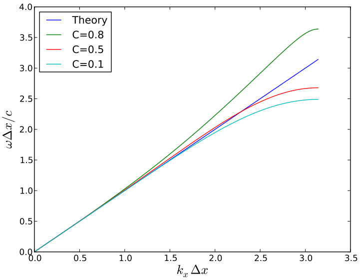
$$\frac{(B_x)^{n+\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}-(B_x)^{n-\frac{1}{2}}_{i,j+\frac{1}{2},k+\frac{1}{2}}}{\Delta t} = -\frac{(E_z)^{n}_{i,j+1,k+\frac{1}{2}}-(E_z)^{n}_{i,j,k+\frac{1}{2}}}{\Delta y} + \frac{(E_y)^{n}_{i,j+\frac{1}{2},k+1}-(E_y)^{n}_{i,j+\frac{1}{2},k}}{\Delta z}$$

$$\frac{(B_y)^{n+\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}-(B_y)^{n-\frac{1}{2}}_{i+\frac{1}{2},j,k+\frac{1}{2}}}{\Delta t} = -\frac{(E_x)^{n}_{i+\frac{1}{2},j,k+1}-(E_x)^{n}_{i+\frac{1}{2},j,k}}{\Delta z} + \frac{(E_z)^{n}_{i+1,j,k+\frac{1}{2}}-(E_z)^{n}_{i,j,k+\frac{1}{2}}}{\Delta x}$$

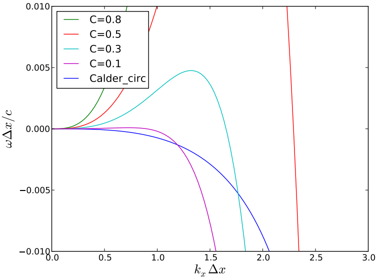
$$\frac{(B_z)^{n+\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k}-(B_z)^{n-\frac{1}{2}}_{i+\frac{1}{2},j+\frac{1}{2},k}}{\Delta t} = -\frac{(E_y)^{n}_{i+1,j+\frac{1}{2},k}-(E_y)^{n}_{i,j+\frac{1}{2},k}}{\Delta x} + \frac{(E_x)^{n}_{i+\frac{1}{2},j+1,k}-(E_x)^{n}_{i+\frac{1}{2},j,k}}{\Delta y}$$

Yee latice convienently allows first order finite difference scheme to produce order 2 accuracy : $\mathcal{O}(\Delta x^2)$.

A plane wave is propagated in vacuum via the discrete equations. We obtain information on stability and accuracy.



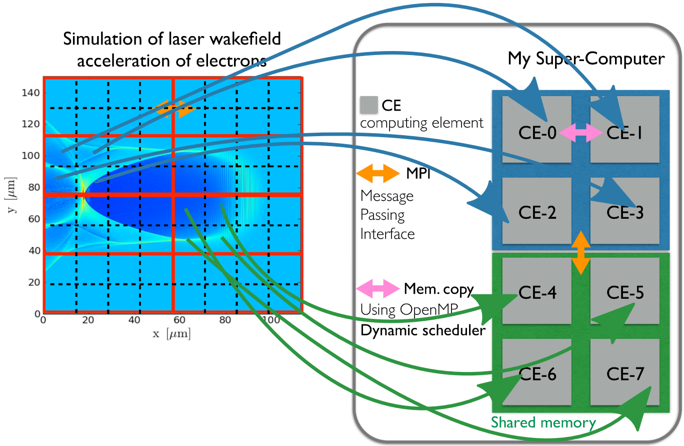
Numerical dispersion in vacuum.



Difference between numerical and theoretical dispersion.

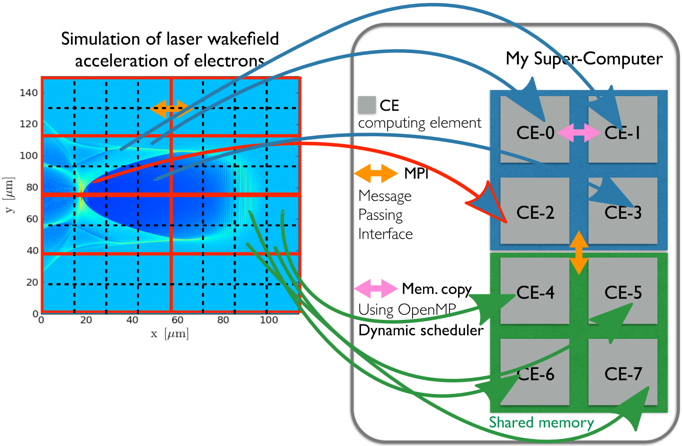
Structure

PIC code are 'easily' parallelized using domain decomposition



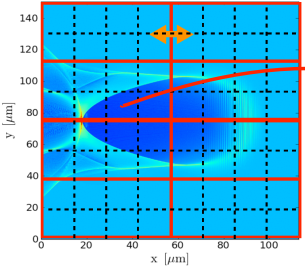
Domain decomposition : MPI + openMP in SMILEI

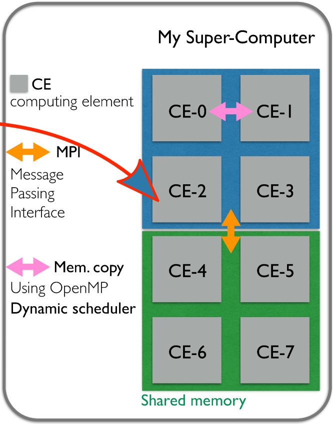
PIC code are ‘easily’ parallelized using domain decomposition



PIC code are 'easily' parallelized using domain decomposition

Simulation of laser wakefield acceleration of electrons

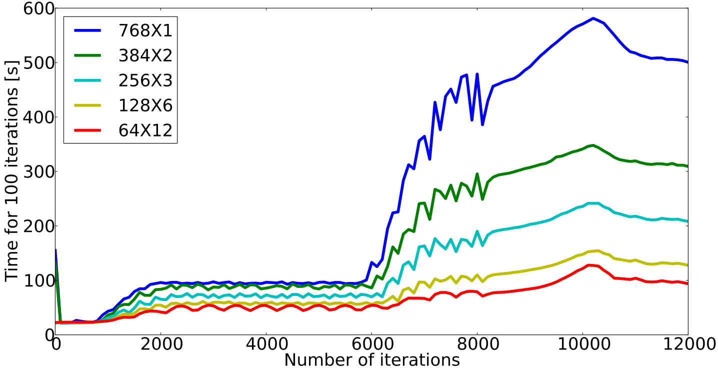




Domain decomp. is not enough!!!

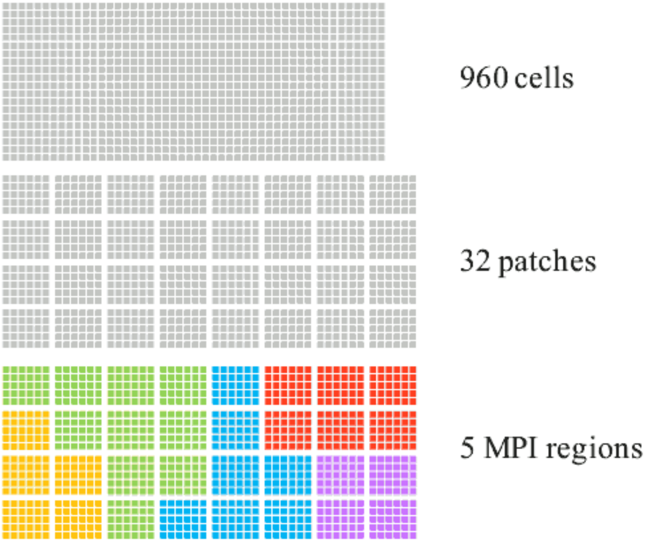
- workload not optimally shared
- not adapted to new architectures!

MPI × *OpenMP*

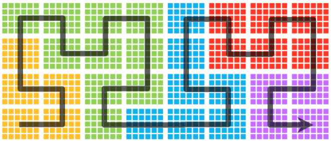


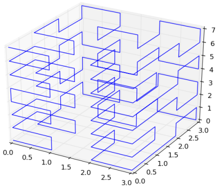
OpenMP dynamic scheduler is able to smooth the load but only at the node level.

Patched base data structure



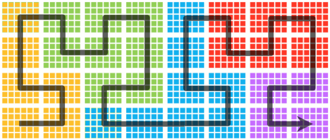
Hilbert ordering

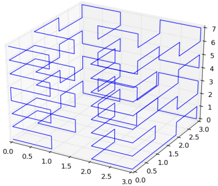




We need a policy to assign patches to MPI processes. To do so, patches are organized along a one dimensional space-filling curve.

1. Continuous curve which goes across all patches.
2. Each patch is visited only once.
3. Two consecutive patches are neighbours.
4. In addition we want compactness !

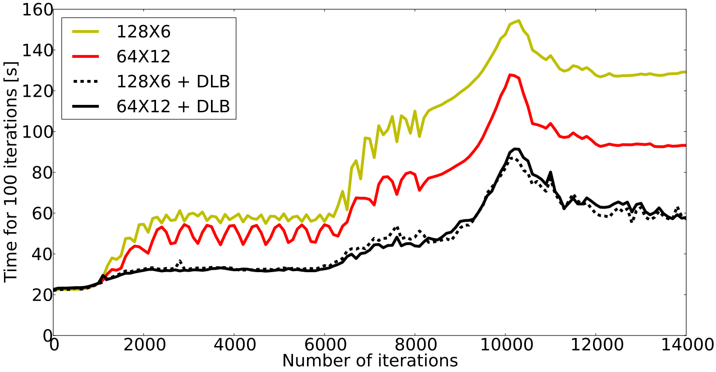




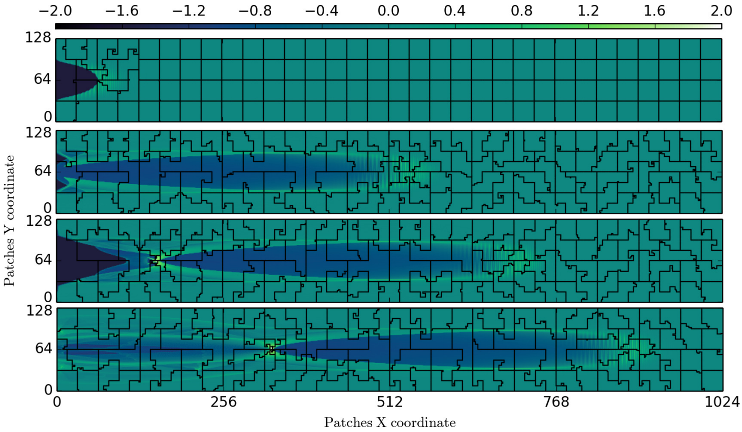
We need a policy to assign patches to MPI processes. To do so, patches are organized along a one dimensional space-filling curve.

1. Continuous curve which goes across all patches.
2. Each patch is visited only once.
3. Two consecutive patches are neighbours.
4. In addition we want compactness !

MPI × *OpenMP*

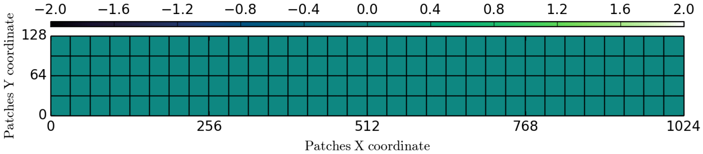


Yellow and red are copied from previous figure.



Color represents the local patch computational load imbalance

$$I_{loc} = \log_{10}\left(L_{loc}/L_{av}\right)$$

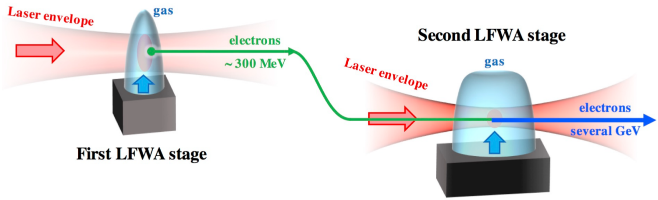


Color represents the local patch computational load imbalance

$$I_{loc} = \log_{10}\left(L_{loc}/L_{av}\right)$$

Structure

Multi-stages acceleration

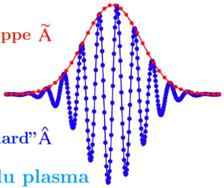


Hypothèse:

$$\hat{A}(\mathbf{x},t) = Re\left[\tilde{A}(\mathbf{x},t)e^{ik_0(x-ct)}\right]$$

Enveloppe Complexe

Laser Enveloppe $\tilde{\mathrm{A}}$



Equation d'Enveloppe:

$$\nabla^2\tilde{A} + 2i\left(\partial_x\tilde{A} + \partial_t\tilde{A}\right) - \partial_t^2\tilde{A} = \chi\tilde{A}$$

Laser "Standard" $\hat{\mathrm{A}}$

$$\chi(\mathbf{x}) = \sum_s \frac{q_s^2}{m_s}\sum_p \frac{w_p}{\bar{\gamma}_p} S(\mathbf{x}-\bar{\mathbf{x}}_p)$$

Susceptibilité du plasma

Equations du Mouvement des Macroparticules:

$$\frac{d\bar{\mathbf{x}}_p}{dt} = \frac{\bar{\mathbf{u}}_\mathbf{p}}{\bar{\gamma}_p}$$

Force Pondéromotrice

$$\bar{\gamma}_p = \sqrt{1 + \bar{\mathbf{u}}_p^2 + \frac{|\tilde{A}(\bar{\mathbf{x}}_p)|^2}{2}}$$

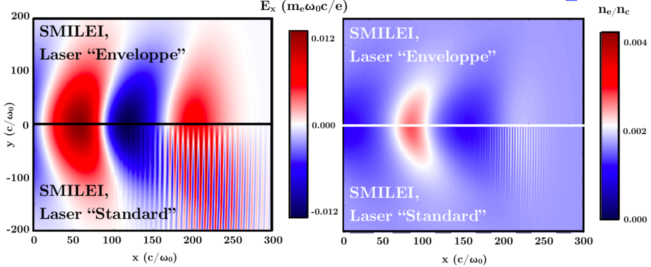
$$\frac{d\bar{\mathbf{u}}_p}{dt} = r_s\left(\bar{\mathbf{E}}_p + \frac{\bar{\mathbf{u}}_p}{\bar{\gamma}_p}\times\bar{\mathbf{B}}_p\right) - r_s^2\frac{1}{4\bar{\gamma}_p}\nabla\left(|\tilde{A}_p|^2\right)$$

$$r_s = q_s/m_s$$

Force de Lorentz

P. Mora and T. M. Antonsen Jr, Physics of Plasmas 4, 217 (1997)
B. Quesnel and P. Mora, Physics Review E 58, 3719 (1998)
S. Sinigardi et al., ALaDyn v2017.1 zenodo (2017)

SMILEI 3D, Laser “Standard” 10.6 heures, 256 noeuds
SMILEI 3D, Laser “Enveloppe” 10 minutes, 50 noeuds
] 1 mm



$L_x = 400\ c/\omega_0$
$L_y = L_z = 640\ c/\omega_0$
Particules par cellule = 8

$$\frac{\text{T-processus Laser “Standard”}}{\text{T-processus Laser “Enveloppe”}} = 320\ !$$

$$F(x,r,\theta) = \tilde{F}^0 + \sum_{\mathrm{m=1}}^{+\infty} \Re\left(\tilde{F}^m(x,r)\right)\cos(m\theta) + \Im\left(\tilde{F}^m(x,r)\right)\sin(m\theta)$$

We can then rewrite Maxwell equations for each Fourier modes :

$$\frac{\partial \tilde{B}_r^m}{\partial t} = \frac{im}{r}\tilde{E}_x^m + \frac{\partial \tilde{E}_\theta^m}{\partial x}$$

Since the modes are independant, the simulation boils down to m 2D simulations.

Grand Challenge GENCI

- New French super computer Irene : 79 488 cores and computational power of 6,86 Pflop/s.
- Preliminary access was granted to 20 applications and Smilei won this “Grand Challenge” : 7 million hours.
- Smilei ran and showed exceptional efficiency on 43200 cores.
- Perfect opportunity to run a partial second stage simulation to validate the enveloppe model on a long distance propagation.

Pour utilisateurs, futurs développeurs:

Smilei) Training Workshop!

http://www.maisondelasimulation.fr/smilei/

Prochaine Edition: Fevrier/Mars 2019

Smilei Training workshop 2017,
Maison de la Simulation