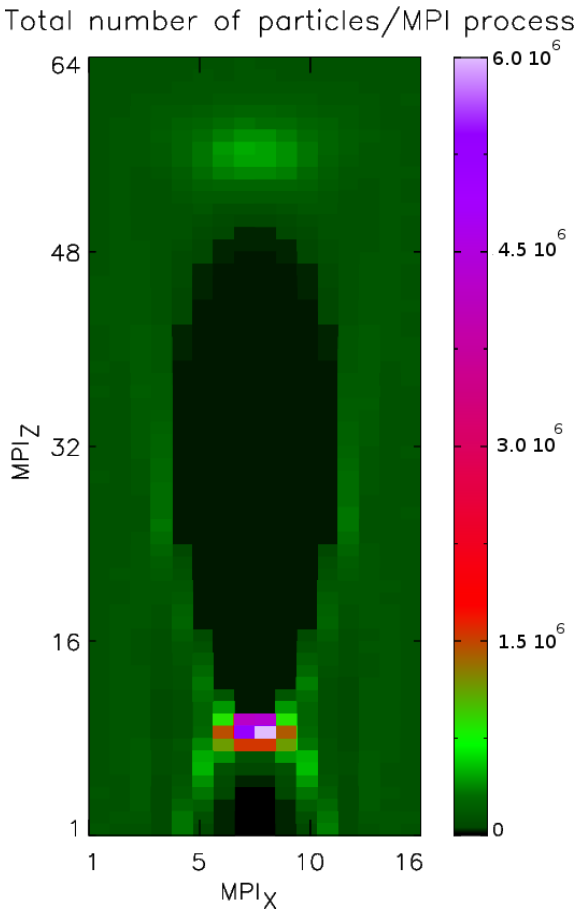
On the importance of being balanced

Arnaud Beck - LLR

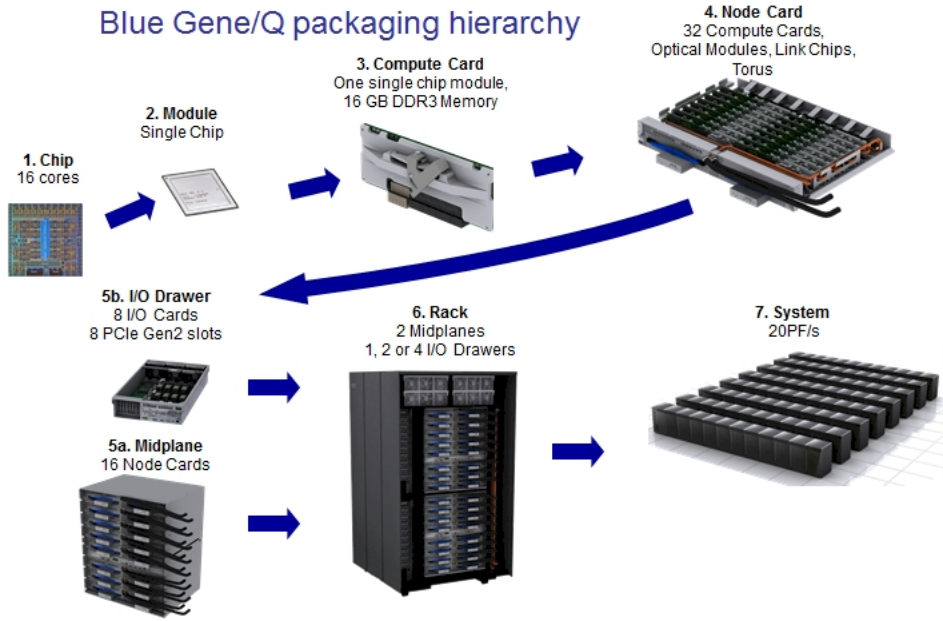
EAAC 2015

The problem

Example of Laser Wakefield Electron Acceleration in full 3D, Photon-Plasma, Niels Bohr Institute.



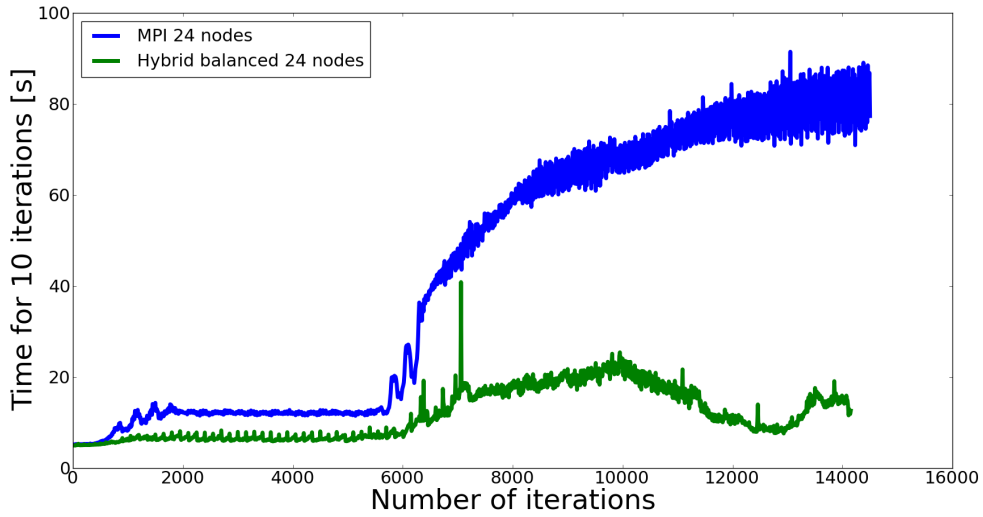
2048 BG/Q nodes.
16X16X64 MPI processes.
8 openMP threads.
Standard domain decomposition.
Imbalance of 40.



Many other cases are sensitive to this problem (magnetic reconnection, laser-solid interaction...)

Our solution

2 identical runs with Smilei on 24 nodes of 24 cores each. 2D simulation, the imbalance rate is approximately 15 here.



Smilei, a collaborative code

Smilei is born from the will of a group of labs to work together on a common and open source PIC code for laser matter interaction in the wake of the Cilex initiative.

The objective is to give access to an HPC simulation tool for laser-matter interaction to the local growing community.

It brings together physicists and numerical experts.

New users are more than welcome.

Disclaimer

Don't expect too much from a 2 year old

Smilei does not support

- 3D (in progress)
- Ionization (in progress)
- High order schemes
- Cylindrical geometry
- GPUs, MIC ...
- PML
- Boosted frame
- Aggressive optimization

Smilei supports

- 2D3V geometry
- Moving window
- Laser injection
- 2^{nd} order schemes
- Collisions
- hdf5 parallel outputs
- MPI+openMP
- Dynamic Load Balancing

Our load balancing strategy

Inspired by previous work by K. Germaschewski (University of New Hampshire, arXiv:1310.7866).

- Divide traditional MPI domains into many smaller structures called "Patch".
- Patches can be seen as sub-MPI domains, with their own set of particles and fields. Operators are all the same, only parallelism is affected.
- Each MPI processes owns a large number of patches.
- Patches are used as a sorting structures and can be treated independantly by the openMP threads.
- Patches are exchanged between MPI processes to ensure load balance.

Space filling curve

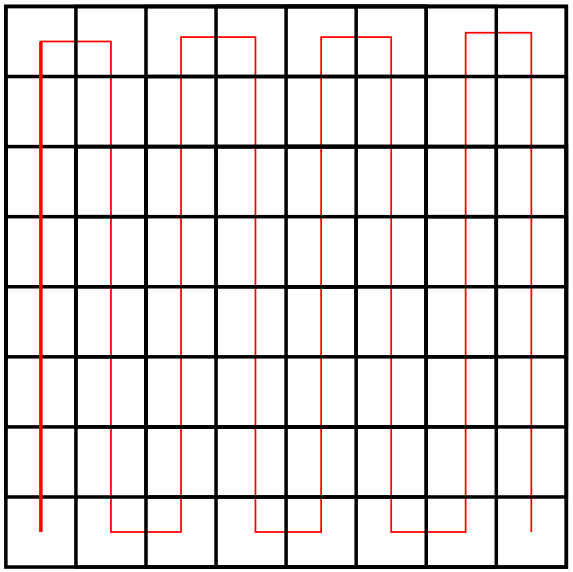
We need a policy to assign patches to MPI processes. To do so, patches are organized along a one dimensional space-filling curve.

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 !

The 1d curve is then devided into "*MPI_Comm_Size*" segments.

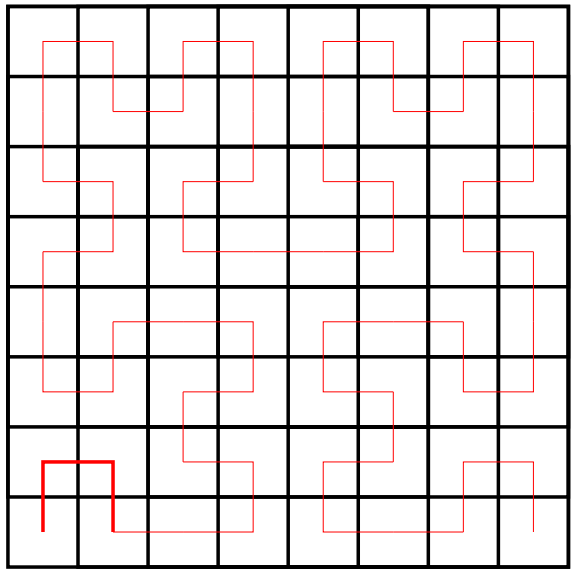
Examples

Naive implementation



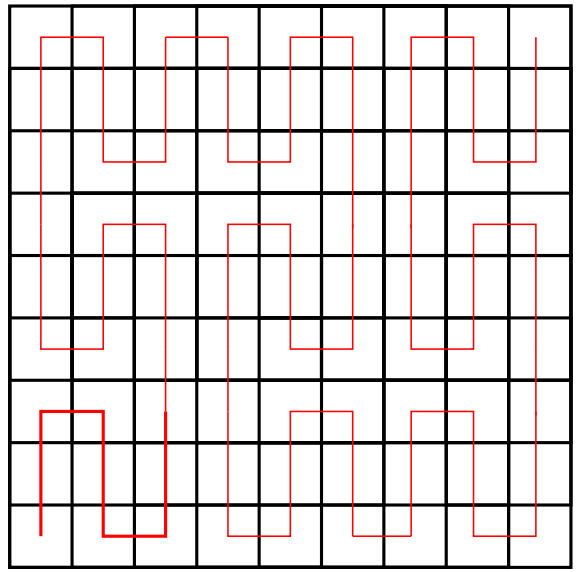
Base n.
Domain size dependant.

Hilbert curve



Base 2.
Pattern rotation at each iteration.

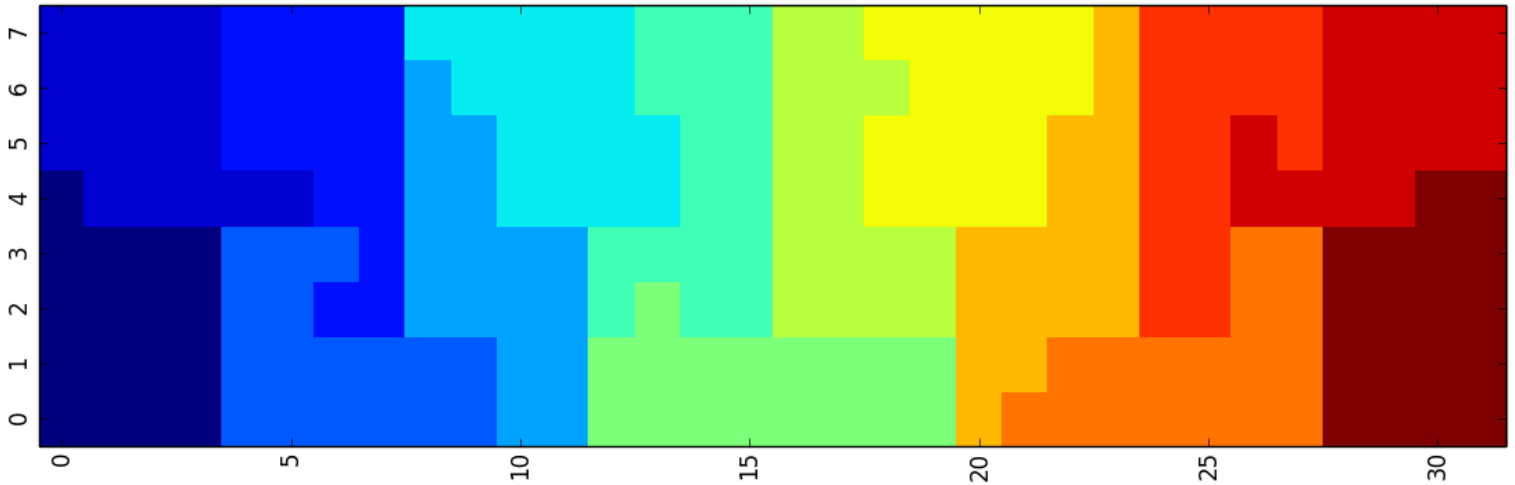
Peano curve



Base 3.
Same pattern at each iteration.

Example of partitioning

32 X 8 patches with 15 MPI processes.



Synchronization between patches is mostly intra-node.
We use a generalization of the Hilbert curve in order to be able to have non square grid.
The 3D manipulation of Hilbert curves is already implemented.

Balancing algorithm

1. Evaluate Total Load.
2. Evaluate optimal load for each process.
3. Claim patches until optimal load is approached.
4. Exchange patches to match the claims if necessary.

Parameters

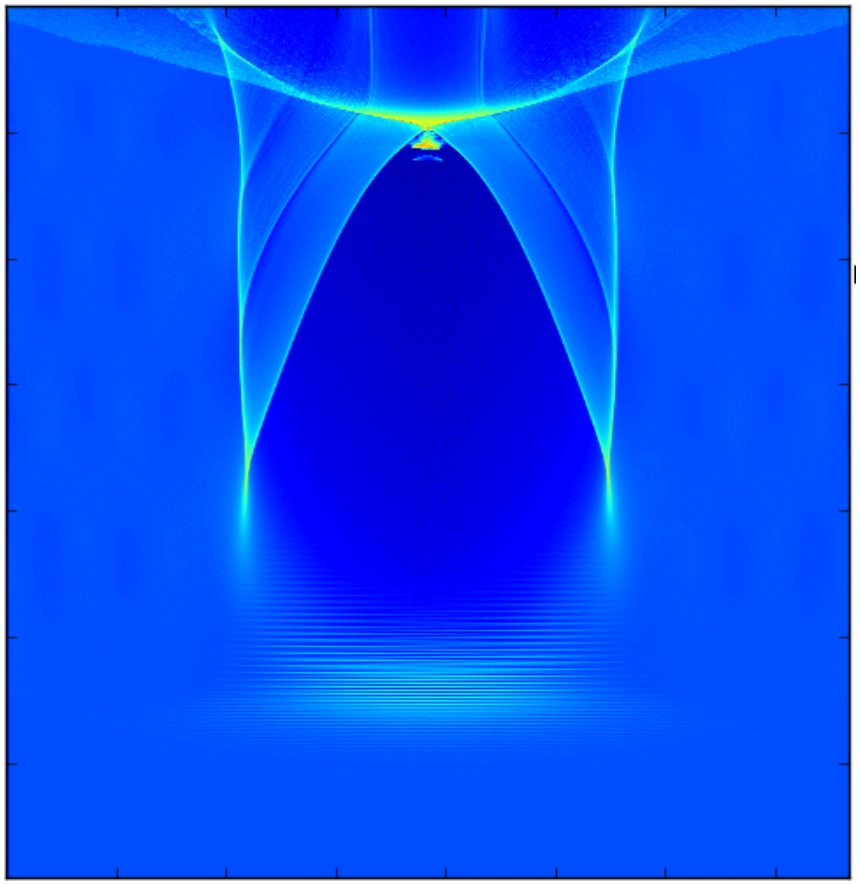
- Load = $N_{part} + \mathbf{L_{cell}} \times Ncell + \mathbf{L_{frozen}} \times N_{partfrozen}$
- **Capability** of each MPI process.
- **Balancing frequency.**

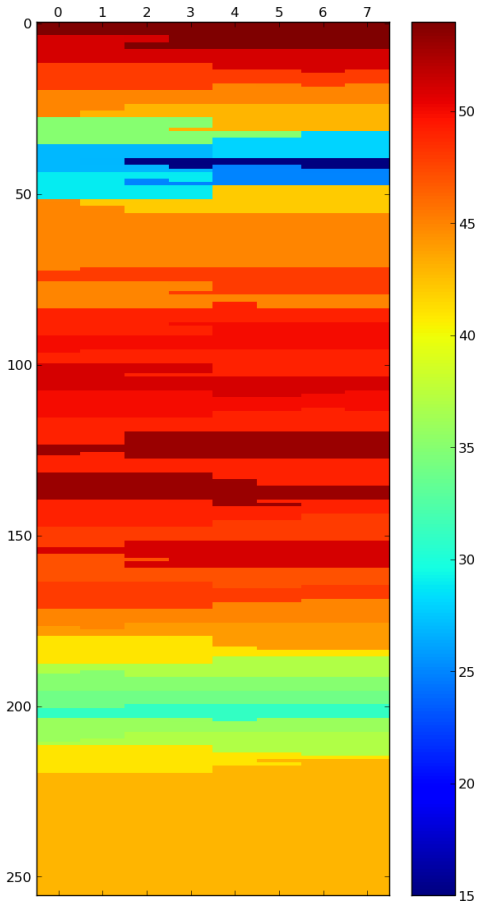
Typically we use $L_{cell} = 2 - 10$, $L_{frozen} = 0.1$, frequency=150 iterations and same capability for all nodes for the moment.

Partioning evolution

Electronic density

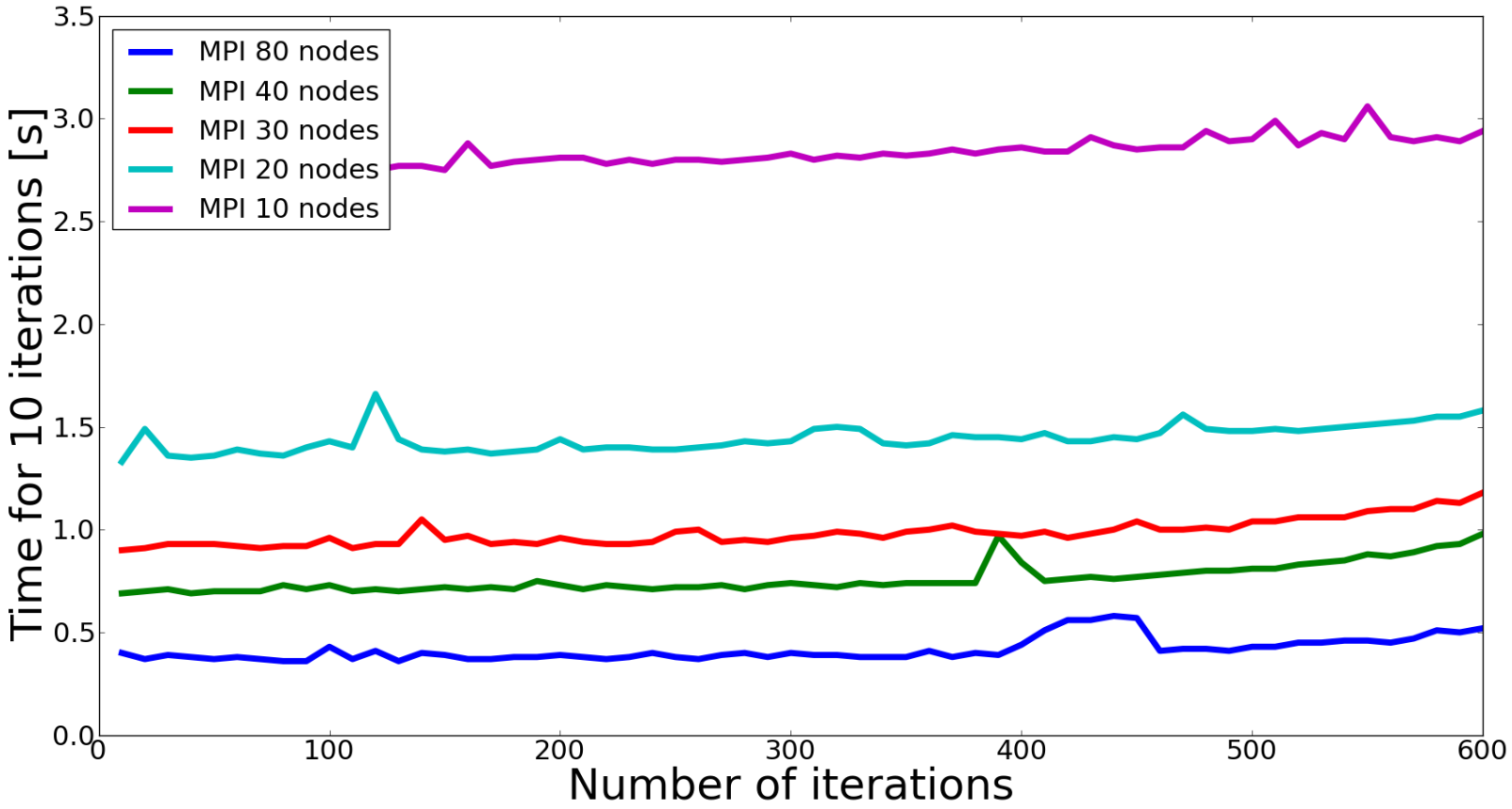
N_{patches} per MPI process





So you think you can scale ?

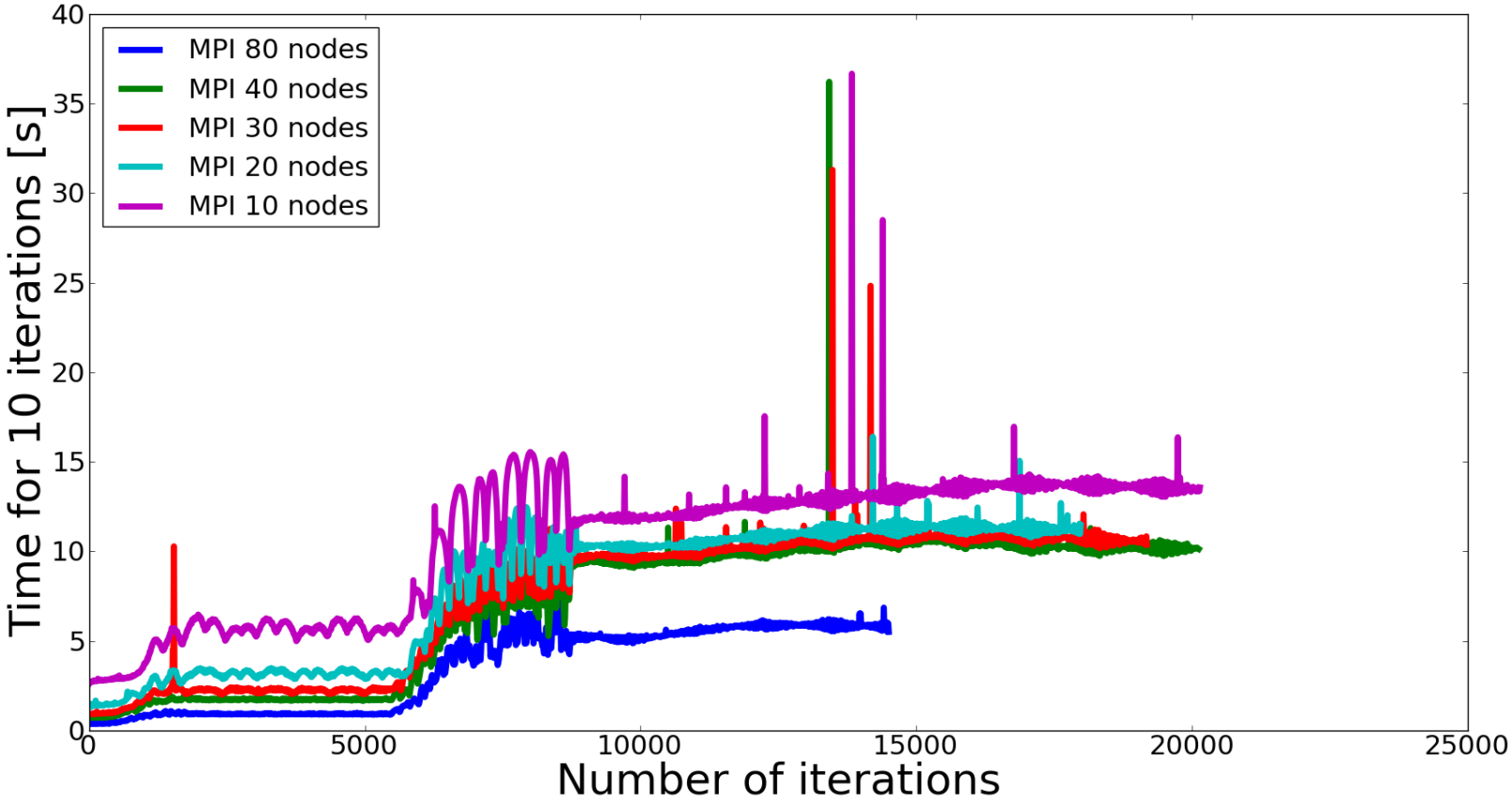
Strong scaling on a 10240×1280 cells domain. 24 MPI processes per node.



In trivial conditions, good strong scaling has been demonstrated up to 50k+ cores (OCCIGEN, system).

But for how long ?

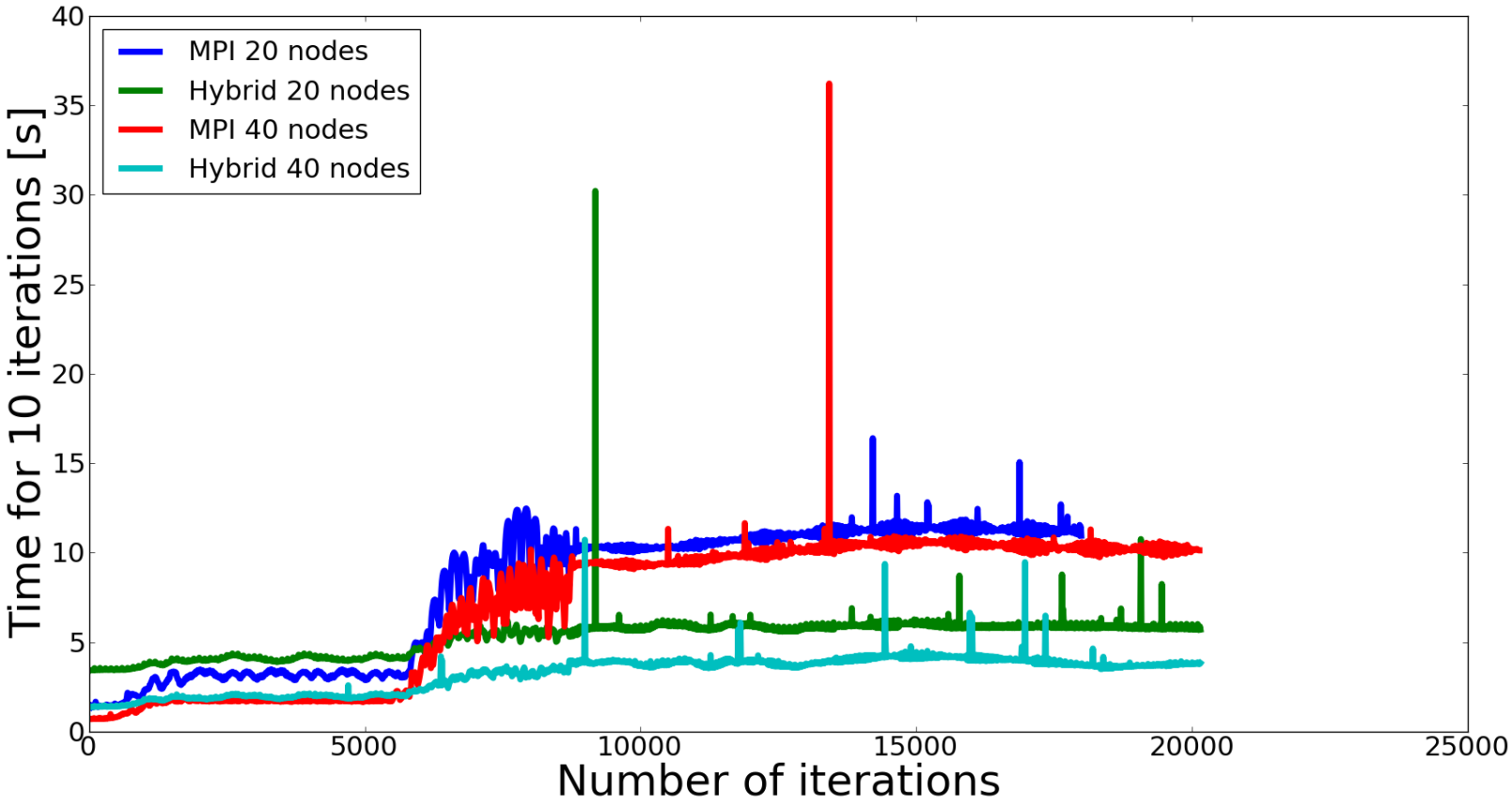
Strong scaling on a 10240×1280 cells domain. 24 MPI processes per node.



The hot spot is too small and you loose your scalability !

openMP to smooth out the load

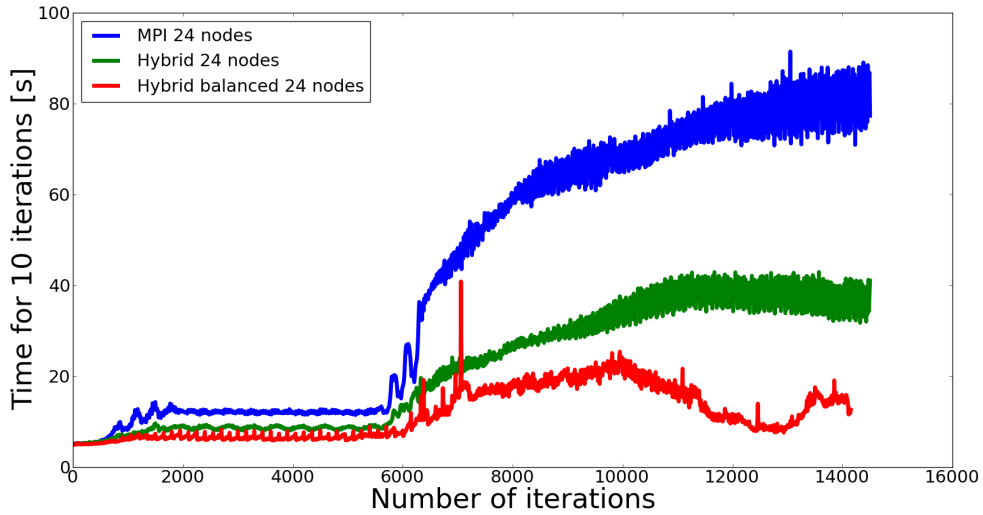
Strong scaling on a 10240 $\times$ 1280 cells domain.
2 MPI processes per node, 12 openMP threads.



The wonders of the dynamic scheduler of openMP.

Limits of openMP

Performances on a 6912×800 cells domain.
Increased imbalance by increasing the number of particles per cell
24 nodes, 2 MPI processes per node, 12 openMP threads.



The balancing power of openMP is limited by the number of threads.
The number of threads is limited by the number of cores per socket.

Still a factor 3 is lost. Fine tuning or smaller patches required ?

Alternatives exist

Other approaches for load balancing have been implemented in other codes.

For example, the split/merge technique is especially interesting even though it is disruptive. Significant advances in this direction have been made at the Niels Bohr Institute (Photon-Plasma code). OSIRIS is also capable of merging particles.

The good news is that different approaches can be complementary ! The perfect dynamic load balance technique is probably a mix of several approaches.

Credits

Smilei webpage:
http://www.maisondelasimulation.fr/projects/Smilei/html/index.html

Maison de la Simulation, Julien Derouillat

LULI, Mickael Grech, Tommaso Vinci, Frédéric Perez, Marco Chiaramello

IDRIS, Marie Fle

LLR, Arnaud Beck

Labex Plas@Par and PALM (through the SimPLE project, grant ANR-10-LABX-0039-PALM)