Software Architecture for parallel particle tracking with the distribution of large amount of data.

Florent BONNIER
3rd year PhD

Phd Advisor:Nahid EMAD (UVSQ/Li-PARAD)
Phd Coadvisor Xavier JUVIGNY (ONERA)

UNIVERSITÉ DE VERSAILLES ST-QUENTIN-EN-YVELINES
université PARIS-SACLAY

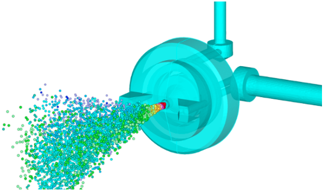
ONERA
THE FRENCH AEROSPACE LAB

retour sur innovation

Context: Tracking particles

- Simulations and video games,
- Track a very huge number of particles,

Several examples that use particle tracking in numerical simulations:



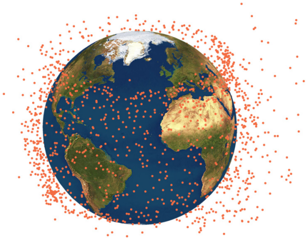
- sand grains flowing around a helicopter,
- gasoline drops in an engine and oozing on a surface,
- tracking chemical species in a fluid,
- objects in a solar system.

Methods to follow particles

- Independent particles,
- Sum all forces applied on particles.

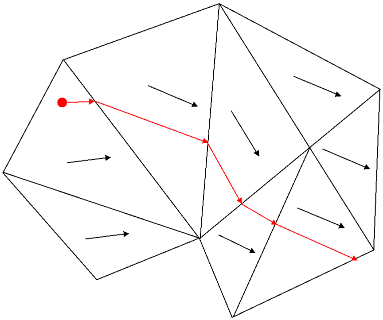
$$m_p \frac{du_p}{dt} = \sum F_p \qquad (1)$$

- multiple environment configuration and computation method depending on the problem (mesh, boundaries, ...).



Representation of space debris in Low Earth Orbit.

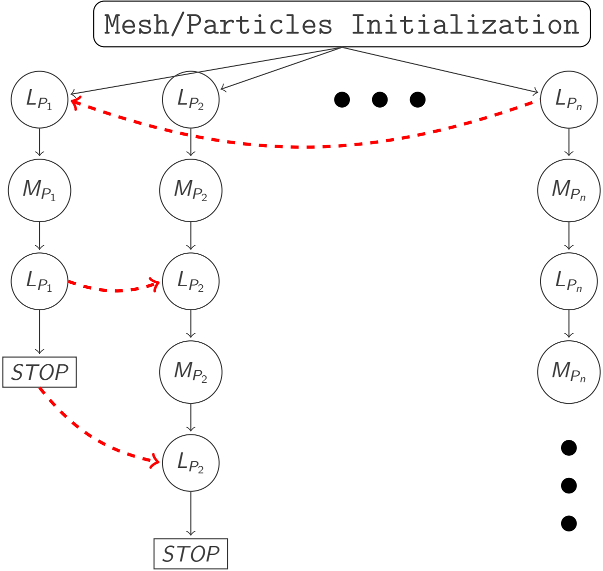
Path of a particle in triangularized mesh



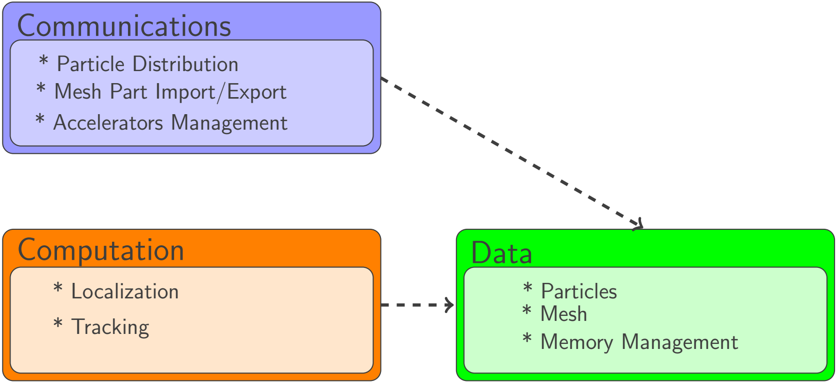
Context of the study:

- Huge partitioned meshes,
- passive tracers.

Graph of tasks



Graph of Components



Communications

* Particle Distribution
* Mesh Part Import/Export
* Accelerators Management

Computation

* Localization
* Tracking

Data

* Particles
* Mesh
* Memory Management

Data representation.

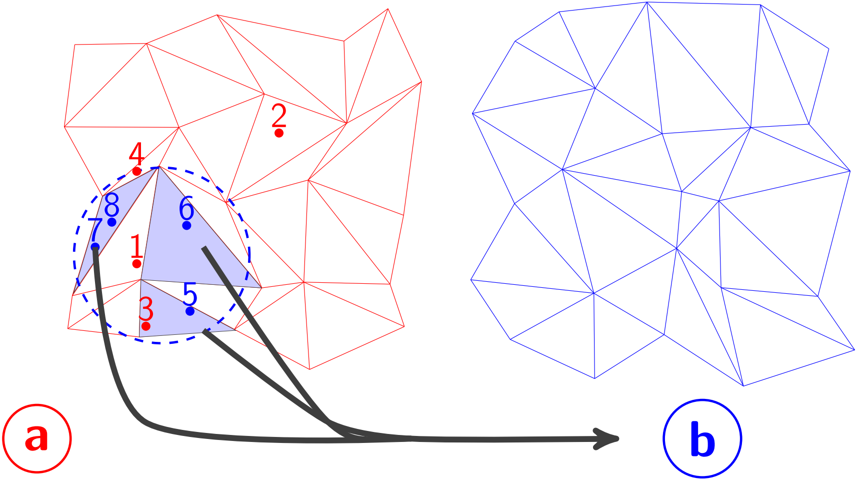
Global Lagrangian context initialized with a mesh representation:

- list of vertices,
- list of cells,
- list of faces and limit conditions,
- flowfield,
- ...

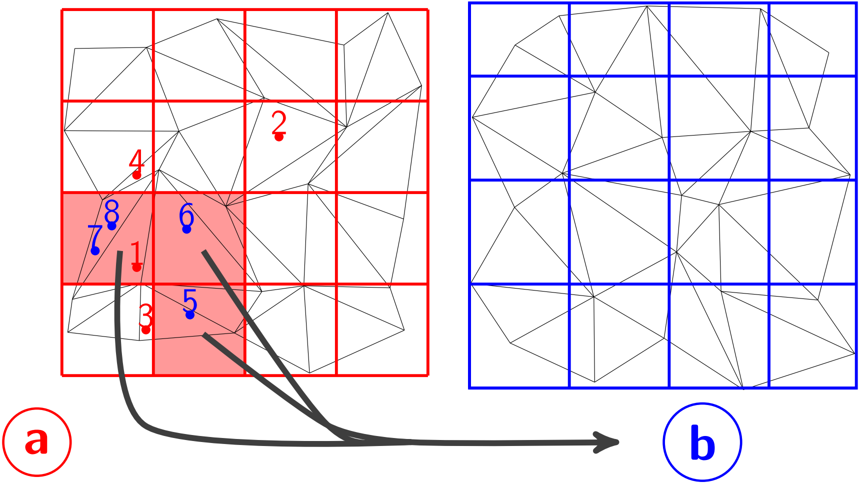
The data representation is based on usual compressed sparse matrix representation:

- data array,
- adress array.

Localize particles on remote memory spaces.



Localize particles on remote memory spaces.



Cells of the structured grid and data representation.

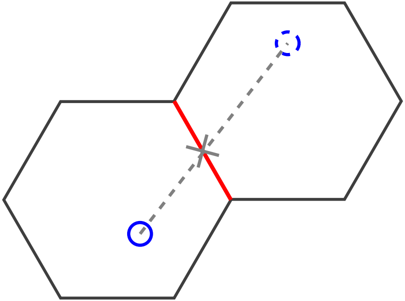
- Each structured cell is an independent mesh,
- ghost cells are initiated,
- structured cells are import/export from/to another process,
- data representation is the same of the computing mesh, so that the same functions are called to import single/multiple structured cells or mesh data,
- a structured cell is named with a unique 32 bits label (16 bits for the process ID and 16 bits for the original position in the process).

Localization during particle tracking.

Intersection Face/Ray computation:

- + allow to localize particles that move to another cell with 1 memory access,
- + fluid integration more precise (no cell skipped),
- - ray/plane intersection computation

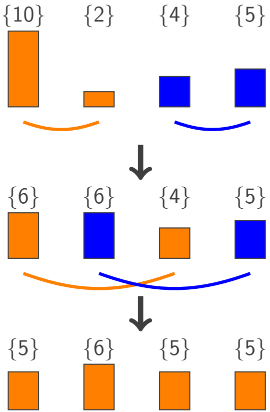
Algorithm used: Möller-Trumbore (2005)



Particle leaving a cell to its next localization.

Particles/Tasks Distribution – Original distribution.

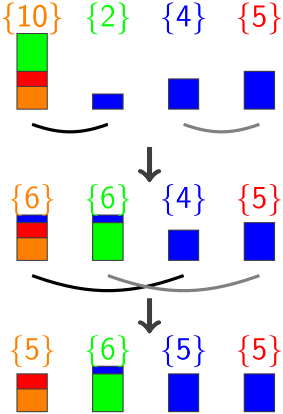
- Matthew J. O'Brien and Patrick S. Brantley algorithm (2013),
- load-balancing between 2 processes,
- all processes are well balanced in $\lceil log_2(nProcs) \rceil$,
- all processes recieve few messages.



Original particle distribution algorithm on 4 processes.

Particles and Tasks Distribution – Customized.

1. particles of the partner process are sent,
2. particles that are not localized in the current process or in one of the future partner are sent,
3. particles that are not localized in the current process are sent,
4. residual particles are sent.



Modified distribution algorithm on 4 processes.

Particle distribution comparison on 4 cores with 6.4 millions particles.

Static Distribution

	RANK0	RANK1	RANK2	RANK3
Loc. in R0	400681	399811	399899	401319
Loc. in R1	399787	401122	400260	399334
Loc. in R2	400059	399804	400171	399167
Loc. in R3	399473	399263	399670	400180

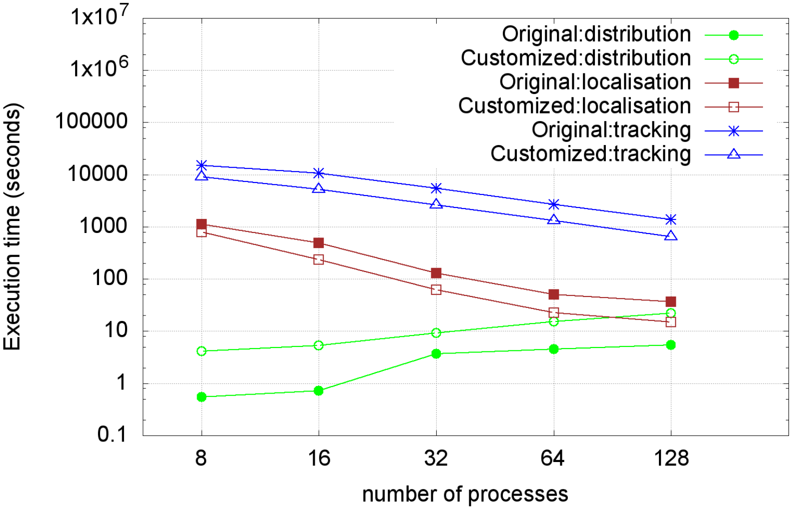
Customized Distribution

	RANK0	RANK1	RANK2	RANK3
Loc. in R0	1600000	0	1261	449
Loc. in R1	0	1600000	0	503
Loc. in R2	0	0	1598739	462
Loc. in R3	0	0		1598586

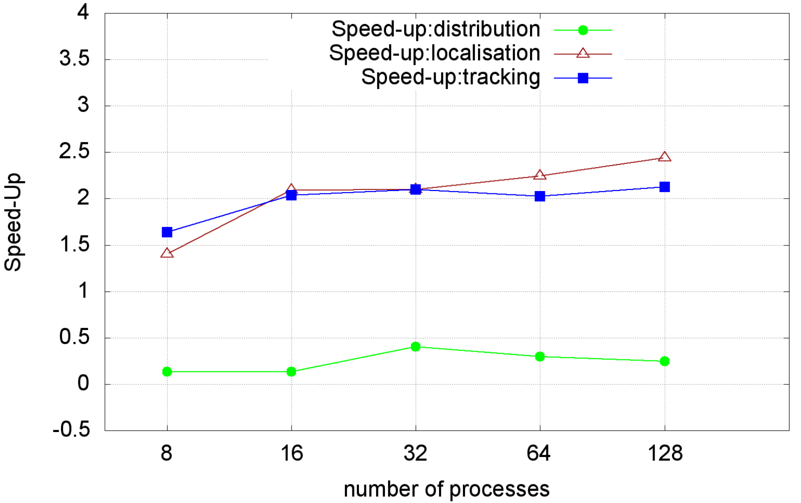
Experimental model

- Cubic mesh (941 192 cubic cells),
- Random initialization of 64×10^6 particles coordinates,
- Random directions,
- Compute particles/faces intersections,
- Particles rebound when they leave the cube,
- Mesh partitionned with Hilbert curve method,
- Set of Intel Xeon X5670 (2 sockets per node).

Distribution impact on different steps.



Speed-up of impact of the particle's distribution.



Conclusion

+ Efficient structures and algorithms are developped,
+ Architecture allows to manage data as needed by efficient algorithms,
+ More experiments are running today.

- Particle distribution is very efficient but need a lot of computation and can be greedy in computing time,
- Particle distribution algorithm works only with a number of processes equal to 2^n.

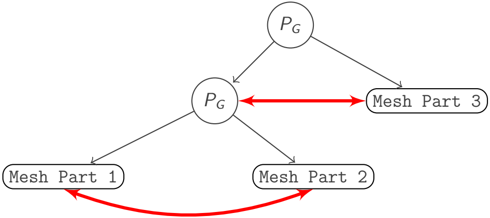
Future Work

- Implement Process grouping,
- Add GPU/Xeon-Phi management functions to components,
- Add and evaluate particle renumbering methods based on particle localization.

Thank you for your attention.

Process grouping.

- Exascale challenge,
- Use multiple level of processes,
- Optimize particle and mesh partitions proximity.


P_G
P_G
Mesh Part 3
Mesh Part 1
Mesh Part 2