Astrophysics Simulations on multi-GPU: FleCSPHg

Julien Loiseau, François Alin, Christophe Jaillet, Arnaud Renard, Michaël Krajecki
University of Reims Champagne-Ardenne, CReSTIC Laboratory EA3804

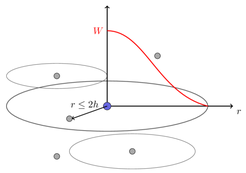
JCAD 2018 - Journées Calcul & Données (Poster initially presented at GTC Europe 2018)

HPC for astrophysics simulations

The recent results of the Laser Interferometer Gravitational-Wave Observatory (LIGO) show the real need of domain scientists for more reliable and complex simulations for Astrophysics events. Those simulations are usually based on grid methods that perfectly suit the regular GPU architectures. We show that tree based method like SPH can also be a very good candidate for multi-GPU computing even with distributed scheme.
FleCSPH is a SPH simulation framework initially developed at the Los Alamos National Laboratory and part of the Los Alamos Ristra Project along with FleCSI.
This study aims to provide a multi-GPU accelerated version of FleCSPH.
This poster presents the difficulties involved with the Smoothed Particles Hydrodynamics method and the gravitation computation based on the Fast Multipole Method.

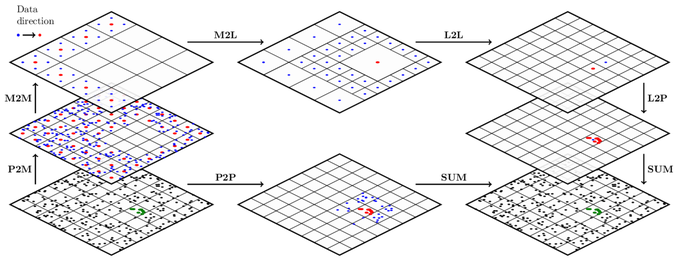
SPH + Gravitation

The Smoothed Particles Hydrodynamics problem is a Lagrangian method used in many fields for particles based simulations. Every particle is associated with a smoothing length h representing its volume/area of action in addition with a smoothing kernel W.



In this study we implemented astrophysical events. We worked on binary neutron stars merger. This requires to compute density, hydrodynamical forces, internal energy and also adapting timestep using the SPH method.
In order to perform astrophysics simulations we need to consider self-gravity. This is based on the Fast Multipole Method: approximations based on a Multipole Acceptance Criterion angle are extended locally using a Taylor Serie. This algorithm is then decomposed in several steps with:

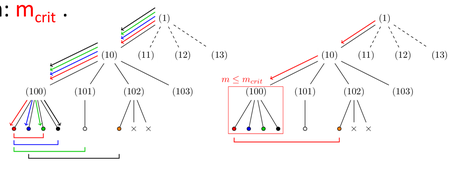
P2M -> M2M -> M2L -> L2L -> L2P and P2P.



Those two methods imply:

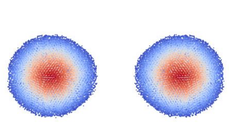
- Efficient load balancing method
- Efficient domain distribution: Morton/Hilbert Ordering
- Reliable and efficient framework for tree construction and search

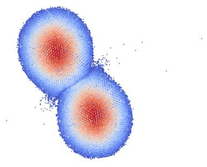
The initial FleCSPH framework is fully distributed and parallel on classical multi-core architectures. The most time consuming step, the tree search, is our target. The current version is already optimized and target groups of particles and compute interaction list for a group of particles based on particles mass with a criterion: m_{crit} .

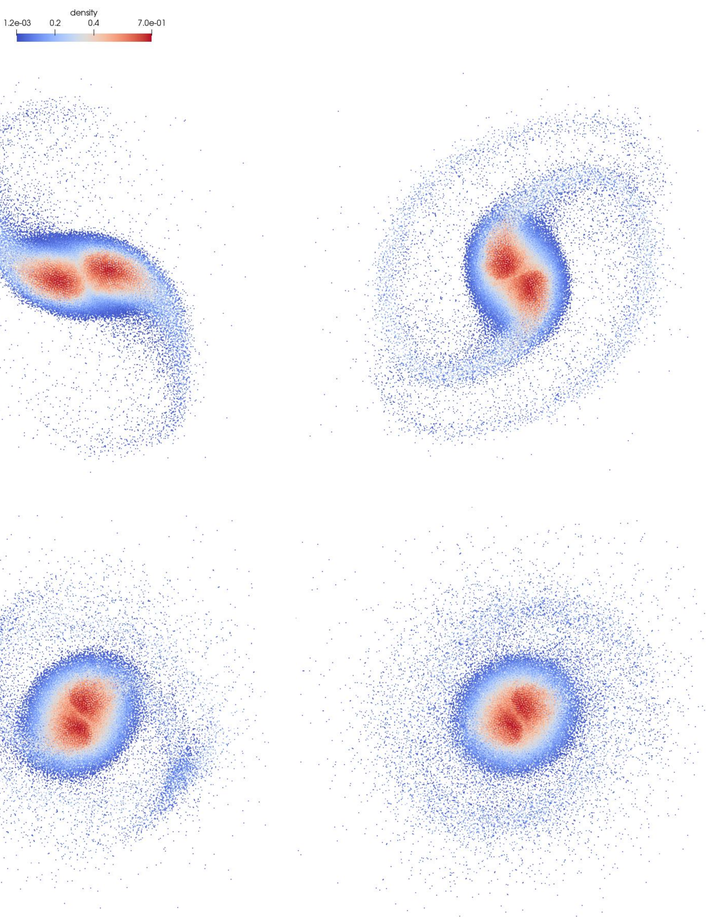


Binary Neutron Stars Coalescence

In order to benchmark our multi-GPU version on both SPH and gravitation using FMM we choose an astrophysics simulation: binary neutron star coalescence. This example provides a high number of particles and cover all aspects of both SPH+FMM.
The simulation presented here have been computed on FleCSPH using our multi-GPU approach. This simulation is based on two stars composed of 40000 particles total and ran on 8 nodes of the NVIDIA's GPU accelerated ROMEO supercomputer center at the University of Reims Champagne-Ardenne.
The SPH method itself and the reliability of the tree search make that the mass, energy, linear and angular momentum are conserved.



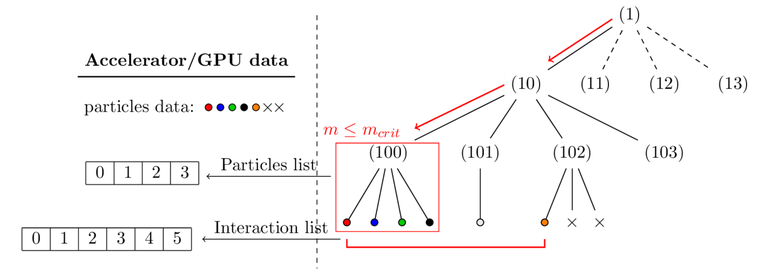




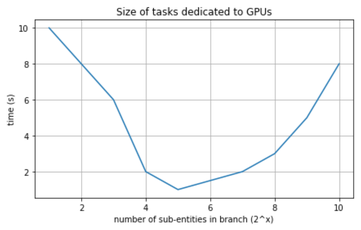
See the simulation on YouTube

FleCSPH on Multi-GPU

The multi-GPU implementation has been developed in addition to the original FleCSPH code. We targeted the most time consuming step, the tree traversal.
Our multi-GPU strategy is based on this scheme:
While the CPU prepares another group of particles and interaction lists the GPU computes the physics part: the one requiring more computational power.



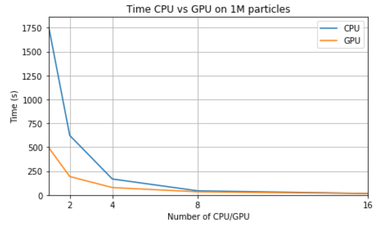
In order to get the best work between the CPUs and GPUs we need to define empirically the tree depth dedicated to the task generation. If the number of tasks is too high regarding the computation itself, there is not enough work for the GPU. At the opposite if the tasks number is too high we fall into the $O(N^2)$ computation.

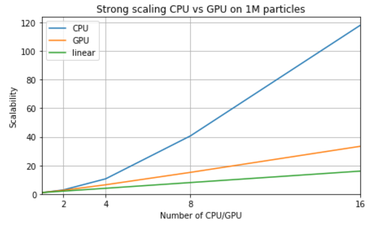


This graph shows our empirical tests for the depth distribution.
Each task on the GPU is done using its own data and computation stream.
When all the kernels are finished the particles are gathered on the host for data exchange between the different processes.

Results

All the results presented here have been computed on the ROMEO supercomputer of the university of Reims Champagne-Ardenne.





This approach gives us an acceleration of up to 3.5x faster using up to 16 K20Xm NVIDIA GPU in addition to the classical processors, E5-2650v2.
The scalability is presented on the right side and gives an hyper-scalability up to 16 GPUs. The scalability of the CPU version is better but the overall time is lower on the accelerated version.
For those tests the number of particles was 1 Million and we are working to solve simulation with billions of bodies.

We would like to thank the ROMEO supercomputer center at the University of Reims Champagne Ardennes.