INF560 Project : Particle interaction

...

Algorithms and sources of parallelism

2 main algorithms:

- bruteforce : computes the interactions between each pair of particle
  - exact but execution time : $O(n^2)$

- Barnes-Hut algorithm : approximation of the force when particles are far enough
  - spread the particles in a quad-tree structure
  - when a particle is far enough from a node, only computes the interaction between this particle and the center mass of the node

Algorithms and sources of parallelism

Constraint : requires to update the position of each particle at each step

- data parallelism : suits perfectly
  - multiple similar data (particles information)
  - one task to perform on these data
- task parallelism and flow parallelism : not so much
  - tasks depends on all the previous results

Brute-force

MPI + OpenMP

- MPI alone :

Particles to compute forces on are fairly shared between MPI tasks

task 1	task 2	task 3	task 4

MPI + OpenMP

- MPI + OpenMP : two options
- share particles between OpenMP tasks

task 1		task 2		task 3		task 4	
1.1	1.2	2.1	2.2	3.1	3.2	4.1	4.2

MPI + OpenMP

- MPI + OpenMP : two options
- share force computation between OpenMP tasks

task 1	task 2	task 3	task 4
1.1	2.1	3.1	4.1
1.2	2.2	3.2	4.2

MPI + OpenMP

Comparison of these two approaches : the first one is faster by 25% with 10 000 particles and 8 OpenMP tasks

Reasons :

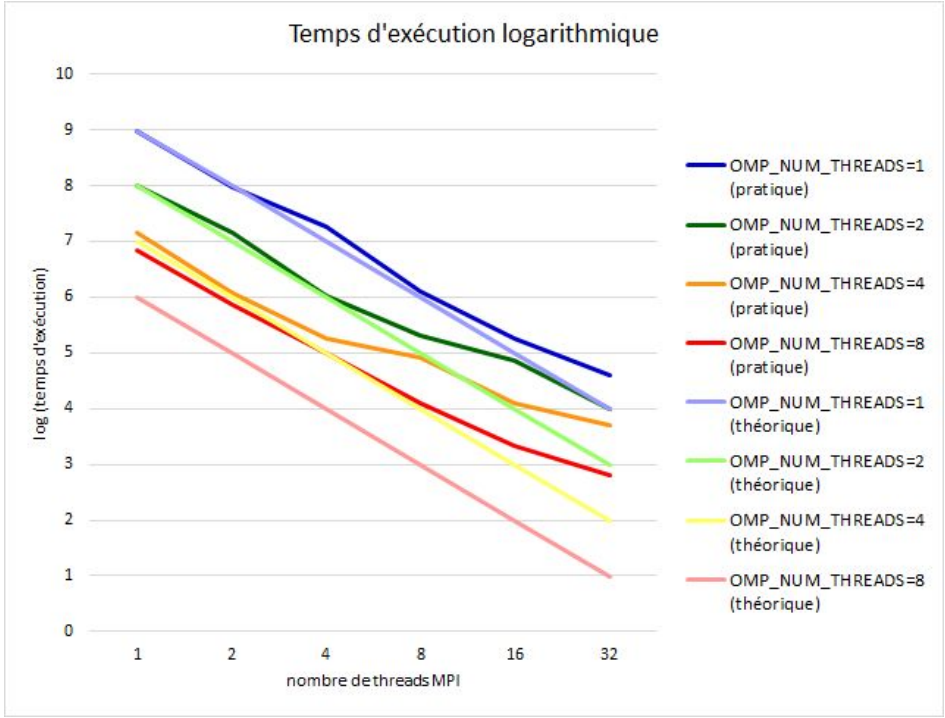
- more data tranfert and use of reduction operation in the second approach
- requires synchronization for each particle

So we chose the first one

MPI + OpenMP

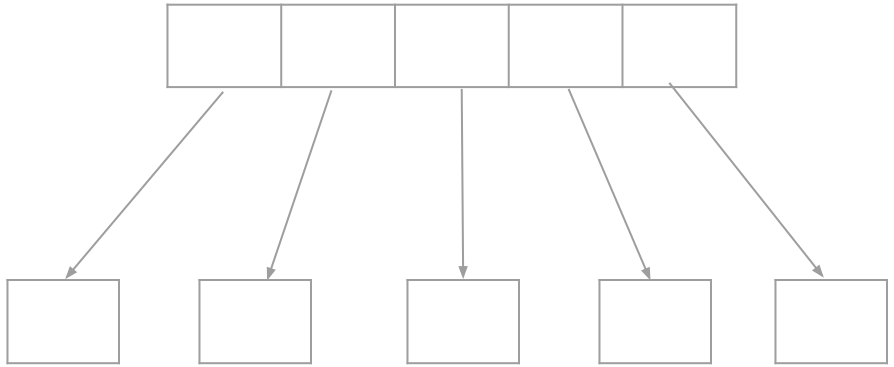
- Performances with 10 000 particles :

Task number		OpenMP			
		1	2	4	8
MPI	1	507	256	143	114
	2	249	143	67	58
	4	154	65	38	32
	8	68	40	30	17
	16	38	29	17	10
	32	24	16	13	7



CUDA + MPI

- CUDA

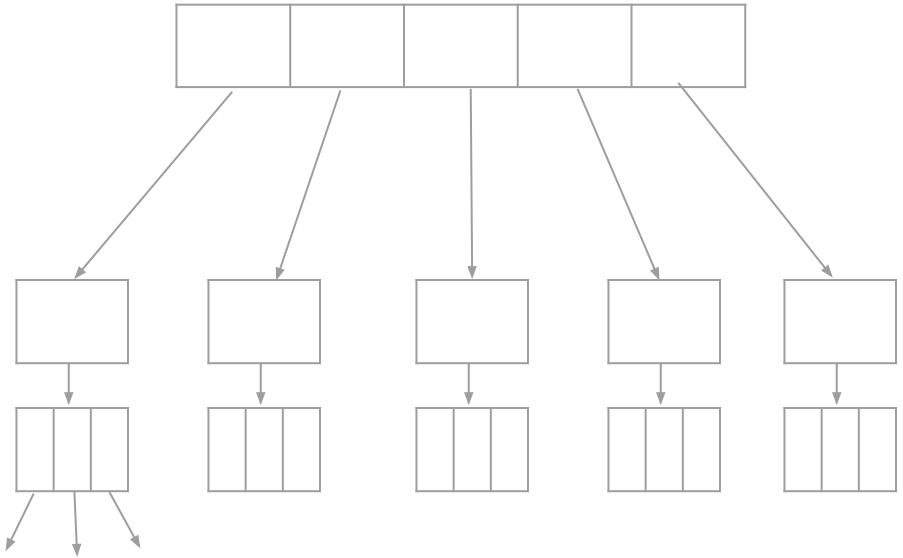


Particles to calculate forces on are shared between CUDA threads

CUDA + MPI

- CUDA

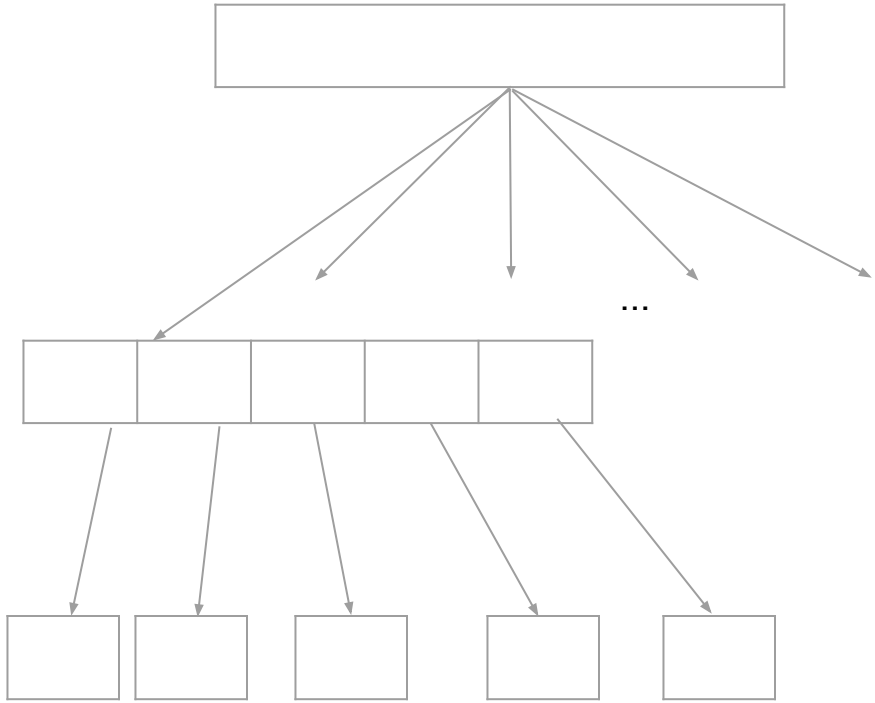
- CUDA + MPI (V1)



Particles to calculate forces on are shared between MPI tasks, and then between CUDA threads

CUDA + MPI

- CUDA
- CUDA + MPI (V1)
- CUDA + MPI (V2)



Particles to calculate forces from are shared between MPI tasks, and particles to calculate forces on between CUDA threads

CUDA + MPI

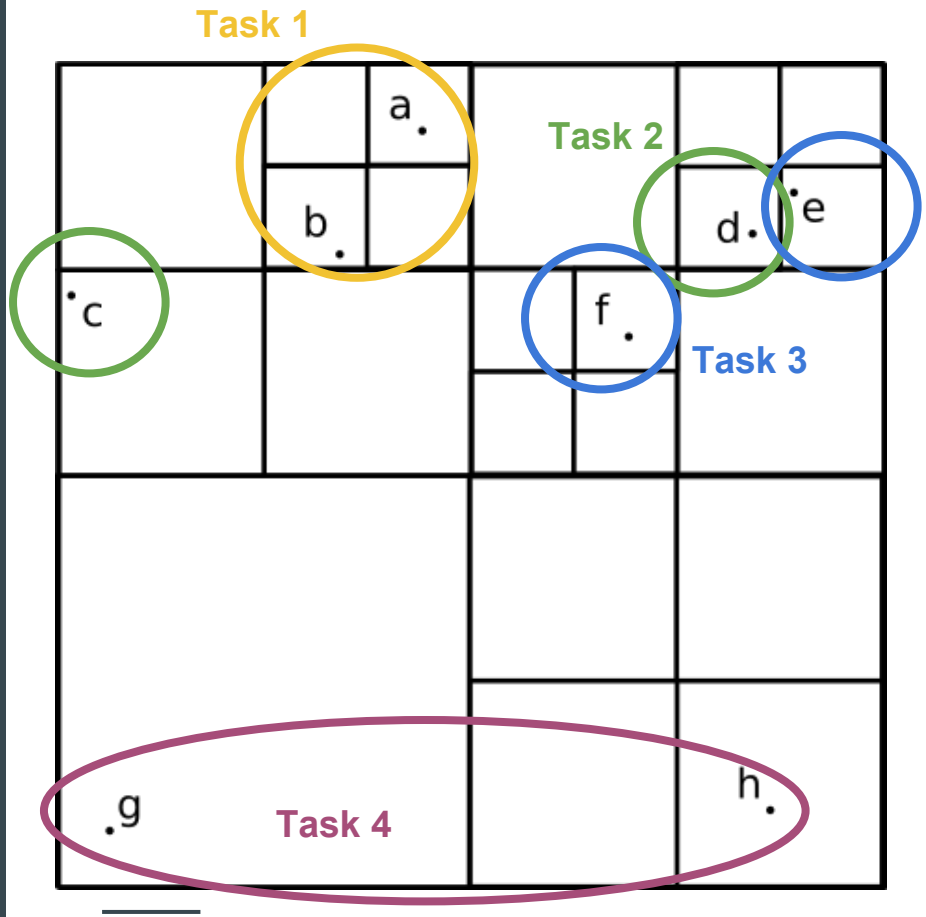
- CUDA
- CUDA + MPI (V1)
- CUDA + MPI (V2)

nparticles	1000	10 000	50 000
CUDA	0,72	77	4012
V1	0,84	78	275
V2	8,98	24	232

Barnes-Hut algorithm

MPI

No clear speed-up



Barnes-Hut algorithm

More approximations

->

More parallelization ?

Demo !

10’000 particles

MPI + OpenMP (32, 8)

CUDA

CUDA + MPI V2 (25)

Sequential and parallel Barnes-Hut algorithm