Simulating Zoom-In initial conditions of cosmological simulations on GPUs

Antonio Ragagnin

With: K. Dolag, M. Wagner, C. Gheller,
C. Roffler, D. Goz, D. Hubber, A. Arth

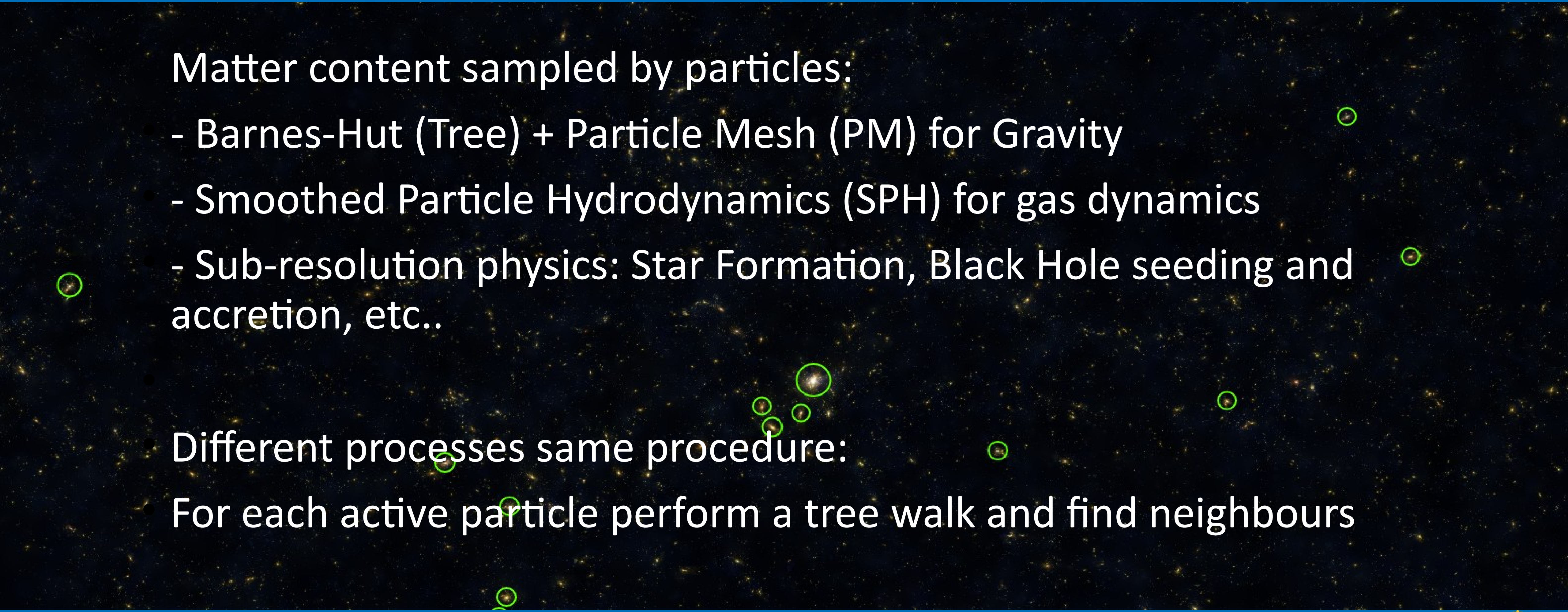
INAF - OATs

Project ID: 754337

Table of Content

- Intro: Gadget code and its challenges with accelerators
- The porting: Our asynchronous approach
- Tests on cosmological simulations
- Preliminary tests on Zoom-in simulations

Intro (pt. 1/4)
N-body simulations with Gadget3



Matter content sampled by particles:

- Barnes-Hut (Tree) + Particle Mesh (PM) for Gravity

- Smoothed Particle Hydrodynamics (SPH) for gas dynamics

- Sub-resolution physics: Star Formation, Black Hole seeding and accretion, etc..

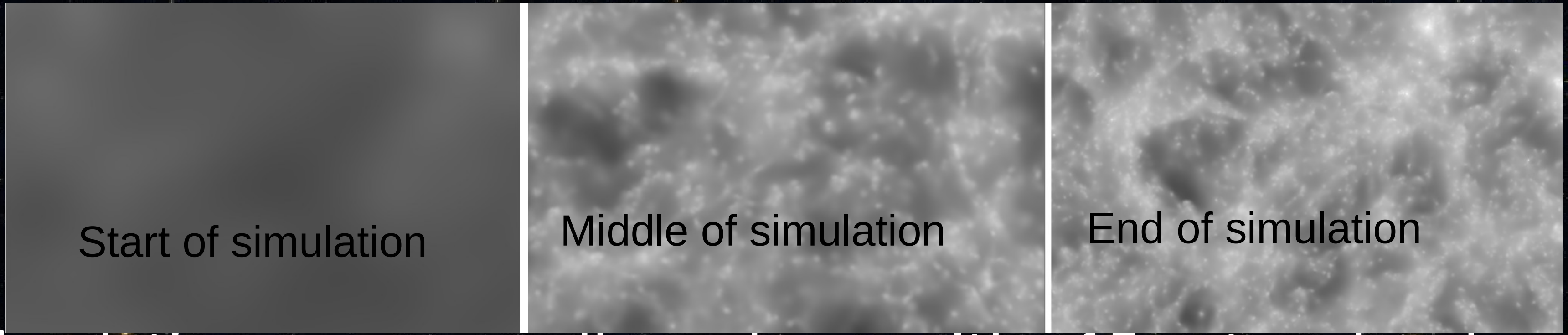
Different processes same procedure:

For each active particle perform a tree walk and find neighbours

Project ID: 754337

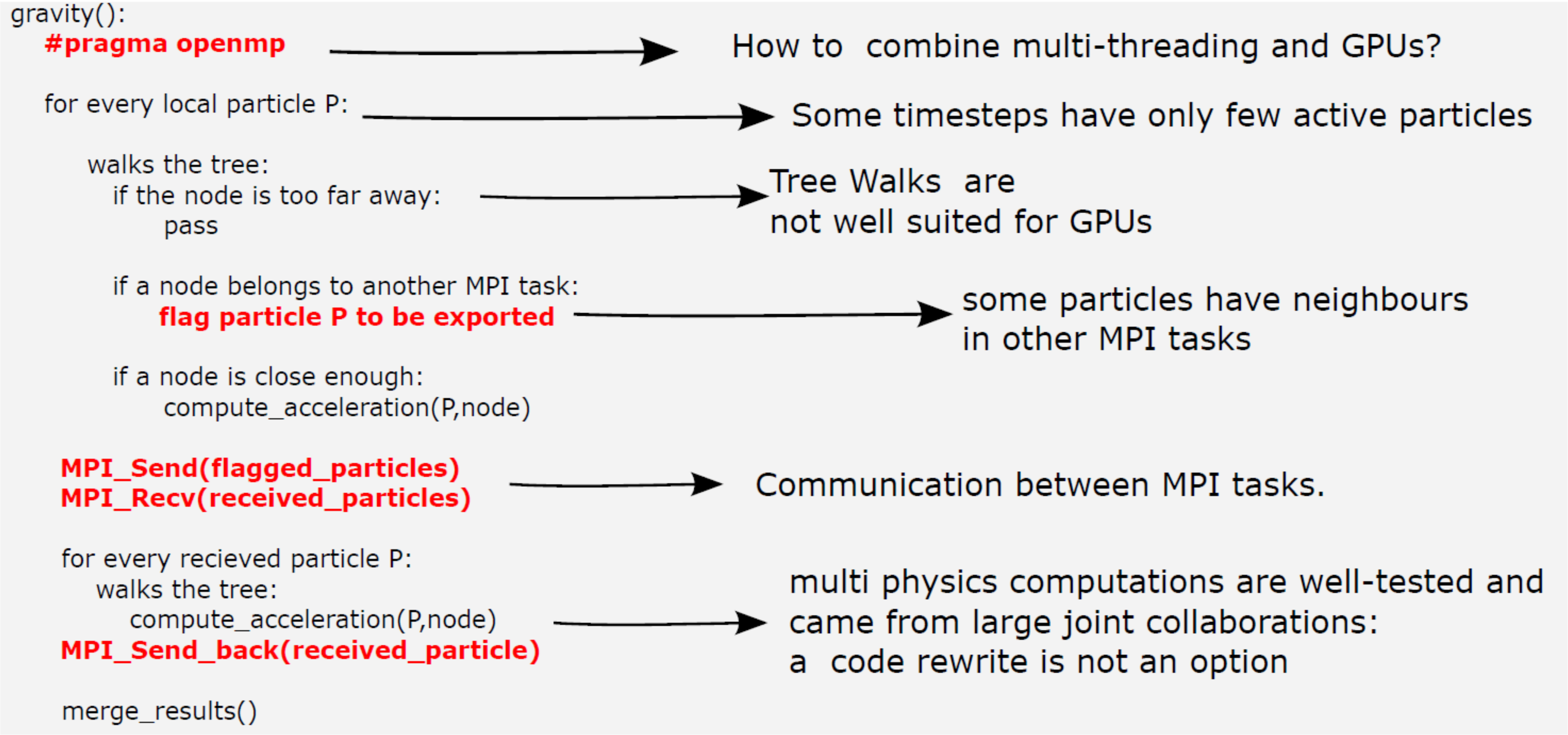
Intro (pt. 2/4)
Initial Conditions

Cosmological simulations cover large volumes with ~<10 octree levels



Zoom-in simulations cover smaller volume with ~15 octree levels

Intro (pt. 3/4)
Gadget Kernel evaluation



```
gravity():
   #pragma openmp

   for every local particle P:

      walks the tree:
         if the node is too far away:
               pass

         if a node belongs to another MPI task:
               flag particle P to be exported

         if a node is close enough:
               compute_acceleration(P,node)

    MPI_Send(flagged_particles)
    MPI_Recv(received_particles)

    for every recieved particle P:
       walks the tree:
          compute_acceleration(P,node)
    MPI_Send_back(received_particle)

    merge_results()
```

How to combine multi-threading and GPUs?

Some timesteps have only few active particles

Tree Walks are
not well suited for GPUs

some particles have neighbours
in other MPI tasks

Communication between MPI tasks.

multi physics computations are well-tested and
came from large joint collaborations:
a code rewrite is not an option

Intro (pt. 4/4): Challenges in accelerating the code

- vectorisation and cache misses (500B/structure)

- blocking MPI communications

- Time-steps with too few active particles won't fully exploit GPU parallelism

- Thread-locking operations at each tree walk

- GPUs memories have less capacity than their host memories

→ keeping all data in GPUs requires more computing nodes

- decennial effort of developers

Problem (pt. 1/4): GPU memory smaller than host memory

- Magneticum Box0/mr (see LRZ extreme scaling) has
  - 1.2 10^7 particles per node,
  - each node was allocating 4GB for the Barnes Hut tree,
  - 22GB for the basic quantities used in gravity
  - and additional 14GB for the SPH-only
- **Solution: brings one module per time to GPU and push/pull only desidered in/out properties**

Problem (pt. 2/4): Adaptive timesteps do not perform

- Particles in void regions evolve with large timesteps
- After nearly half of the simulation time, it is very common to have time-bins with only one or very few active particles.
- time-steps with such a low amount of active particles won't benefit from the single instruction multiple thread (SIMT) paradigm of GPUs,
- **we decided to keep small timebins (with less than a given threshold N_min = 10^4 active particles) to run on the CPU only.**

Problem (pt. 3/4): blocking MPI

- **We decided to overlap the CPU and the GPU computation** in the following way:
  - while the GPU loops over the active particles and computes local interactions,
  - the CPU takes care of walking the tree for each active particle in order to perform all MPI send/receive of guest particles.

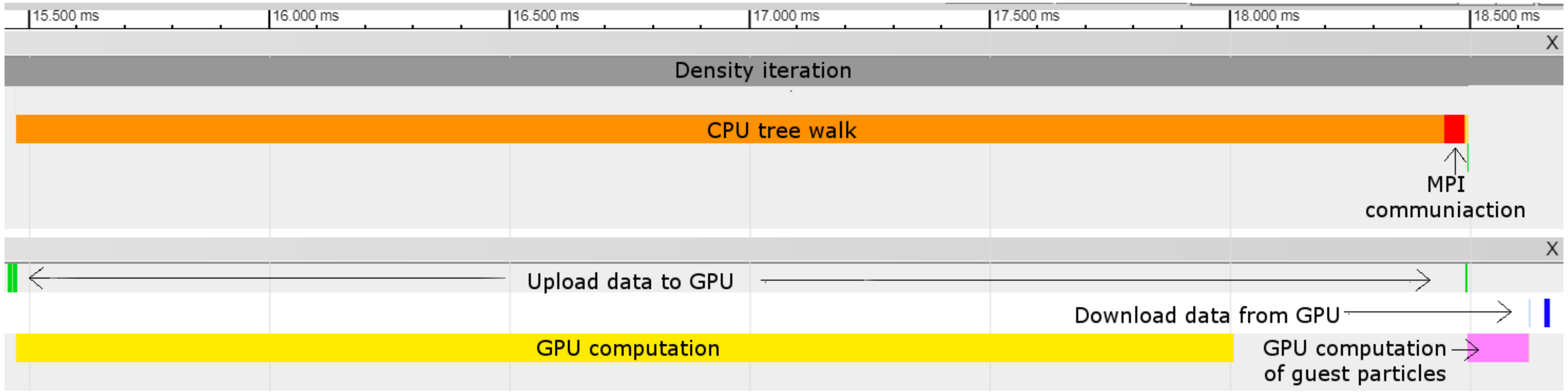
Problem (pt. 4/4): all neughbours may not fit GPU memory

- In SPH it is not well known a-priori the amount of neighbours of a given SPH particle (especially in zoom-in simulations).
- **Solution: GPU treewalks performed in chunks of N_chunk=32 neighbours.**

Tests (pt. 1/5):
One density iteration

- CPU version: 1 Power9 node, with 4 sockets each with 10 threads
- GPU version: same with two Tesla V100 GPU + NVLink
- Iteration time:
  - GPU:3.2s,
  - CPU: 11.4s

GPU profiling (NVTools):

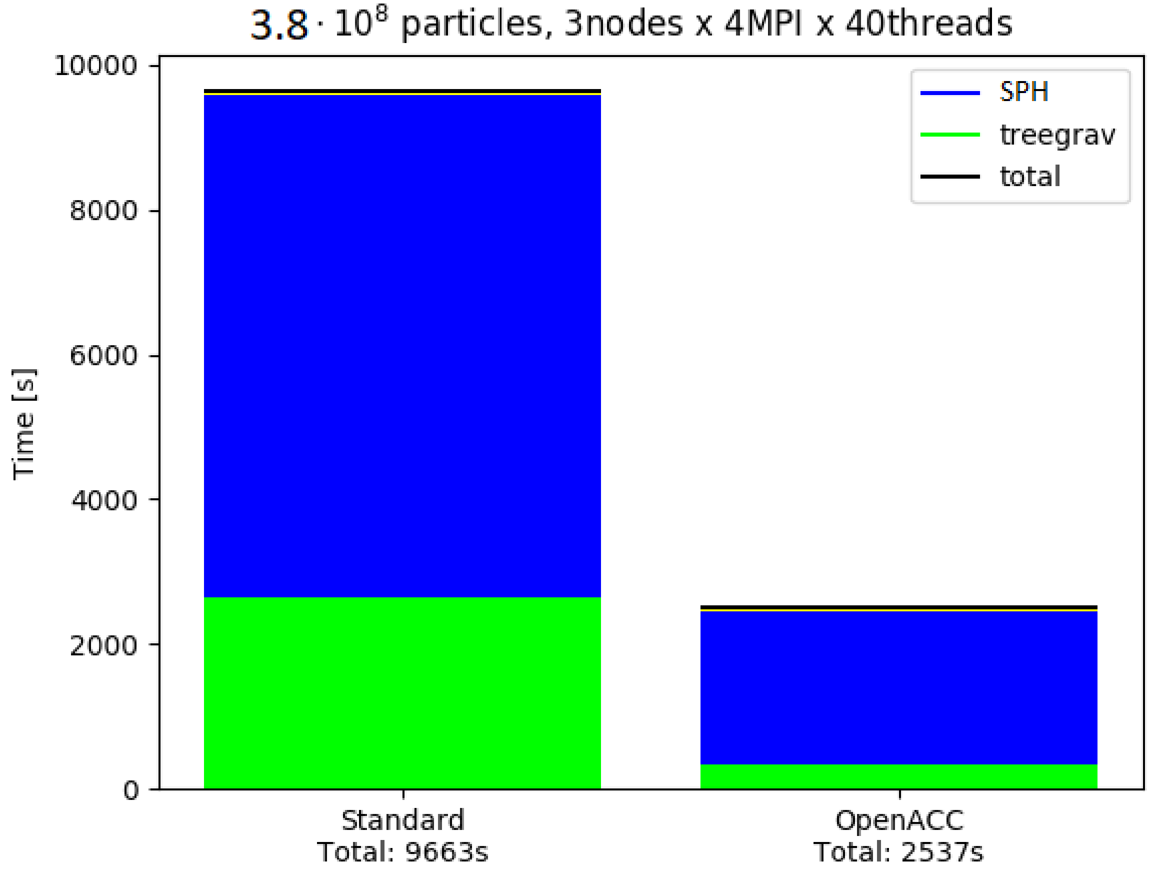


Problem (pt. 2/5): Unified memory

- Unified Memory (with PGI compiler) gives almost same computing time,
- plus you do not take care of memory movements!

Problem (pt. 3/5): one full timestep

- GPU: 1600s
- CPU: 5400s (with a speedup of 3.3)

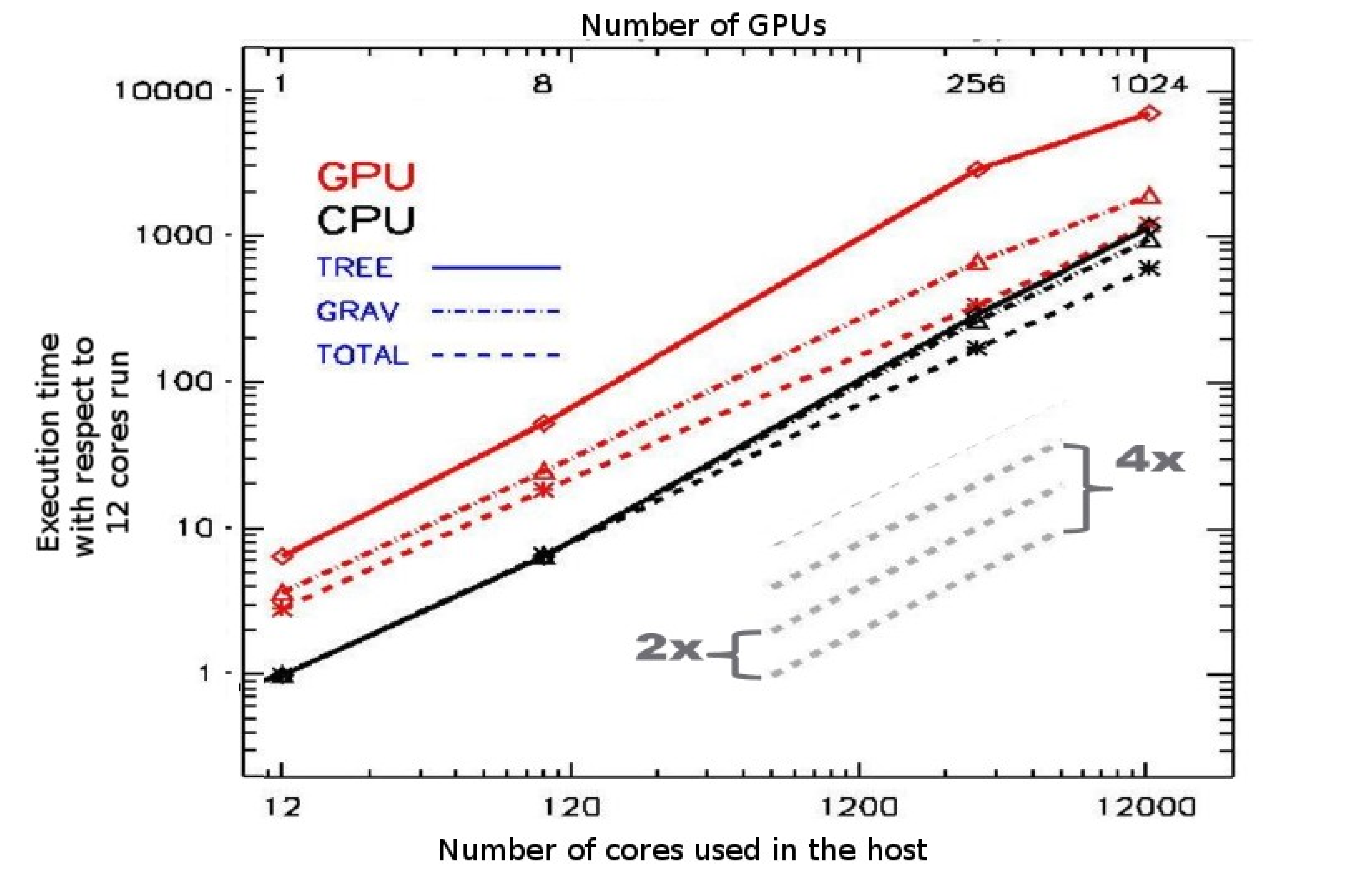


Project ID: 754337

Problem (pt. 4/5): full run

- On Piz Daint
  - CPU: 8 MPI tasks, each with 8 OpenMP threads
  - GPU: same with 8 GPU P100 + PCI express
- Speedup:
  - Barnes-Hut speedup: 1.8
  - SPH speedup: 2.6
  - Thermal conduction speedup: 3.0
  - Total speedup: 2.1

Problem (pt. 5/5): Gravity scaling @ EuroHack2017



Zoom-in sims (pt. 1/3): Difference between cosmo sim

- Compared to cosmological simulations they go from ~10 to 15 octree level
- Will this additional treewalk overhead slow down?

EUROEXA FUNDED BY THE EUROPEAN UNION

Zoom-in sims (pt. 2/3): 10x results

- Testcase with with 6 10e7 particles of small volume with 10x resolution on 64MPI ranks

- Total speedup: 1.65
- gravity speedup: 1.4
- and SPH speedup: 1.6

Code spends 33% of time in serial parts (tree build, domain decomposition, sf, PM)

Zoom-in sims (pt. 3/3): 250x DM only results

- Testcase with with 1.3 x 10e9 particles of small volume with 250x resolution on 128MPI ranks

- Total speedup: 1.4
- gravity speedup: 3 ← because of many particles per nodes

New bottleneck in zoom-in cosmological simulations on GPUs: Particle-Mesh (done fftw)

Conclusions

- We offload to the GPU only one module per time (to maximize the number of particle per each host);
- We overlap GPU and CPU computations (as the CPU takes care of neighbour exchanges).
- Unified Memory is good (no obvious since with OpenAcc one tune copy)
- Total speedup ~2 on various configurations of cosmological simulations
- Zoom-in simulations can reach speedup of 2 on ported-modules. But Particle Mesh dominates execution time → porting it to GPUs at next EuroHack?

Many Thanks
Please Connect at
antonio.ragagnin@inaf.it

Antonio Ragagnin

INAF, OATs

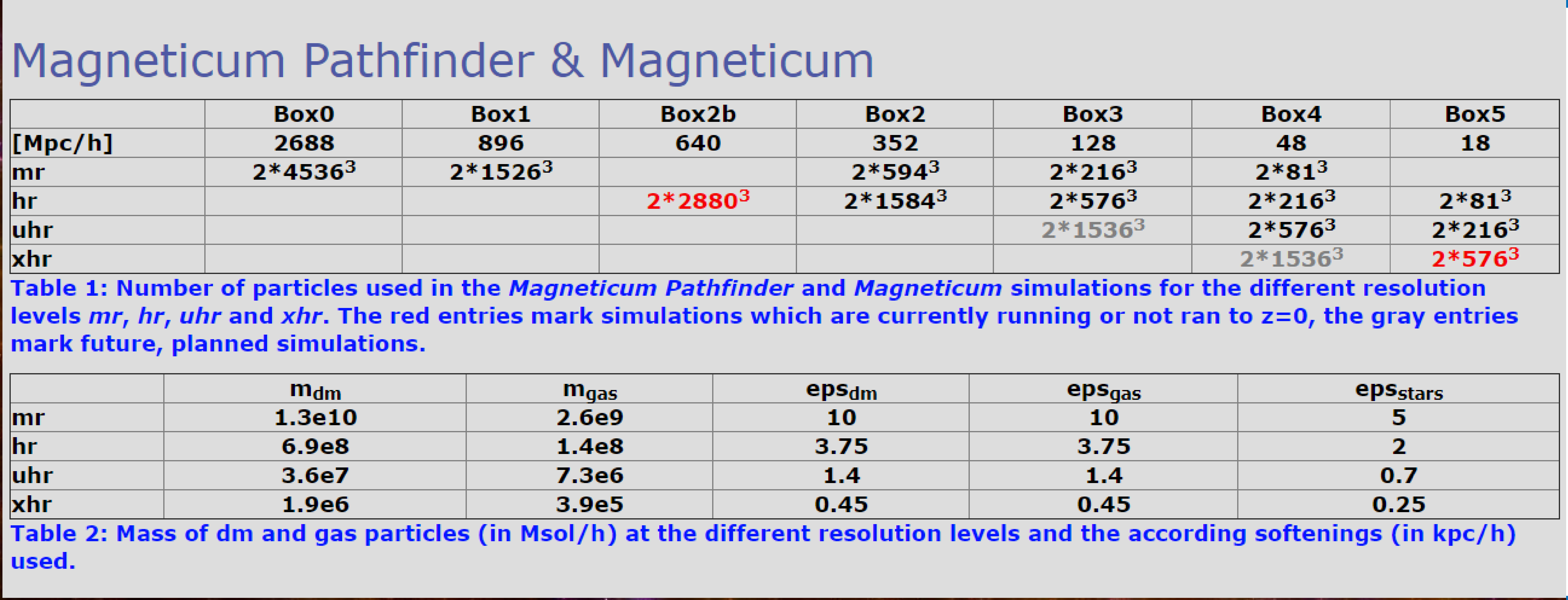
EuroEXA

Project ID: 754337

Conclusions

Backup slides

Magneticum project (PI Klaus Dolag)



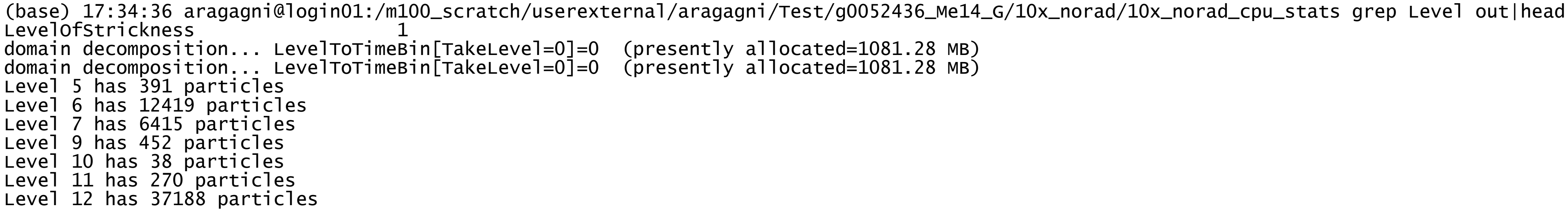
Magneticum Pathfinder & Magneticum

	Box0	Box1	Box2b	Box2	Box3	Box4	Box5
[Mpc/h]	2688	896	640	352	128	48	18
mr	$2*4536^3$	$2*1526^3$		$2*594^3$	$2*216^3$	$2*81^3$	
hr			$2*2880^3$	$2*1584^3$	$2*576^3$	$2*216^3$	$2*81^3$
uhr					$2*1536^3$	$2*576^3$	$2*216^3$
xhr						$2*1536^3$	$2*576^3$

Table 1: Number of particles used in the *Magneticum Pathfinder* and *Magneticum* simulations for the different resolution levels *mr*, *hr*, *uhr* and *xhr*. The red entries mark simulations which are currently running or not ran to z=0, the gray entries mark future, planned simulations.

	m_{dm}	m_{gas}	eps_{dm}	eps_{gas}	eps_{stars}
mr	1.3e10	2.6e9	10	10	5
hr	6.9e8	1.4e8	3.75	3.75	2
uhr	3.6e7	7.3e6	1.4	1.4	0.7
xhr	1.9e6	3.9e5	0.45	0.45	0.25

Table 2: Mass of dm and gas particles (in Msol/h) at the different resolution levels and the according softenings (in kpc/h) used.

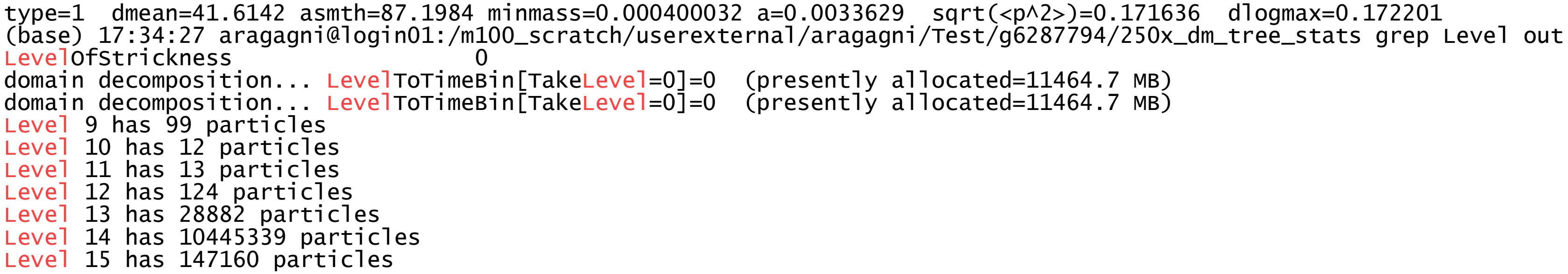


```
(base) 17:34:36 aragagni@login01:/m100_scratch/userexternal/aragagni/Test/g0052436_Me14_G/10x_norad/10x_norad_cpu_stats grep Level out|head
LevelOfStrickness                  1
domain decomposition... LevelToTimeBin[TakeLevel=0]=0  (presently allocated=1081.28 MB)
domain decomposition... LevelToTimeBin[TakeLevel=0]=0  (presently allocated=1081.28 MB)
Level 5 has 391 particles
Level 6 has 12419 particles
Level 7 has 6415 particles
Level 9 has 452 particles
Level 10 has 38 particles
Level 11 has 270 particles
Level 12 has 37188 particles
```

```
 ID=586846149 return 0 ninteractions 1270 n_tree_hits 2058 type=1

 ID=1351628618 return 0 ninteractions 8 n_tree_hits 81 type=2

 ThisTask 9 ID=1351094406 return 0 ninteractions 14 n_tree_hits 84 type=3
```



```
type=1  dmean=41.6142 asmth=87.1984 minmass=0.000400032 a=0.0033629  sqrt(<p^2>)=0.171636  dlogmax=0.172201
(base) 17:34:27 aragagni@login01:/m100_scratch/userexternal/aragagni/Test/g6287794/250x_dm_tree_stats grep Level out
LevelOfStrickness                  0
domain decomposition... LevelToTimeBin[TakeLevel=0]=0  (presently allocated=11464.7 MB)
domain decomposition... LevelToTimeBin[TakeLevel=0]=0  (presently allocated=11464.7 MB)
Level 9 has 99 particles
Level 10 has 12 particles
Level 11 has 13 particles
Level 12 has 124 particles
Level 13 has 28882 particles
Level 14 has 10445339 particles
Level 15 has 147160 particles
```

```
 ID=44540621 return 0 ninteractions 1009 n_tree_hits 1714 type=1
 ID=56865874 return 0 ninteractions 344 n_tree_hits 990 type=2

 ThisTask 9 ID=1351112758 return 0 ninteractions 21 n_tree_hits 107 type=3
```

Many Thanks
Please Connect at
antonio.ragagnin@inaf.it

Antonio Ragagnin

INAF, OATs

EuroEXA

Project ID: 754337