CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

RAMSES on the GPU: An OpenACC-Based Approach

Claudio Gheller
(ETHZ-CSCS)
Giacomo Rosilho de Souza
(EPFL Lausanne)
Romain Teyssier
(University of Zurich)
Markus Wetzstein
(ETHZ-CSCS)

PRACE

PRACE-2IP project
EU 7th Framework Programme
RI-283493

Cosmological Simulations

Numerical simulations represent an extraordinary tool to study and solve astrophysical problems.

They require:

- **Sophisticated simulation codes,** including all necessary physics and adopting suitable and effective algorithms
- **Data processing, analysis and visualization tools** necessary to process the enormous amount of generated information
- **High-end HPC systems,** that provide the necessary computing power

What & Why GPUs

GPUs are hardware components born for graphics;

They are now "widely" used for computing;

On suitable algorithms, GPUs are much faster than CPUs so they can dramatically reduce the time to solution:

- Data parallel algorithms (each piece of data is processed independently from the others) are privileged;
- High flops/bytes ratio are favored;
- Memory intensive (size, access) algorithms can be hard to implement and/or optimize;
- Asynchronous operations supported and must be exploited;
- Code development is not so hard, but getting a fast code can require a huge effort.

The RAMSES code: overview

- **RAMSES (R.Teyssier, A&A, 385, 2002): code to study of astrophysical problems**
- **It treats at the same time various components (dark energy, dark matter, baryonic matter, photons)**
- **Includes a variety of physical processes (gravity, magnetohydrodynamics, chemical reactions, star formation, supernova and AGN feedback, etc.)**

- **Open Source**
- **Fortran 90**
- **Code "size": about 70000 lines**
- **MPI parallel (public version)**
- **OpenMP support (restricted access)**

- **http://irfu.cea.fr/Phocea/Vie_des_labos/Ast/ast_sstechnique.php?id_ast=904**

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

RAMSES workflow

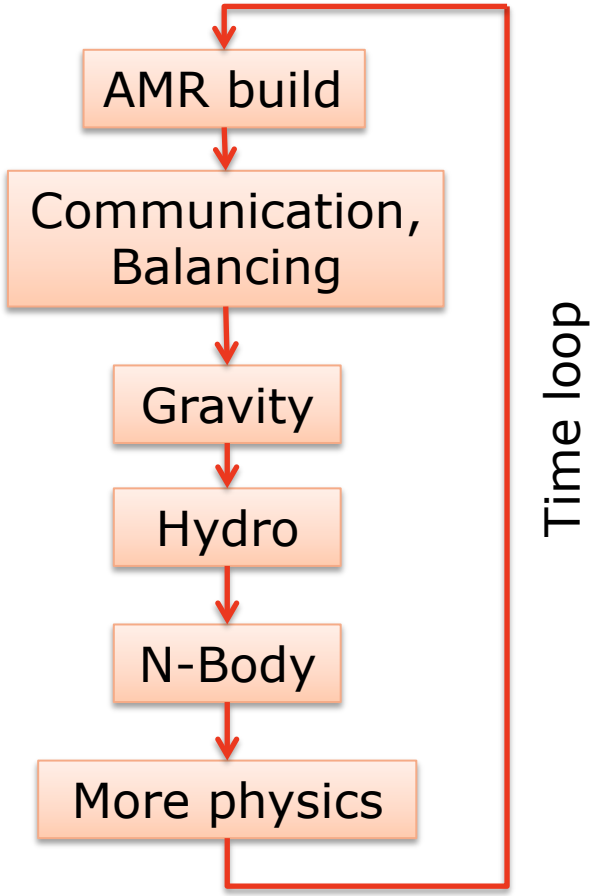
3D Eulerian Adaptive Mesh Refinement codes.

The code solves:

- **dark matter - N-body particle-mesh technique.**
- **gravity - multigrid technique.**
- **Hydrodynamics: various shock capturing methods.**
- **A number of additional physics processes**

Spatial discretization through and adaptive cartesian mesh

AMR provides high resolution ONLY where this is strictly necessary



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

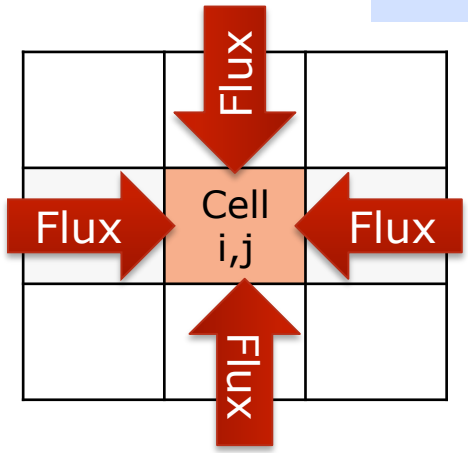
RAMSES: solving fluid dynamics

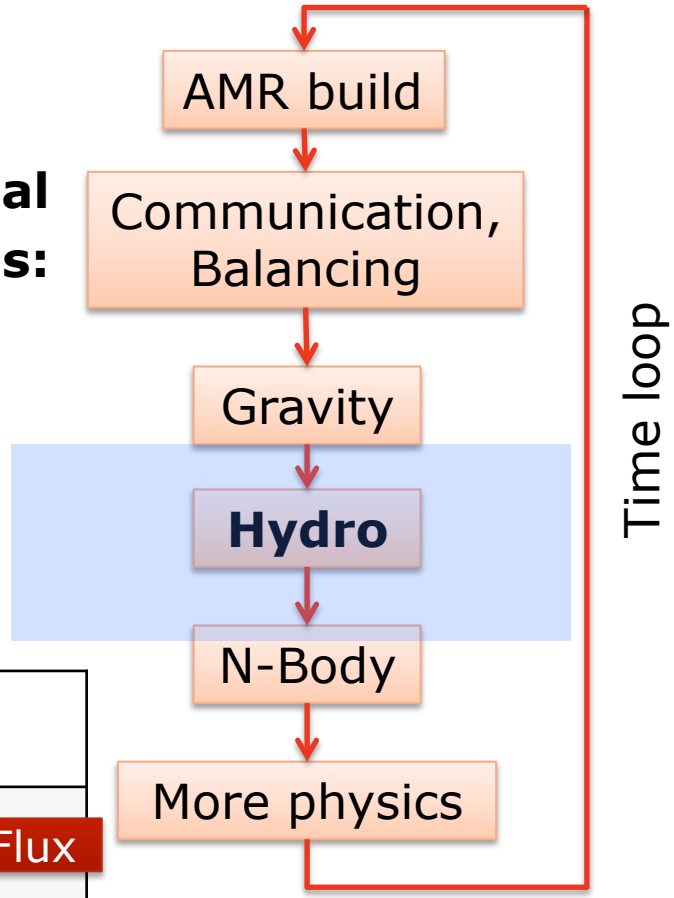
- **Fluid dynamics is one of the key kernels;**
- **It is also among the most computational demanding;**
- **fluid dynamics is solved on a computational mesh solving three conservation equations: mass, momentum and energy:**

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0$$

$$\frac{\partial}{\partial t}(\rho \mathbf{u}) + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) + \nabla p = -\rho \nabla \phi$$

$$\frac{\partial}{\partial t}(\rho e) + \nabla \cdot [\rho \mathbf{u}(e + p/\rho)] = -\rho \mathbf{u} \cdot \nabla \phi$$



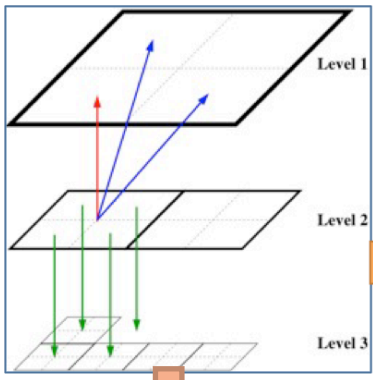


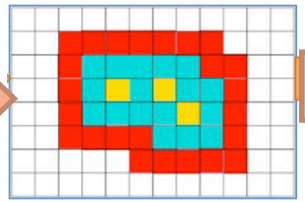
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

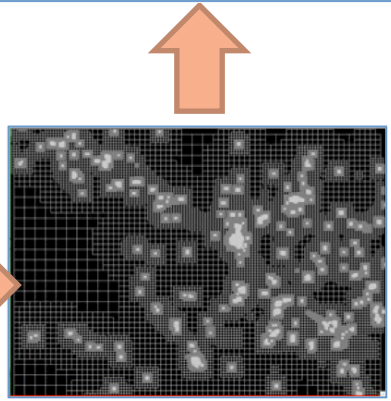
RAMSES AMR Mesh

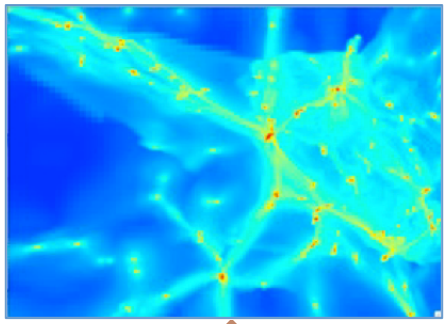
Fully Threaded Tree with Cartesian mesh

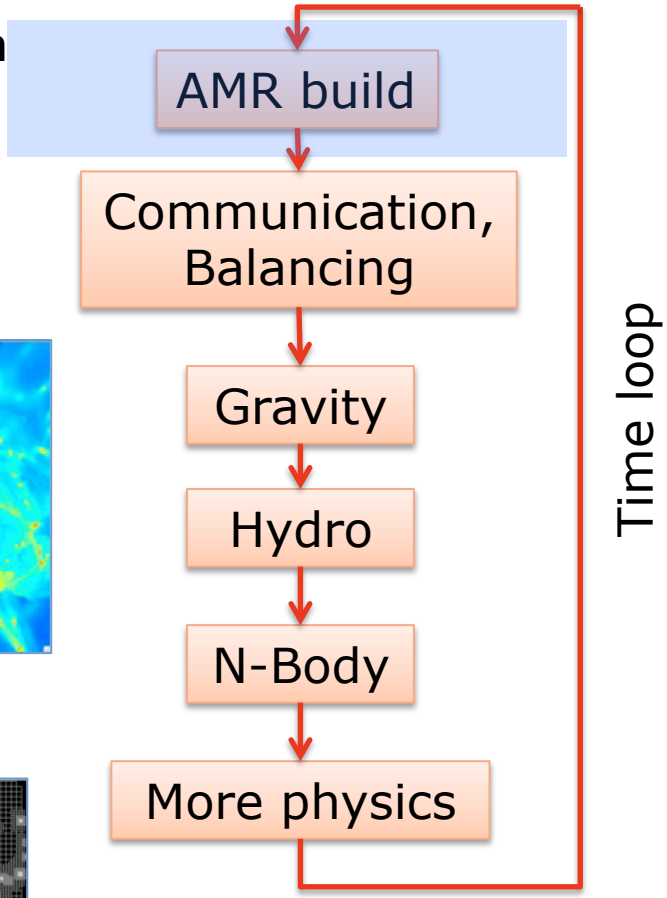
- **CELL BY CELL refinement**
- **COMPLEX data structure**
- **IRREGULAR memory distribution**





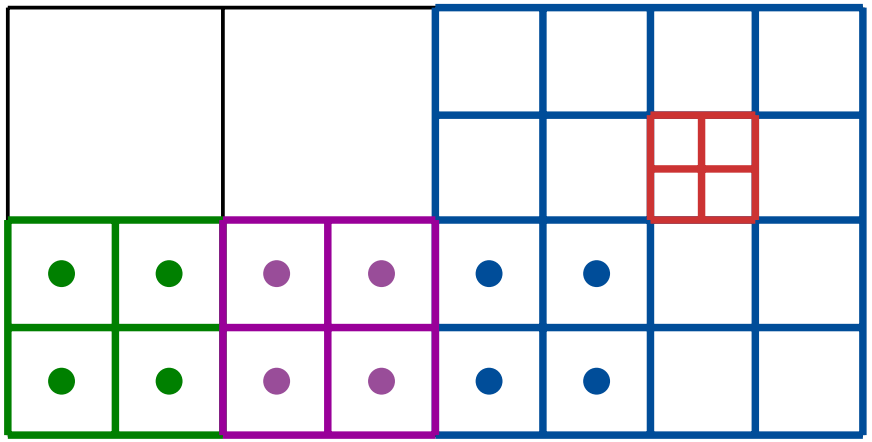






Ramses memory management

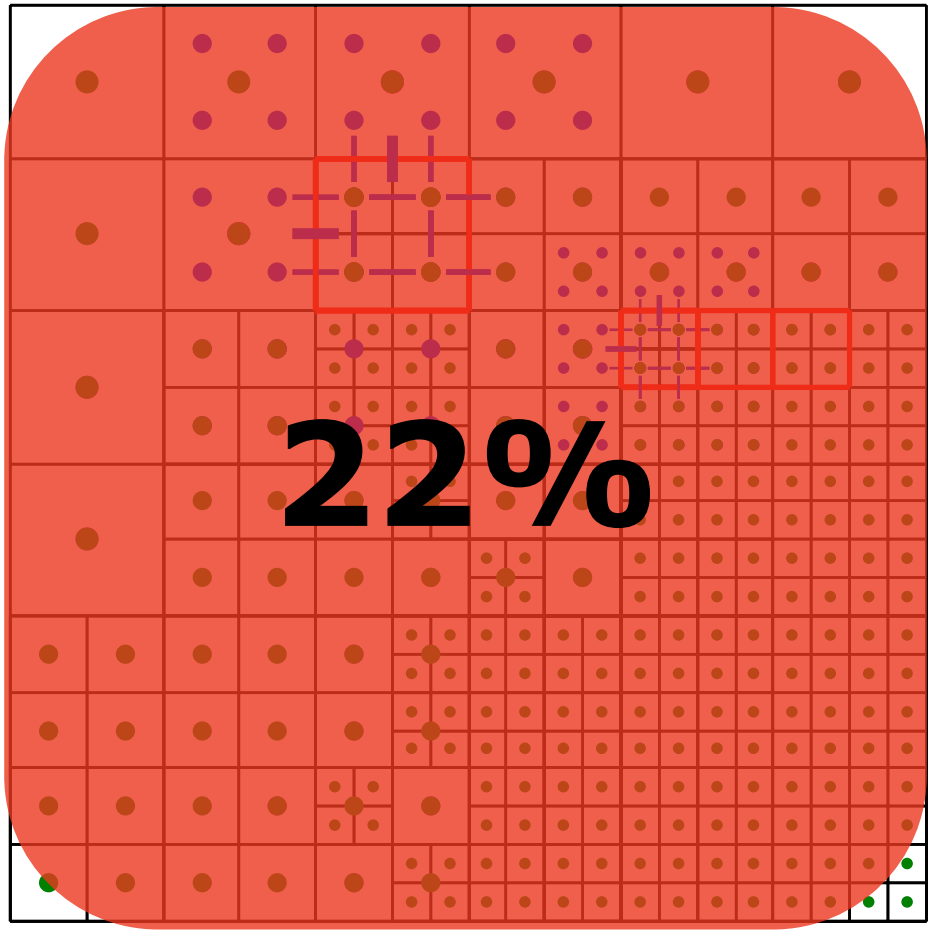
- **Non contiguous memory**
- **Different levels of refinement are mixed**
- **Cell position in memory is unpredictable**



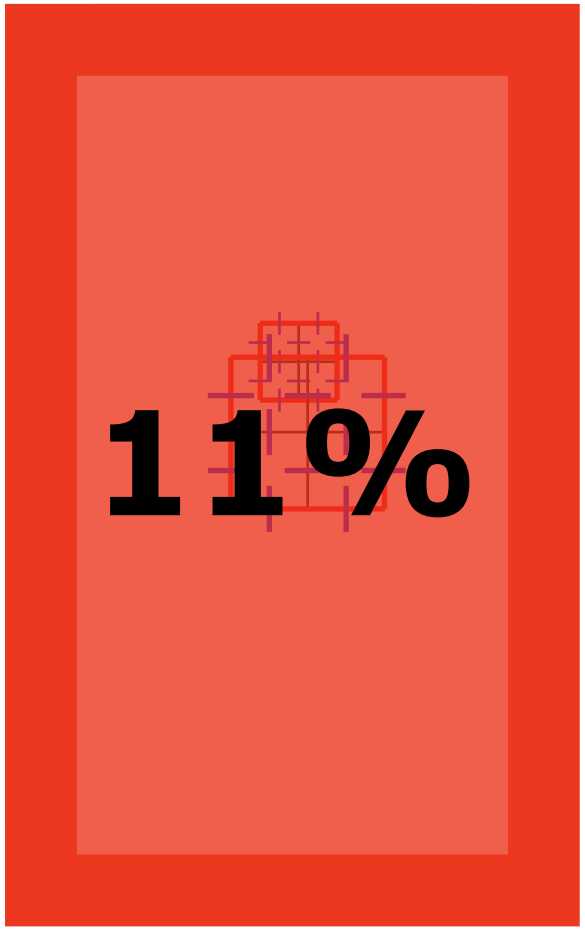
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Ramses hydro kernel & AMR

AMR grid



Equations Solver



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Ramses on the GPU

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

RAMSES: GPU hydro solver

Original code and OpenACC port profiling:

```
|-----------------------------------------------------------
| 100.0% | 8775.076792 |USER
||----------------------------------------------------------
|| 17.2% | 1510.795560|godfine1_
|| 14.6% | 1277.922684|get3cubefather_
|| 8.1% | 712.218697 |gauss_seidel_mg_fine_
|| 6.1% | 534.160606 |interpolate_and_correct_fine_
|| 6.0% | 523.049099 |make_virtual_fine_dp_
|| 5.5% | 485.875381 |make_virtual_reverse_dp_
|| 4.5% | 393.035509 |cmp_residual_mg_fine_
|| 3.5% | 305.048373 |interpol_phi_
|| 3.4% | 295.687628 |interpol_hydro_
|| 3.0% | 261.753846 |unsplit_
|| 2.7% | 238.224640 |cmpflxm_
|| 2.1% | 182.118400 |ctoprim_
|| 2.0% | 176.510382 |build_parent_comms_mg_
|| 2.0% | 174.412867 |gauss_seidel_mg_coarse_
|| 1.5% | 133.619786 |riemann_llf_
|| 1.4% | 125.394321 |synchro_hydro_fine_
|| 1.3% | 111.616723 |restrict_residual_fine_reverse_
|| 1.3% | 110.802651 |uslope_
|| 1.1% | 94.464831 |getnborfather_
|| 1.1% | 93.612857 |interpolate_and_correct_coarse_
|| 1.1% | 92.368937 |make_virtual_mg_dp_
|| 1.0% | 90.738933 |get3cubefather_godfine_
|| 1.0% | 85.241909 |make_fine_bc_rhs_
|=========================================
```

```
|-----------------------------------------------------------
| 100.0% | 5718.250985 |USER
||----------------------------------------------------------
|| 22.3% | 1276.678459|get3cubefather_
|| 12.8% | 732.494081 |gauss_seidel_mg_fine_
|| 9.4% | 536.284852 |interpolate_and_correct_fine_
|| 9.1% | 522.834264 |make_virtual_fine_dp_
|| 6.9% | 394.130303 |cmp_residual_mg_fine_
|| 5.3% | 304.073254 |interpol_phi_
|| 3.1% | 176.423148 |gauss_seidel_mg_coarse_
|| 3.1% | 176.036009 |build_parent_comms_mg_
|| 2.3% | 129.439764 |make_virtual_reverse_dp_
|| 2.2% | 125.986470 |synchro_hydro_fine_
|| 2.0% | 111.681912 |restrict_residual_fine_reverse_
|| 1.7% | 95.785096 |interpolate_and_correct_coarse_
|| 1.6% | 92.444358 |make_virtual_mg_dp_
|| 1.5% | 87.124500 |make_fine_bc_rhs_
|| 1.5% | 83.867781 |cic_cell_
|| 1.0% | 56.125077 |gradient_phi_
|| 1.0% | 54.569618 |courant_fine_
|=========================================
```

33% VS 3%

CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Our development/testing/target system

"Piz Daint" CRAY XC30 system @ CSCS (N.6 in Top500)

Nodes:

5272 CPUs 8-core Intel SandyBridge equipped with:

- 32 GB DDR3 memory
- One NVIDIA Tesla K20X GPU with 6 GB of GDDR5 memory

Overall system

- 42176 cores and 5272 GPUs
- 170+32 TB
- Interconnect: Aries routing and communications ASIC, and Dragonfly network topology
- Peak performance: 7.787 Petaflops

© CSCS 2013 - Claudio Gheller

The programming model: OpenACC

- **Directive based API (corresponds to "OpenMP" for parallel programming)**

OpenACC (http://www.openacc-standard.org/)

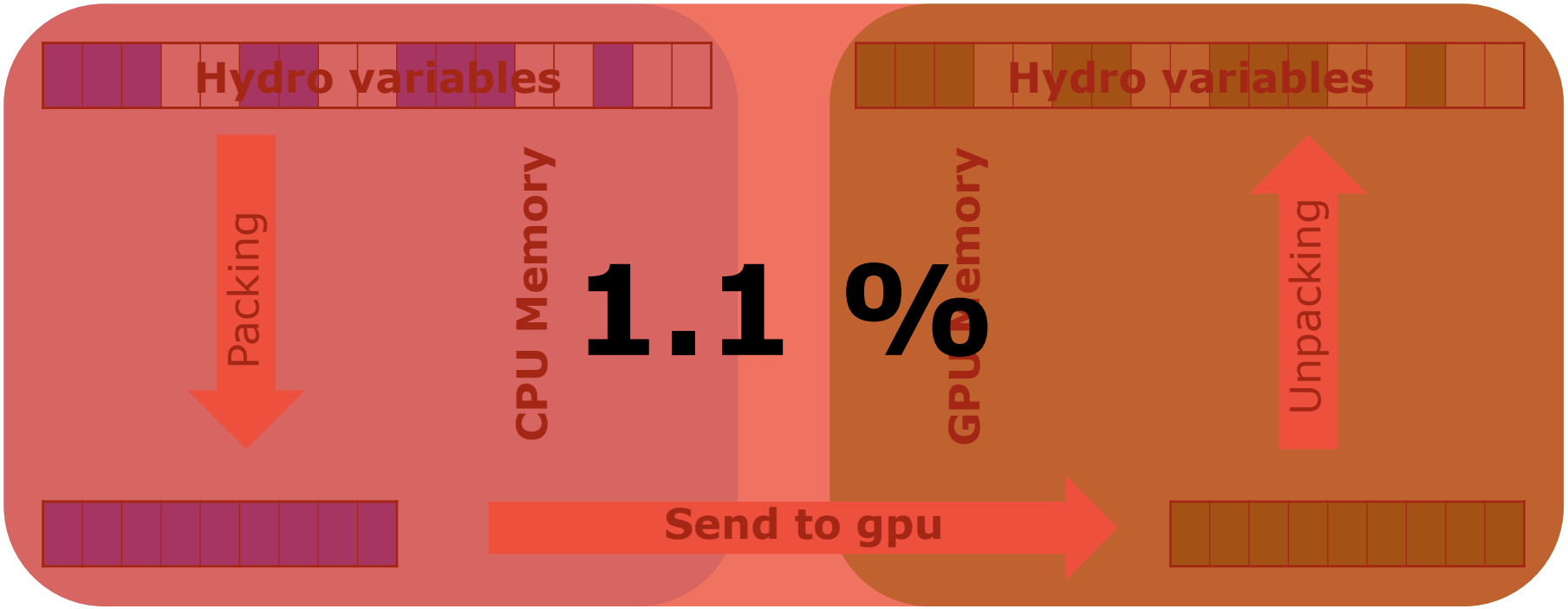
- ✓ Supported by CRAY and PGI (slightly different standards, but converging)
- ✓ Finally converging (hopefully) to OpenMP
- ✓ "Easier" code development – supports incremental development
- ✓ Suitable to Fortran
- ✓ Performance tuning not so easy (possible performance sacrifice – goal is 80% CUDA)
- ✓ Can be "combined" with CUDA code

```
PROGRAM main
  INTEGER :: a(N)
  <stuff>
!$acc parallel loop
  DO i = 1,N
   a(i) = i
  ENDDO
!$acc end parallel loop
!$acc parallel loop
  DO i = 1,N
   a(i) = 2*a(i)
  ENDDO
!$acc end parallel loop
  <stuff>
END PROGRAM main
```

Moving data to/from the GPU

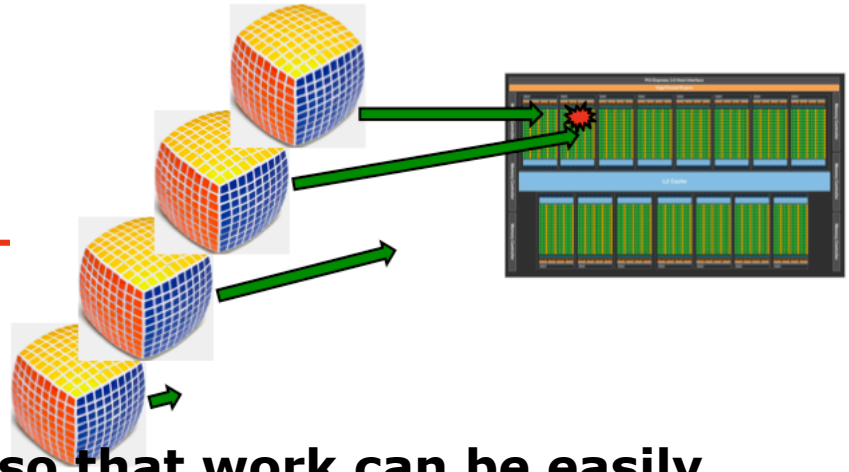
Send data to the gpu.

- **AMR grid's data is stored "random" in memory.**
- **Pack-unpack strategy level by level**
- **Has to be done every time step.**



CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

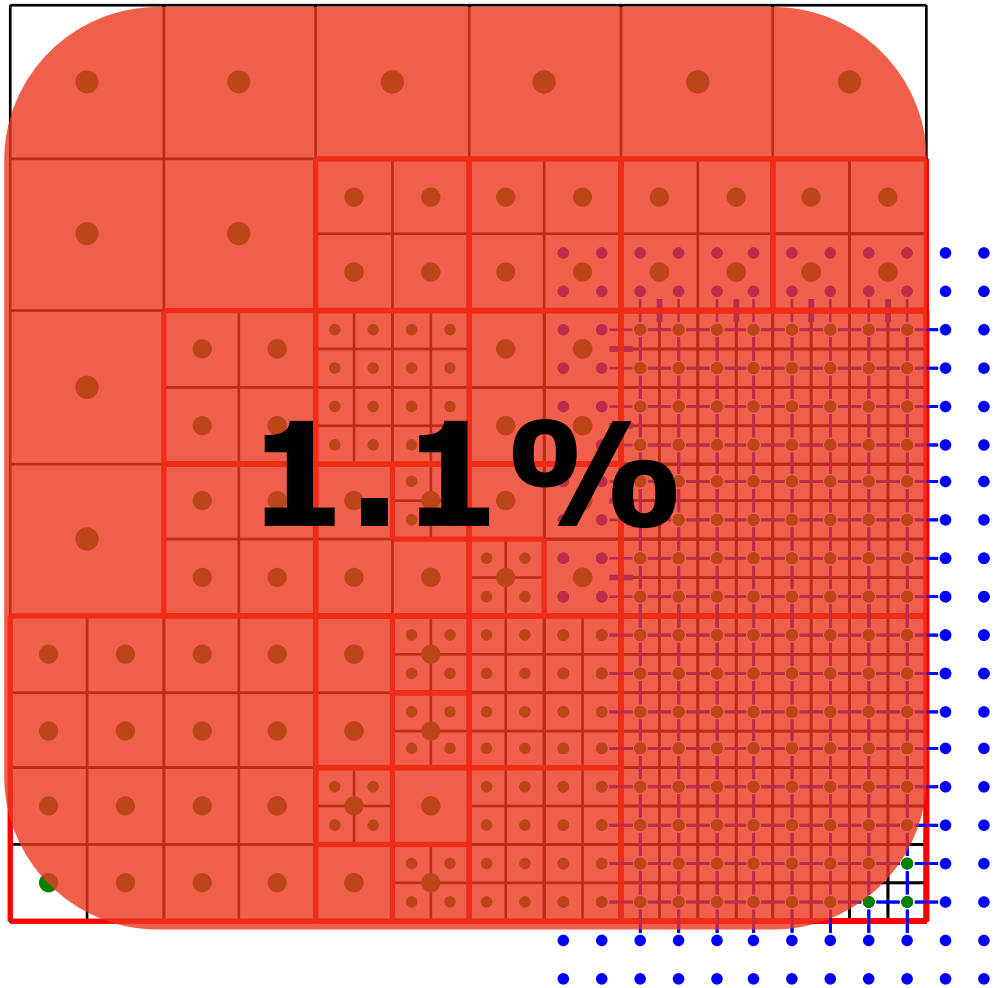
On board the GPU…



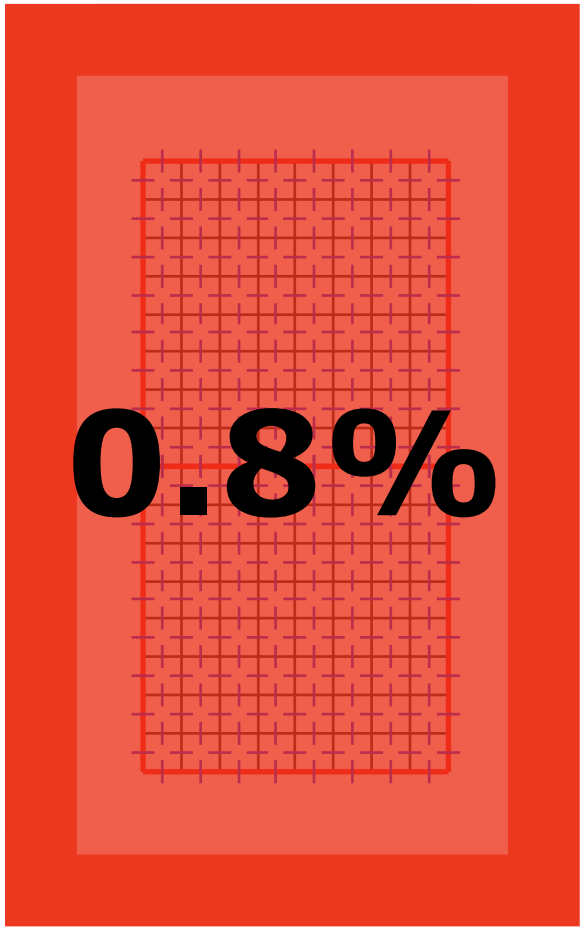
1. **Reorganization of memory in spatially contiguous patches, so that work can be easily split in blocks and coalescing memory can be exploited**
2. **Patches are grouped and pushed to the GPU cores. Groups size can be tuned in order to improve the occupancy**
3. **Patches build-up strongly benefits of the high memory bandwidth**
4. **Nested loops collapse used wherever possible**
5. **Gang and vector based work scheduling adopted (no particular benefit in using worker scheduling)**
6. **Offload data only when and where necessary (but this can be still improved – ongoing work)**

GPU implementation

AMR grid



Equations Solver



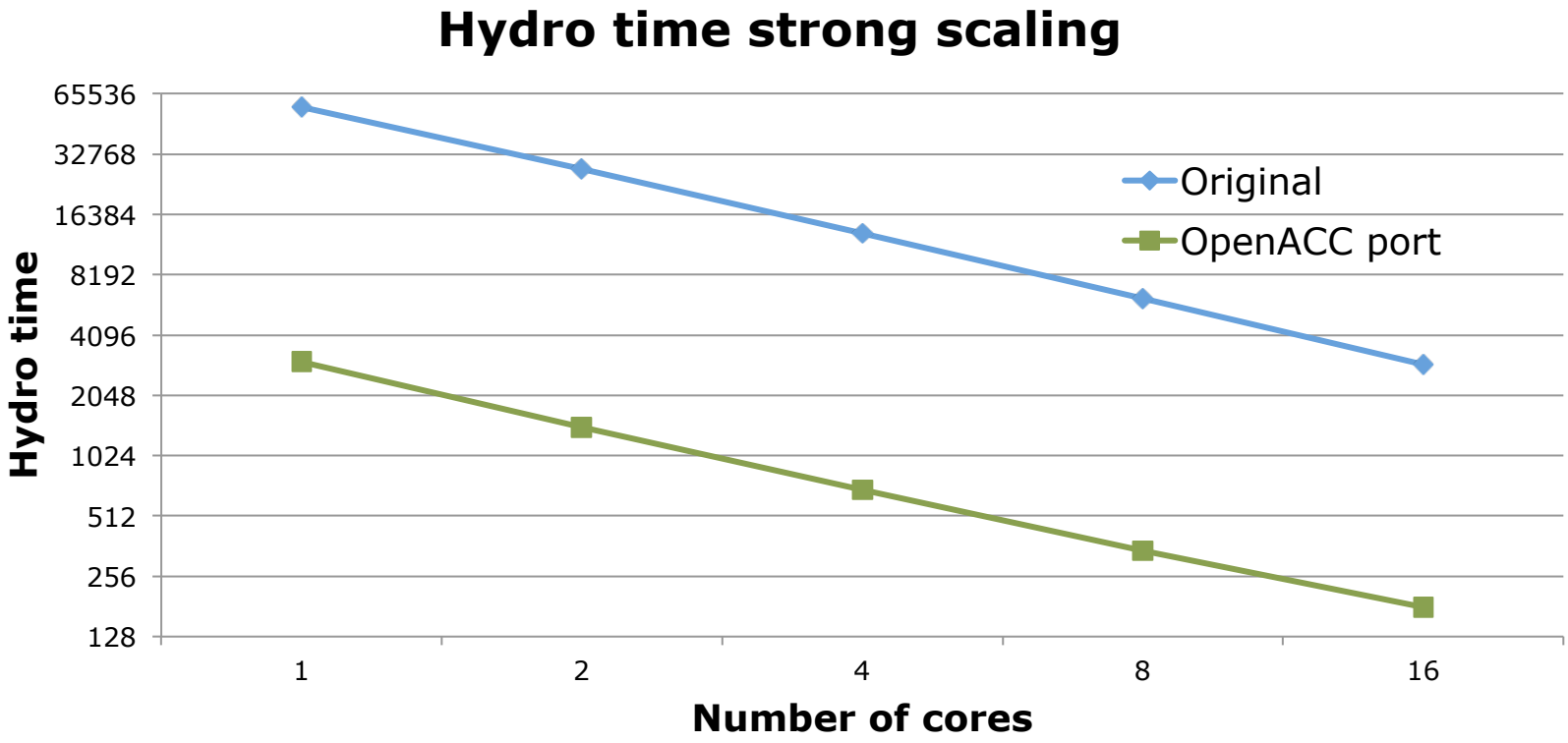
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Performance analysis

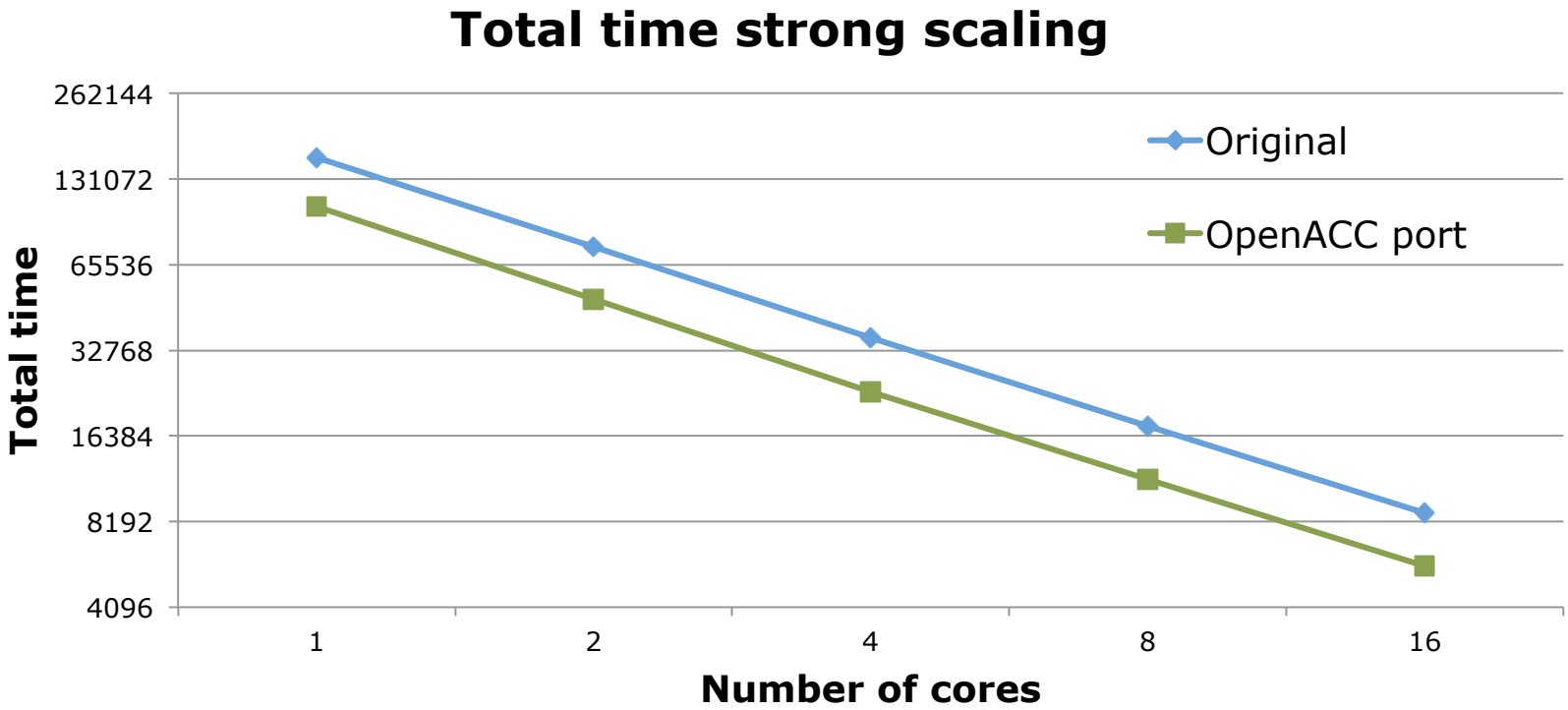
Cosmological test with 3 levels of refinement
Levels 6 to 8

Cosmo 3 Levels (6-8)	T_tot	T_hydro		T_god_fine	T_copy	T_tot speedup	T_hydro speedup		T_god/ T_copy
		Sec	Percent				1 core vs 1gpu	1 cpu VS 1gpu	
orig_V10_N1	155662	218	36.1 %	56218					
orig_V10_N2	75905	27625	36.4	27625					
orig_V10_N4	36147	13207	36.5	13207					
orig_V10_N8	17	6243	35.2 %	6243					
orig_V10_N16	8775	2918	33.3	2918					
ACCyes_C1000_N1	104811	009	2.9 %	2270		1.49	.68	2.07	3.07
ACCyes_C1000_N2	49718	1425	2.9	1040	385	1.53	19.39	2.05	2.70
ACCyes_C1000_N4	23372	693	3.0	485	208	1.55	19.07		2.33
ACCyes_C1000_N8	11543	344	3.0	231	113	1.54	18.15		2.03
ACCyes_C1000_N16	5718	179	3.1	115	64	1.53	16.26		1.79

Performance results



Performance results



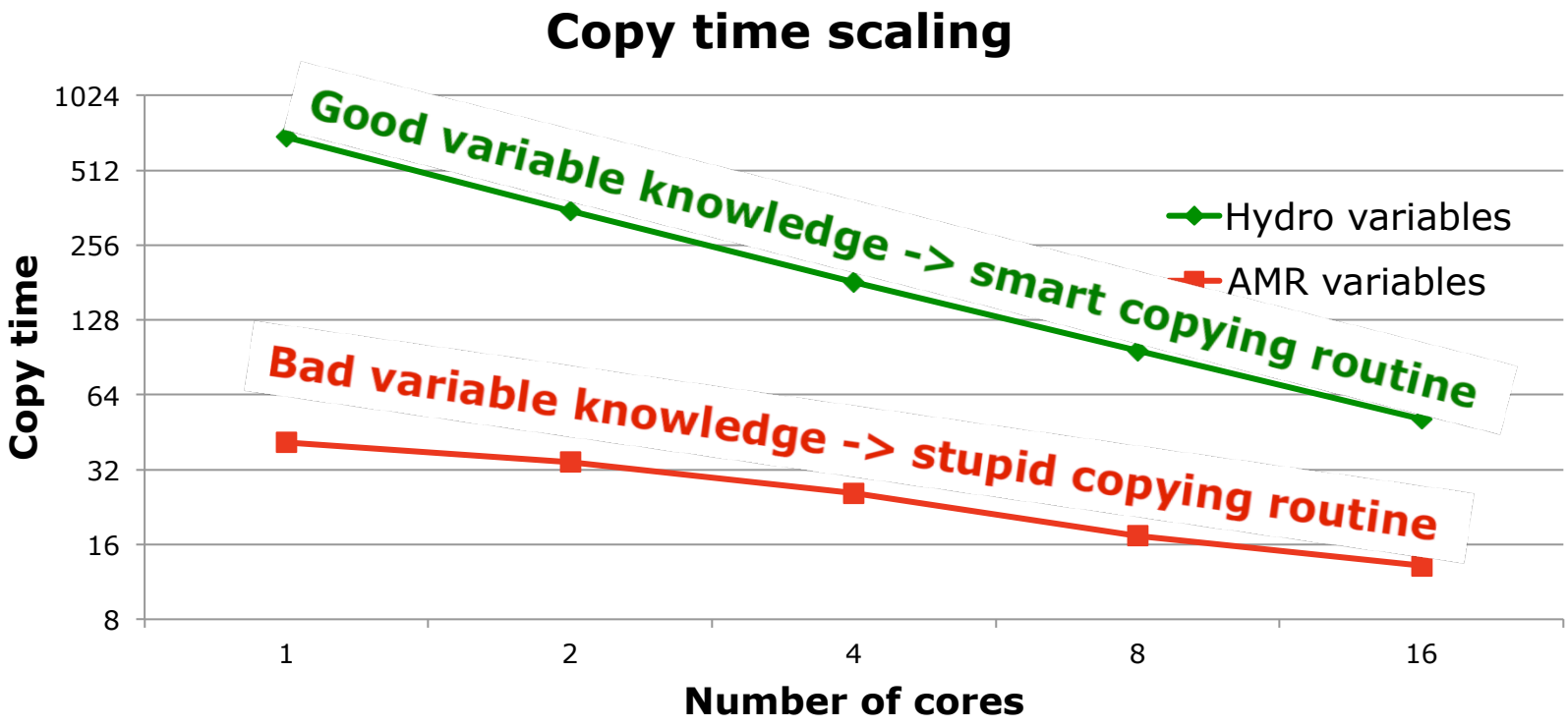
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

Performance results

Hydro vars + AMR vars

Cosmo 3 Levels (6-8)	T_tot	T_hydro		T_god_fine	T_copy	T_tot speedup	T_hydro speedup		T_god/ T_copy
		Sec	Percent				1core VS 1gpu	1 cpu VS 1gpu	
orig_V10_N1	155662	56218	36.1	56218					
orig_V10_N2	75905	27625	36.4	27625					
orig_V10_N4	36147	13207	36.5	13207					
orig_V10_N8	17755	6243	35.2	6243					
orig_V10_N16	8775	2918	3						
ACCyes_C1000_N1	104811	3009		2270	739	.49	18.68	2.0	3.07
ACCyes_C1000_N2	49718	1425		1040	385	.53	19.39	2.0	2.70
ACCyes_C1000_N4	23372	693		485	208	.55	19.07		2.33
ACCyes_C1000_N8	11543	344		231	113	.54	18.15		2.03
ACCyes_C1000_N16	5718	179		115	64	.53	16.26		1.79

Performance results

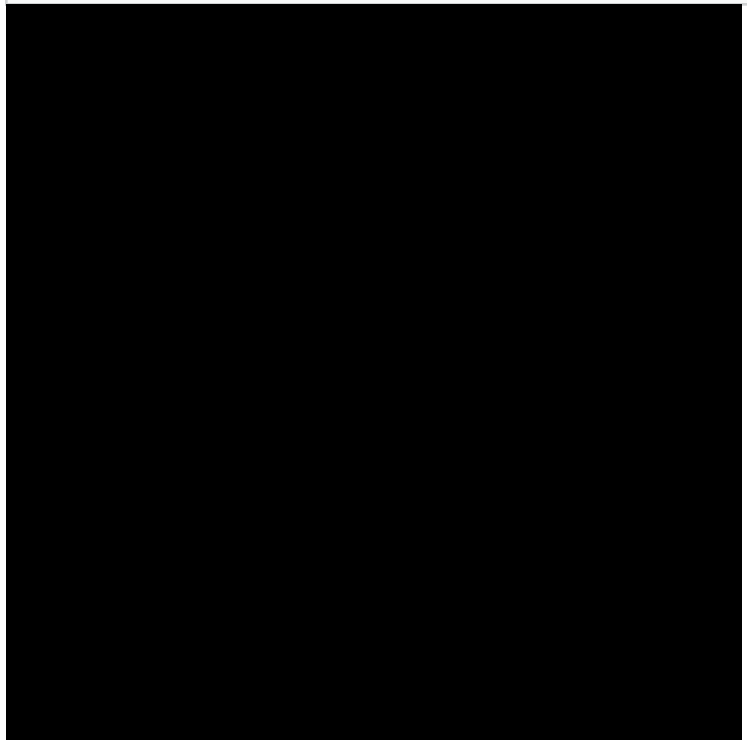


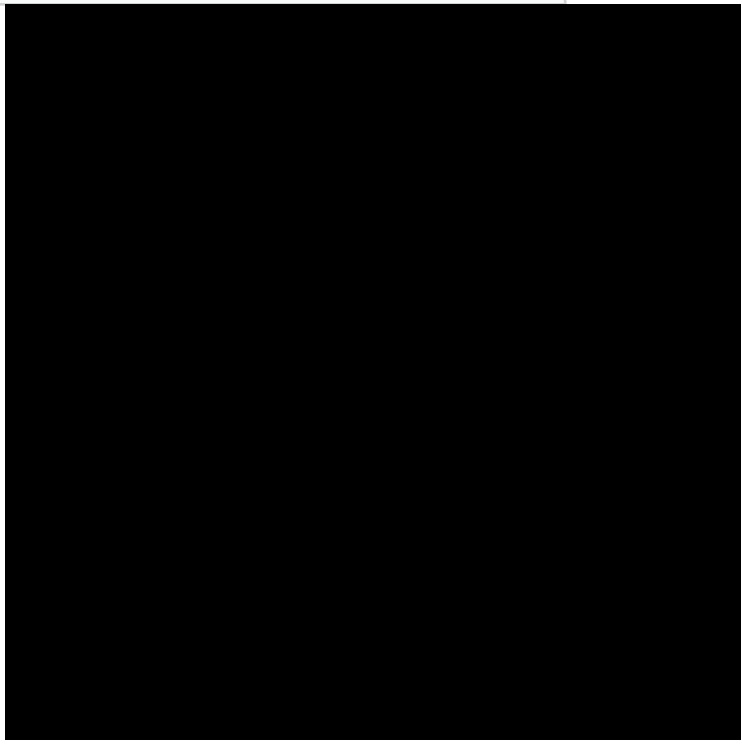
CSCS
Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

A small simulation…

"Small" cosmological simulation with hydro, gravity and cooling: box size = 100 Mpc (10^{19}km), memory 3 GB

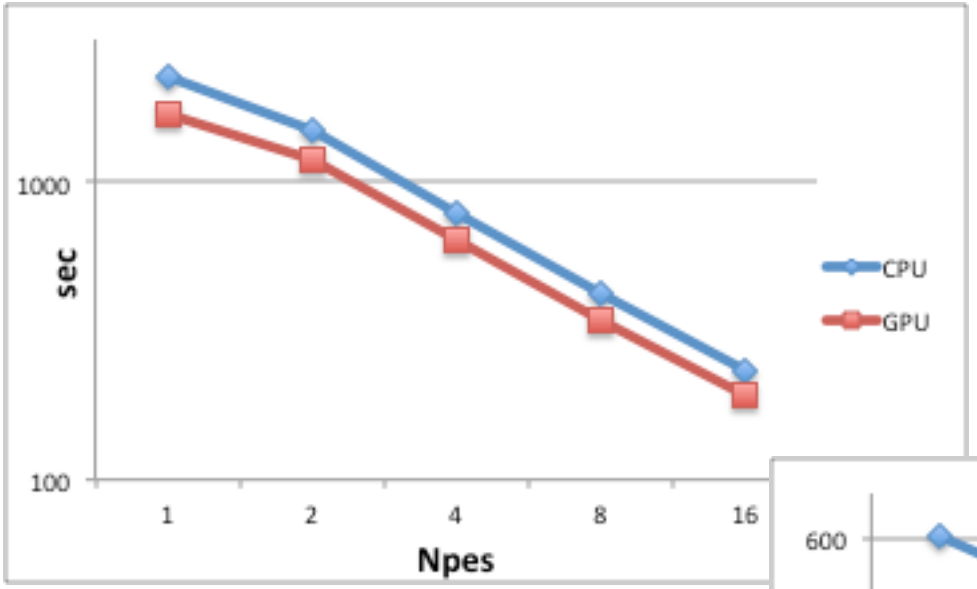
	Level id	Eff. Mesh size	Spatial resolution (Mpc)
Base level	7	256^3	0.39
Eff. level	12	8192^3	0.012
Max level	15	65536^3	0.0015





Visualization made with Splotch (https://github.com/splotchviz/splotch) S4516 @ GTC (by Mel Krokos)

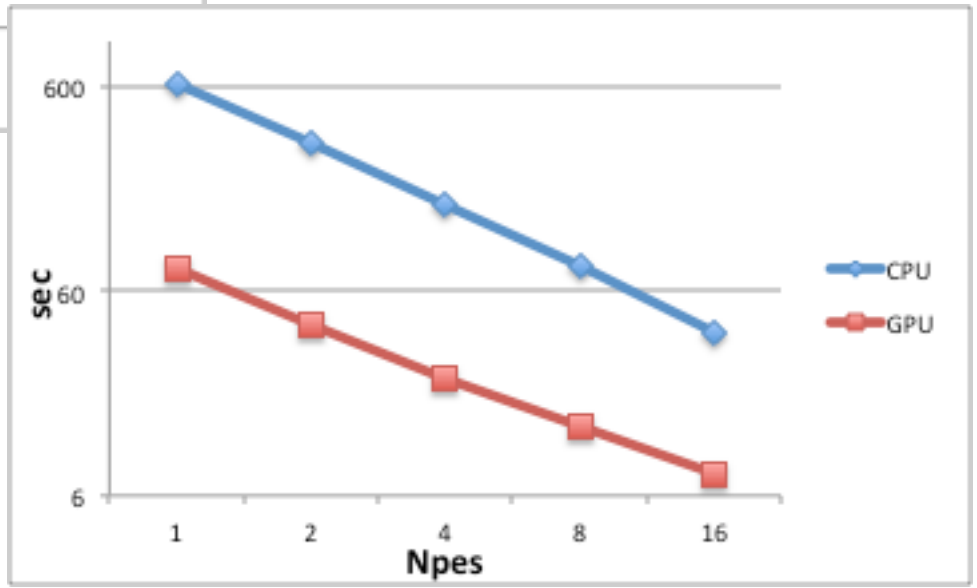
Results



Scalability of the CPU and GPU versions (Total time)

Fraction of time saved using the GPU

0.2431415
0.20057766
0.19113076
0.18720094
0.17725265



Scalability of the CPU and GPU versions (Hydro time)

© CSCS 2013 - Claudio Gheller

...and a big simulation...

"Big" cosmological simulation with hydro, gravity and cooling: box size = 100 Mpc (10^{19}km)

	Level id	Eff. Mesh size	Spatial resolution (Mpc)
Base level	9	1024^3	0.09765625
Eff. level	14	32768^3	0.003051758
Max level	17	262144^3	0.00038147

Memory = 240 GB
AMR structure at timestep 100 →

180000544 cells
703127 cells/256
351563 cells/512

Mesh structure
Level 1 has 1 grids (0, 1, 0,)
Level 2 has 8 grids (0, 1, 0,)
Level 3 has 64 grids (0, 1, 0,)
Level 4 has 512 grids (1, 3, 2,)
Level 5 has 4096 grids (10, 19, 16,)
Level 6 has 32768 grids (80, 149, 128,)
Level 7 has 262144 grids (648, 1190, 1024,)
Level 8 has 2097152 grids (5192, 9526, 8192,)
Level 9 has 16777216 grids (41541, 76208, 65536,)
Level 10 has 2608087 grids (1971, 27866, 10187,)
Level 11 has 536556 grids (0, 8409, 2095,)
Level 12 has 155720 grids (0, 2717, 608,)
Level 13 has 24607 grids (0, 675, 96,)
Level 14 has 1137 grids (0, 90, 4,)

© CSCS 2013 - Claudio Gheller

Results

HYDRO

Computing devices	CPU (sec.)	GPU (sec.)
256	80.72	10.46
512	39.81	5.16

TOTAL

Computing devices	CPU (sec.)	GPU (sec.)
256	701.18	590.35
512	423.12	358.75

Summary

Objective:
RAMSES enabling to GPUs

Methodology
Incremental approach exploiting RAMSES'modular architecture and OpenACC programming model

Current achievement:
Hydro kernel ported on GPU: final optimization being completed

Coming steps:
- **Enable the cooling and radiative transfer module to GPU**
- **Enable the MHD module to GPU**
- **Move MPI stuff related to hydro variables to the GPU**

Challenges
- **Enable the gravitational solver to the GPU**
- **Redesign data structures**