Finanziato dall'Unione europea NextGenerationEU
Ministero dell'Università e della Ricerca
Italiadomani PIANO NAZIONALE DI RIPRESA E RESILIENZA
ICSC Centro Nazionale di Ricerca in HPC, Big Data and Quantum Computing

RAMSES GPU

Presented by:
Raffaele Pascale

Collaborators:
Francesco Calura, Claudio Gheller, Emanuele De Rubeis, Donatella Romano, Valentina Cesare

Spoke 3 Technical Workshop, Trieste October 9 / 11, 2023

Scientific Rationale

In modern astrophysics, hydrodynamical *N*-body simulations are powerful and versatile tools for **testing theories of galaxy formation and evolution.**

To understand the physical processes involved, simulations with **increasingly high spatial resolution** are required, leading to a dramatic **escalation of computational**.

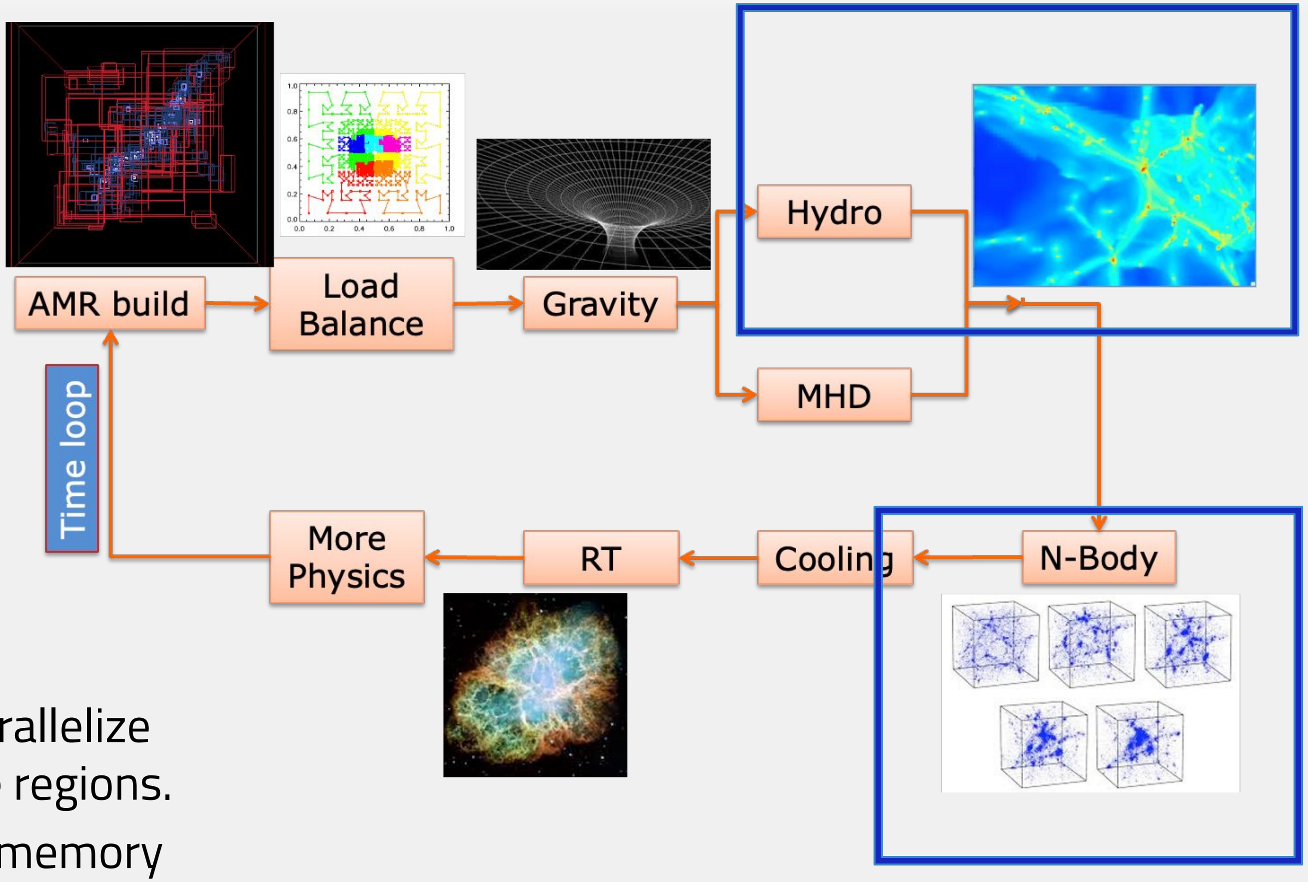
Solution

Porting and optimization on GPU architecture of specific (time consuming) modules of N-body hydrodynamical codes to cut computational time

Application to (MINI)RAMSES:
RAMSES is an Eulerian code, specifically designed for cosmological simulations. It is written in Fortran90 and it exploits **AMR** (Adaptive mesh refinement), i.e. the spatial resolution is increased in regions where specific criteria are satisfied (e.g. mass density)

Technical Objectives

- Performances enhancement and reduction of computational time: GPU porting of hydrodynamical and N-body modules (with an overall speeding factor >1?)



Methodologies

- Profiling methods.
- Utilize OpenACC-based directives to parallelize time-consuming loops and critical code regions.
- Apply optimization techniques such as memory management, kernel optimization, and reduction of communication between CPU and GPU

Timescale, Milestones and KPIs

Analysis and Preparation:
- Investigation of MINIRAMSES to identify sections suitable for GPU parallelization - *partially done (hydro and Nbody)*

Initial Parallelization (Nbody and Hydro): >3 months (each)
- Identification of suitable modules to port on GPU (*partially done - Profiling*)
- Porting of identified part on GPU based on specific libraries (OpenACC directives)

Testing (Nbody and Hydro): >4 months (each)
- Implementation and run of suitable tests (hydrodynamical sedevo3d test + cosmological simulation).
- Evaluation of initial performance and identification of any issues or bugs
- Optimization of the code on GPU to maximize performance

Integration (Nbody and Hydro):
- Integration in principal version of the code
- Execution of tests to evaluate scalability

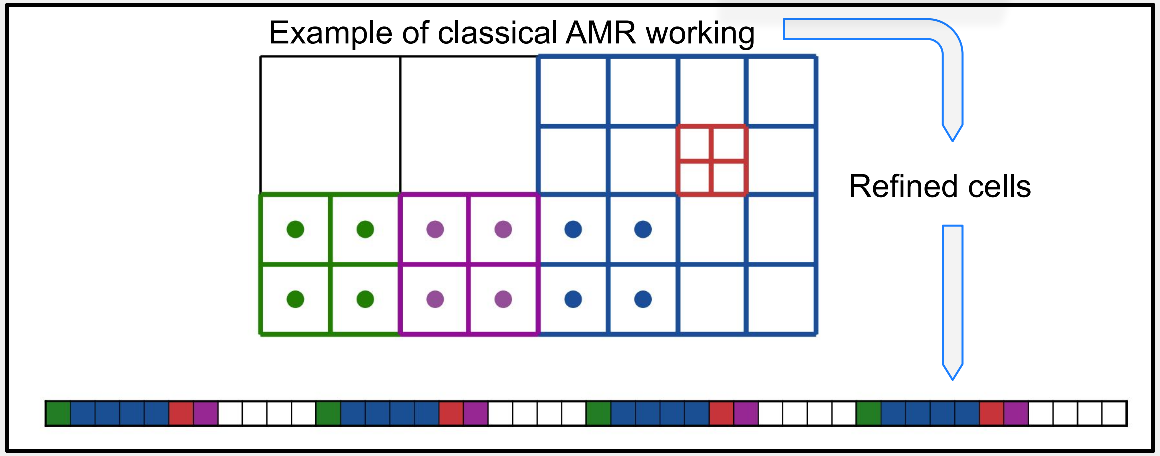
Validation and verification of correcteness of results w.r.t. original version of the code.

Accomplished Work, Results

- Project started during June 2023, taking advantage of the **hackathon event @ Cineca**
- Focus on MINIRAMSES, abridged version of RAMSES.

Designed for GPU parallelization since minimized memory access

RAMSES



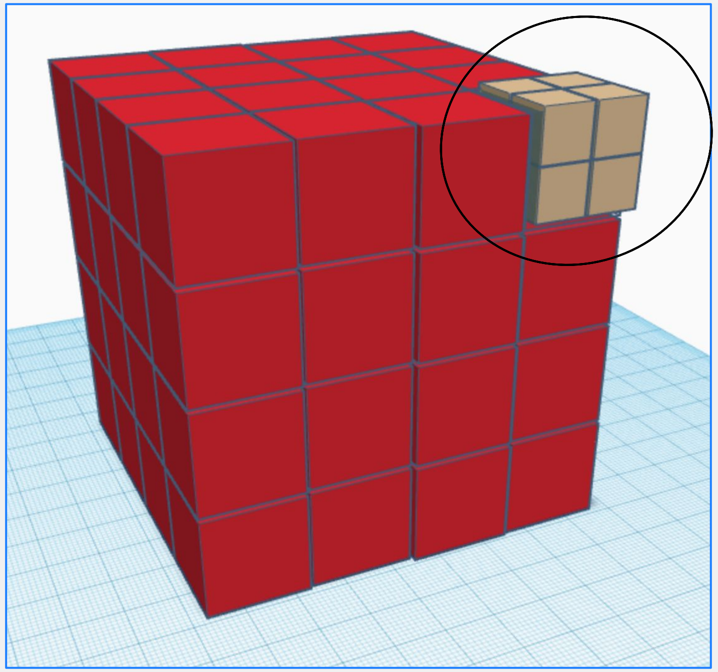
Saved in non-contiguous memory

Accomplished Work, Results

- Project started during June 2023, taking advantage of the **hackathon event @ Cineca**
- Focus on MINIRAMSES, abridged version of RAMSES.

Designed for GPU parallelization since minimized memory access

MINIRAMSES



Super-ocs: groups of adiaject ocs saved in contiguous memory locations

Number of octs per super-octs: 8^n

Designed for GPU parallelization

Accomplished Work, Results

- Project started during June 2023, taking advantage of the **hackathon event @ Cineca**
- Focus on MINIRAMSES, abridged version of RAMSES.

- Profiling of the code based on an hydrodynamical test (sedov 3d)
- Identification of time consuming modules (hydrodynamic solver)

__libc_start_main	·	99,99	/usr/lib64/power9/libc-2.28.so
generic_start_main	·	99,99	/usr/lib64/power9/libc-2.28.so
main	·	99,99	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
MAIN_	·	99,99	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
mdl_init_	·	99,99	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
mdl_init_master	·	99,97	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
adaptive_loop	·	99,97	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
amr_step_m_amr_step_	·	85,91	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
godunov_fine_module_godunov_fine_	·	62,93	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
godunov_fine_module_godfine1_	·	62,93	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
unsplit_	·	41,55	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
cmpflxm_	0,00	23,13	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
riemann_llf_	19,81	19,81	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
cmpflxm_	3,32	3,32	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
unsplit_	2,57	18,43	/m100_scrat[illegible]o0/mini-ramses/bin/ramses3d
trace3d_	8,52	8,52	/m100_scra[illegible]o0/mini-ramses/bin/ramses3d
uslope_	5,01	5,01	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
ctoprim_	2,32	2,32	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
godunov_fine_module_godfine1_	19,27	19,30	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
nbors_utils_get_grid_	·	1,60	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
newdt_fine_module_m_newdt_fine_	·	11,49	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
synchro_hydro_fine_module_m_sync...	·	7,34	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
amr_step_m_amr_step_	·	1,64	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
godunov_fine_module_set_unew_	·	1,28	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
godunov_fine_module_set_uold_	·	1,20	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
m_init_refine_adaptive_	·	9,05	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d
init_refine_basegrid_module_m_init_ref...	·	4,94	/m100_scratch/userexternal/dromano0/mini-ramses/bin/ramses3d

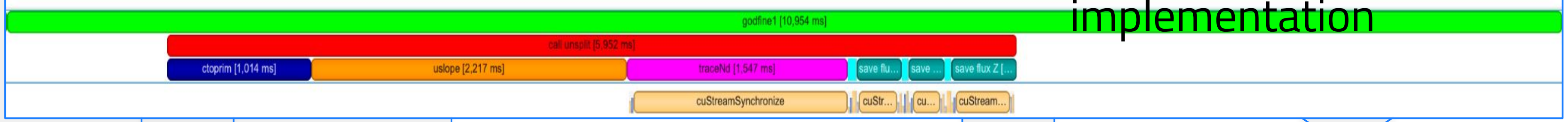
Sedov 3D with 1 task on 1 CPU
(about 89 s of execution time)

Accomplished Work, Results

- Execution time of GPU implementation of hydro modules takes a factor of 3 less

- **Major problem: memory management between Host and Device!**

CPU implementation



GPU implementation

Next Steps and Expected Results (by next checkpoint: April 2024)

- Produce a partial documentation of the code.
- get GPU hours. ISCRA C proposal submitted to get GPU hours on Leonardo
- Complete optimization of the hydrodynamical modules on GPU: better management of data movements. Change of paradigm: from unified memory to explicit data transfers.