The MPI+CUDA Gaia AVU-GSR Parallel Solver towards next-generation Exascale Infrastructures and new Green Computing frontiers

V. Cesare[1,*], U. Becciani[1], A. Vecchiato[2], M. G. Lattanzi[2], F. Pitari[3], M. Raciti[1], G. Tudisco[1], M. Aldinucci[4], B. Bucciarelli[2]
(1) INAF, OA-Catania; (2) INAF, OA-Torino; (3) CINECA; (4) UniTO, Dipartimento di Informatica

*valentina.cesare@inaf.it

CSN 5 - Forum della Ricerca Sperimentale e Tecnologica in INAF
June 23rd 2022

1. Gaia AVU-GSR solver target

- In production for the **ESA Gaia mission** since 2014, under a **INAF-CINECA** agreement, with the **ASI** support
- Derivation of positions and proper motions of ~10^8 stars in the Milky Way observed with the Gaia satellite, **with a μas accuracy**.

2. Code structure and MPI+OpenMP parallelization

$$\mathbf{A} \times \boldsymbol{x} = \boldsymbol{b}$$

Coefficient matrix:

- Large and sparse ($N_{obs} \times N_{unk} \simeq 10^{11} \times 10^{8}$ elements)
- Computation with a dense matrix $\mathbf{A_d}$ ($\sim 10^{11} \times 10^{1}$ elements)

Solution array: $\sim 10^{8} \times 10^{1}$ elements

Known terms array: $\sim 10^{11} \times 10^{1}$ elements

Becciani et al. (2014)

2. Code structure and MPI+OpenMP parallelization

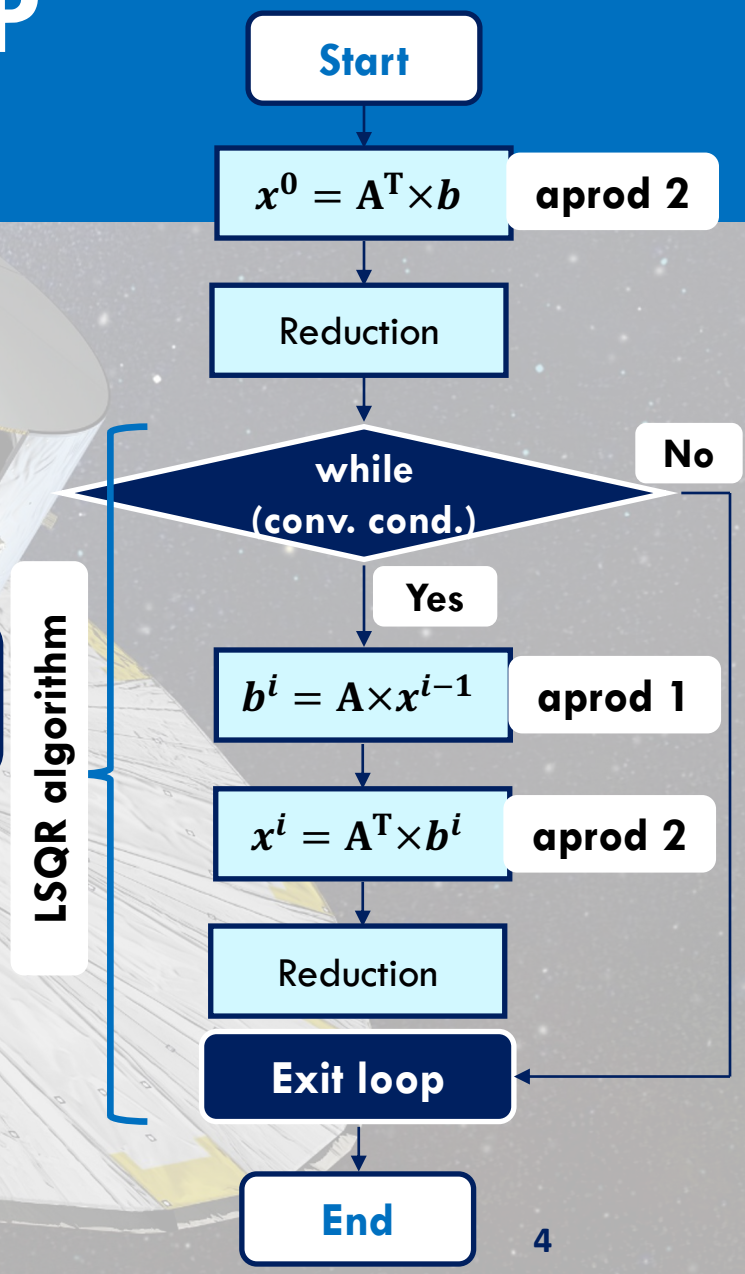
Coefficient matrix:

- Large and sparse ($N_{obs} \times N_{unk} \simeq 10^{11} \times 10^{8}$ elements)
- Computation with a dense matrix $\mathbf{A_d}$ ($\sim 10^{11} \times 10^{1}$ elements)

$$\mathbf{A} \times \boldsymbol{x} = \boldsymbol{b}$$

Solution array: $\sim 10^{8} \times 10^{1}$ elements

Known terms array: $\sim 10^{11} \times 10^{1}$ elements



Becciani et al. (2014)

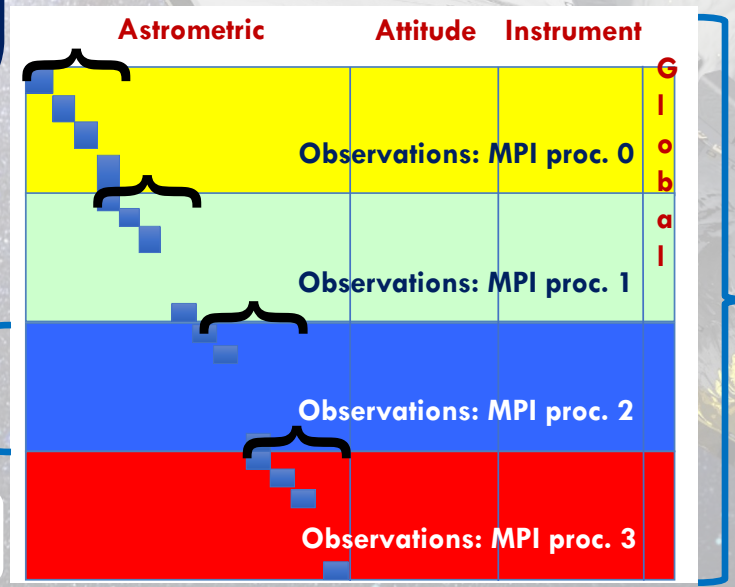
2. Code structure and MPI+OpenMP parallelization

Coefficient matrix:
- Large and sparse ($N_{obs} \times N_{unk} \simeq 10^{11} \times 10^{8}$ elements)
- Computation with a dense matrix $\mathbf{A_d}$ ($\sim 10^{11} \times 10^{1}$ elements)

$$\mathbf{A} \times \boldsymbol{x} = \boldsymbol{b}$$

Solution array: $\sim 10^{8} \times 10^{1}$ elements

Known terms array: $\sim 10^{11} \times 10^{1}$ elements



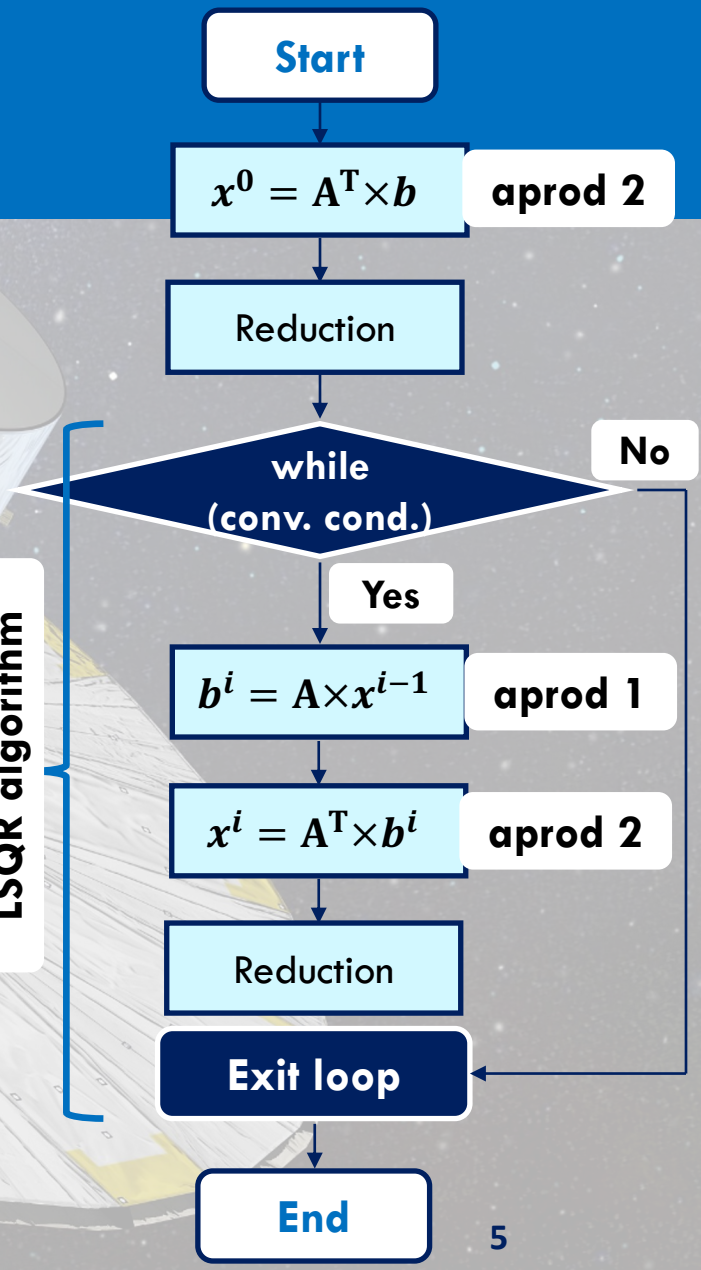
OpenMP threads

$N_{obs} \sim 10^{11}$

$N_{unk} \sim 10^{8}$

Coefficient matrix structure and parallelization.

Becciani et al. (2014)

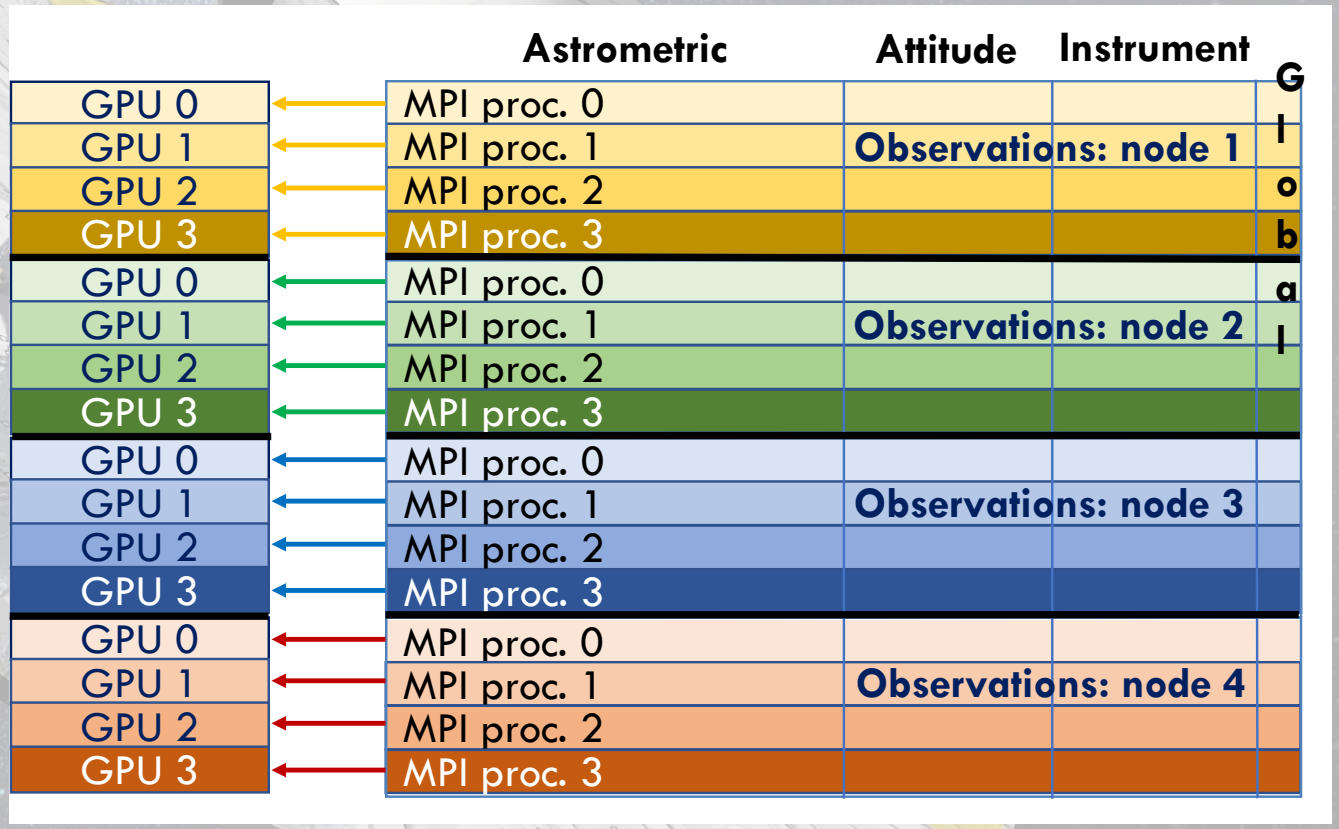


3. The CUDA porting

Cesare et al., in preparation

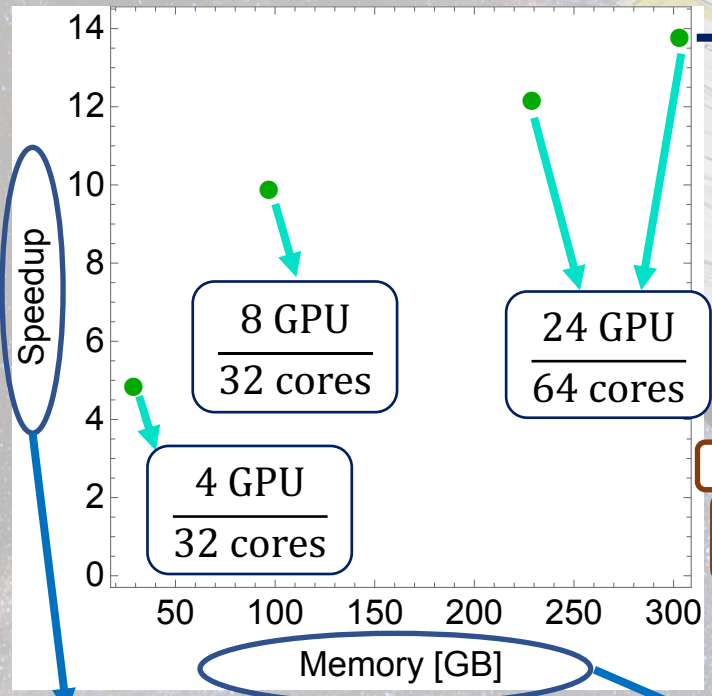
3.1 Multi-GPU computation

- MPI processes assigned to the GPUs of the node in a round-robin fashion
- Tests on CINECA supercomputer **Marconi 100**, with **4 NVIDIA Volta V100 GPUs per node**:
  - 16 GB of memory
  - 84x2048 maximum concurrent threads



Coefficient matrix of the CUDA code parallelized on 4 nodes of a cluster.

3.2 Performance gain



13.81!!!

~9.2x wrt a preliminary OpenACC porting (~1.5x wrt CPU code)

Cesare et al., in press

Cesare et al., sub. to A&C

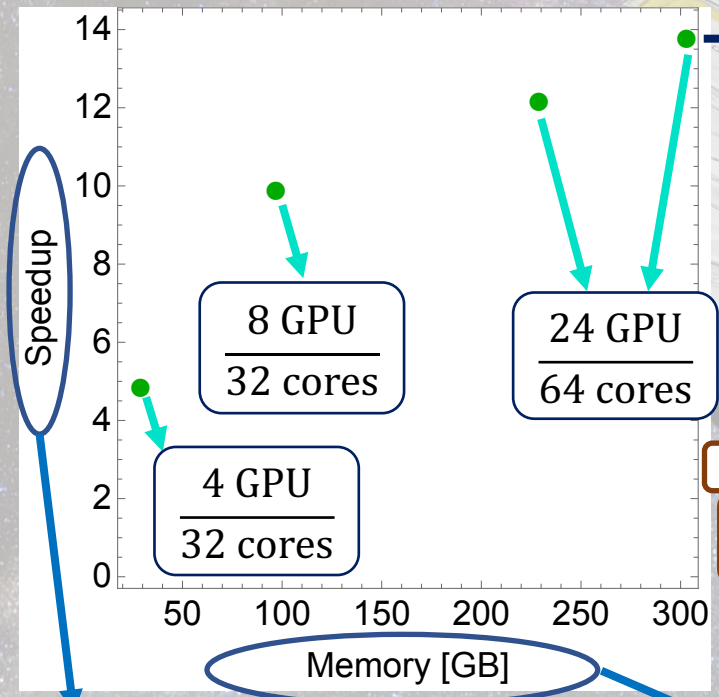
Total memory occupied by the system

Iteration time [s] CPU / Iteration time [s] CUDA

23/06/22

- Speedup increasing with a more efficient utilization of the GPUs and with the system size
- **Speedup of ~14 for the largest system!**

3.2 Performance gain



13.81!!!

~9.2x wrt a preliminary OpenACC porting (~1.5x wrt CPU code)

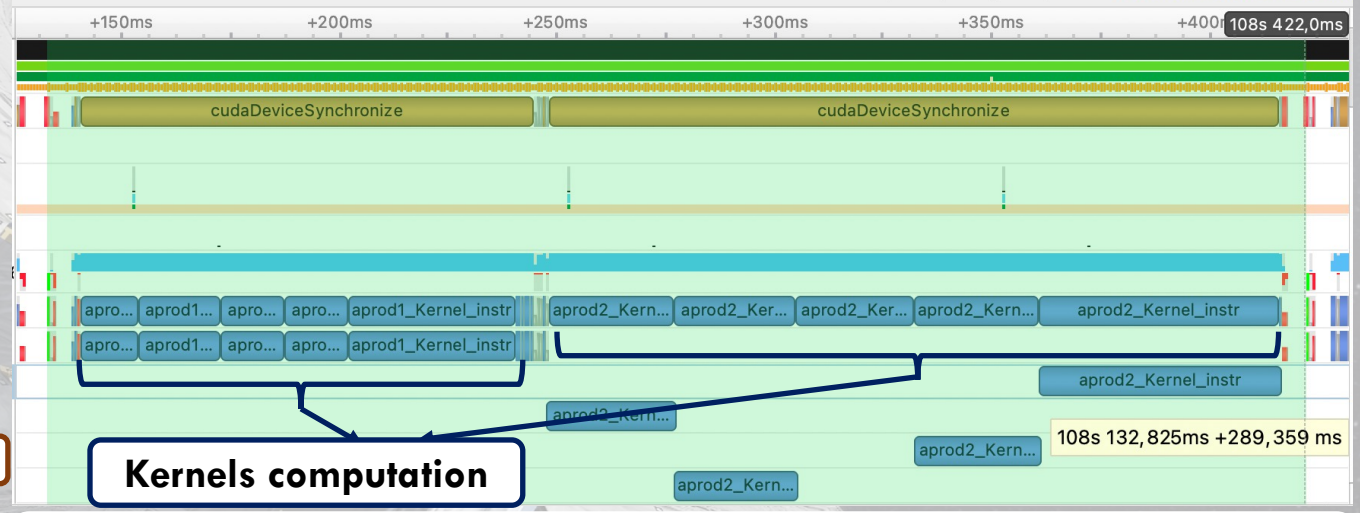
Cesare et al., in press

Cesare et al., sub. to A&C

$$\frac{\text{Iteration time [s] CPU}}{\text{Iteration time [s] CUDA}}$$

Total memory occupied by the system

23/06/22



Output of NVIDIA Nsight system profiler for a 50 GB system parallelized on 4 MPI tasks on one node of Marconi100, for the CUDA code.

- Speedup increasing with a more efficient utilization of the GPUs and with the system size
- **Speedup of ~14 for the largest system!**

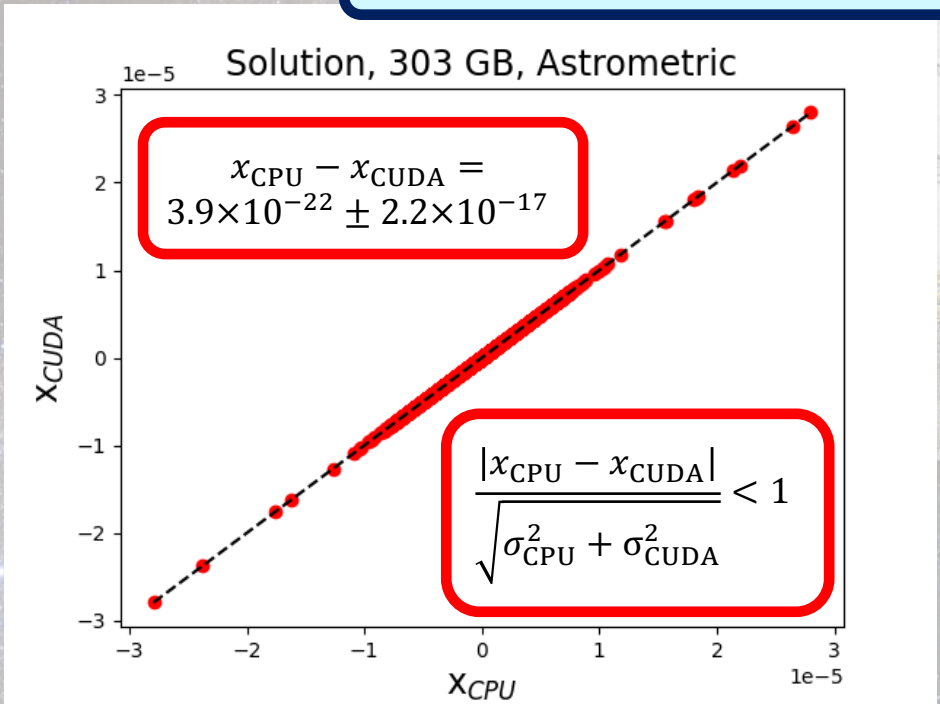
Calculation time dominated by **kernel computation** and not by **data copies and CPU computation**:

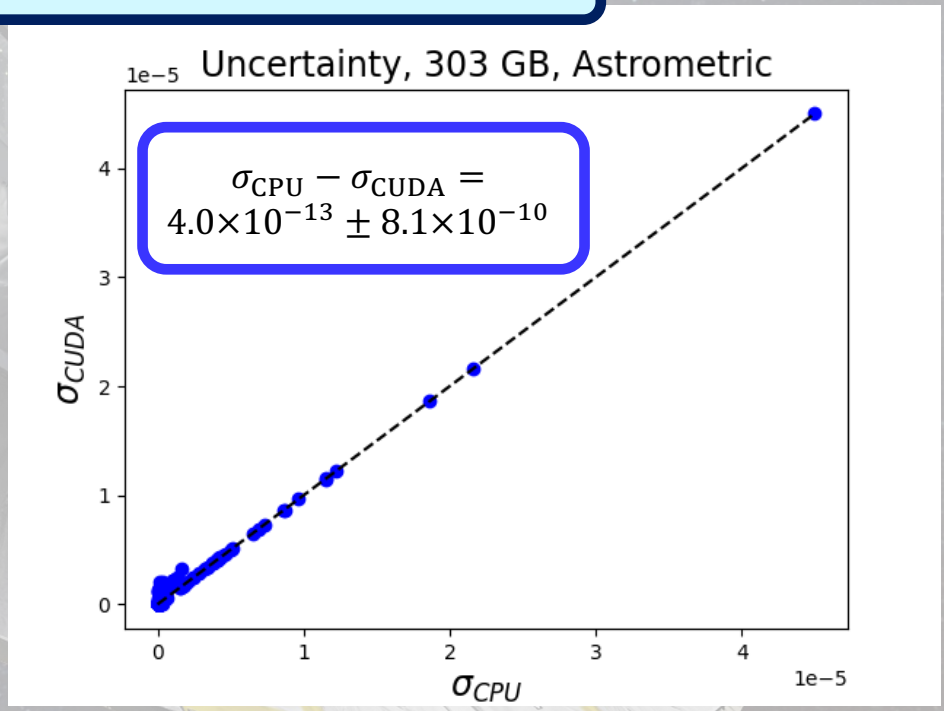
- $\frac{t_{\text{Kernel}}}{t_{\text{Iter}}} \times 100 = 93.5\%$
- $\frac{t_{\text{Copies+CPU}}}{t_{\text{Iter}}} \times 100 = 6.46\%$

3.3 Numerical stability

Comparison between the solutions and their uncertainties found by the CPU and the CUDA codes for a set of different systems.

Example for a system that occupies a memory of 303 GB:





Solutions consistent within 1σ

Differences of the uncertainties consistent with zero

4. Conclusions and outlooks

- Next months: run this application on the pre-exascale platform Leonardo of CINECA:
  - Less performant CPUs
  - Next-generation **A100 GPUs** (4 per node)
    - **Higher memory** each (64-80 GB)
    - **Larger number of Streaming Multiprocessors** ⟹ More threads allowed to run concurrently

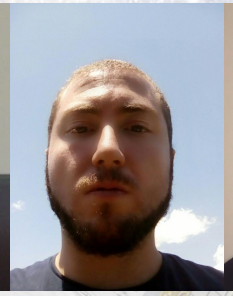
Possible reduction of the power consumption ⟹ **Green computing milestone**

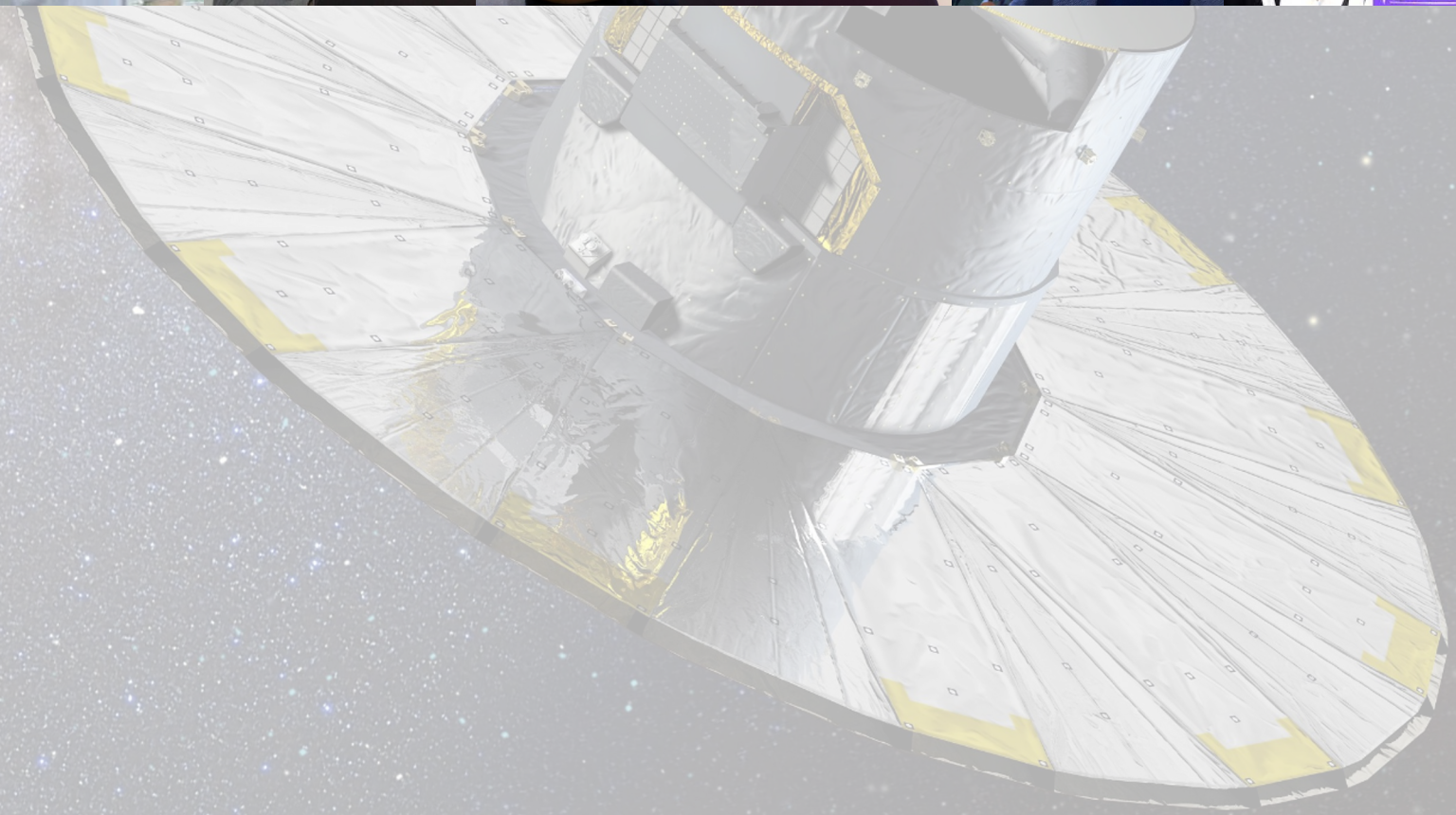
- Extension of the pre-exascale behaviour of this application to other codes having a similar structure, based on the LSQR algorithm, employed in several contexts (e.g. geophysics, medicine, tomography, industry, astronomy)

V. Cesare | U. Becciani | A. Vecchiato | M.G. Lattanzi | F. Pitari | M. Raciti | G. Tudisco | M. Aldinucci | B. Bucciarelli





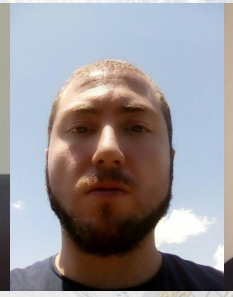


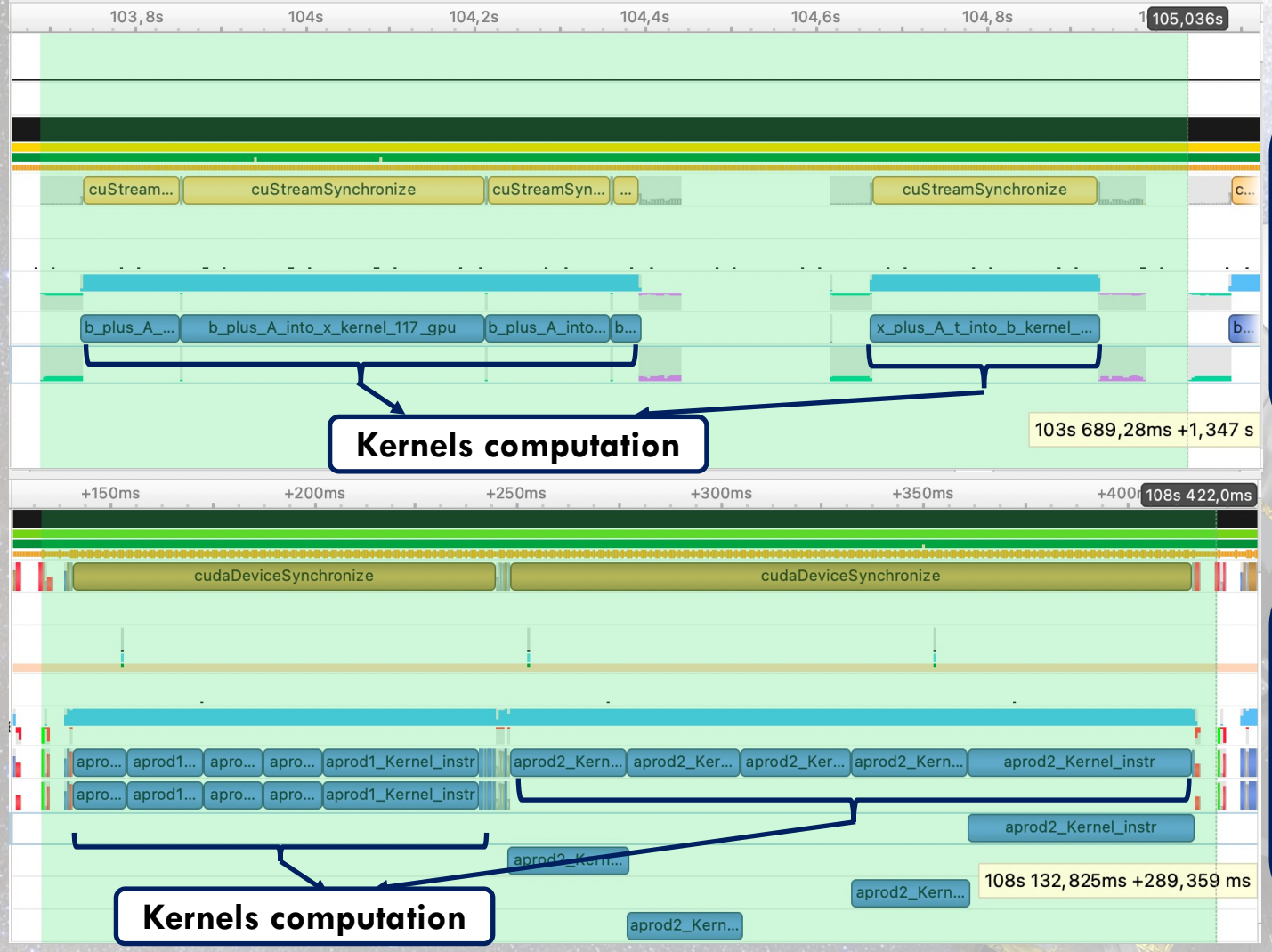
V. Cesare | U. Becciani | A. Vecchiato | M.G. Lattanzi | F. Pitari | M. Raciti | G. Tudisco | M. Aldinucci | B. Bucciarelli








EXTRA SLIDES



Calculation time dominated by **kernel computation** and not **by data copies and CPU computation**:

- $\frac{t_{\text{Kernel}}}{t_{\text{Iter}}} \times 100 = 68.5\%$
- $\frac{t_{\text{Copies+CPU}}}{t_{\text{Iter}}} \times 100 = 31.5\%$

Calculation time dominated by **kernel computation** and not by **data copies and CPU computation**:

- $\frac{t_{\text{Kernel}}}{t_{\text{Iter}}} \times 100 = 93.5\%$
- $\frac{t_{\text{Copies+CPU}}}{t_{\text{Iter}}} \times 100 = 6.46\%$

Better exploitation of **kernel computation** wrt **data copies and CPU computation** in the CUDA code compared to the OpenACC code.