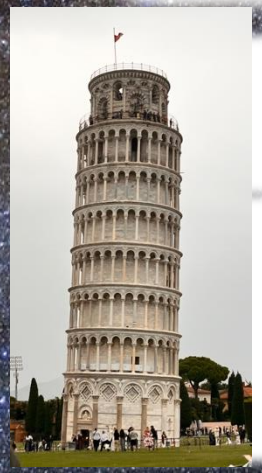
The Gaia AVU-GSR solver: CPU+GPU parallel code toward Exascale systems



V. Cesare[1], U. Becciani[1], A. Vecchiato[2], M. G. Lattanzi[2], M. Aldinucci[3], B. Bucciarelli[2]
(1) INAF-OACT, (2) INAF-OATO, (3) UniTO

Workshop "Scientific HPC in the pre-Exascale era"
part of ITADATA 2024
Pisa, Italy
September 18th 2024

25/09/2024

Table of contents

1. The ESA Gaia mission

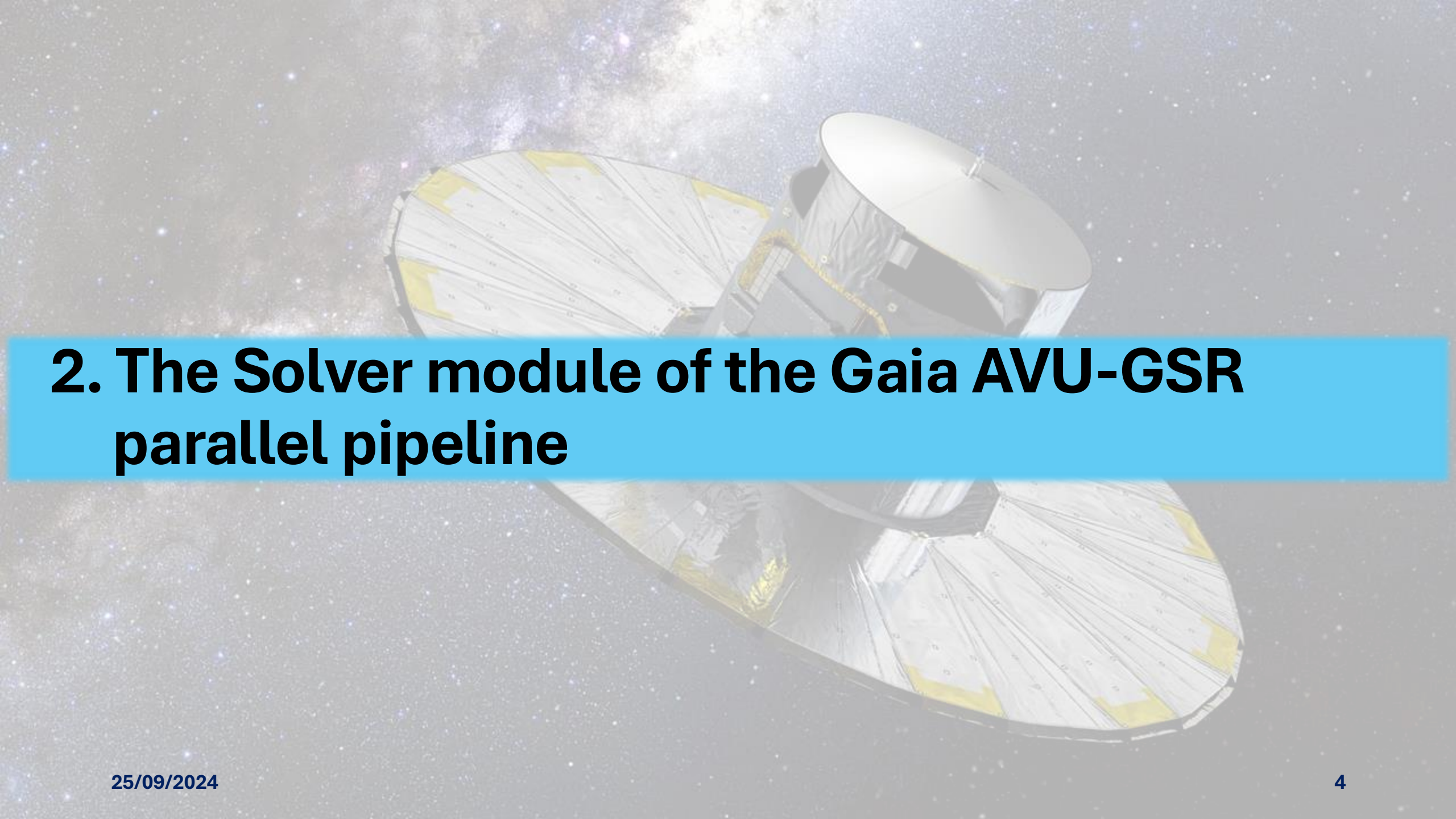
Target of the Solver module of the AVU-GSR pipeline:
Derivation of positions and proper motions of ~10^8 stars (primary stars) in the Milky Way observed with the Gaia satellite with a total of 10^{11} observations, with a [10,100] μas accuracy.

Gaia launch from Guyana Space Center– ESA/CNES/Arianespace

The Gaia mission

- **Developed by**: European Space Agency (ESA)
- **Duration:** December 19th 2013 – 2018 (extended to 2025).
- **Data Release 3:** Published on June 13th 2022
- **Objectives:**
  - Astrometry: map of positions and proper motions of the stars in our Galaxy
  - Origin and evolution of the Milky Way
  - Test of theories of gravity
- **Website:** https://sci.esa.int/web/gaia


Gaia spacecraft - ESA-D. Ducros (2013)



2. The Solver module of the Gaia AVU-GSR parallel pipeline

Coefficient matrix:

- Large and sparse ($N_{obs} \times N_{unk} \sim 10^{11} \times (5 \times 10^8)$ elements → **~400 EB!!!**)
- Computation with a dense matrix $\mathbf{A_d}$ (~$10^{11} \times 24$ elements → **~19 TB**)

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

Solution array: ~ 5×10^8 elements → **~4 GB**

Known terms array: ~10^{11} elements → **~400 GB**

10-50 TB of memory: **Big Data problem**

Becciani et al. (2014)

Coefficient matrix:

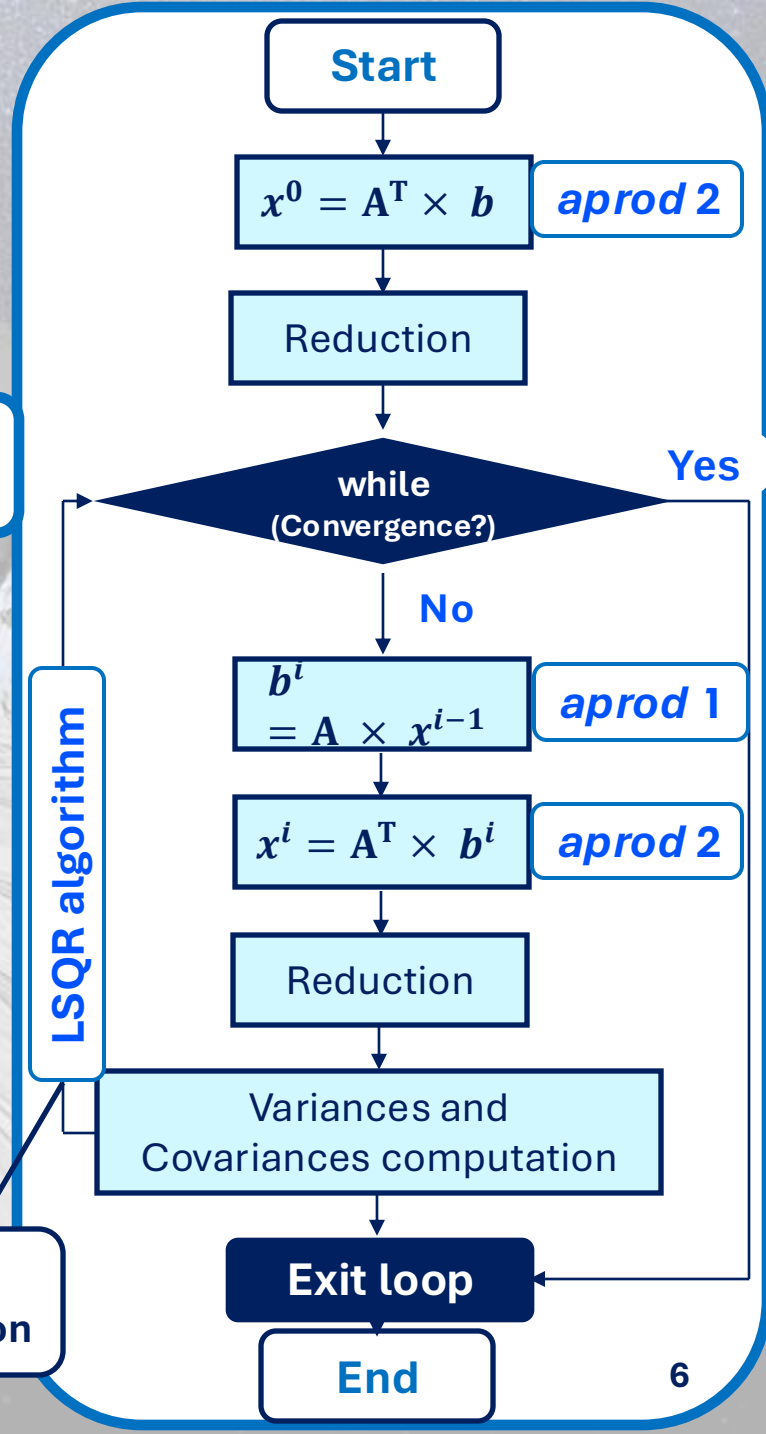
- Large and sparse ($N_{obs} \times N_{unk} \sim 10^{11} \times (5 \times 10^8)$ elements → **~400 EB!!!**)
- Computation with a dense matrix $\mathbf{A_d}$ (~$10^{11} \times 24$ elements → **~19 TB**)

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

Solution array: ~ 5×10^8 elements → **~4 GB**

Known terms array: ~10^{11} elements → **~400 GB**

10-50 TB of memory: **Big Data problem**



Becciani et al. (2014)

Coefficient matrix:

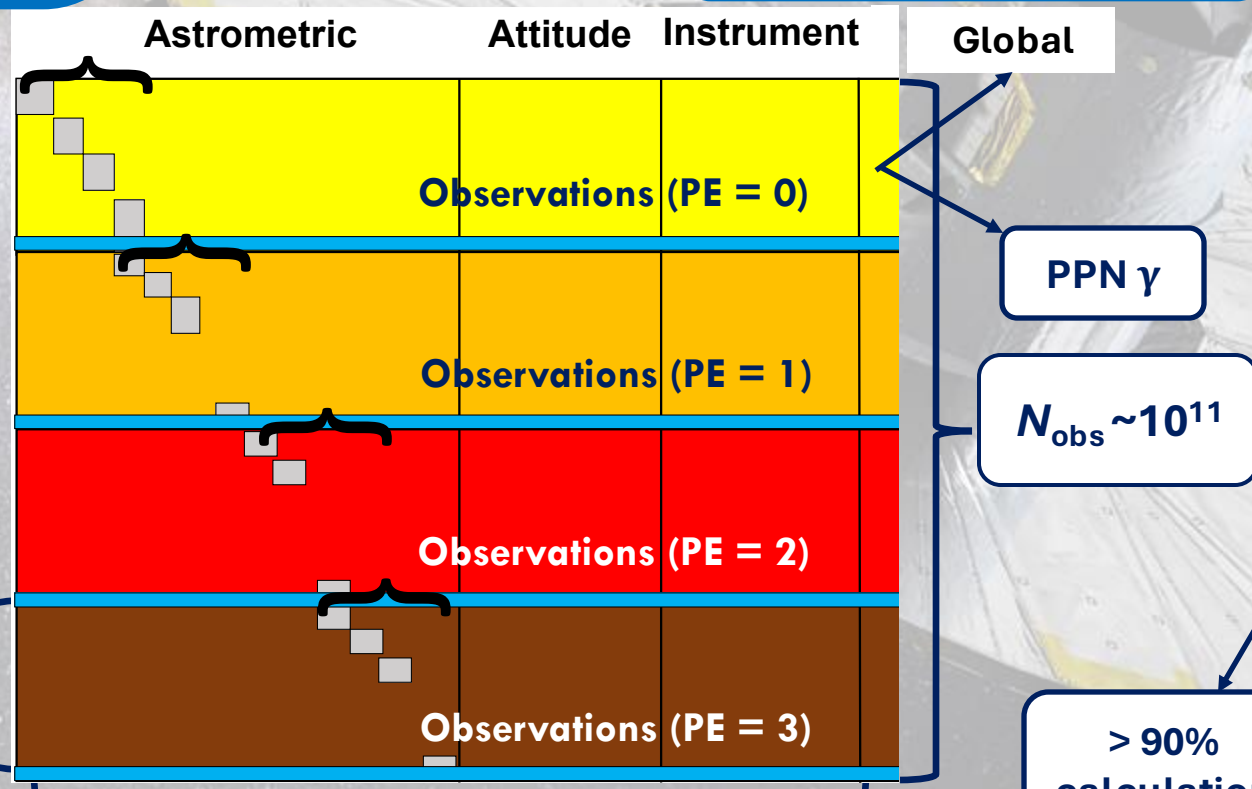
- Large and sparse ($N_{obs} \times N_{unk} \sim 10^{11} \times (5 \times 10^8)$ elements → **~400 EB!!!**)
- Computation with a dense matrix $\mathbf{A_d}$ (~$10^{11} \times 24$ elements → **~19 TB**)

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

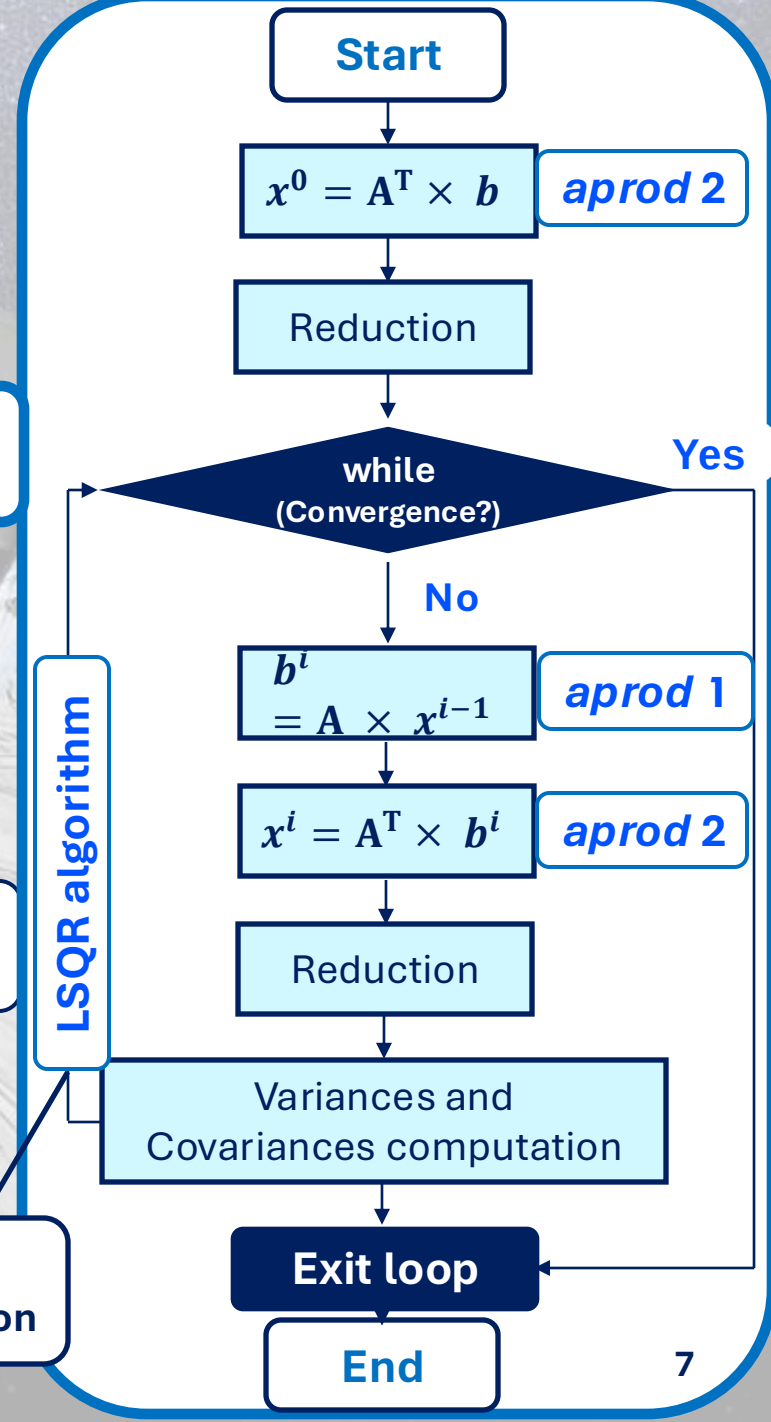
Solution array: ~ 5×10^8 elements → **~4 GB**

Known terms array: ~10^{11} elements → **~400 GB**

10-50 TB of memory: **Big Data problem**



OpenMP threads (CPU version) or CUDA (GPU version) → **14x + 2x speedup over the CPU version.**



> 90% calculation

Becciani et al. (2014)

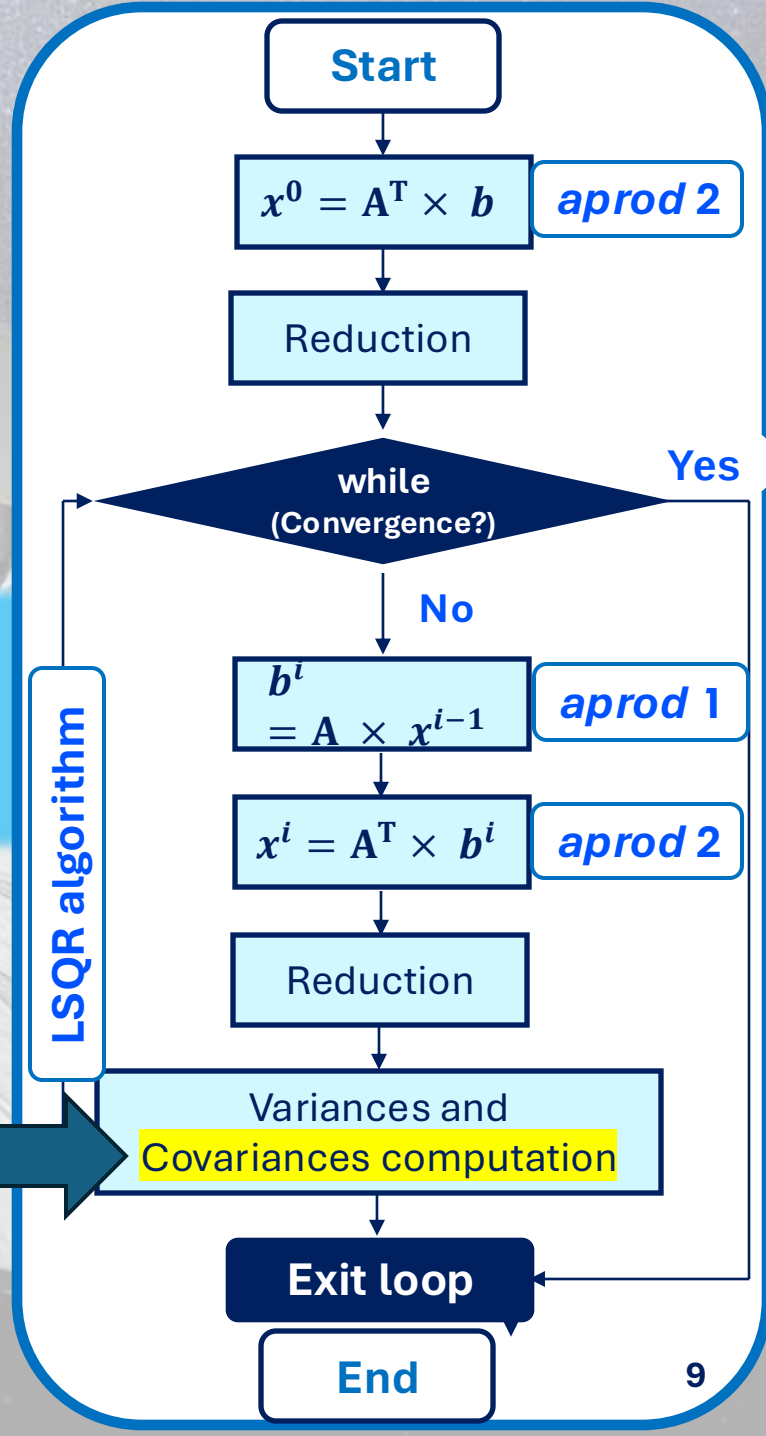
25/09/2024

Malenza, et. al. (2024)

2.1 Computational resources

- Calculation requiring a parallelization over many THIN-like nodes.
- Maximal occupancy of each node resources.
- Application **dominated by computation** and **minimal MPI communications** (~10% of the total execution time) (Malenza, et al., 2024).
- Memory per node limited by the GPU memory of the node since the code runs on GPU.

3. The covariances computation



3.1 Computational problem

- Covariances quantify the correlations between couples of unknowns and the problem faced by Gaia mission presents natural correlations.
- Calculation of the variances-covariances matrix in the Gaia AVU-GSR solver cannot be faced with standard approaches.
- $N_{\mathrm{cov}} = N_{\mathrm{unk}} \times (N_{\mathrm{unk}} - 1)$ total number of covariances (N_{unk} = Number of unknowns).
- $N_{\mathrm{unk}} \sim 5 \times 10^8$ at the end of the Gaia mission. ⇒ ~1 EB of memory. ⇒ Unresolvable problem on existing infrastructures.⇒ Computation of a subset of the total covariances.

N_{unk}

N_{unk}

$$\begin{pmatrix} \sigma^2(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots \\ \mathrm{Cov}(X_1, X_2) & \sigma^2(X_2) & \cdots \\ \cdots & \cdots & \cdots \end{pmatrix}$$

3.1 Computational problem

- Covariances quantify the correlations between couples of unknowns and the problem faced by Gaia mission presents natural correlations.
- Calculation of the variances-covariances matrix in the Gaia AVU-GSR solver cannot be faced with standard approaches.
- $N_{\mathrm{cov}} = N_{\mathrm{unk}} \times (N_{\mathrm{unk}} - 1)$ total number of covariances (N_{unk} = Number of unknowns).
- $N_{\mathrm{unk}} \sim 5 \times 10^8$ at the end of the Gaia mission. ⇒ ~1 EB of memory. ⇒ Unresolvable problem on existing infrastructures.⇒ Computation of a subset of the total covariances.

Covariances calculation:

for $j \leftarrow 0$ to N_{cov} do

$$Cov^{itn}[j] \mathrel{+}= factor^{itn} \times x^{itn}[j_1] \times x^{itn}[j_2]$$

- $\boldsymbol{itn}$**:** LSQR iteration index.
- $0 \leq j_1, j_2 < N_{\mathrm{unk}}$: couples of covariances indexes randomly generated by a separate program.
- $0 \leq j < N_{\mathrm{cov}}$: index of the covariances array, $\overrightarrow{Cov}$.

3.1 Computational problem

- Covariances quantify the correlations between couples of unknowns and the problem faced by Gaia mission presents natural correlations.
- Calculation of the variances-covariances matrix in the Gaia AVU-GSR solver cannot be faced with standard approaches.
- $N_{\mathrm{cov}} = N_{\mathrm{unk}} \times (N_{\mathrm{unk}} - 1)$ total number of covariances (N_{unk} = Number of unknowns).
- $N_{\mathrm{unk}} \sim 5 \times 10^8$ at the end of the Gaia mission. ⇒ ~1 EB of memory. ⇒ Unresolvable problem on existing infrastructures.⇒ Computation of a subset of the total covariances.

Covariances calculation:

for $j \leftarrow 0$ to N_{cov} do

$$\boldsymbol{Cov^{itn}[j] \mathrel{+}= factor^{itn} \times x^{itn}[j_1] \times x^{itn}[j_2]}$$

- $\boldsymbol{itn}$: LSQR iteration index.
- $0 \leq j_1, j_2 < N_{\mathrm{unk}}$: couples of covariances indexes randomly generated by a separate program.
- $0 \leq j < N_{\mathrm{cov}}$: index of the covariances array, $\overrightarrow{Cov}$.

Computational problem:

j_1 and j_2 are global indexes, whereas $\vec{x}$ is a local array → All the MPI processes would have to broadcast to all the other MPI processes the entire $\vec{x}$ array to evaluate $\boldsymbol{x^{itn}[j_1]}$ and $\boldsymbol{x^{itn}[j_2]}$.

Parabolic increase of MPI communications and substantial performance loss.

3.2 The I/O pipeline

LSQR program

Simultaneous start

Covariances program

- **Every $itnCovCP$ iterations, each MPI process prints to file the information of the local $\vec{x}$ related to these iterations.**

- Execution not started **until the files of the first cycle of $itnCovCP$ iterations are not printed** by LSQR program.

3.2 The I/O pipeline

LSQR program

Simultaneous start

Covariances program

- **Every $itnCovCP$ iterations, each MPI process prints to file the information of the local $\vec{x}$ related to these iterations.**

- Execution not started **until the files of the first cycle of $itnCovCP$ iterations are not printed** by LSQR program.
- **Files reading** and **calculation of the correspondent covariances for every cycle of $itnCovCP$ iterations.**
- **Files deletion** before the next printing cycle.

3.2 The I/O pipeline

LSQR program

Simultaneous start

Covariances program

- **Every $itnCovCP$ iterations, each MPI process prints to file the information of the local $\vec{x}$ related to these iterations**.
- Files of the next cycle not printed **until the files of the previous cycle are not deleted** by the covariances program.

- Execution not started **until the files of the first cycle of $itnCovCP$ iterations are not printed** by LSQR program.
- **Files reading** and **calculation of the correspondent covariances for every cycle of $itnCovCP$ iterations.**
- **Files deletion** before the next printing cycle.

3.2 The I/O pipeline

LSQR program ⟷ **Simultaneous start** ⟷ **Covariances program**

- **Every $itnCovCP$ iterations, each MPI process prints to file the information of the local $\vec{x}$ related to these iterations**.
- Files of the next cycle not printed **until the files of the previous cycle are not deleted** by the covariances program.

- Execution not started **until the files of the first cycle of $itnCovCP$ iterations are not printed** by LSQR program.
- **Files reading** and **calculation of the correspondent covariances for every cycle of $itnCovCP$ iterations.**
- **Files deletion** before the next printing cycle.

PIPELINE EFFICIENT IF READING + COVARIANCES PHASE IS SHORTER THAN ITERATION + PRINTING PHASE

3.2 The I/O pipeline

LSQR program

Simultaneous start

Covariances program

- **Every $itnCovCP$ iterations, each MPI process prints to file the information of the local $\vec{x}$ related to these iterations.**
- Files of the next cycle not printed **until the files of the previous cycle are not deleted** by the covariances program.

- Execution not started **until the files of the first cycle of $itnCovCP$ iterations are not printed** by LSQR program.
- **Files reading** and **calculation of the correspondent covariances for every cycle of $itnCovCP$ iterations.**
- **Files deletion** before the next printing cycle.

PIPELINE EFFICIENT IF READING + COVARIANCES PHASE IS SHORTER THAN ITERATION + PRINTING PHASE

- **Execution on GPU and on N nodes,** according to the memory of the system.

- **Sequential execution** on the CPU.

$N + 1$ TOTAL COMPUTATIONAL NODES

3.3 The two versions of the I/O pipeline

Version n. 1

- Each MPI process prints one file every $itnCovCP$ iterations **with the entire local information of $\vec{x}$.**

Version n. 2

- Each MPI process prints two files every $itnCovCP$ iterations **with the local information of $\vec{x}$ related to the covariances to compute.**

3.3 The two versions of the I/O pipeline

Version n. 1

- Each MPI process prints one file every $itnCovCP$ iterations **with the entire local information of $\vec{x}$.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},1} = 8 \times itnCovCP \times N_{\text{unk}}$$

8: double type

Version n. 2

- Each MPI process prints two files every $itnCovCP$ iterations **with the local information of $\vec{x}$ related to the covariances to compute.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},2} = 8 \times itnCovCP \times N_{\text{cov.indexes}}$$

8: double type

$N_{\text{cov.indexes}}$: $2 \times N_{\text{cov}}$

3.3 The two versions of the I/O pipeline

Version n. 1

- Each MPI process prints one file every $itnCovCP$ iterations **with the entire local information of $\vec{x}$.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},1} = 8 \times itnCovCP \times N_{\text{unk}}$$

The print phase only depends on the number of unknowns of the system.

Version n. 2

- Each MPI process prints two files every $itnCovCP$ iterations **with the local information of $\vec{x}$ related to the covariances to compute.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},2} = 8 \times itnCovCP \times N_{\text{cov.indexes}}$$

The print phase only depends on the number of covariances indexes.

3.3 The two versions of the I/O pipeline

Version n. 1

- Each MPI process prints one file every $itnCovCP$ iterations **with the entire local information of $\vec{x}$.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},1} = 8 \times itnCovCP \times N_{\text{unk}}$$

The print phase only depends on the number of unknowns of the system.

Version n. 2

- Each MPI process prints two files every $itnCovCP$ iterations **with the local information of $\vec{x}$ related to the covariances to compute.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},2} = 8 \times itnCovCP \times N_{\text{cov.indexes}}$$

The print phase only depends on the number of covariances indexes.

PRINT PHASE MORE EFFICIENT IN VERSION N. 1 WHEN $N_{\text{cov.indexes}} > N_{\text{unk}}$.

3.3 The two versions of the I/O pipeline

Version n. 1

- Each MPI process prints one file every $itnCovCP$ iterations **with the entire local information of $\vec{x}$.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},1} = 8 \times itnCovCP \times N_{\text{unk}}$$

The print phase only depends on the number of unknowns of the system.

Version n. 2

- Each MPI process prints two files every $itnCovCP$ iterations **with the local information of $\vec{x}$ related to the covariances to compute.**
- Total size printed every $itnCovCP$ iterations:

$$Size_{\text{Cycle},2} = 8 \times itnCovCP \times N_{\text{cov.indexes}}$$

The print phase only depends on the number of covariances indexes.

PRINT PHASE MORE EFFICIENT IN VERSION N. 1 WHEN $N_{\text{cov.indexes}} > N_{\text{unk}}$.

How many and which covariances to calculate is set in the covariances program.

How many and which covariances to calculate is already set in the LSQR program.

4. Results

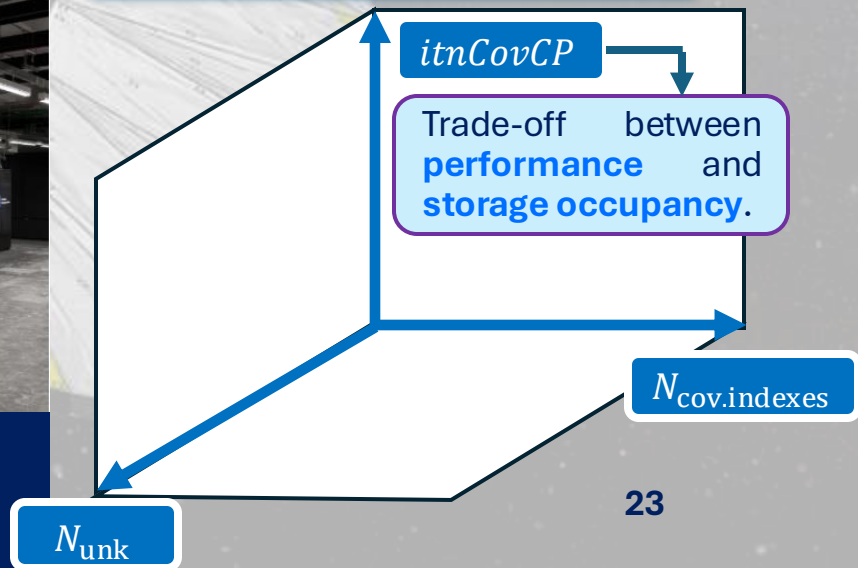
Computing platform: Leonardo CINECA infrastructure

- Platform used for the production of the AVU-GSR pipeline.
- 1 x CPU Intel Xeon 8358 32 cores, 2.6 GHz. 512 GB in total.
- **4 x 64 GB GPU NVIDIA Ampère A100. 256 GB in total.**
- **Storage: capacity tier of 106 PB and with a 620 GB/s bandwidth** and fast tier of 5.4 PB and with a 1.4 TB/s bandwidth.

Tests performed with synthetic data, distributed in the system as the real data, to directly control the memory occupied by the considered systems.

Leonardo supercomputer: https://leonardo-supercomputer.cineca.eu/it/leonardo-hpc-system/

Parameter space to be explored for the tests



TEST N. 1

Settings

- System of 244 GB (95% GPU memory), $N_{\text{unk}} = 2.55 \times 10^6$, 1 node of Leonardo, 4 MPI processes
- $N_{\text{cov. indexes}} \in [10^1, 10^8]$, 1 dex intervals.

Results.

- $t_{\text{Read+Cov,Cycle}} < t_{\text{Iter+Print,Cycle}}$ **(pipeline efficient)** up to $N_{\text{cov. indexes}} \sim 1.6 \times 10^7$ for both versions and up to $N_{\text{cov. indexes}} \sim 6.3 \times 10^7$ for version n. 1.

TEST N. 2

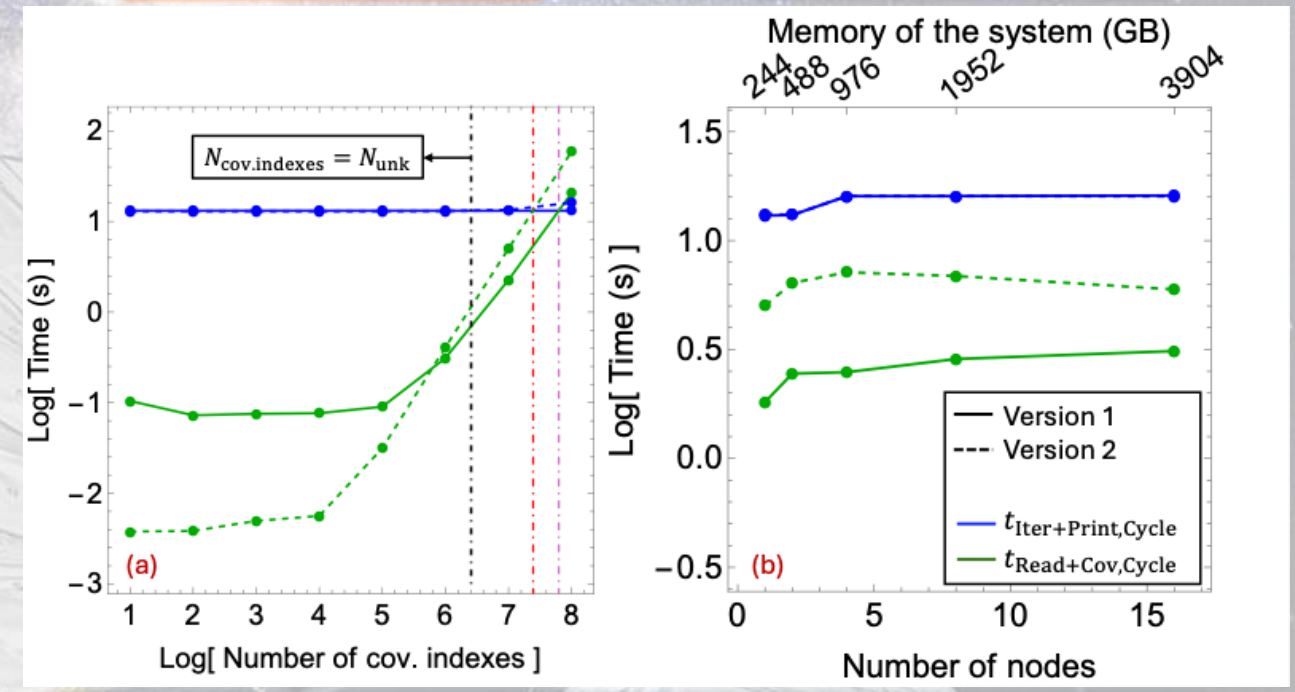
Settings

- System of 244 GB per node, 4 MPI processes per node, 1-16 nodes; N_{unk} does not necessarily increase with the number of nodes.
- $N_{\text{cov.Indexes}} = 10^7 \Rightarrow N_{\text{cov}} = 5 \times 10^6$.

Results

- Good scalability.
- $t_{\text{Read+Cov,Cycle}} < t_{\text{Iter+Print,Cycle}}$ always **(pipeline efficient)**.

TEST N. 1 **TEST N. 2**



100 LSQR iterations, $itnCovCP = 20 \Rightarrow$ 5 cycles.

LSQR program:

- $t_{\text{Iter+Print,Cycle}}$: Duration of a cycle of $itnCovCP$ iterations ($itnCovCP$ LSQR iterations + 1 print).

Covariances program:

- $t_{\text{Read+Cov,Cycle}}$: Duration of a cycle of $itnCovCP$ iterations (reading + covariances calculation).

4.1 Storage considerations

- When covariances will be computed, typically $N_{\text{cov. indexes}} > N_{\text{unk}} \Rightarrow$ Version n. 1 of the pipeline will be preferred.

- Computation of covariances **not planned in all future production runs**.

- Possibly, the AVU-GSR solver and the covariances program will be executed in sequence, preserving all the files, **to decide at a second stage which covariances to compute, not to slowdown the AVU-GSR production** (possible only with version n. 1).

- **More covariances sets** computable from the same set of output files.

- System memory = 3.9 TB (16 nodes), $Size_{\text{Cycle},1} = 2.2$ GB, number of iterations $\sim 3 \times 10^5 \Rightarrow$ total size printed at the end of the run = 33 TB $\sim$ $\mathbf{10^{-4}}$ **total storage capacity of Leonardo capacity tier (no storage issue)**.

5. Conclusions and Outlooks

- We have presented the Gaia AVU-GSR solver, to find the astrometric parameters of $\sim 10^8$ stars in the Milky Way
- We have explored the performance of the covariances calculation along the $N_{\text{cov.indexes}}$ and $Mem =$ "Memory occupied by the system" axes, **verifying its efficiency up to** $\boldsymbol{N_{\text{cov. indexes}} \sim 1.6 \times 10^7 \Rightarrow N_{\text{cov}} \sim 8.0 \times 10^6}$ for both pipelines' versions.

Outlooks about covariances calculation:

- Extension of test n. 2 up to **256 nodes** of Leonardo (**62 TB system, comparable to final expected sizes**).
- Performance exploration along the $\boldsymbol{N_{\text{unk}}}$ **axis**
- Performance exploration along the $\boldsymbol{itnCovCP}$ axis
- Performance tests with **real instead of synthetic data**.

The results obtained in this work are part of the targets of an INAF Grant (in collaboration with Prof. Marco Aldinucci of the University of Turin) and of Italian National Center of HPC, Big Data, and Quantum Computing.

- This work has been supported by the Spoke 1 “FutureHPC & BigData” of the ICSC — Centro Nazionale di Ricerca in High Performance Computing, Big Data and Quantum Computing — and hosting entity, funded by European Union — Next GenerationEU.
- This work was also supported by the Italian Space Agency (ASI) [grant No.: 2018-24-HH.0], in support of the Italian participation to the Gaia mission, and by Consorzio Interuniversitario Nazionale per l’Informatica (CINI), under the project EUPEX, EC H2020 RIA, EuroHPC-02-2020 [Grant Agreement: 101033975].
- The results reported in this work are part of foreground knowledge of the project “Investigation of Scalability, Numerical Stability, and Green Computing of LSQR-based applications involving Big Data in perspective of Exascale systems: the ESA Gaia mission case study” minigrant awarded to Dr. Valentina Cesare by INAF.

V. Cesare

U. Becciani

A. Vecchiato

M. G. Lattanzi

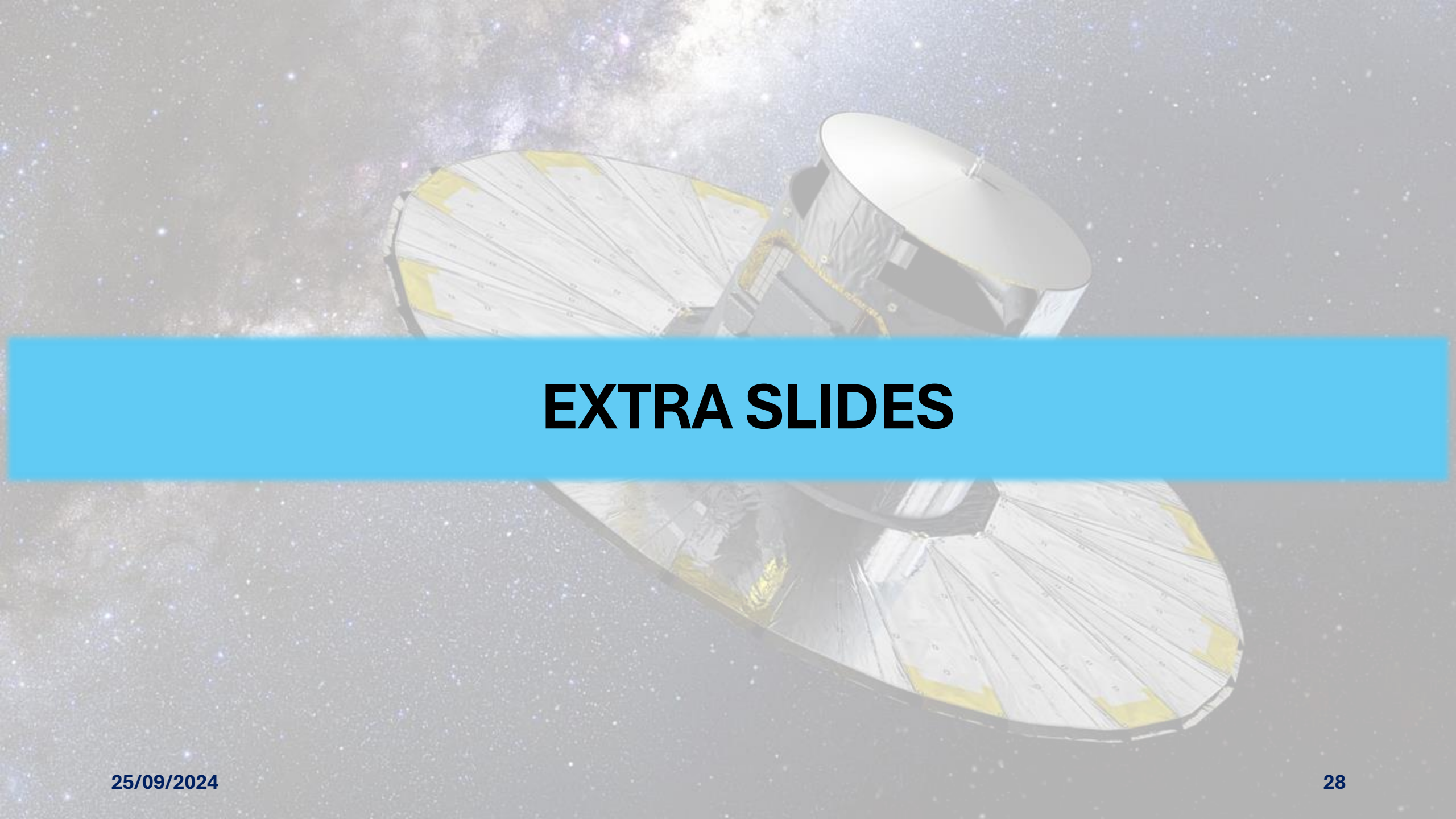
M. Aldinucci

B. Bucciarelli

Thank you for your attention!!!

☺

Questions?



EXTRA SLIDES

A. Global performance of the covariances pipeline

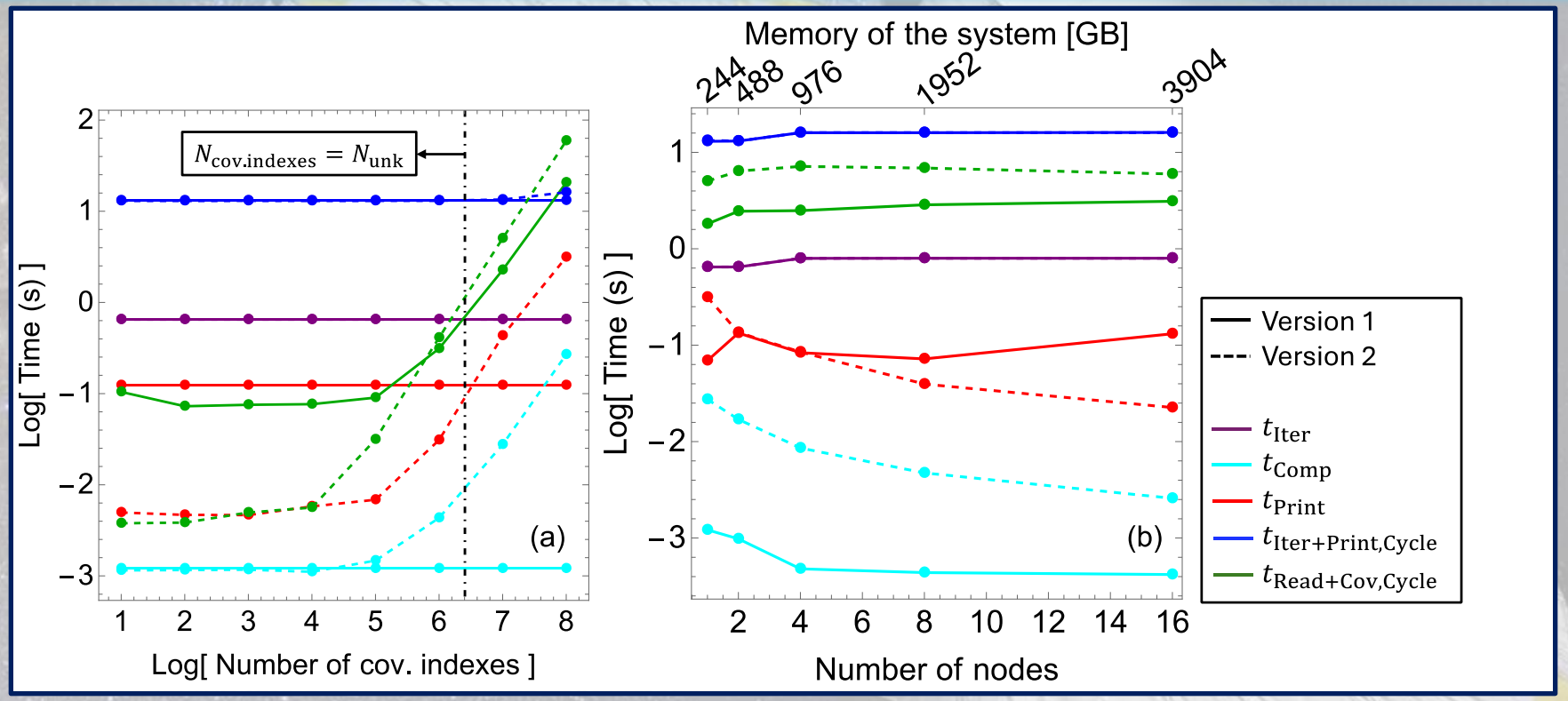


Figure 1 from Cesare, et al., (2024), https://doi.org/10.1117/12.3018102

B. Important caveats

- System of 3.9 TB (16 nodes of Leonardo, 244 GB per node), ~40% of the size expected at the end of the Gaia mission: $N_{\mathrm{unk}} = 1.38 \times 10^{7}$.
- $N_{\mathrm{cov.indexes}} = 10^{7}$.
- $itnCovCP = 20$.
- Number of iterations for convergence: $N_{\mathrm{iter}} \sim 5 \times 10^{6}$.

Version n. 1

55 TB printed at the end of the run
(2.2 GB per cycle, with a total of 1.6×10^{6} files.)

Version n. 2

40 TB printed at the end of the run
(1.6 GB per cycle, with a total of 3.2×10^{6} files.)

B. Important caveats

- System of 3.9 TB (16 nodes of Leonardo, 244 GB per node), ~40% of the size expected at the end of the Gaia mission: $N_{\mathrm{unk}} = 1.38 \times 10^{7}$.
- $N_{\mathrm{cov.indexes}} = 10^{7}$.
- $itnCovCP = 20$.
- Number of iterations for convergence: $N_{\mathrm{iter}} \sim 5 \times 10^{6}$.

Version n. 1

55 TB printed at the end of the run
(2.2 GB per cycle, with a total of 1.6×10^{6} files.)

Version n. 2

40 TB printed at the end of the run
(1.6 GB per cycle, with a total of 3.2×10^{6} files.)

- **We do not expect to calculate covariances in each production run.**
- Possibility of running LSQR program first, not deleting files, and running covariances program afterwards with **version n. 1 of the pipeline,** to decide at a second stage how many and which covariances to calculate according to scientific needs.