CINECA

Software Track - SPUMA benchmarking on 236M cells occDrivAerStaticMesh test case

01 / 07 / 2025

Simone Bnà (s.bn@cineca.it)

Giuseppe Giaquinto (g.giaquinto@cineca.it)

Tommaso Zanelli (t.zanelli@cineca.it)

WHAT IS SPUMA

CINECA

- **SPUMA** (**S**imulation **P**rocessing on **U**nified **M**emory **A**ccelerators) is a **minimally invasive** porting of OpenFOAM to GPU under development by Cineca.
- It is **portable** across different hardware (e.g. x86, Nvidia, AMD)
- Native OpenFOAM smoothers (Gauss-Seidel, DIC) cannot be trivially parallelized on GPU accelerators
- Additional algorithms were implemented to provide **scalable** parallel smoothers
  - Chebyshev
  - Two-stage Gauss-Seidel
  - Richardson

Chebyshev polynomial smoother

Given a polynomial degree n, the $n+1$ approximate solution to $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ is given by:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^{n} + \alpha_0^n(\boldsymbol{x}^{n} - \boldsymbol{x}^{n-1}) + \alpha_1^n \boldsymbol{D}^{-1}(\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{n})$$

Where: $\boldsymbol{D} = \mathrm{diag}(\boldsymbol{A})$, $\alpha_0^0 = 0$, $\alpha_1^0 = \rho^n \rho^{n-1}$, $\alpha_1^n = \frac{4\rho^n}{\lambda_{\max} - \lambda_{\min}}$,

$$\rho^0 = \frac{\lambda_{\max} - \lambda_{\min}}{\lambda_{\max} + \lambda_{\min}} \text{ and } \rho^n = \left(2\frac{\lambda_{\max} + \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}} - \rho^{n-1}\right)^{-1}$$

The **Chebyshev polynomial** is constructed to strongly damp eigenvalues in the range $[\lambda_{\min}, \lambda_{\max}]$.

In the actual implementation, the value of $\lambda_{\max}$ is either set a priori or estimated via the **Gershgorin theorem**, while $\lambda_{\min}$ is estimated as $\lambda_{\min} = \frac{1}{8}\lambda_{\max}$

Two-stage Gauss-Seidel smoother

In the **Gauss-Seidel** method the approximate solution to the linear system system $\boldsymbol{A}\,\boldsymbol{x} = \boldsymbol{b}$ is given, at each iteration, by:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^{n} + \boldsymbol{L}^{-1}\,(\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{n})$$

Where $\boldsymbol{L}$ is the lower triangular portion of $\boldsymbol{A}$.

$\boldsymbol{L}^{-1}$ can be approximated as:

$$\sum_{j=0}^{s} (-\boldsymbol{D}^{-1}\,\boldsymbol{L})^{j}\,\boldsymbol{D}^{-1}$$

For $s = 1$ the **two-stage Gauss-Seidel** method is obtained, and the n + 1 iteration of $\boldsymbol{x}$ is given by:

$$\boldsymbol{x}^{n+1} = \boldsymbol{x}^{n} + (\boldsymbol{I} - \boldsymbol{D}^{-1}\,\boldsymbol{L})\boldsymbol{D}^{-1}(\boldsymbol{b} - \boldsymbol{A}\,\boldsymbol{x}^{n})$$

HARDWARE USED

CINECA

- Benchmarking tests were run on Cineca's Leonardo cluster.
- Leonardo partitions are:

Booster

- **116** BullSequana XH2000 Direct Liquid cooling racks
- **240** PFLOPs HPL Linpack Performance (Rmax)
- **3456** computing node servers
- **13824 NVIDIA Ampere** GPUs (**A100**)
- **884** TB of HBM2

Data Centric

16 BullSequana XH2000 Direct Liquid cooling racks

9 PFLOPs HPL Linpack Performance (Rmax)

1536 computing node servers

3072 Intel **Sapphire Rapids 8480+** CPUs

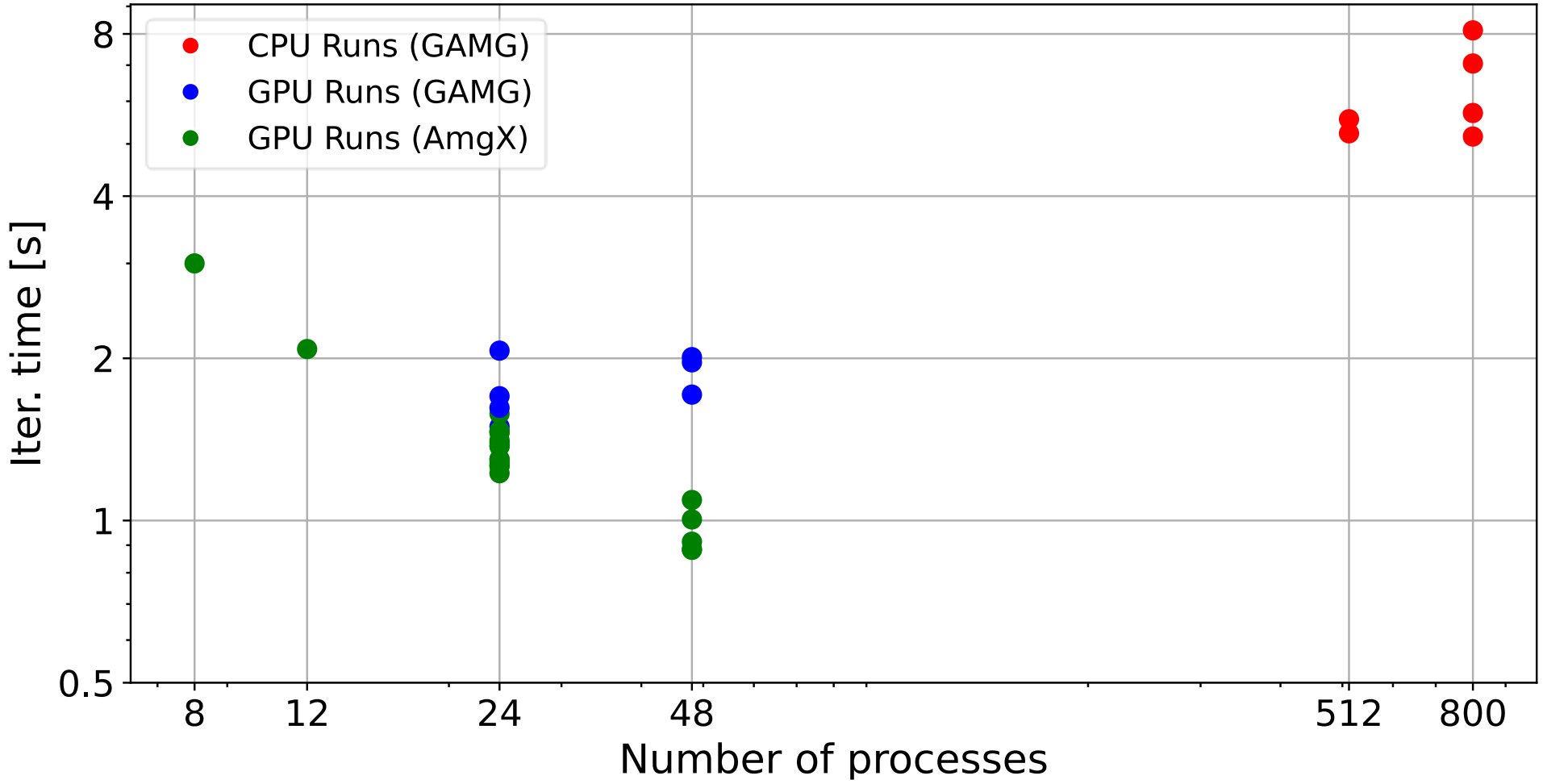
786 TB DDR5 Memory

5.8 PB of local NVM

BENCHMARK RESULTS

CINECA

Summary



BENCHMARK RESULTS

CINECA

Notes

- Fixed iteration runs took consistently longer than **fixed tolerance** runs.
- For GPU runs, the best setup was:
  - **AmgX** (PCG solver, AMG preconditioner, Richardson smoother, direct solver for coarsest level) for the pressure equation
  - **Two-stage Gauss-Seidel** for Velocity and turbulence
- Mixed precision tested with native solvers only (no AmgX) **up to ~17%** performance improvement
- GAMG did not scale effectively from 24 to 48 GPUs
- Fastest run (48 GPUs) took **3615 seconds** total