Performance analysis of parallel-in-time techniques in modern supercomputers

J. Plana-Riu[1], F.X. Trias[1], À. Alsalti-Baldellou[2], G. Colomer[1], A. Oliva[1]

[1]Heat and Mass Transfer Technological Centre
Technical University of Catalonia
[2]Department of Civil, Environmental and Architectural Engineering (ICEA)
University of Padova

June 6th 2024

Introduction

Improving memory-boundness

- According to the roofline model[1], CFD simulations are memory-bound, i.e. performance bounded by data rate instead of flops.
- Let's introduce the arithmetic intensity concept...

$$AI = \frac{\text{Floating point operations}}{\text{Data to transfer}}$$

- Reducing amount of data to transfer, AI will increase "for free".
  - Exploiting symmetries[2]
  - Running multiple flow states simultaneoslly[3]
- How is it done?

[1] S. Williams et al., "Roofline: an insightful visual performance for multicore architectures," *Commun. ACM* **52**, 2009.

[2] À. Alsalti-Baldellou et al., "Lighter and faster simulations on domains with symmetries," *Comput. Fluids* **275**, 2024.

[3] B.I. Krasnopolsky, "An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles," *Comput. Phys. Commun.* **229**, 2018

Introduction

Improving memory-boundness

- Bottleneck is usually the sparse matrix-vector product (`SpMV`) (or equivalent in stencil-based implementations).
- Assume we need to do more than one `SpMV`:

Introduction

Improving memory-boundness

- Bottleneck is usually the sparse matrix-vector product (SpMV) (or equivalent in stencil-based implementations).
- Assume we need to do more than one SpMV:

$$\begin{pmatrix} A & \\ & A \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} \qquad\qquad A(u_1 \quad u_2)$$

2 SpMV

1 SpMM with 2 RHS

Introduction

Improving memory-boundness

- Bottleneck is usually the sparse matrix-vector product (SpMV) (or equivalent in stencil-based implementations).
- Assume we need to do more than one SpMV:

$$\begin{pmatrix} A & \\ & A \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix}$$

2 SpMV

$$A(u_1 \quad u_2)$$

1 SpMM with 2 RHS

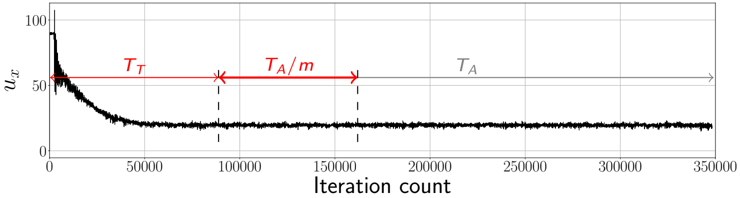
Comparison between operations

- Number of floating point operations is the same in both cases
- Need to transfer A only once, AI for SpMM is higher.

Parallel-in-time

Exploiting multiple right-hand sides...

- Instead of running a long simulation, why not running m shorter ones in the same device, and average the results?



- Exploiting the speed-up in `SpMM` can lead to speed-up in the simulation.

Parallel-in-time

- Running a more expensive case for $T_T + \frac{T_A}{m}\ldots$
  - T_T will now be more expensive
  - The speed-up in the averaging part is **key**

Key parameter, *times ratio*

$$\beta = \frac{T_A}{T_T}$$

- The bigger is β, the bigger the effectiveness of the method.
- According to Kransopolski[3],

$$P_m = \frac{1+\beta}{m+\beta}\frac{5m}{5m-3\theta(m-1)}$$

  - θ, weight of the Poisson equation in the iteration
  - Speed-up of the iteration, $P_{m,ite}$
  - Extension to the whole simulation

Parallel-in-time

- Running a more expensive case for $T_T + \frac{T_A}{m}$...
  - T_T will now be more expensive
  - The speed-up in the averaging part is **key**

Key parameter, *times ratio*

$$\beta = \frac{T_A}{T_T}$$

- The bigger is β, the bigger the effectiveness of the method.
- According to Kransopolski[3],

$$P_m = \frac{1+\beta}{m+\beta} \frac{5m}{5m - 3\theta(m-1)}$$

  - θ, weight of the Poisson equation in the iteration
  - Speed-up of the iteration, $P_{m,ite}$
  - Extension to the whole simulation

Methodology

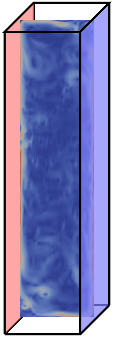
Case definition

Differentially heated cavity of aspect ratio 4

$$\begin{aligned} M\boldsymbol{u}_s &= 0_c, \\ \Omega \frac{d\boldsymbol{u}_c}{dt} + C(\boldsymbol{u}_s)\boldsymbol{u}_c - \frac{\text{Pr}}{\text{Ra}^{1/2}} D\boldsymbol{u}_c + \Omega G_c \boldsymbol{p_c} + \Omega \boldsymbol{f}_c &= 0_c, \\ \Omega \frac{d\boldsymbol{\theta}_c}{dt} + C(\boldsymbol{u}_s)\boldsymbol{\theta}_c - \frac{1}{\text{Ra}^{1/2}} D\boldsymbol{\theta}_c &= 0_c \end{aligned}$$

- 3rd order Heun Runge-Kutta[4], SAT[5]
- Ra=10^{10}, Pr=0.71, $\boldsymbol{f}_c = (0, \text{Pr}\boldsymbol{\theta}, 0)$

$\theta_H = 0.5$



$\theta_C = -0.5$

[4] B. Sanderse and B. Koren, "Accuracy analysis of explicit Runge-Kutta methods applied to the incompressible Navier-Stokes equations," *J. Comput. Phys.* **231**, 2012

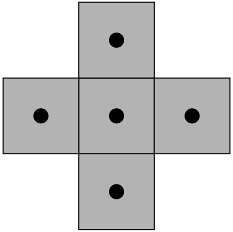
[5] F.X. Trias and O. Lehmkuhl, "A self-adaptive strategy for the time integration of Navier-Stokes equations," *Numer. Heat Transfer* B **60**, 2011

Methodology

Construction of test case

Characteristics

- Run in 1 MN5 GPP-HighMem partition node:
  - 2x Intel Xeon Platinum 8480+ 56C 2GHz
  - 1024GB of RAM memory
  - 2x54 OpenMP threads within the socket + 2 update threads
  - 2 MPI processes (1x socket)
- Mesh:
  - 400k: 220x880x220, 394k cells/CPU
  - 300k: 201x804x201, 300k cells/CPU
  - 200k: 176x704x176, 202k cells/CPU
- Run in three different discretizations for the Laplacian operator: 7p,13p, and 27p
- 1, 2, 4, 8, and 16 RHS

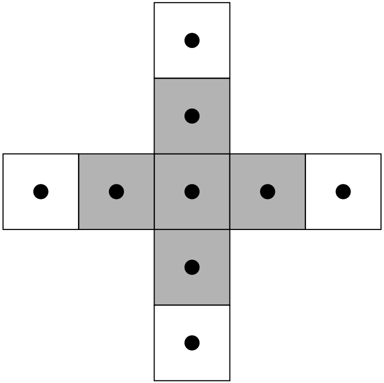


Methodology

Construction of test case

Characteristics

- Run in 1 MN5 GPP-HighMem partition node:
  - 2x Intel Xeon Platinum 8480+ 56C 2GHz
  - 1024GB of RAM memory
  - 2x54 OpenMP threads within the socket + 2 update threads
  - 2 MPI processes (1x socket)
- Mesh:
  - 400k: 220x880x220, 394k cells/CPU
  - 300k: 201x804x201, 300k cells/CPU
  - 200k: 176x704x176, 202k cells/CPU
- Run in three different discretizations for the Laplacian operator: 7p,13p, and 27p
- 1, 2, 4, 8, and 16 RHS

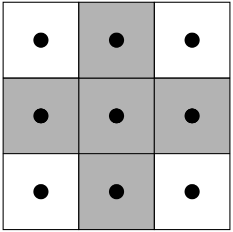


Methodology

Construction of test case

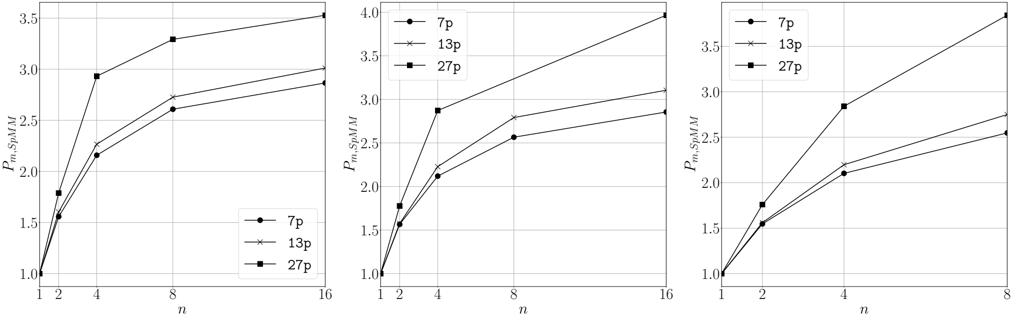
Characteristics

- Run in 1 MN5 GPP-HighMem partition node:
  - 2x Intel Xeon Platinum 8480+ 56C 2GHz
  - 1024GB of RAM memory
  - 2x54 OpenMP threads within the socket + 2 update threads
  - 2 MPI processes (1x socket)
- Mesh:
  - 400k: 220x880x220, 394k cells/CPU
  - 300k: 201x804x201, 300k cells/CPU
  - 200k: 176x704x176, 202k cells/CPU
- Run in three different discretizations for the Laplacian operator: 7p,13p, and 27p
- 1, 2, 4, 8, and 16 RHS



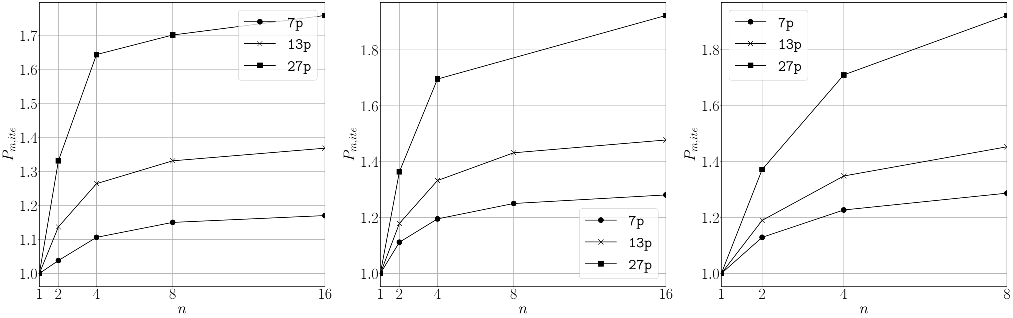
Results

Dependency on the discretization of the Laplacian operator. Speedup in SpMM



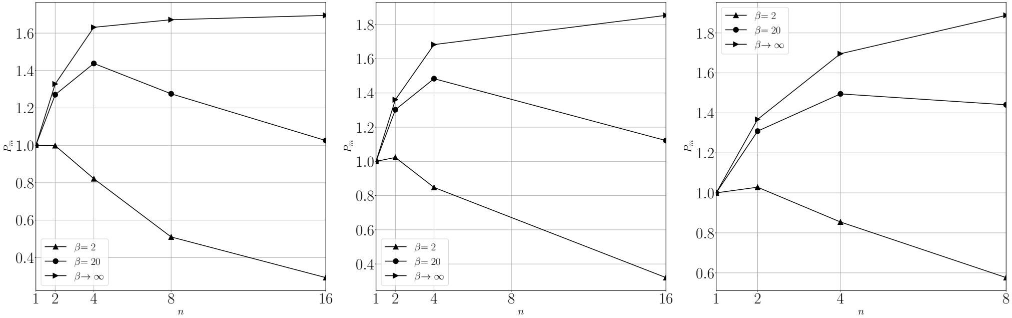
Results

Dependency on the discretization of the Laplacian operator. Speedup in the whole iteration



Results

Dependency on the discretization of the Laplacian operator. Speedup in the whole simulation, 27p



Results

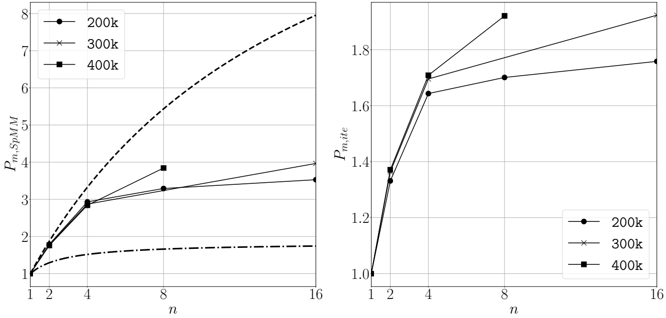
Dependency on the discretization of the Laplacian operators. Concluding remarks

- In general, good behaviour in acceleration of `SpMM`.
- Speed-up translates to the whole iteration.
- Denser `SpMM` provide better results.
- Saturation exceeds capacity of a single node.
  - `200k`: 5889 doubles/cell
  - `300k`: 3965 doubles/cell
  - `400k`: 3019 doubles/cell
- Simulation speed-ups max out for `400k, 27p`; 1.65.
  - Cannot be compared with Krasnopolsky[3] estimates as test cases are different. Smaller speed-up values are expected as heavy `SpMM` are more "diluded".

[3] B.I. Krasnopolsky, "An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles," *Comput. Phys. Commun.* **229**, 2018

Results

Dependency on the load of the node. Speed-up in SpMM and iteration



Conclusions

- This work presents an alternative to long simulations in which extracting statistics of the flow, based on the works of Krasnopolsky[3] and Alsalti-Baldellou et al.[2].
- Strategy to increment the arithmetic intensity of the sparse matrix-vector products (`SpMV`) by converting them to sparse matrix-matrix products (`SpMM`).
- Tested in a differentially heated cavity, $Ra=10^{10}$, $Pr=0.71$, uniform mesh.
- Different loads of the CPU's were tested (`200k,300k,400k`).
- Different discretizations of the Laplacian operator were tested (`7p,13p,27p`).
- The more dense the sparse matrix, the bigger the benefits of the method.
- The more the CPU is loaded, the bigger the benefits of the method.
- Speed-ups of around a maximum value of 1.65 in the whole simulations have been estimated.

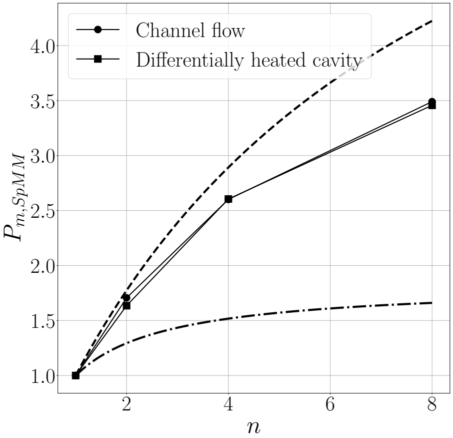
[2] À. Alsalti-Baldellou et al., "Lighter and faster simulations on domains with symmetries," *Comput. Fluids* **275**, 2024.

[3] B.I. Krasnopolsky, "An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles," *Comput. Phys. Commun.* **229**, 2018

Conclusions

On-going and future work

- Testing the method to GPU-accelerated nodes in equivalent loads.
- Apply the method to the same test cases of Krasnopolsky[3].
- SPOILER ALERT!!! Applying the new results from the presentation of X.Álvarez-Farré.

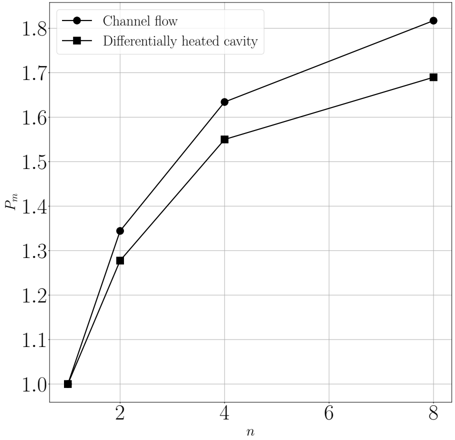


[3] B.I. Krasnopolsky, "An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles," *Comput. Phys. Commun.* **229**, 2018

Conclusions

On-going and future work

- Testing the method to GPU-accelerated nodes in equivalent loads.
- Apply the method to the same test cases of Krasnopolsky[3].
- SPOILER ALERT!!! Applying the new results from the presentation of X.Álvarez-Farré.



[3] B.I. Krasnopolsky, "An approach for accelerating incompressible turbulent flow simulations based on simultaneous modelling of multiple ensembles," *Comput. Phys. Commun.* **229**, 2018