PARALLEL CFD 2011
23rd International Conference on
Parallel Computational Fluid Dynamics 2011

Hybrid MPI+OpenMP parallelization of an FFT-based 3D Poisson solver that can reach 10^5 CPU cores

A. V. Gorobets[1,3], F. X. Trias[1], R. Borrell[2], M. Soria[1] and A. Oliva[1]

1 Heat and Mass Transfer Technological Center
Technical University of Catalonia, Barcelona, Spain

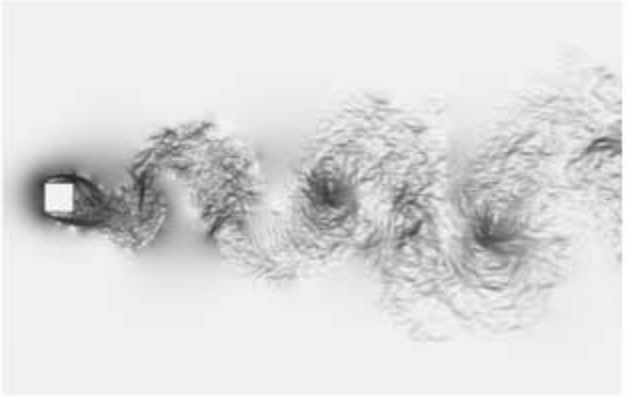
2 Termo Fluids S.L.
Barcelona, Spain

3 Keldysh institute of applied mathematics RAS
Moscow, Russia

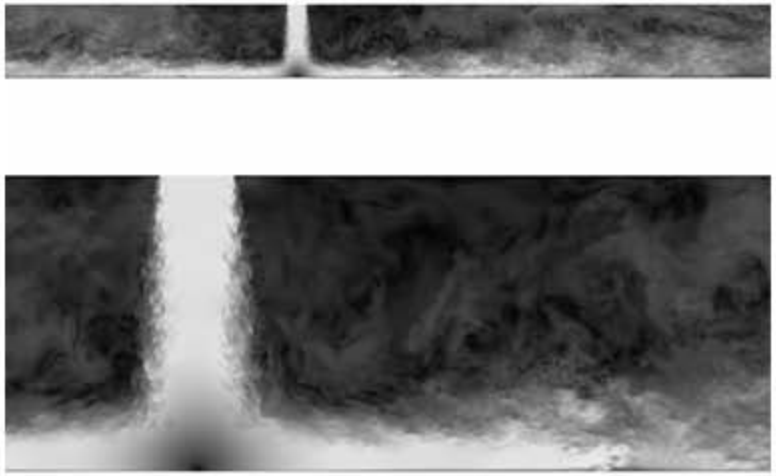
Applications with high computing power demands

Direct numerical simulations

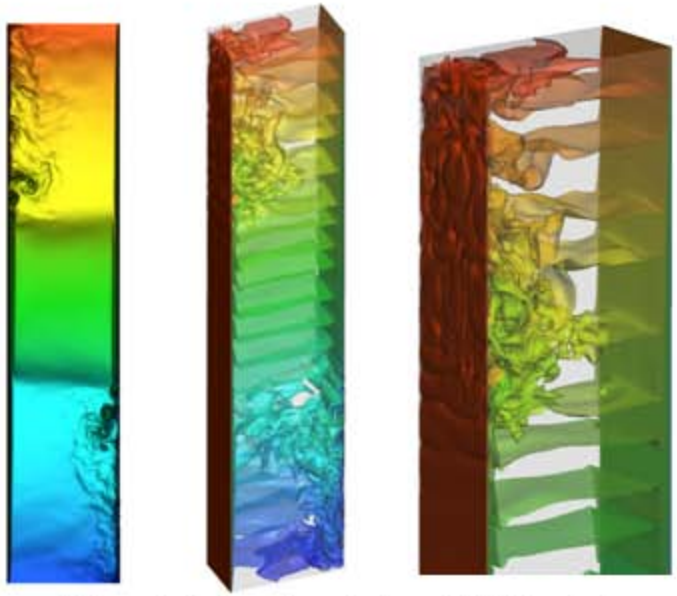
- Incompressible turbulent flows with heat transfer
- High order numerical schemes
- Simplified geometry but complex physical phenomena
- High space and time resolution, long time integration period
- Validation basis for turbulence models



Square cylinder Re=22K (mesh 75M nodes)



Impinging jet Re=20000 (mesh 100M nodes)



Differentially heated cavity (mesh 110M nodes)

A CFD algorithm for incompressible flows

- **Navier-Stokes system to solve:**

$$\nabla \cdot \mathbf{u} = 0,$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = \frac{\mathrm{Pr}}{\sqrt{\mathrm{Ra}}} \nabla^2 \mathbf{u} - \nabla p + \mathbf{f},$$

$$\frac{\partial T}{\partial t} + (\mathbf{u} \cdot \nabla) T = \frac{1}{\sqrt{\mathrm{Ra}}} \nabla^2 T.$$

- **Discrete system for pressure-velocity coupling:**

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = \frac{3}{2}\mathbf{R}^n - \frac{1}{2}\mathbf{R}^{n-1} - Gp^{n+1},$$

$$M\mathbf{u}^{n+1} = 0,$$

where $\mathbf{R}(\mathbf{u}) = -C(\mathbf{u})\mathbf{u} - D\mathbf{u} + f$

- **Fractional step projection method:**

Predictor velocity: $\mathbf{u}^p = \mathbf{u}^n + \Delta t\left(\frac{3}{2}\mathbf{R}^n - \frac{1}{2}\mathbf{R}^{n-1}\right)$

Unknown velocity: $\mathbf{u}^{n+1} = \mathbf{u}^p - G\tilde{p}$, where $\tilde{p} = \Delta t p^{n+1}$

Mass conservation equation: $M\mathbf{u}^{n+1} = M\mathbf{u}^p - GM\tilde{p} = 0$

$$M\mathbf{u}^{n+1} = M\mathbf{u}^p - GM\tilde{p} = -M\Omega M^* \tilde{p} = \boxed{L\tilde{p} = M\mathbf{u}^p}$$

The Poisson equation

The algorithm of the time step

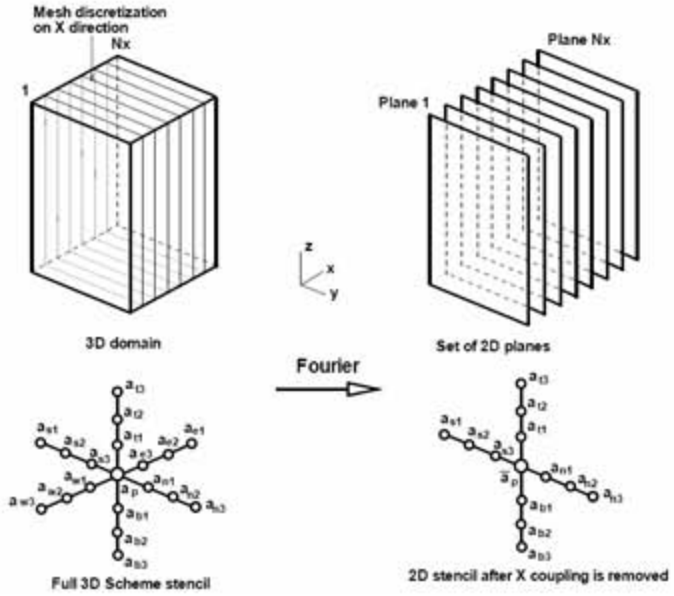
1. Predictor velocity field, $\mathbf{u}^p$, is obtained explicitly
2. Correction, $\tilde{p}$, is obtained from the Poisson equation
3. Resulting velocity field, $\mathbf{u}^{n+1}$, is obtained
4. Energy equation is solved explicitly

The Poisson solver for cases with one periodic direction

- The method is based on a time-consuming preprocessing
- 3D geometry is restricted to extruded 2D shapes with a constant mesh step
- Allows flexible configuration for different parallel systems, CPU groups and problem sizes
- The main scalability limitation comes from the interface system of the Schur complement based method

The algorithm of the solver

1. Fourier diagonalization using FFT uncouples 3D problem into set of independent 2D problems (planes)
2. The Schur complement based direct method is used to solve planes that correspond to lower Fourier frequencies
3. The preconditioned conjugate gradient (PCG) method is used to solve the remaining planes
4. Inverse FFT to restores solution of the 3D problem.

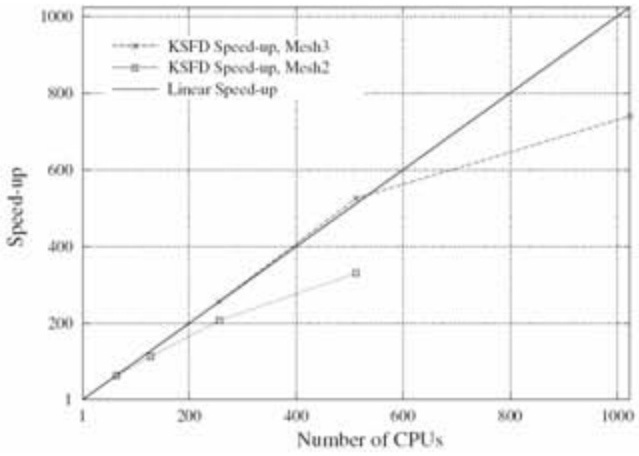


Uncoupling of a 3D domain using FFT

The MPI-only parallelization

Domain decomposition

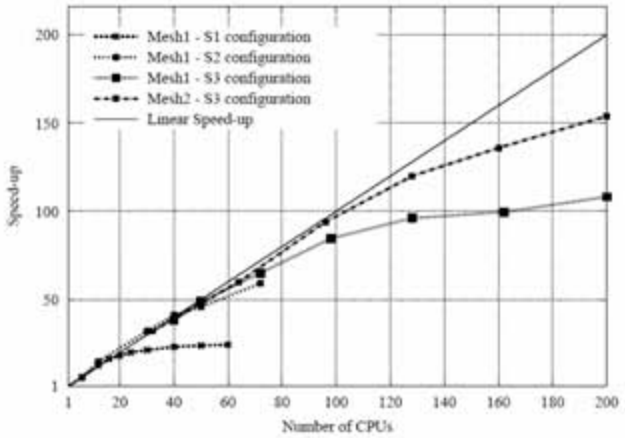
- Domain is decomposed into $P = P_x \times P_{yz}$ parts in all 3 directions
- Periodic direction is decomposed in P_x parts:
  FFT is replicated within 1D groups and each FFT is sequential.
- Each plane is decomposed into P_{yz} parts
- First D planes are solved with the direct method, the remaining planes with the iterative method



Speedup in the periodic direction

Speeup with MPI on MareNostrum

- A DNS case of a differentially heated cavity
  $Ra=10^{11}$, $Pr = 0.71$, aspect ratio 4, **4-th order scheme**
- Meshes:
  Mesh1 is 1.7M nodes, Mesh2 – 11M nodes, Mesh3 – 111M nodes
- Solver configurations
  S1: Direct solver, S2: D = 5, S3: D = 1



Speedup in non-periodic directions

Gorobets, F. X. Trias, M. Soria and A. Oliva, "A scalable parallel Poisson solver for three-dimensional problems with one periodic direction", Computers & Fluids journal, 39 (2010) 525-538, Elsevier

The MPI-only parallelization

Limitations in non-periodic directions

- **Decomposition in non-periodic directions is limited by the Schur method**
  It can be say around **200-300** subdomains for 4-th order
  and around **500-800** for 2-nd order

- **PCG method suffers from too big CPU groups mainly due to reduction communications (cost O(log(P_{yz})) messages) for scalar products**

Limitations in the periodic direction

- **Decomposition in periodic direction is limited by FFT**
  that is replicated within 1D groups requiring an expensive group communication of O(P_x) messages
  It's works well till P_x around **8**

So it is rather difficult to go beyond around ~2000 CPUs for 4-th and ~6000 for 2-nd

Gorobets, F. X. Trias, M. Soria and A. Oliva, "A scalable parallel Poisson solver for three-dimensional problems with one periodic direction", Computers & Fluids journal, 39 (2010) 525-538, Elsevier

Typical supercomputers

Lomonosov, MSU

Network	Infiniband
CPUs	Intel EM64T Xeon 55xx 2930 MHz
Number of cores	**35360**
Nodes	**8 cores** (2 x 4-core CPUs), 12Gb of RAM
Rmax, Tflops	350
Location	Moscow, Russia

MVS-100000, JSC of RAS

Network	Infiniband
CPUs	Intel EM64T Xeon 53xx 3000 MHz
Number of cores	**11680**
Nodes	**8 cores** (2 x 4-core CPUs), 8Gb of RAM
Rmax, Tflops	107
Location	Moscow, Russia

MareNostrum, BSC

Network	Myrinet
CPUs	IBM Power PC 970MP processors at 2.3 GHz
Number of cores	**10240**
Nodes	**4 cores** (2 x 2-core CPUs), 8Gb of RAM
Rmax, Tflops	64
Location	Barcelona, Spain

Additional parallelization with OpenMP

The second level of parallelism

- OpenMP provides parallelization within shared memory model inside of multi-core nodes
- The total number of CPU cores engaded is now $P = P_t \times P_x \times P_{yz}$, where P_t is the number of threads
- In doing so P_x or *I* and P_{yz} can be reduced for the same P healing above mentioned limitations

Advantages

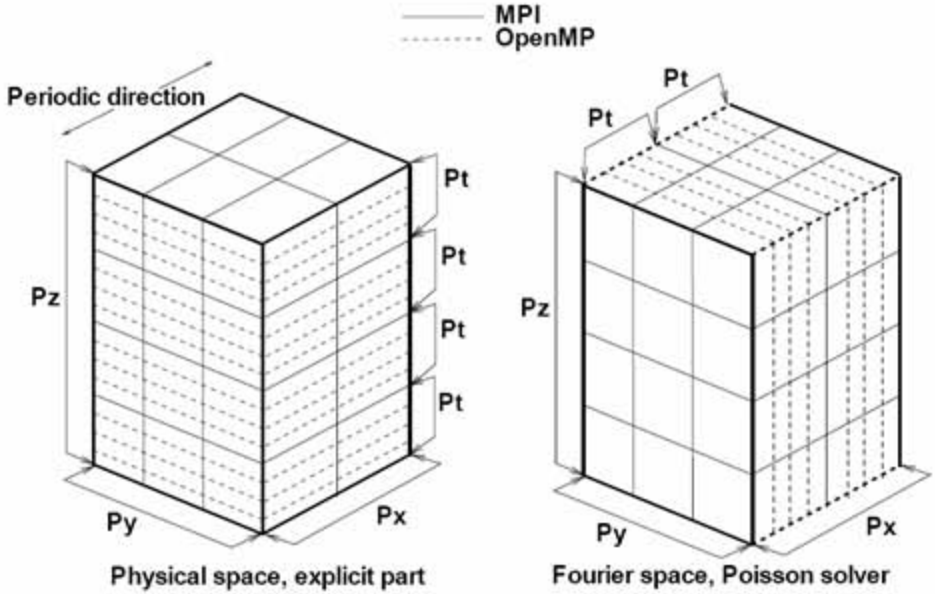
- Schur limitations: he size of interface in Schur solver decreases with P_{yz}
  hence decrease in memory requirements, preprocessing stage cost, communications, etc
- FFT limitations: similarly parallel efficiency grows when P_x reduces
- PCG and explicit parts: the size of halos and the number of communicating processes decreases
- Communications go faster since no multiple processes sharing network hardware of nodes
- More RAM memory available for MPI processes allowing to solve bigger problems
- Easy to change P varying P_t – no need for redecomposition of restarts neither redoing preprocessing stage

Finally efficient range of the number of CPUs can be extended ~P_t times

Additional parallelization with OpenMP

Implementation details

- Explicit part is parallelized by decomposing loops over subdomain nodes
- FFT stages (1,4) are parallelized in the same way – set of subvectors is divided between threads
  FFT itself stays sequential!
- The set of independent planes is divided between threads on stages 2, 3
- I/O and MPI communications are performed by the master thread only

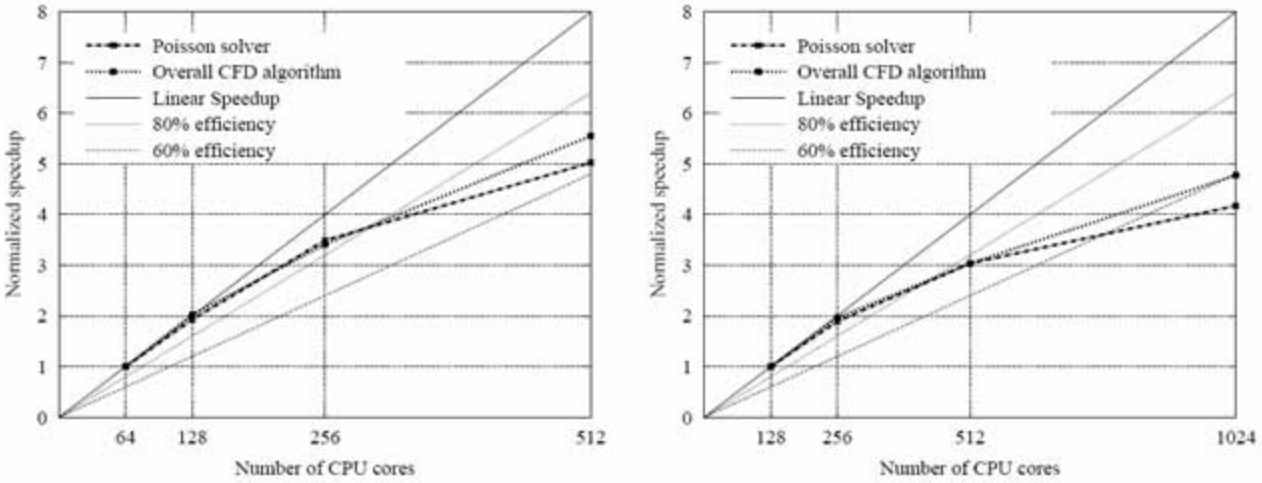


Speedups with OpenMP

Tests on MVS-100000 supercomputer

Meshes used for DHC test case

	N_x	N_y	N_z		N	Ra	Pr	Order
Mesh1	128	192	462	$\approx$	11.4×10^6	10^{11}	0.71	4^{th}
Mesh2	256	800	1600	$\approx$	327.7×10^6	10^{11}	0.71	2^{nd}
Mesh3	256	1400	2800	$\approx$	1003.5×10^6	10^{11}	0.71	2^{nd}

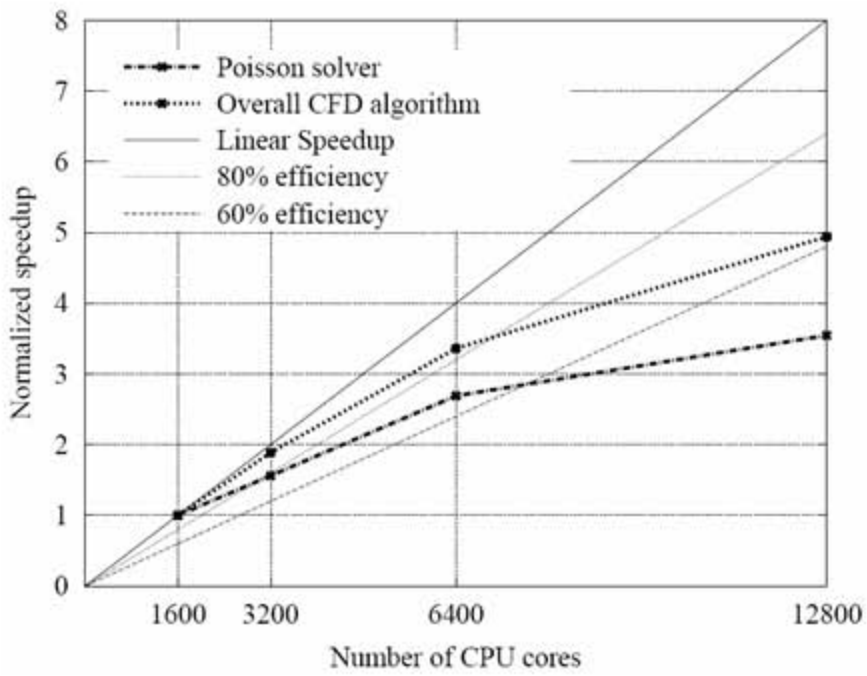


Speedup with OpenMP varying P_t for MPI groups of 64 and 128 processes, Mesh1

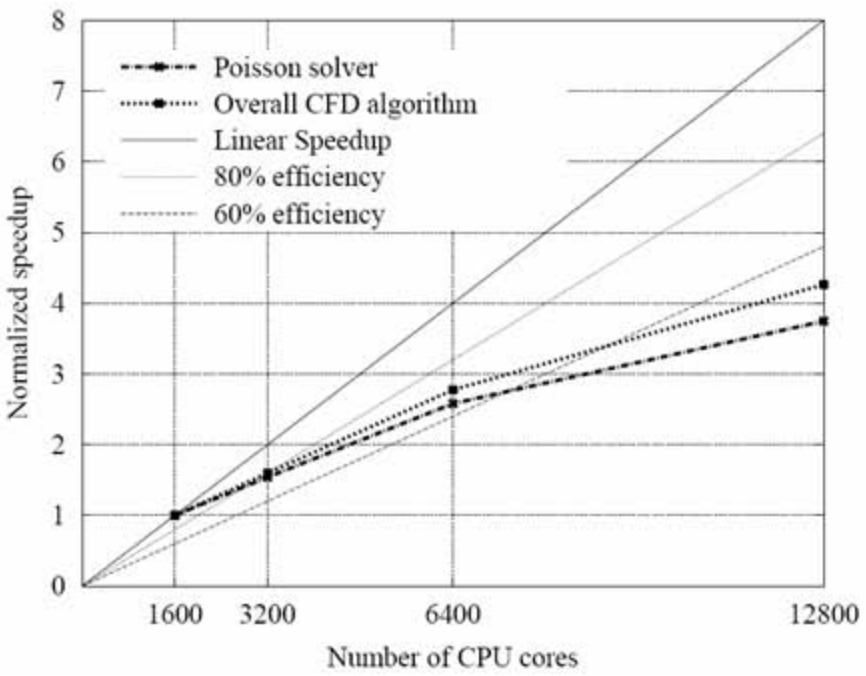
A. Gorobets, F. X. Trias, R. Borrell, O. Lehmkuhl, A. Oliva, "Hybrid MPI+OpenMP parallelization of an FFT-based 3D Poisson solver with one periodic direction", Elsevier, Computers & Fluids (2011)

Speedups with MPI (P_x)

Bigger tests on Lomonosov supercomputer



Mesh 330 millions of nodes



Mesh 1 billion of nodes

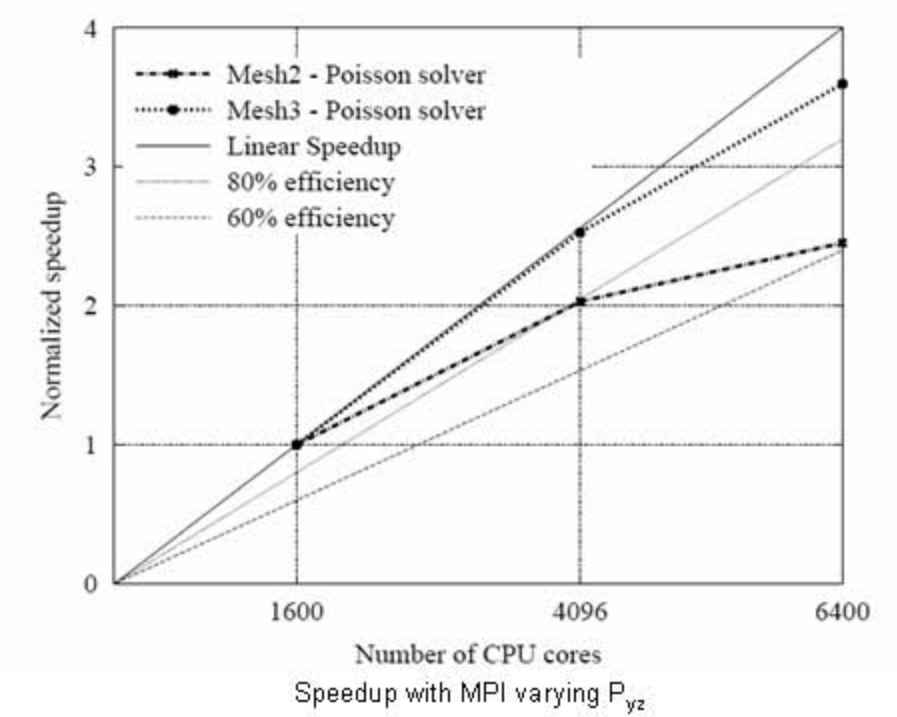
Speedups with MPI varying P_x

A. Gorobets, F. X. Trias, R. Borrell, O. Lehmkuhl, A. Oliva, "Hybrid MPI+OpenMP parallelization of an FFT-based 3D Poisson solver with one periodic direction", Elsevier, Computers & Fluids (2011)

Speedups with MPI (P_{yz}) and estimations of new limitations

Tests with bigger meshes on Lomonosov supercomputer

- Of course with P_x and P_t solver reaches its limits and parallelism is exhausted in the tests
- P_{yz} = 200 is still far from its limits for the number of CPU cores 12800 that was available for tests



Speedup with MPI varying P_{yz}

- Till P_{yz} = 800 it seems to work well, hence we can estimate limits as at least 800x8x8 = **51200 CPU cores**

Symmetric extension

Symmetric extension for cases with mesh symmetries

- The original system to solve: $\mathbf{Ax}=\mathbf{b}, \quad \mathbf{A}=\frac{1}{2}\begin{pmatrix}\mathbf{A}_P & \mathbf{A}_N \\ \mathbf{A}_N & \mathbf{A}_P\end{pmatrix}$

- Changing basis in the following way:

$$\mathbf{x}^{\pm}=\mathbf{S}\mathbf{x}, \quad \mathbf{S}=\frac{1}{2}\begin{pmatrix}\mathbf{I}_{N/2} & \mathbf{I}_{N/2} \\ \mathbf{I}_{N/2} & -\mathbf{I}_{N/2}\end{pmatrix}, \quad \mathbf{x}=(\mathbf{x}_0,\mathbf{x}_1)^T, \quad \mathbf{x}^{\pm}=(\mathbf{x}^+,\mathbf{x}^-)^T$$

The vector is decomposed into symmetric and skew-symmetric parts: $\mathbf{x}^+=\frac{1}{2}(\mathbf{x}_0+\mathbf{x}_1), \quad \mathbf{x}^-=\frac{1}{2}(\mathbf{x}_0-\mathbf{x}_1)$

- Applying change of basis to the linear system:

$$\mathbf{A}^{\pm}\mathbf{x}^{\pm}=\mathbf{b}^{\pm}, \quad \mathbf{A}^{\pm}=\mathbf{S}\mathbf{A}\mathbf{S}^{-1}=\frac{1}{2}\begin{pmatrix}\mathbf{A}^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{A}^-\end{pmatrix}, \quad \mathbf{A}^+=\mathbf{A}^P+\mathbf{A}^N, \quad \mathbf{A}^-=\mathbf{A}^P-\mathbf{A}^N$$

The algorithm is following:

1. Transform the right-hand-side sub-vector, $\mathbf{b}^{\pm}=\mathbf{S}\mathbf{b}$
2. Solve the two decoupled systems, $\mathbf{A}^+\mathbf{x}^+=\mathbf{b}^+, \quad \mathbf{A}^-\mathbf{x}^-=\mathbf{b}^-$
3. Reconstruct solution, $\mathbf{x}=\mathbf{S}^{-1}\mathbf{x}^{\pm 1}$

- Stages 1 and 3 require a point-to-point communication

Symmetric extension

This way the CPU number can be multiplied

- Since we can solve 2 independent twice smaller systems
  we can use 2 MPI groups engaging **twice more CPUs**
- The procedure can be applied recursively and if there are 2 symmetric directions
  we can use **4 times more CPUs**

This leads to the new estimation:

More than **100000 CPU** cores can be used in case of 1 symmetry
and more than **200000** in case of 2 symmetries

Applications: DNS of incompressible flows

DNS of an impinging jet

- Re = 20000
- Mesh size 102M
- 4-th order scheme
- Running on MareNostrum supercomputer on 320 CPU cores

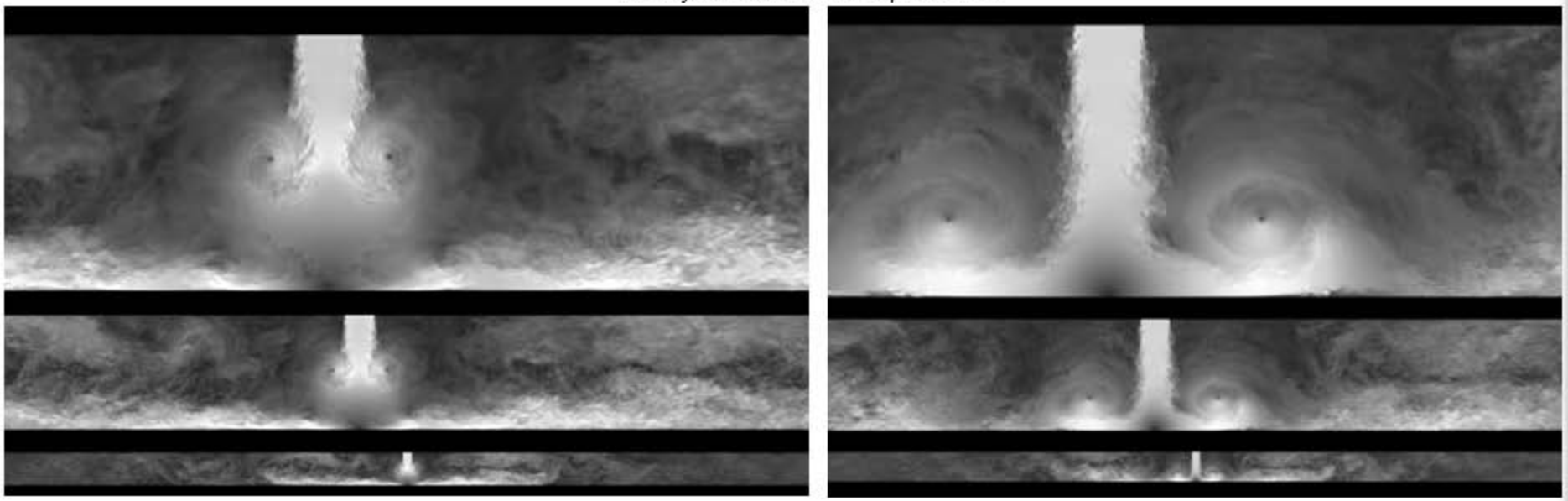
Boundaries (velocity):

1. Periodic in span-wise
2. No-slip walls from top and bottom
3. Given stationary profile on inlet
4. Convective BC on outlet

Boundaries (temperature):

1. Adiabatic top
2. Given heat flux bottom / Isotermal
3. Fixed temperature on inlet

Velocity module in the mid-span section



J.E. Jaramillo, F.X. Trias, A. Gorobets, C.D. Perez-Segarra, and A. Oliva. DNS and RANS modelling of a Turbulent Plane Impinging Jet. International Journal of Heat and MassTransfer, (submitted).

Applications: DNS of incompressible flows

DNS of an impinging jet

J.E. Jaramillo, F.X. Trias, A. Gorobets, C.D. Perez-Segarra, and A. Oliva. DNS and RANS modelling of a Turbulent Plane Impinging Jet. International Journal of Heat and MassTransfer, (submitted).

Applications: DNS of incompressible flows

DNS of differentially heated cavities at aspect ratio 4, 5, Ra = 10^{10} ~10^{12}, meshes up to 110M nodes

F.X. Trias, R.W.C.P. Verstappen, A. Gorobets, M. Soria, A. Oliva, "Parameter-free symmetry-preserving regularization modelling of a turbulent differentially heated cavity", Elsevier, Computers & Fluids (2010), doi:10.1016/j.compfluid.2010.06.016

Conclusions and future actions

- The use of the two-level MPI+OpenMP parallelization extends the limits of the number of CPUs far beyond the resources available at the moment and in the near future
- The Poisson solver and the whole CFD code for incompressible flows with one periodic direction is ready to run on more than 100000 CPU cores with meshes > 10^9 nodes

- The next step is to adapt the solver for heterogeneous GPU-based systems
- The three-level MPI+OpenMP+OpenCL parallel model that combines MIMD and SIMD parallelism on each heterogeneous node is being implemented
- The new code will work well on both AMD (ATI) and NVidia hardware