DE LA RECHERCHE À L'INDUSTRIE

High performance calculation enhancement using massive parallelization with TRUST and TrioCFD

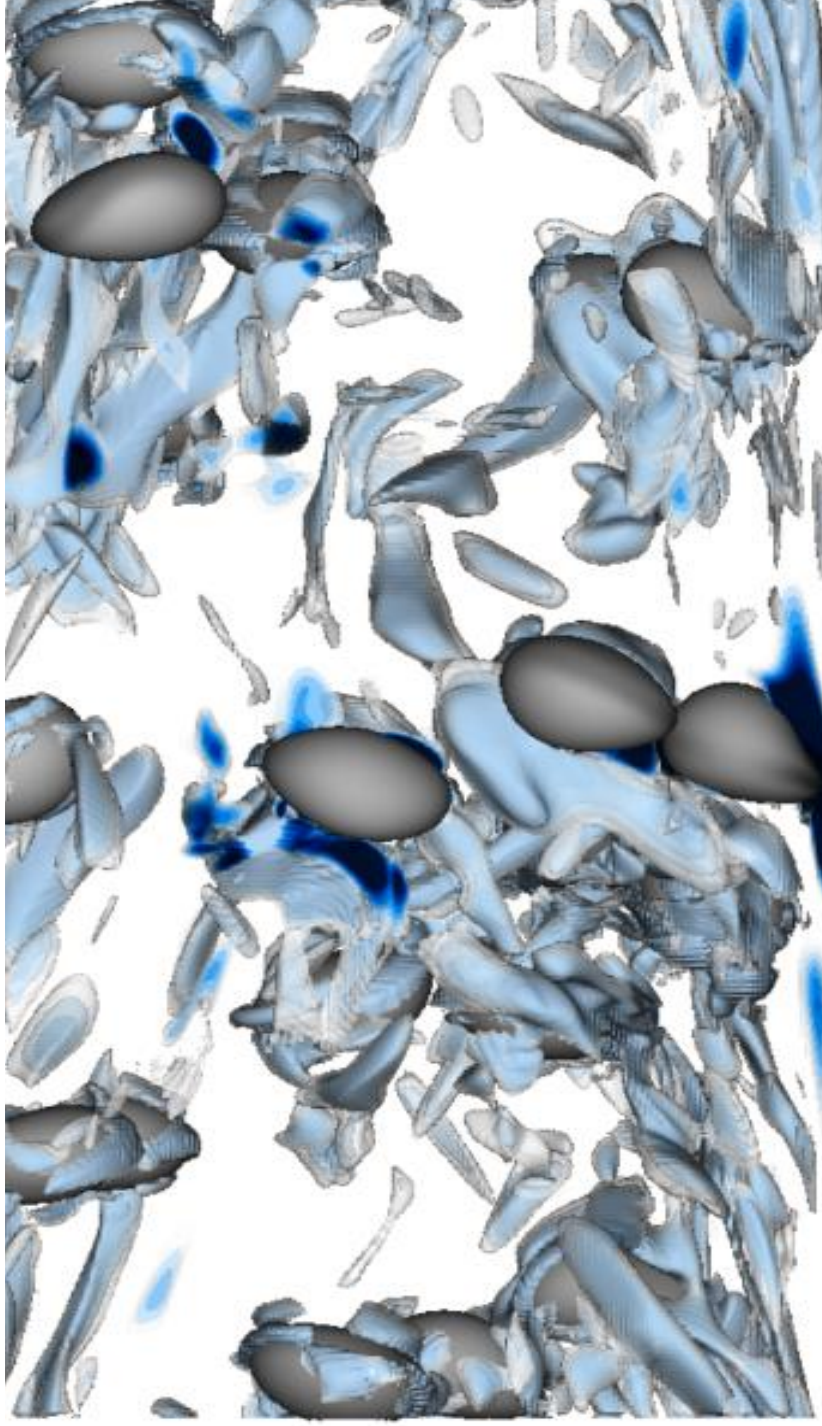
R. Nop, A. Bruneton, J. Darona, E. Saikali, U. Bieder, P. Ledac, A. Khizar, N. Dorville, E. Adam

Université Paris-Saclay, CEA, Service de Thermo-hydraulique et de Mécanique des Fluides, 91191, Gif-sur-Yvette, France.

Outline

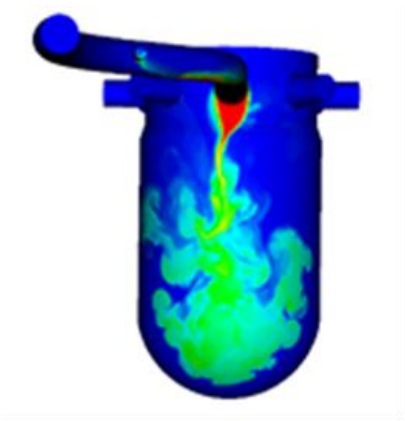
1. TRUST and TrioCFD
2. Parallel computing: Overview, performance and simplicity
3. Examples of massive parallel computations
4. Towards GPU-accelerated simulations

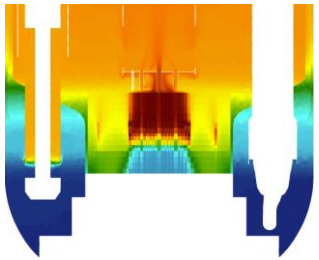
1. TRUST and TrioCFD

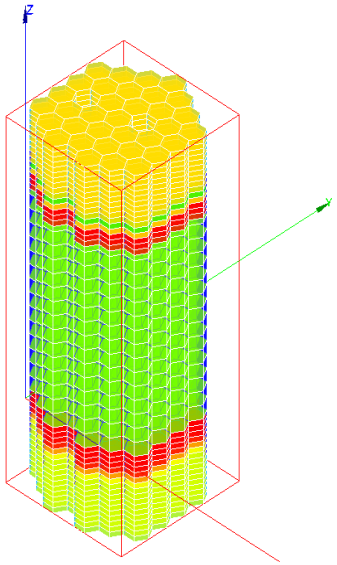


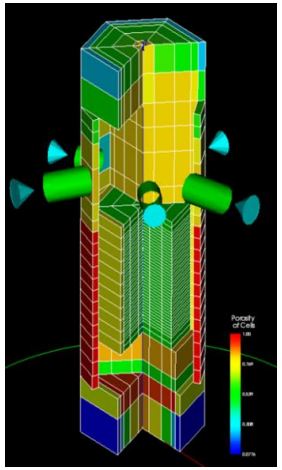
Development strategy for the thermal-hydraulics codes

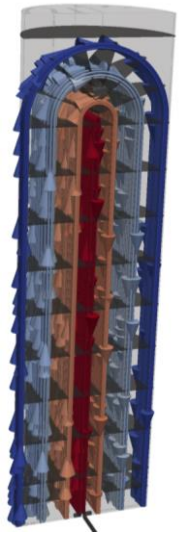
CFD	Component Sodium reactors	Component Core	Component	Component Steam generators	Reactive Transport

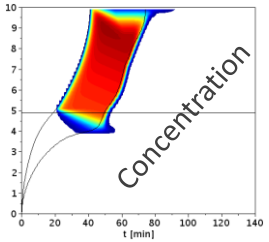












TrioCFD	TrioMC	Flica 5	Cathare3D	Genepi+	Scorpio

TRUST

Software Engineering (C++, Git, CMake, Tuleap)

Tools available in the TRUST platform

TrioCFD is based on the TRUST platform of CEA :

Discretization

Finite Volume Difference **VDF**
Finite Volume Element **VEF**
Polyhedral volumes **PolyMac**

Boundary conditions

Wall, inlet, outlet, symmetry, periodic …

Numerical schemes

- Time integration:
  - ↪ Explicit : Euler, Runge-Kutta, Adams-Bashforth
  - ↪ Semi-implicit: diffusion terme is taken implicitly
  - ↪ Implicit : Euler backward, Runge-Kutta, Crank-Nicholson
- Space discretization of the convection term:
  - ↪ Up to 4th order
  - ↪ upwind, centered, MUSCL, QUICK, ...

Meshing

↪ TRUST internal tools (simple cases)
↪ SALOME, ANSYS-IcemCFD
↪ Gmesh

Post-processing

↪ On the important variables
↪ On the physical properties
↪ On fields and local values
↪ Visualization : SALOME, VisIT, GnuPlot

Evaluate the impact of a TrioCFD development on TRUST platform and other TRUST apps

Automatic V&V

Validation datasets (.prm)
Validation report in pdf
Automatic pdf comparison + expert judgment

High Performance Computing

Massively parallel (MPI), Excellent CPU and GPU performance
Tested on many French and European HPC platforms

TrioCFD at a glance

1995 Trio_U → 2015 TrioCFD

https://github.com/cea-trust-platform/TrioCFD-code

TRUST@cea.fr or TRIOU@cea.fr

https://triocfd.cea.fr/Pages/Presentation/TrioCFD_code.aspx

All LINUX 32/64 bits distributions

CODE DESIGN

- Base on TRUST
- Programmation C++
- ~ 1500 classes
- Computations : datasets
- Command lines

FIELDS OF INTEREST

- Hydraulics
- Thermal-Hydraulics
- Incomp. and Dilatable fluids
- Two phase flows
  - ↪ Front-Tracking
  - ↪ incomp. diffuse interface
  - ↪ Homogeneous Eq Model (IN PROGRESS)
- Fluid-structure interaction
- Chemistry

MANAGEMENT

- Bug Tracker tuleap
- Code management git
- Verification
- Validation Jenkins
- Documentation doxygen & LaTeX

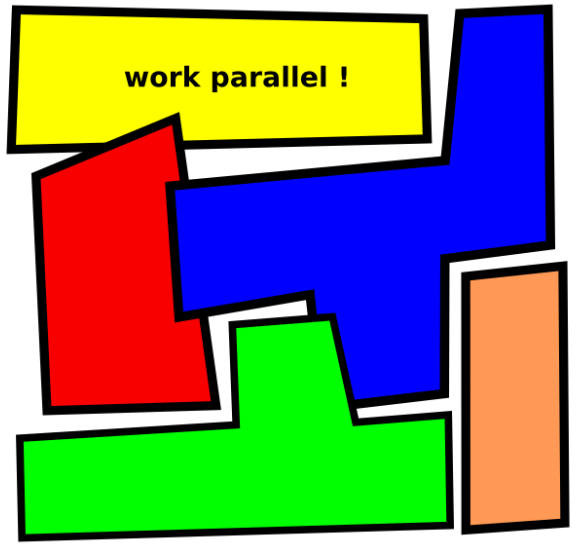
2. Parallel computing
Overview, performance and simplicity

Parallel strategy

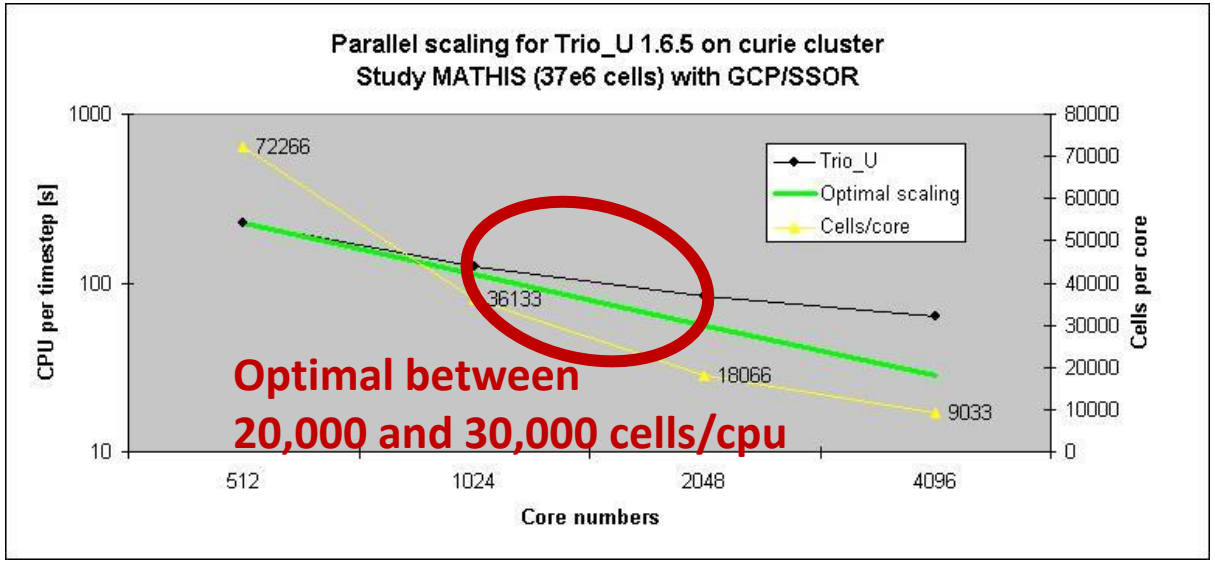
- **Domain splitting** (***not Domain Decomposition***)
  - Partitioning of the mesh (SCOTCH, METIS)
  - Minimizing the interface between sub-domains
  - Ghost cell management

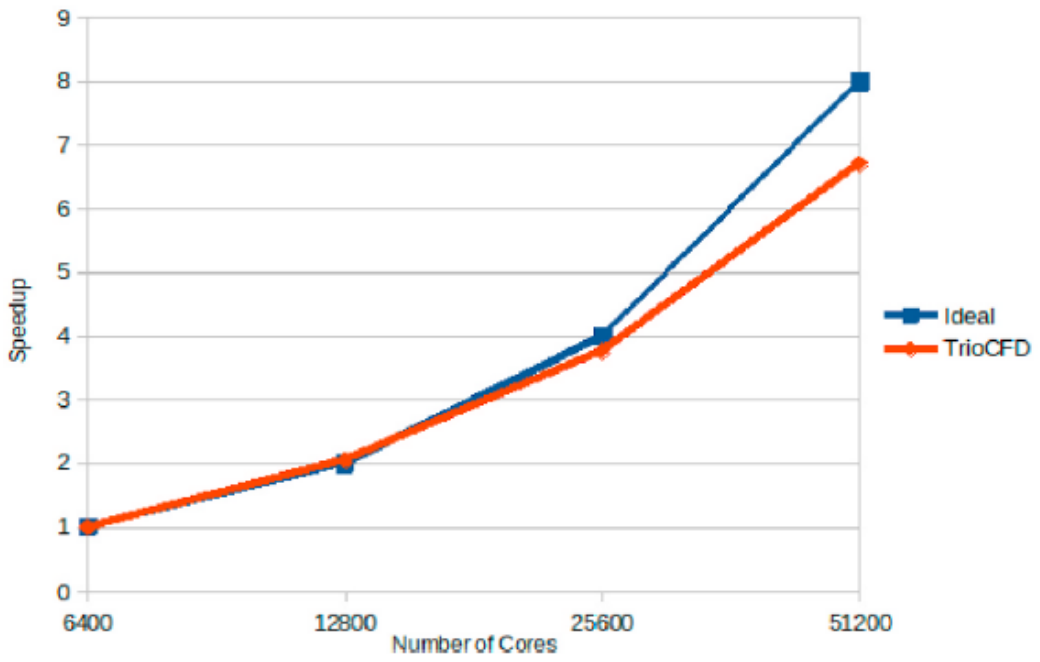
- **Message passing for the stencil computations**
  - MPI (OpenMPI or MPICH)
  - Asynchronous calls (MPI_Isend, MPI_Irecv, …)

- **No hybrid programing**
  - No multi-threading or OpenMP-like strategy
  - Initial philosophy: MPI can do it all … To be tried again, surely!



Performance

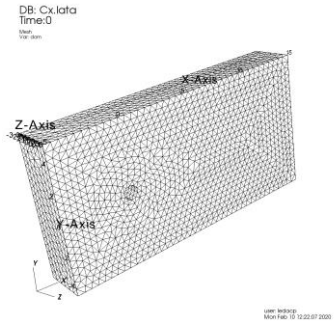


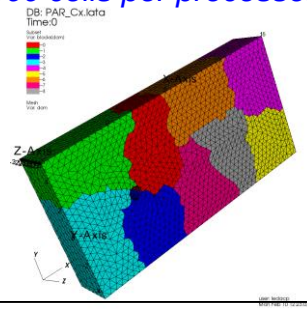


Sequential to parallel: Simplicity of use

Partitioner

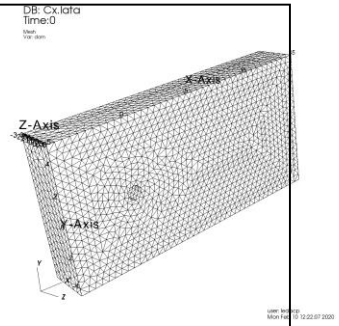
```
dimension 3

# Domain definition #
Domaine dom

# BEGIN MESH #
Read_file dom Obstacle.geom
# END MESH #


# BEGIN PARTITION #
Partition dom
{
        /* Choose Nb_parts so to have ~ 25000 cells per processor */
        Partition_tool metis { nb_parts 9 }
        Larg_joint 2
        zones_name DOM
}
End
# END PARTITION #
```





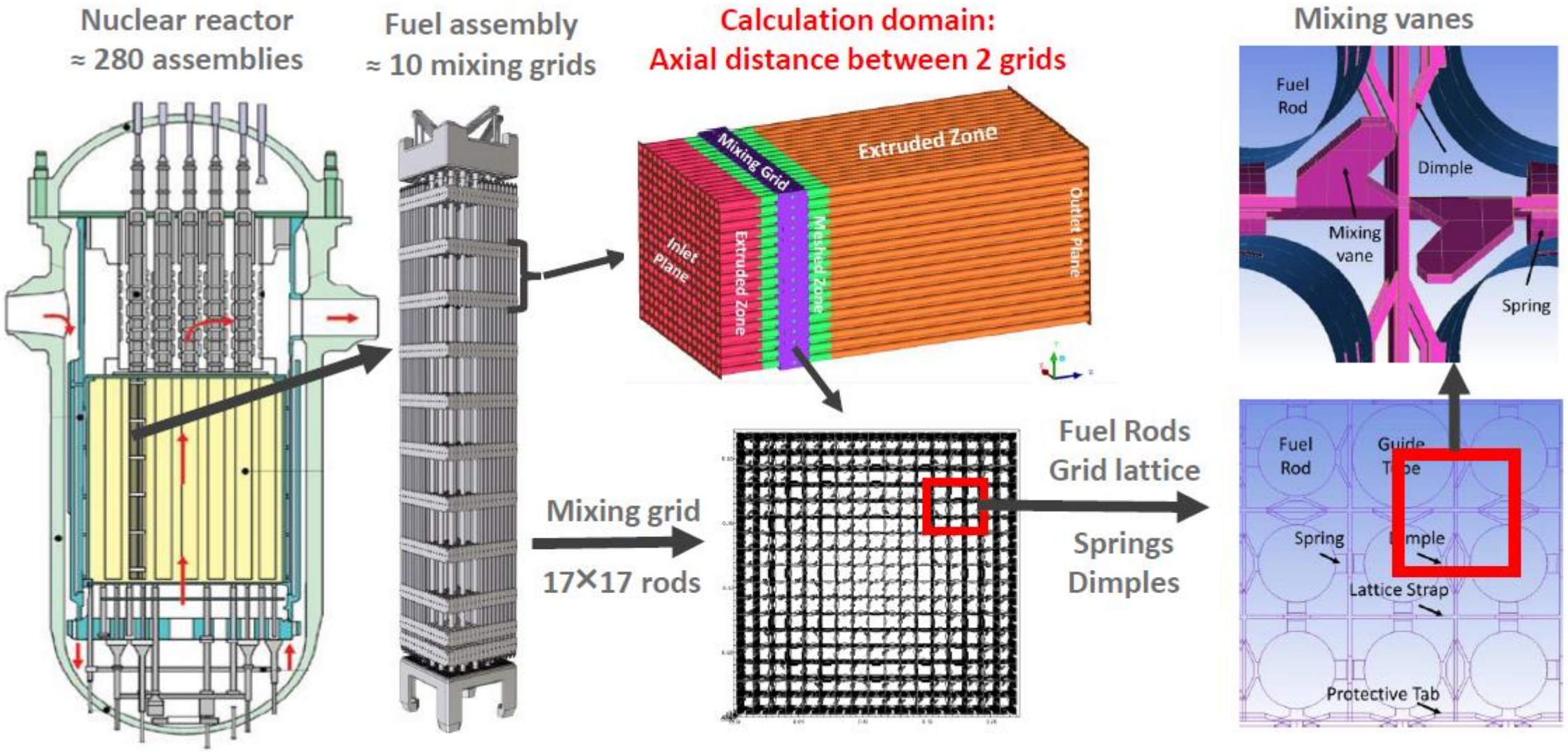
Dataset

```
dimension 3

Domaine dom


VDF ma_discretisation

Scheme_euler_explicit mon_schema
Read mon_schema { ... }

Pb_hydraulique pb

Fluide_Incompressible milieu
Read milieu { ... }

Associate pb dom
Associate pb mon_schema
Associate pb milieu
Discretize pb ma_discretisation

Read pb
{
        Navier_Stokes_standard { ... }
        Post_processing { ... }
}
Solve pb
End
```



3. Examples of parallel computations

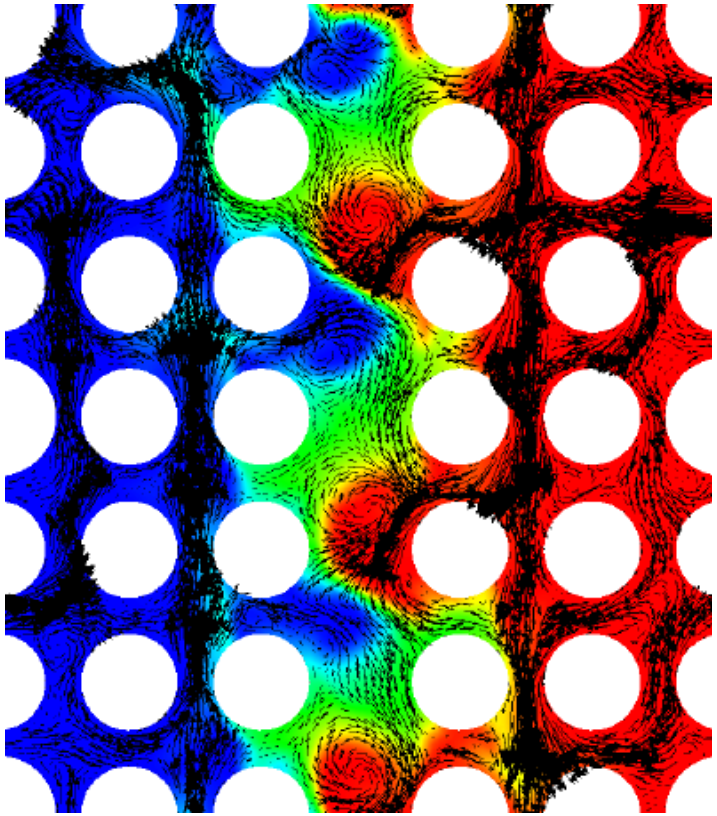
1 billion – cell calculation : RANS simulation with TrioCFD

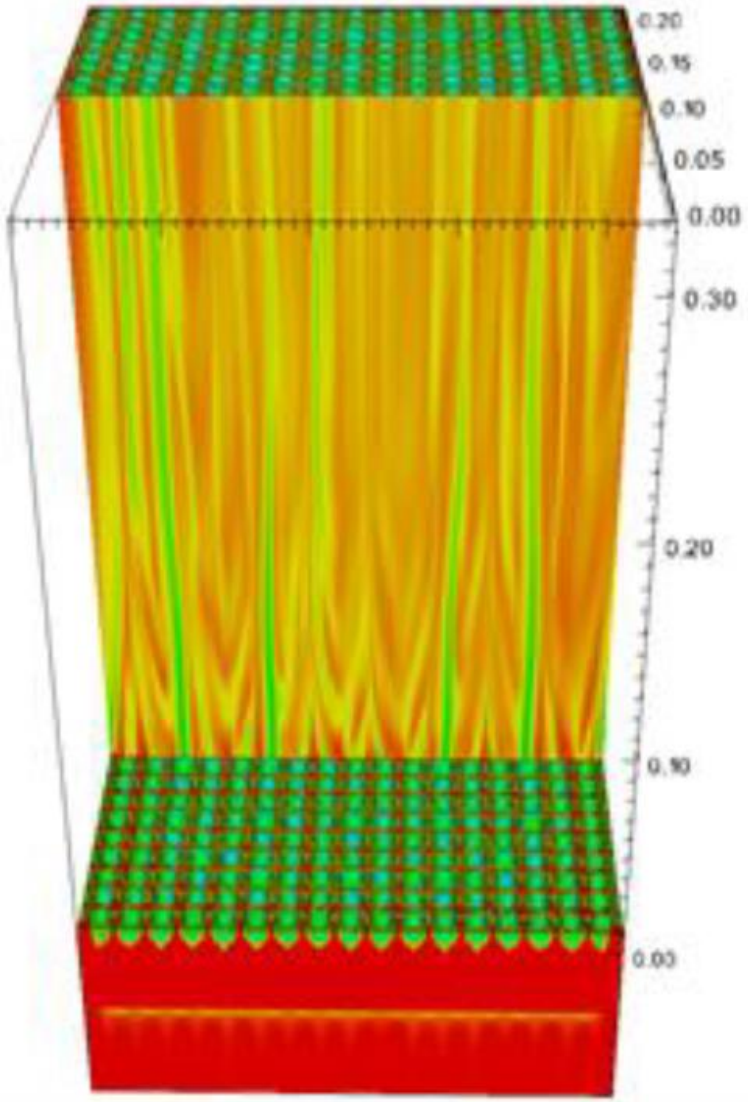


1 billion – cell calculation : RANS simulation with TrioCFD

Thermohydraulic details

- Mean velocity : 5.35 m/s
- Sub-channel Re: ~ 600,000
- Turbulence modeling k-ε



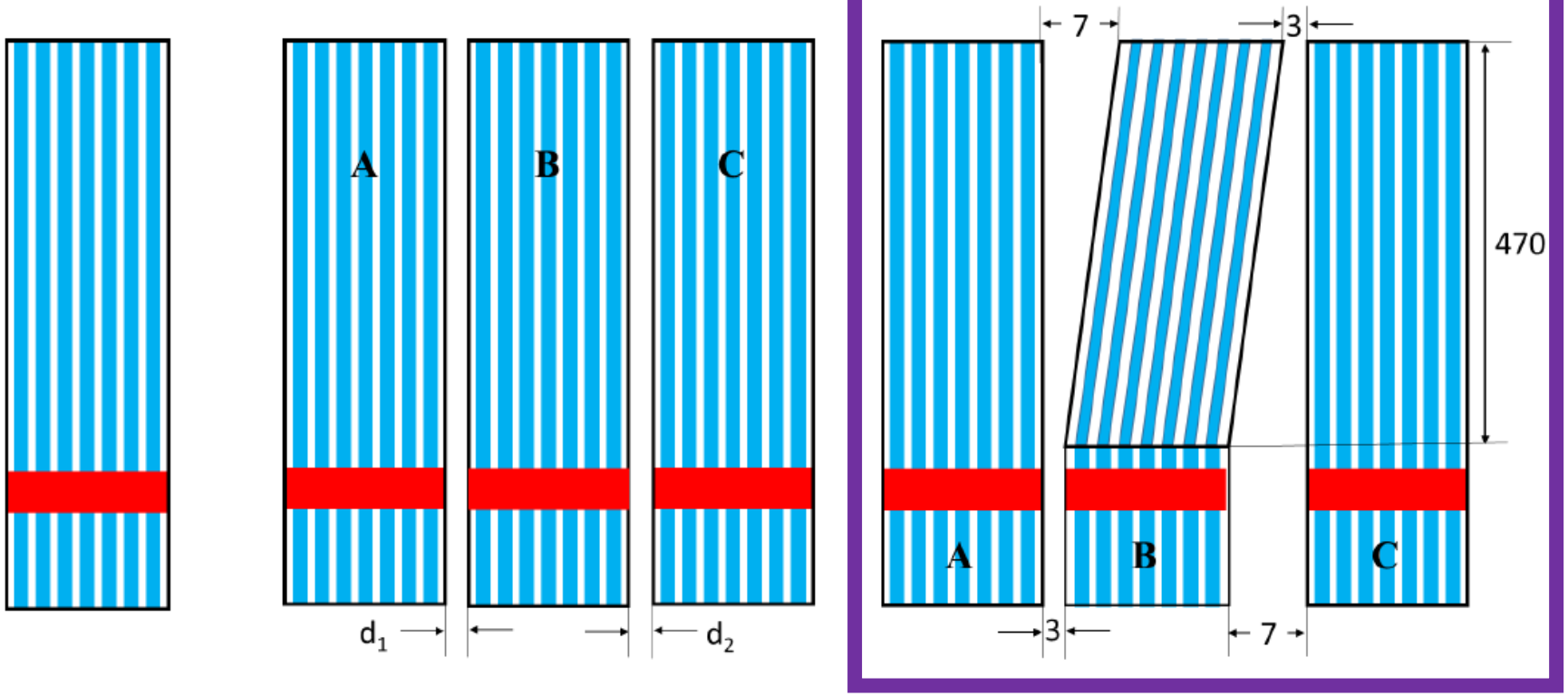


Configurations of analyzed assemblies

one assembly upright rods

three assemblies in line upright rods

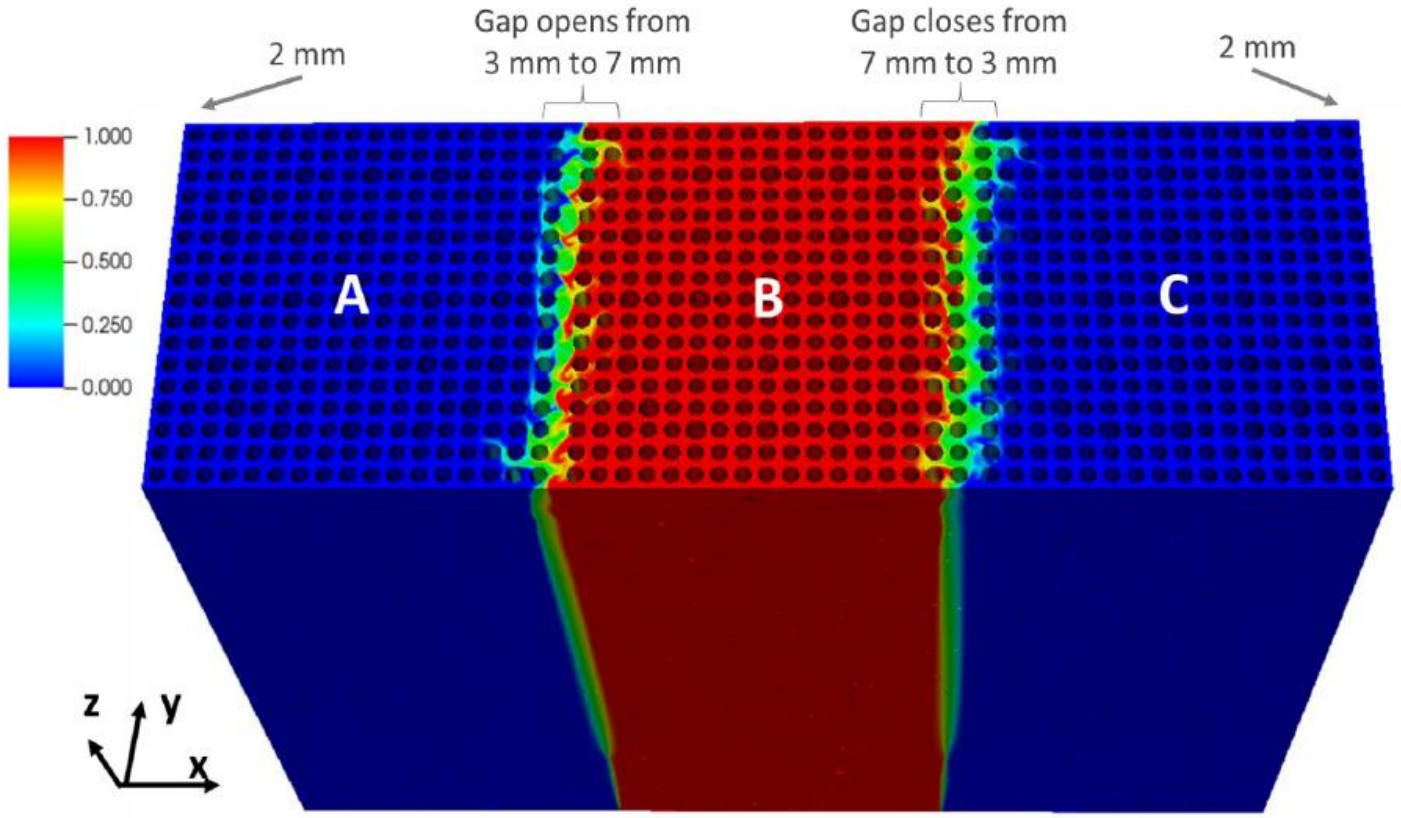
three assemblies in line central assembly with inclined rods



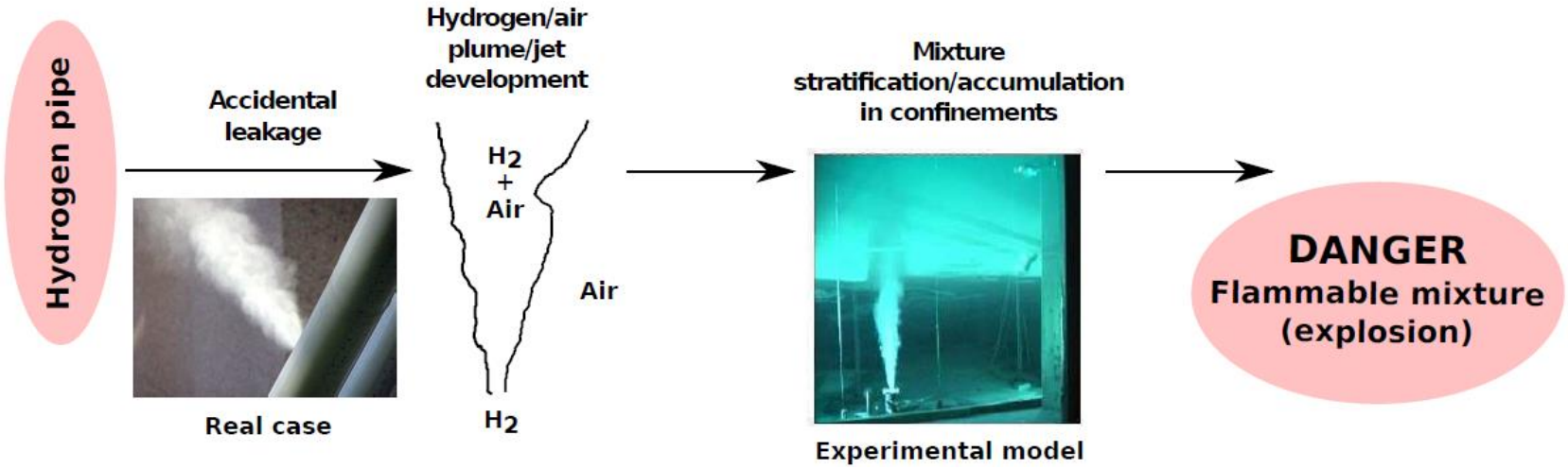
1 billion cells calculation

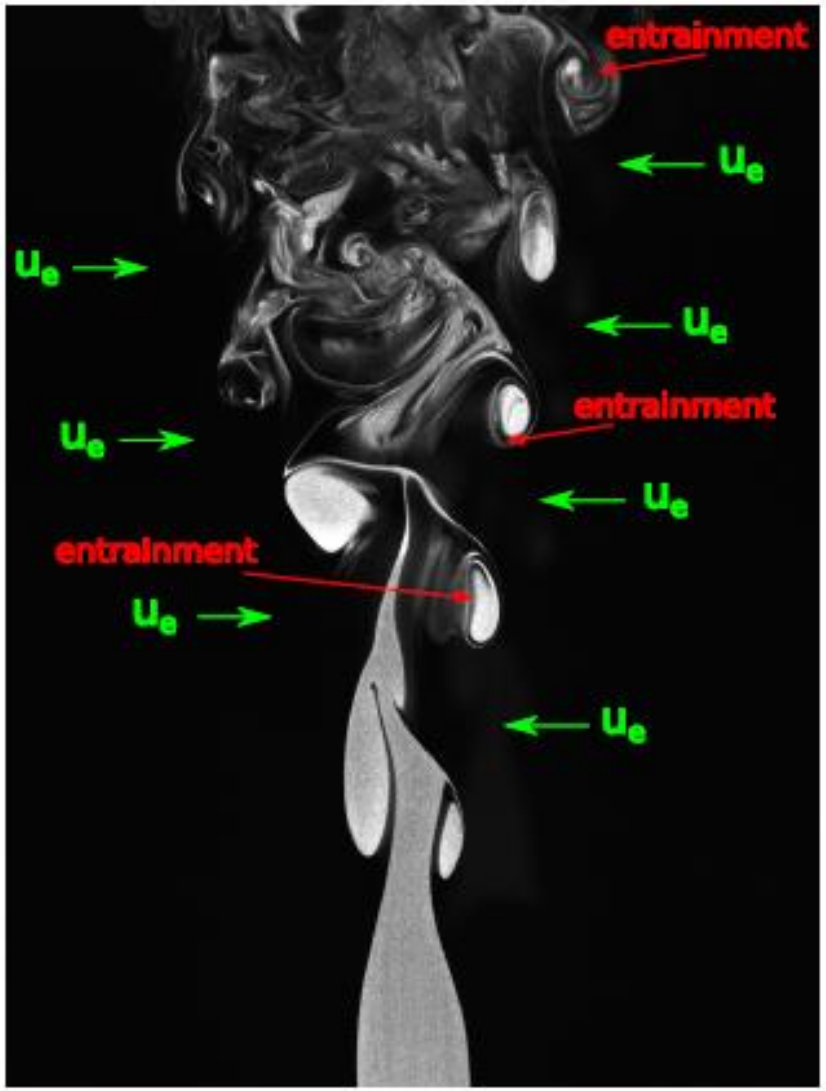
1 billion – cell calculation : RANS simulation with TrioCFD

- 1.032 Billion cells
- 16 680 CPUs
- Converged in 50h → 840 000 CPU hours

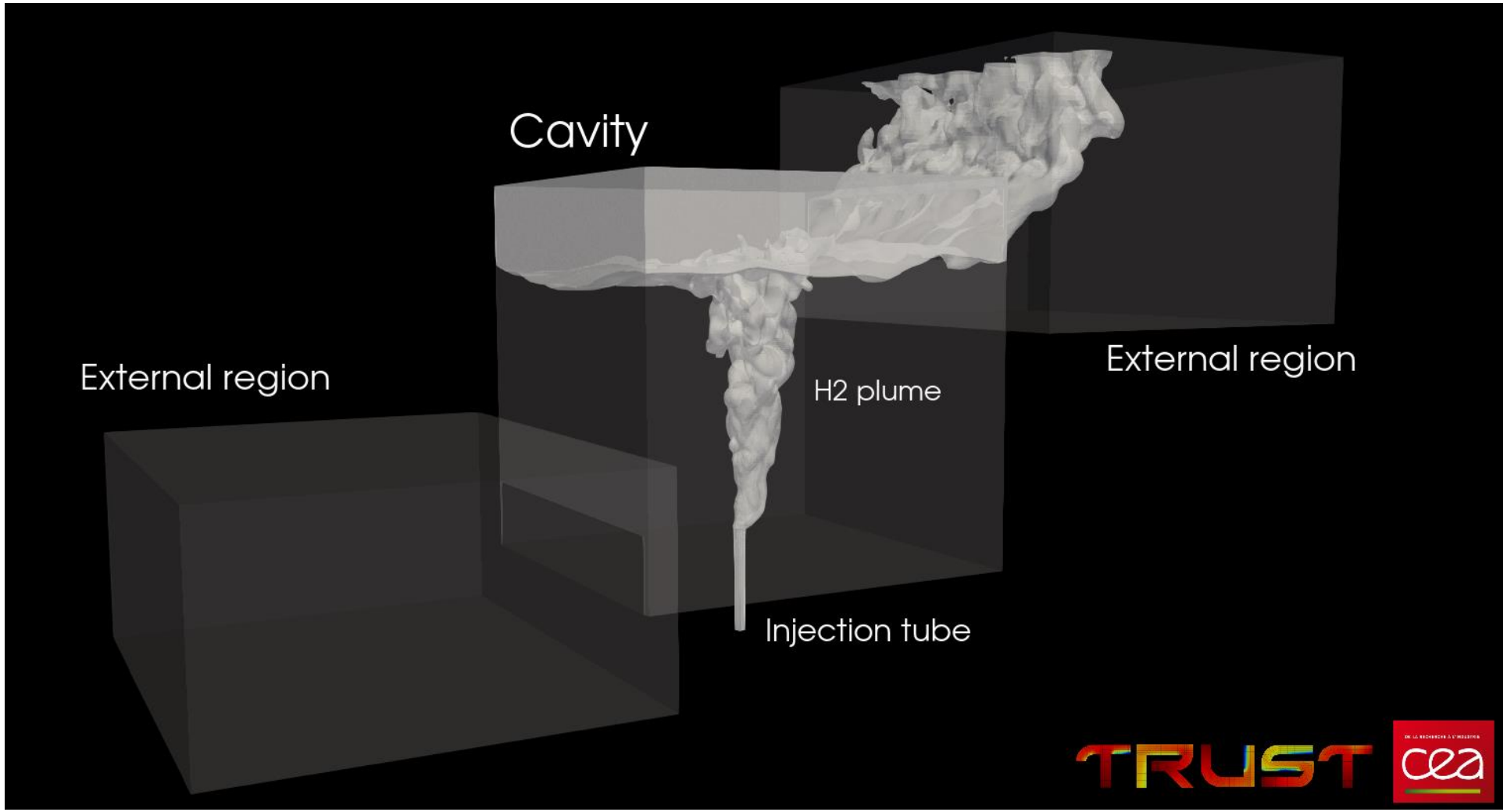


2 billion – cell calculation: DNS simulation with TRUST

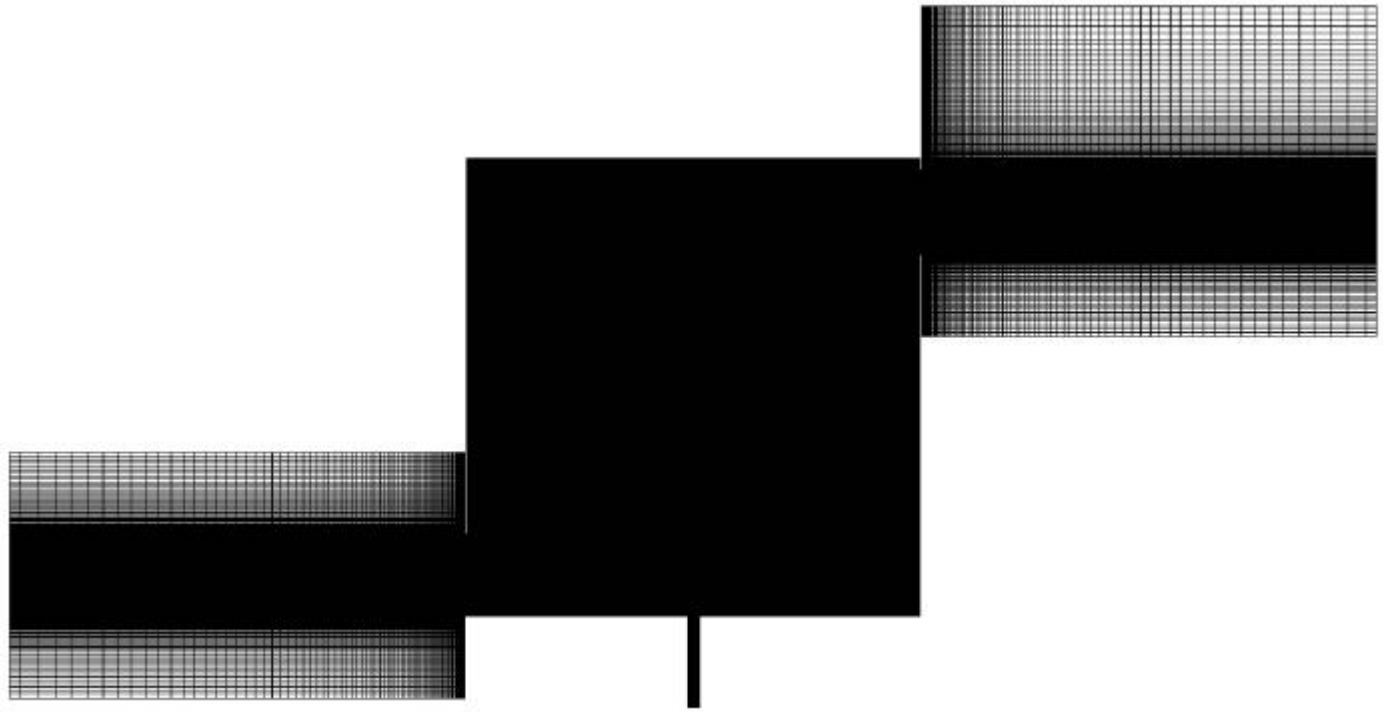




2 billion – cell calculation: DNS simulation with TRUST

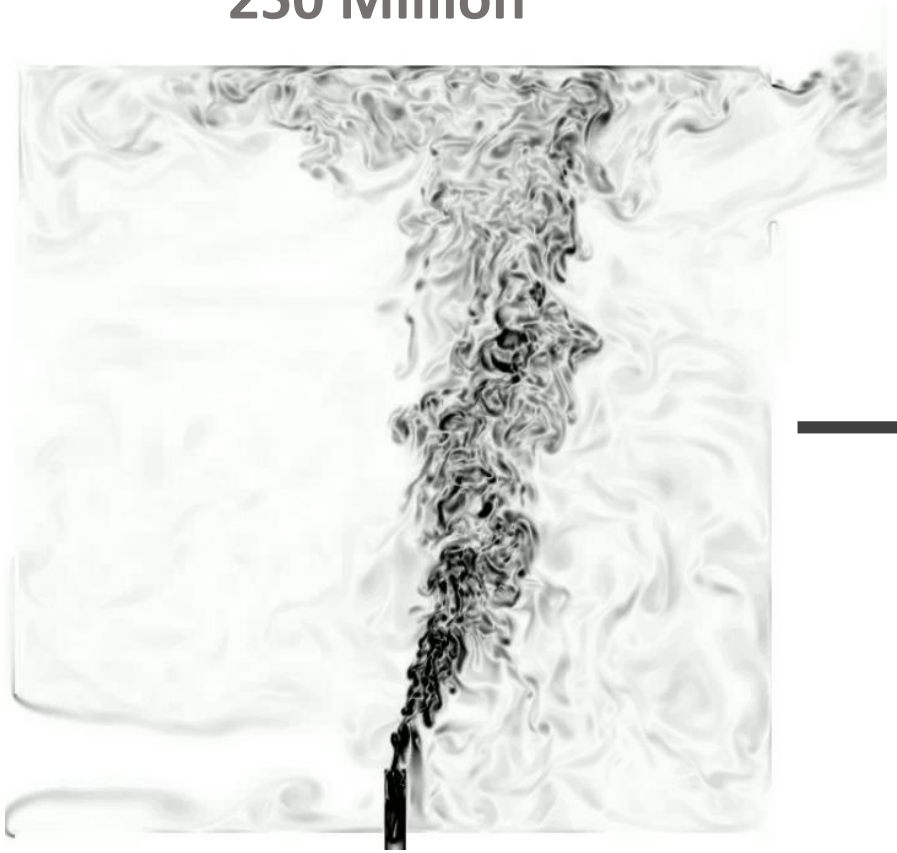


2 billion – cell calculation: DNS simulation with TRUST



2 billion – cell calculation: DNS simulation with TRUST

250 Million



Physical time: 3.5min
to reach steady-state

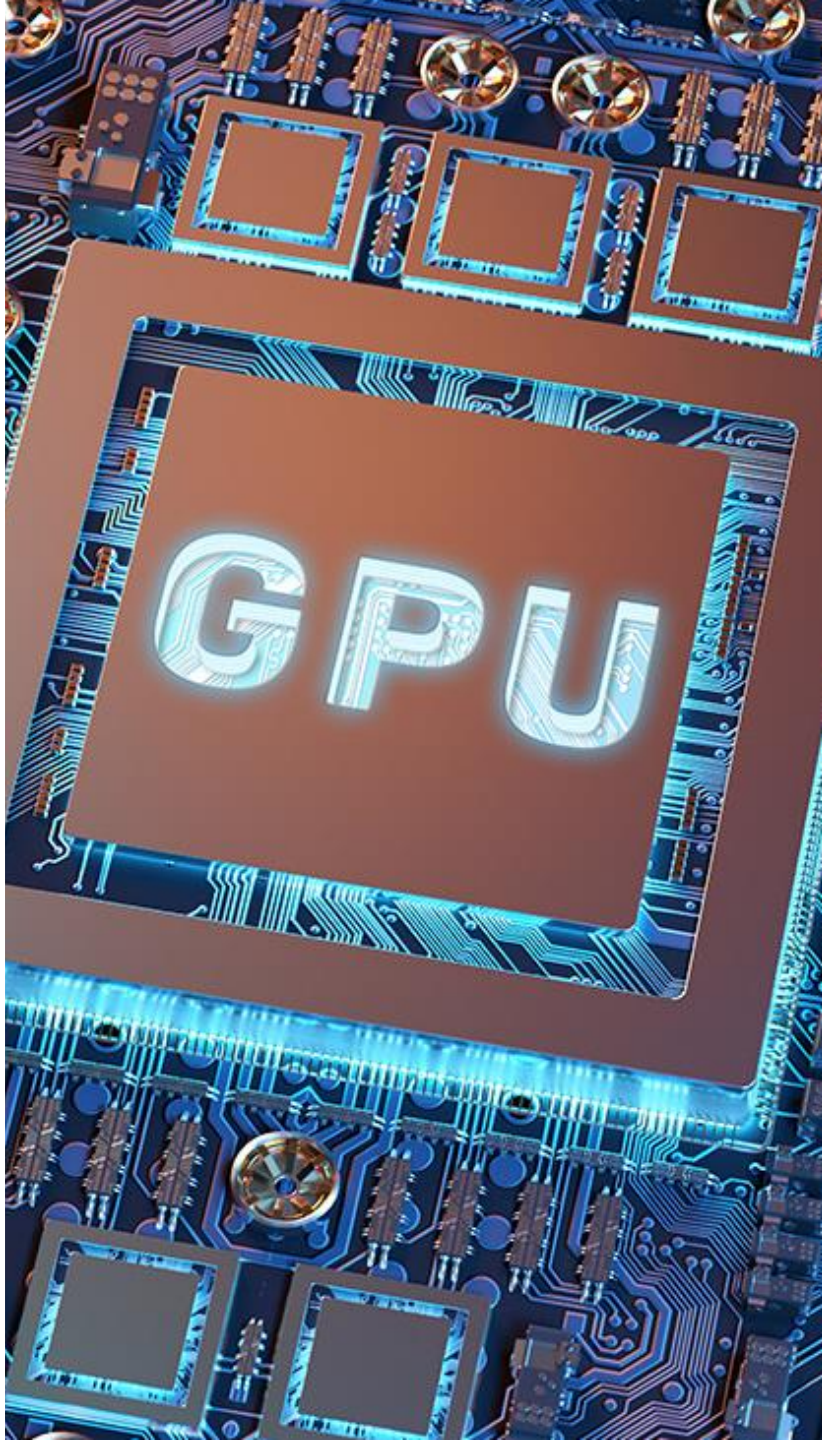
5,000 CPUs

2 Billions

Physical time: 0.5min
50,000 CPUs

Total of
12 M CPU-hours

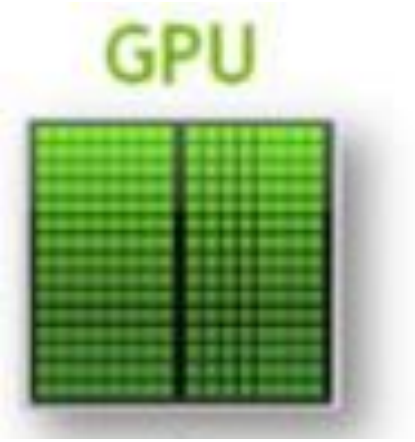
4. Computation enhancement with GPUs



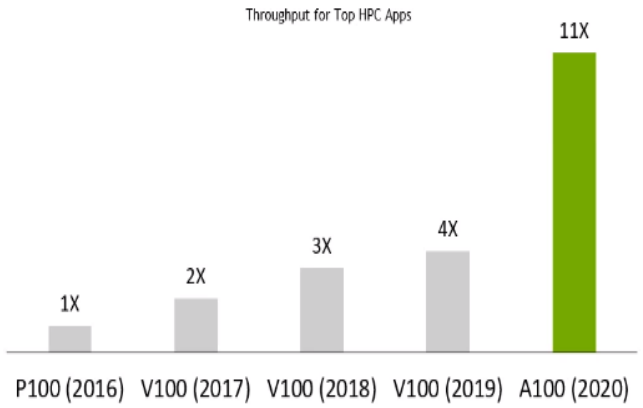
Why developing simulation with GPU?

More efficient for parallel computing

and handling of large amount of data

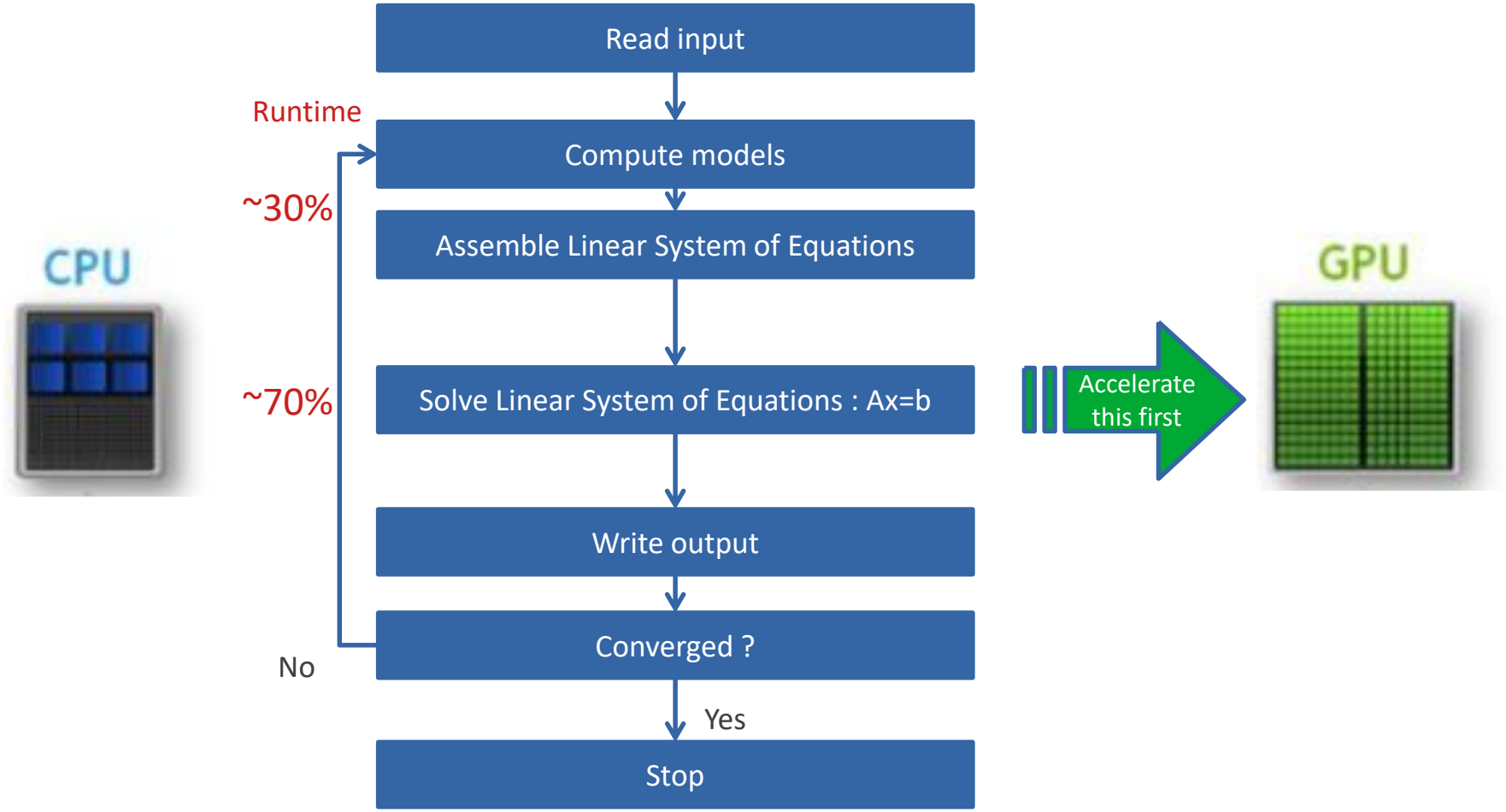


11X MORE HPC PERFORMANCE IN FOUR YEARS



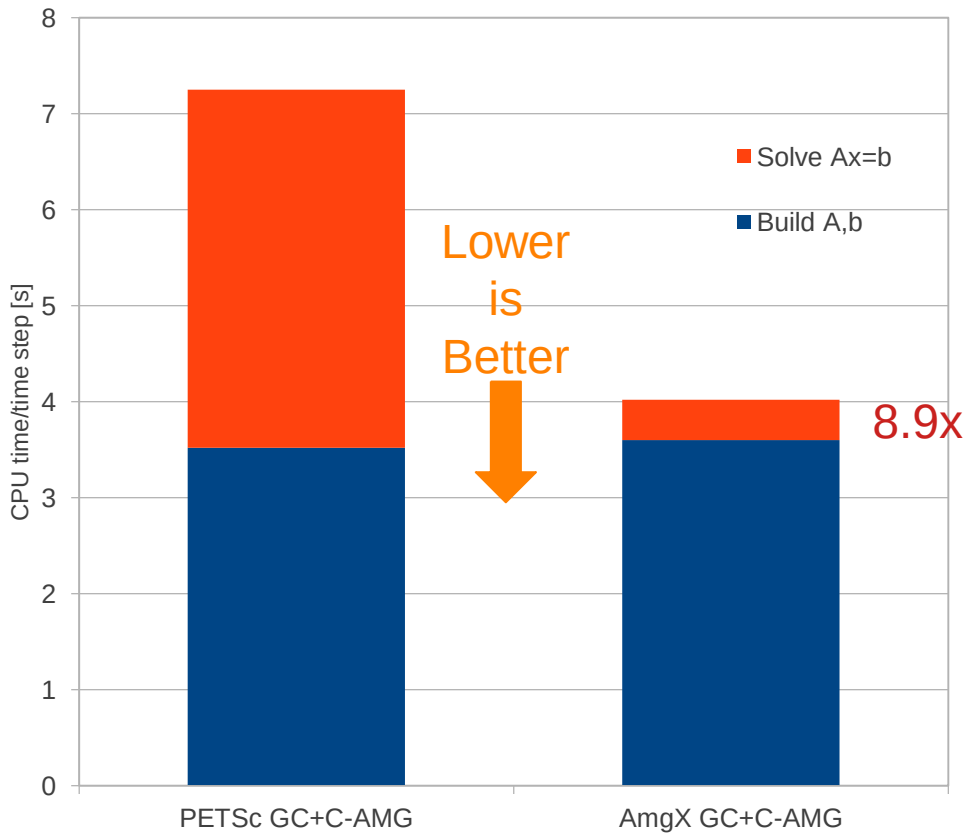
Which strategy for TrioCFD computing on GPU ?

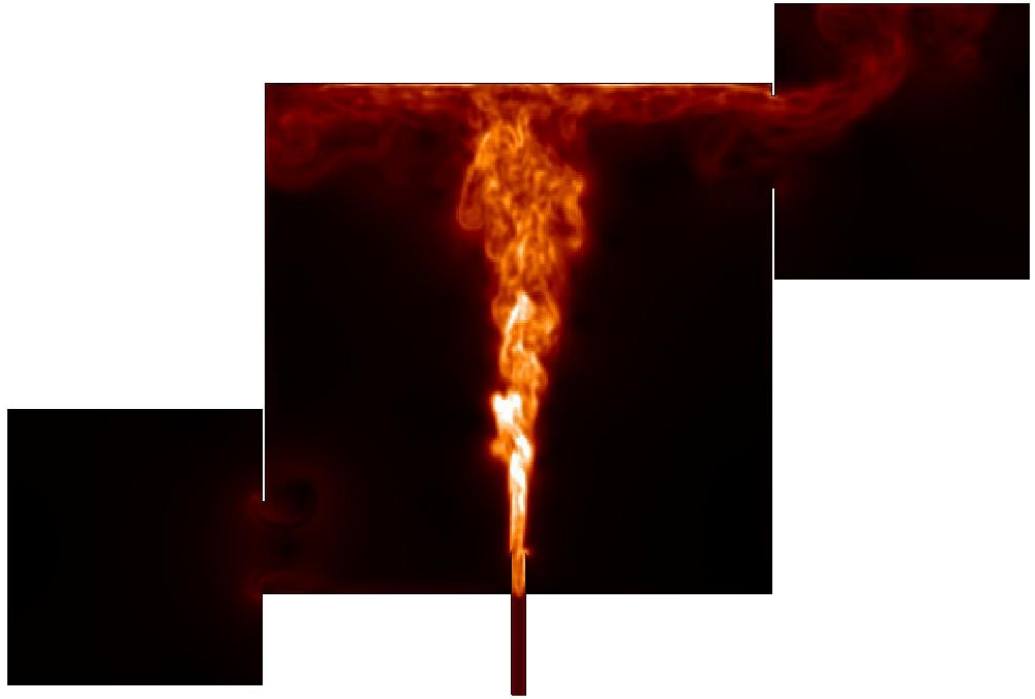
- Detect the most CPU expensive algorithms candidate to GPU



- Benefit firstly from dedicated linear algebra libraries (choice of AmgX open source library formerly initiated by NVIDIA)
- Introduce secondly parallel directives (OpenACC, OpenMP) for the most CPU expensive loops

DNS simulation

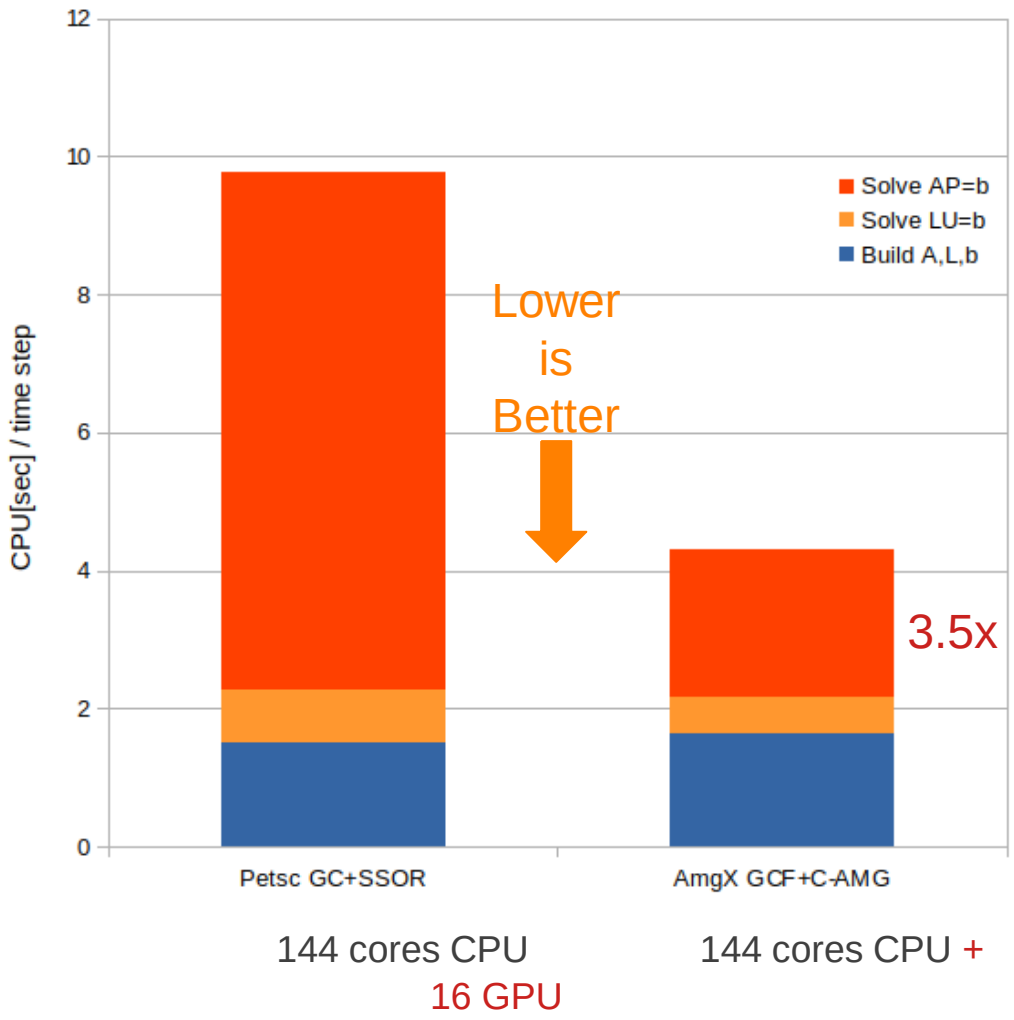


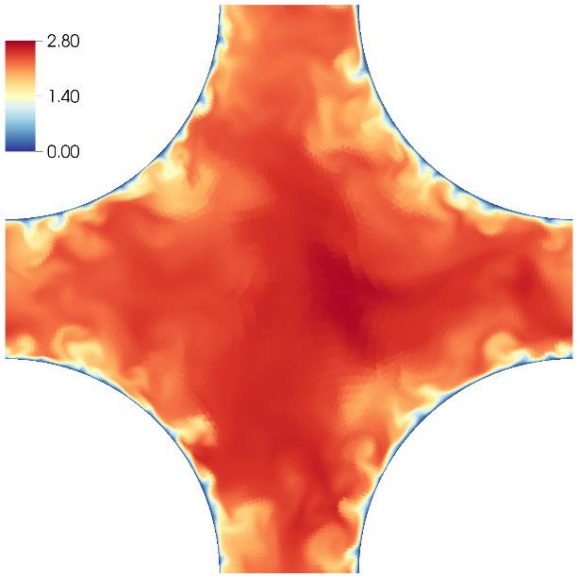


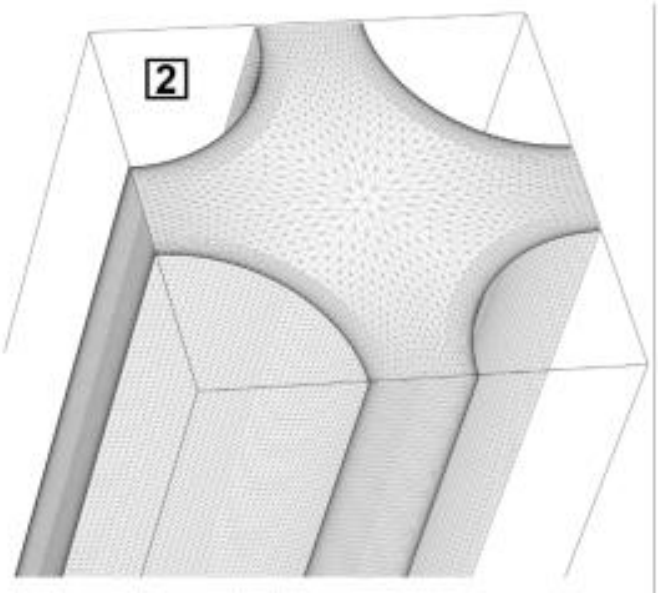
- Mini-GAMELAN geometry
- Structured mesh (VDF)
- 80M cells (250K/core)
- Unsteady DNS
- GC + C-AMG solver
- 50% time into solver

- **1.8x acceleration for the simulation**

LES simulation







- Tube bundle geometry
- Tetra mesh (VEF)
- 4.5M cells (31250/core)
- LES model
- GC + C-AMG solver on GPU
- Implicit diffusion solver on CPU
- 75% time into solvers

- **2.6x acceleration for the simulation**

TRUST/TrioCFD roadmap for GPU computing

2014

- First use of GPU in Trio_U

2020

- Test AmgX, NVidia GPU library
  - Multi-node GPU, more solvers available
- Validate TRUST/TrioCFD on ARM architecture

2021

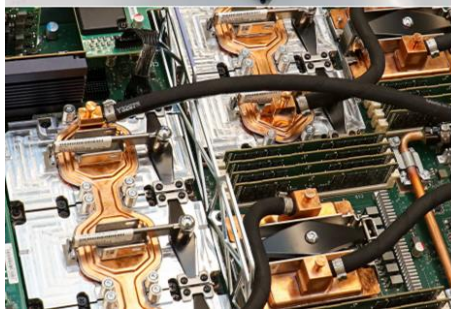
- Add AmgX library to TRUST/TrioCFD (1.8.3)
- Nvidia Hackathon participation
  - Challenge TRUST team to evaluate OpenACC approach (parallel pragma directives)

2022

- TRUST/TrioCFD currently ported on AMD MI250 GPU with OpenMP offloading (AdAstra will be 10th of top500)

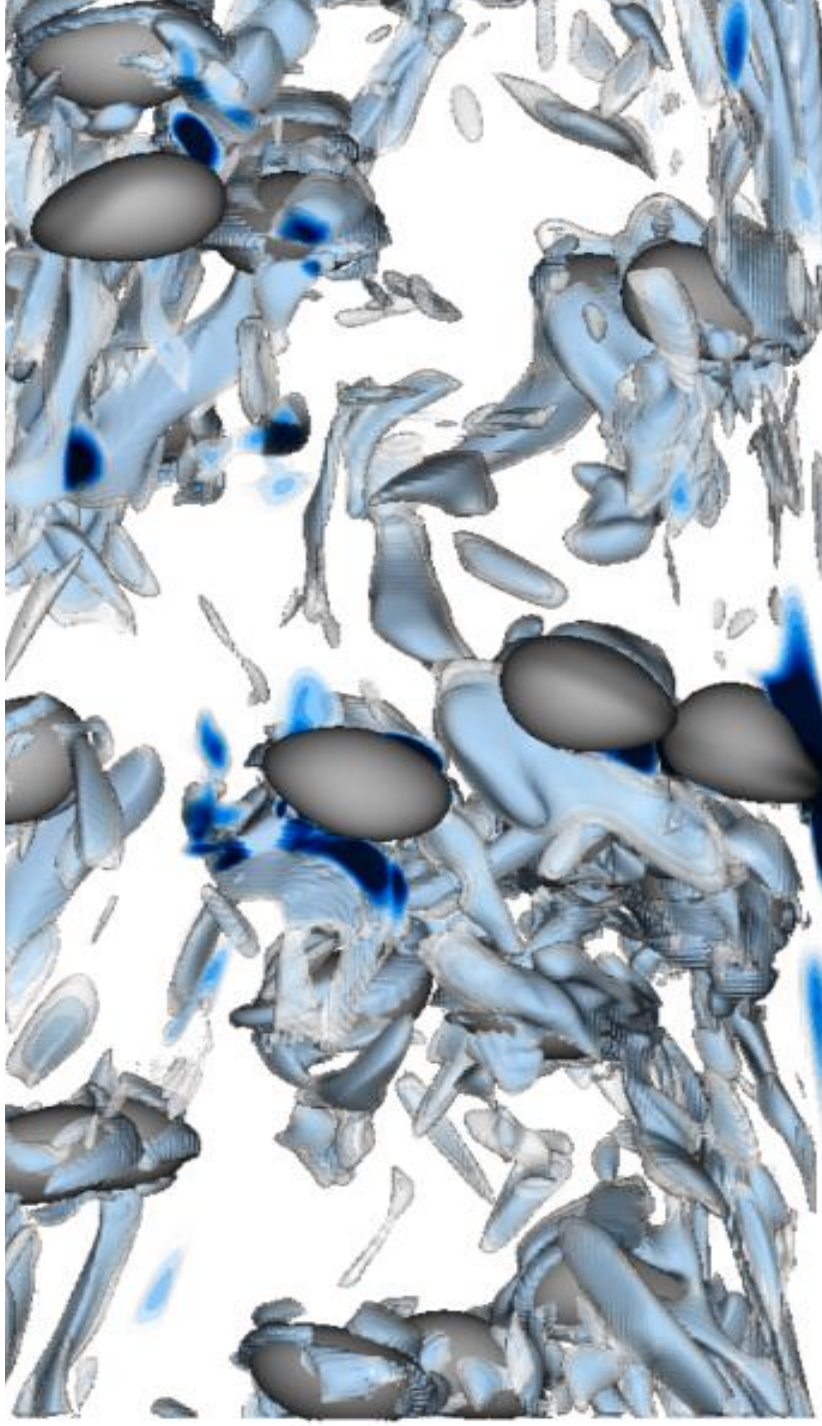
2024

First run on french exascale supercomputer



Conclusion

1. TRUST and TrioCFD
2. Parallel computing: Overview, performance and simplicity
3. Examples of massive parallel computations
4. Towards GPU-accelerated simulations



Conclusion

- TRUST : Thermal-hydraulic IT Platform
  TrioCFD : CFD code base on TRUST

  → Both Open-Source

- The codes are made for parallel computations → Massively Parallel / Simplicity of use

- Work in progress to enhance GPU-accelerated simulations → High potential

Thanks for you attention

Wednesday 22th, 2022

Raksmy NOP