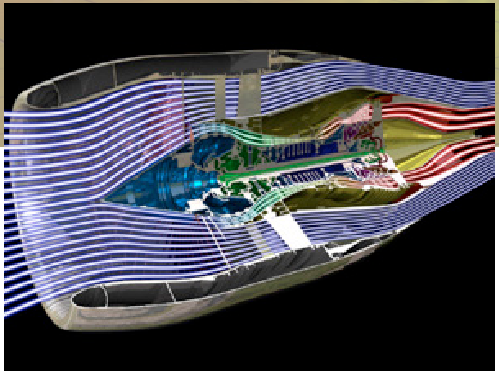
PARTNERSHIP FOR
ADVANCED COMPUTING IN EUROPE

# High Performance Computing of gas turbine flows: current and future trends



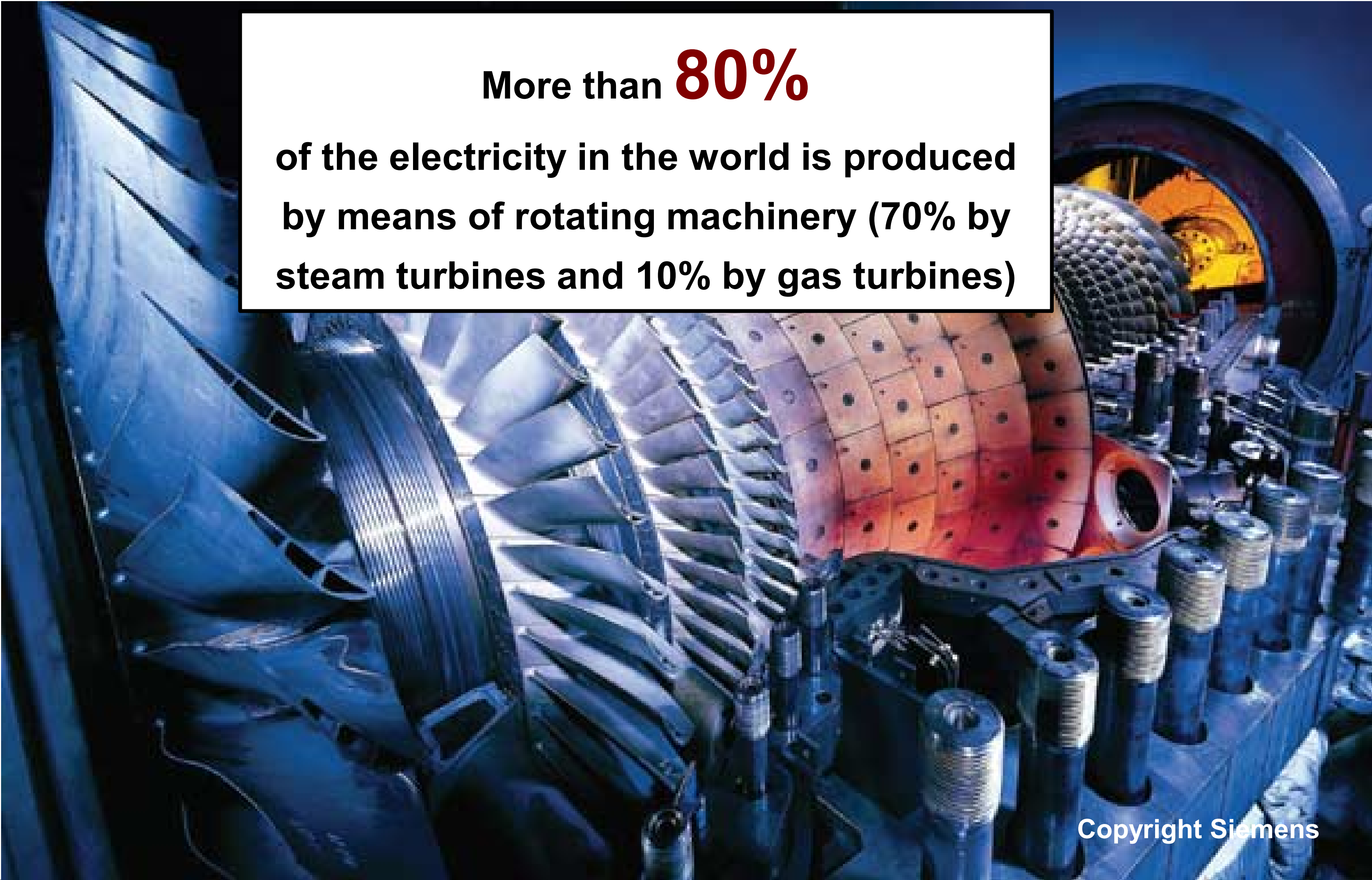
**Nicolas Gourdain, CERFACS**
http://www.cerfacs.fr
**Michel Gazaix, ONERA**
http://www.onera.fr

PRACE Autumn School 2013 - Industry Oriented HPC Simulations, September 21-27,
University of Ljubljana, Faculty of Mechanical Engineering, Ljubljana, Slovenia

# Context

**ENERGY = COMBUSTION**

# Context



More than **80%**

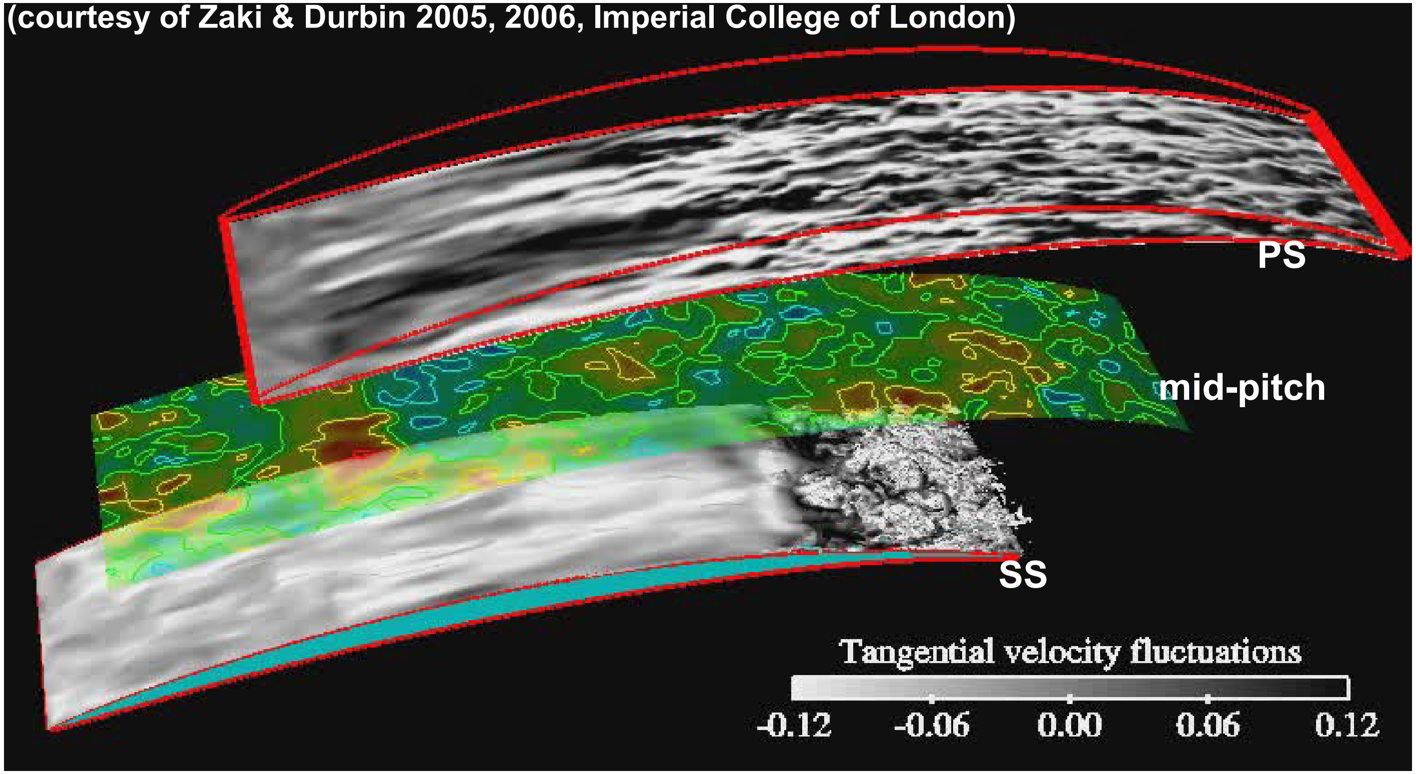
**of the electricity in the world is produced by means of rotating machinery (70% by steam turbines and 10% by gas turbines)**

Copyright Siemens

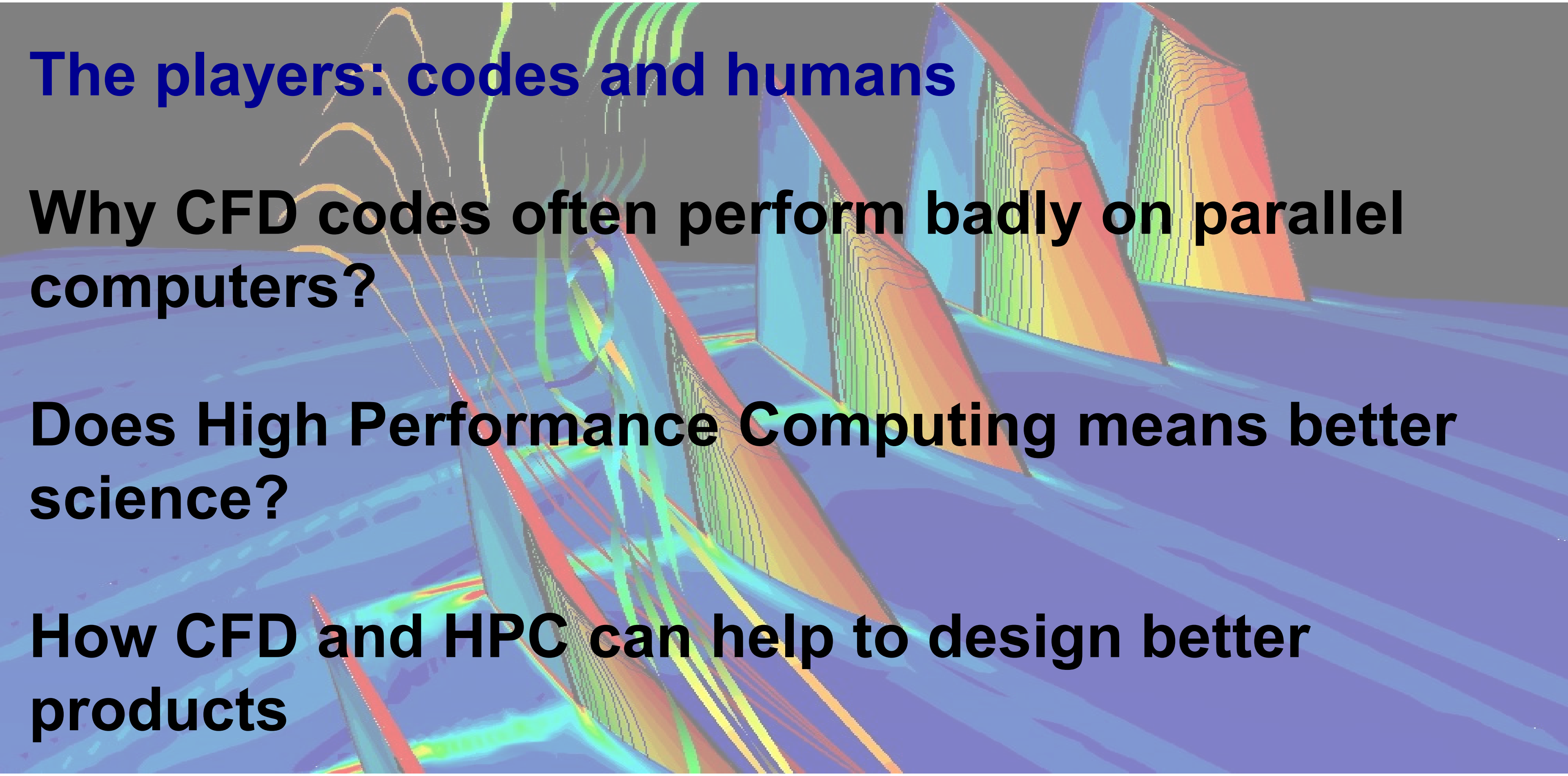
# Why such flows are so challenging for CFD?

"The (turbomachinery) flow is a veritable-fluid-dynamical "zoo", characterized by separation, reattachment, transition, relaminarization, retransition, etc. all often occurring in the same flow." *Jahanmiri, Chalmers University, 2011*

**Transition on a typical compressor airfoil at design conditions (LES)**


(courtesy of Zaki & Durbin 2005, 2006, Imperial College of London)

# Overview



**The players: codes and humans**

**Why CFD codes often perform badly on parallel computers?**

**Does High Performance Computing means better science?**

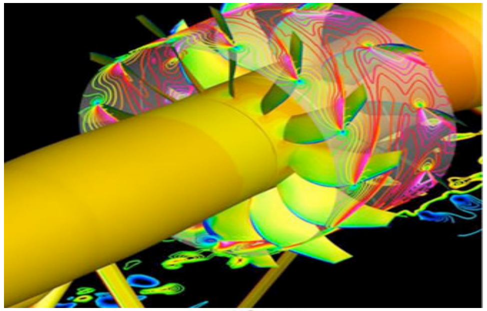
**How CFD and HPC can help to design better products**

# ONERA: the French aerospace lab.

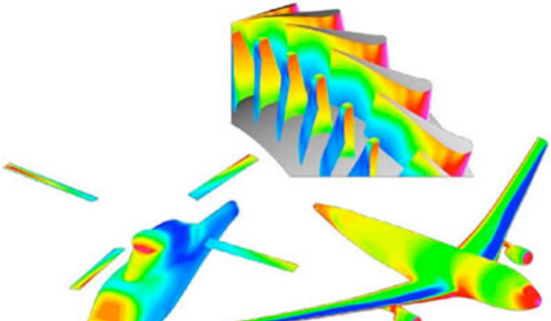
**About 2000 people in three competence centers:**

**Aeronautics, space and defense**

- **Promote breaking technologies**



- **Science, from labs to industry**



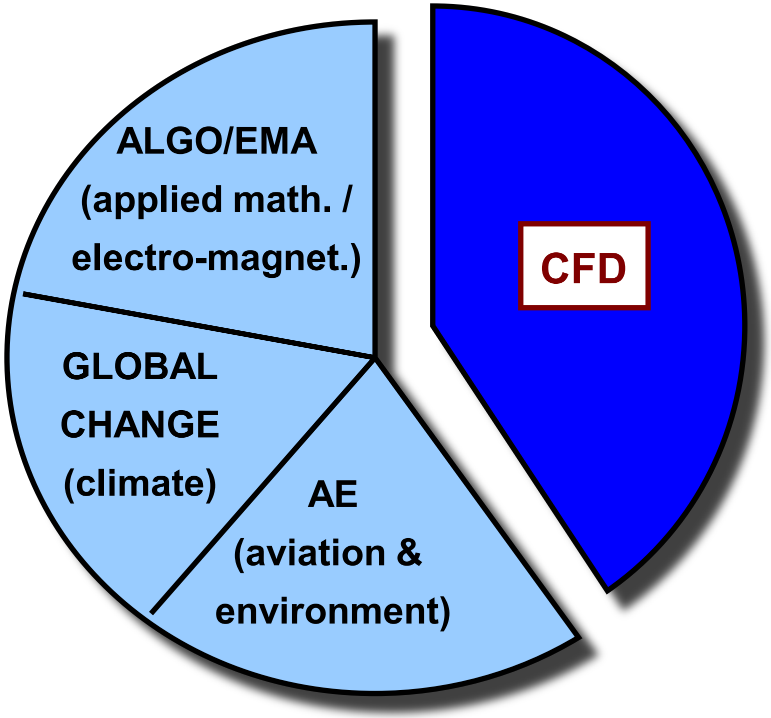
- **Train highly skilled engineers and scientists**

## CERFACS has seven shareholders

EDF
cnes
CENTRE NATIONAL D'ÉTUDES SPATIALES
TOTAL
EADS
METEO FRANCE
Toujours un temps d'avance
SAFRAN
AEROSPACE · DEFENCE · SECURITY
ONERA
THE FRENCH AEROSPACE LAB

## 120 people in 4 scientific teams

## Located in Toulouse (France)



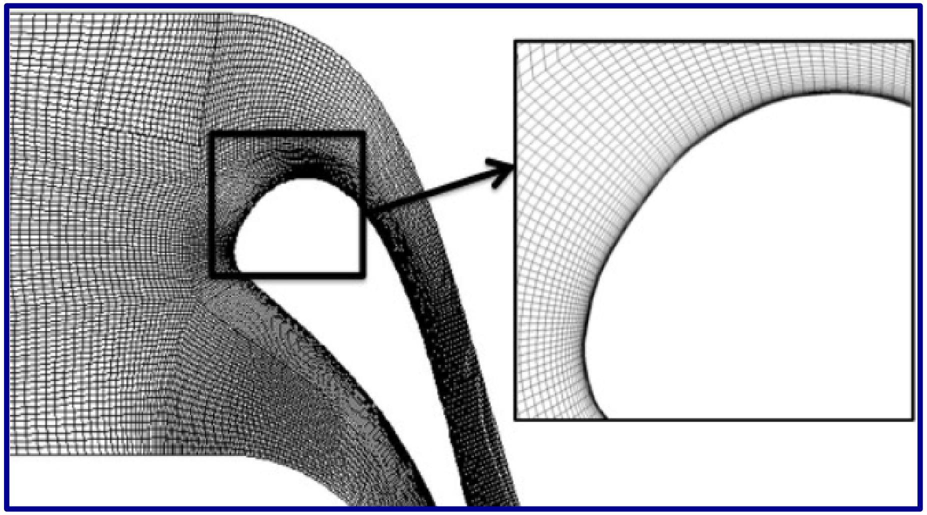
- **30 to 40** publications per year (peer-reviewed journals)
- **10 to 15** Ph.D. thesis per year (80% go to industry after their PhD)

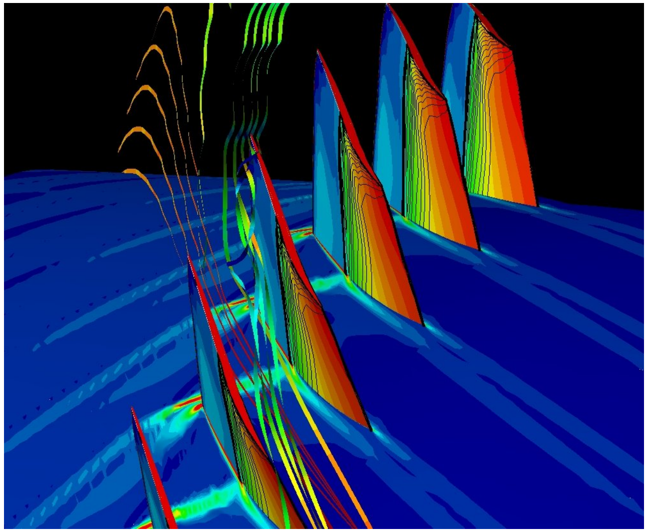
# Presentation of CFD solvers

**elsA** → **a structured code for aerodynamics**

**(ensemble logiciel pour la simulation en Aérodynamique)**

- Mainly developed by ONERA and CERFACS [1] and used by industrial partners (AIRBUS, EUROCOPTER, SAFRAN, EDF, etc.),
- Compressible finite volume flow solver with multi-blocks structured meshes (Chimera, non-coincident interfaces, etc.)
- **Massively parallel** capabilities (MPI) [2]: o($10^4$) cores
- **(U)RANS, DES and LES** approaches



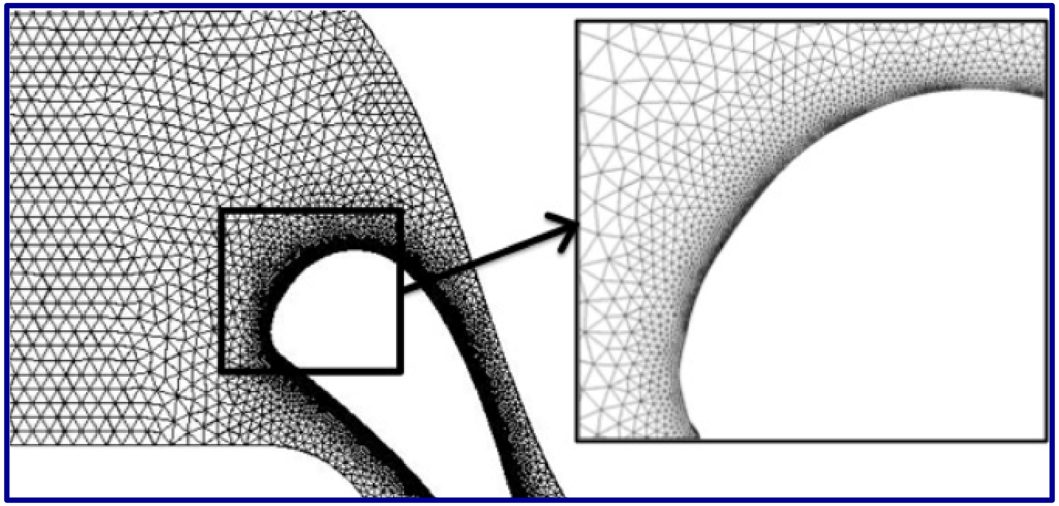


[1] Cambier and Veuillot, AIAA, 2008

[2] Gourdain *et al.* J. Comp. Sc. Discovery, 2009

# *AVBP* ⟶ an unstructured code for reactive flows

- Mainly developed by CERFACS and IFP [1] and used by academic and industrial partners (SAFRAN, etc.)
- Fully compressible turbulent reacting flows
- Unstructured hexaedral, tetraedral, prisms & hybrid meshes
- **Only DNS and LES** approaches
- **Massively parallel** capabilities (MPI, ParMETIS): o($10^5$) cores
- A dedicated version is available for turbomachinery (Wang *et al.*, ASME 2013)



[1] Garcia, PhD, 2008

# These CFD codes are not commercial software with HMI (graphics interfaces), but …

# … they are used to design this

© AIRBUS S.A.S. 2013 - photo by e'm company / H. GOUSSÉ

# … they are used to design

# or this

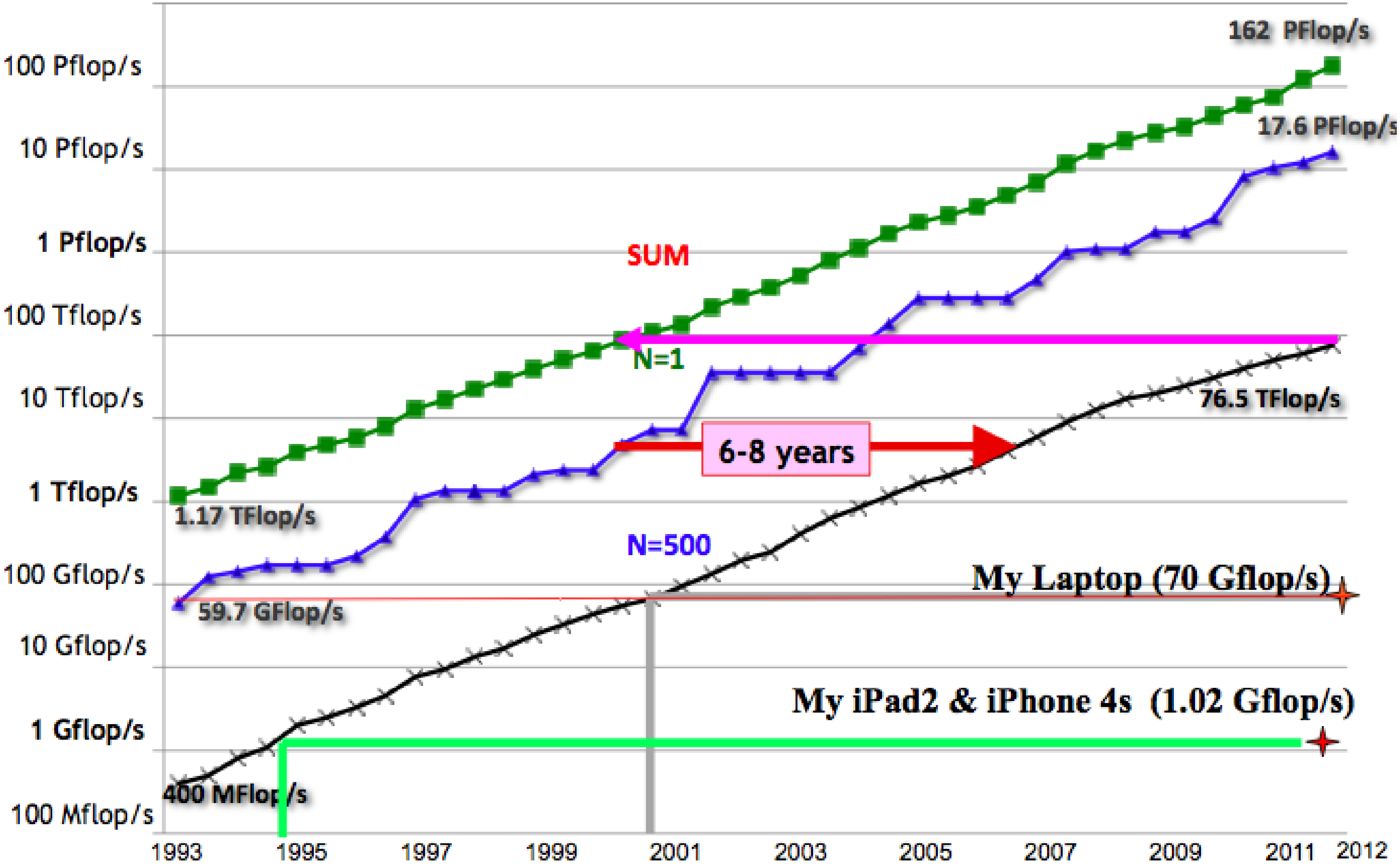
# Actually a large range of aeronautics products!

© AIRBUS S.A.S. 2013 - photo by e'm company / H. GOUSSÉ

# What does more CPU power really mean for industry?

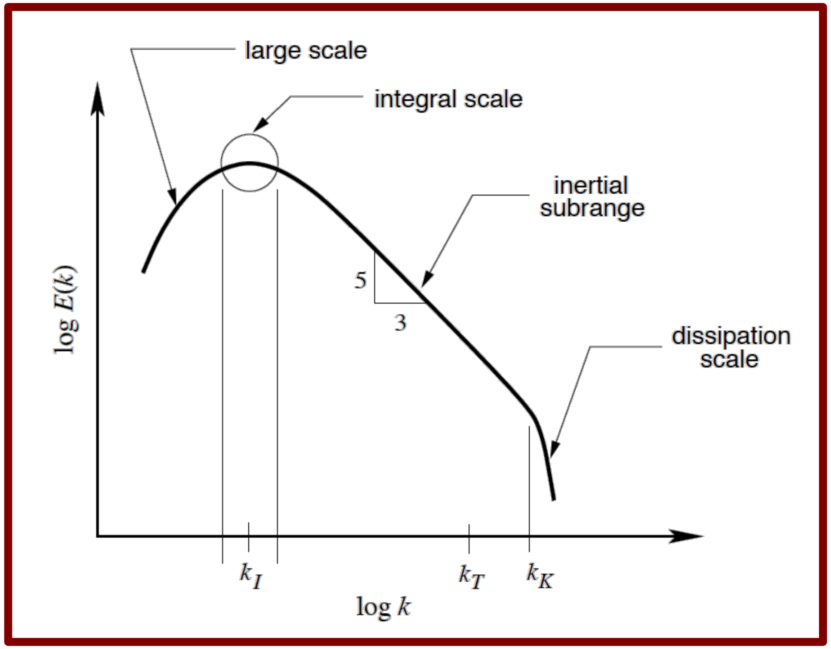
# Evolution of the computing power

Data from www.top500.org



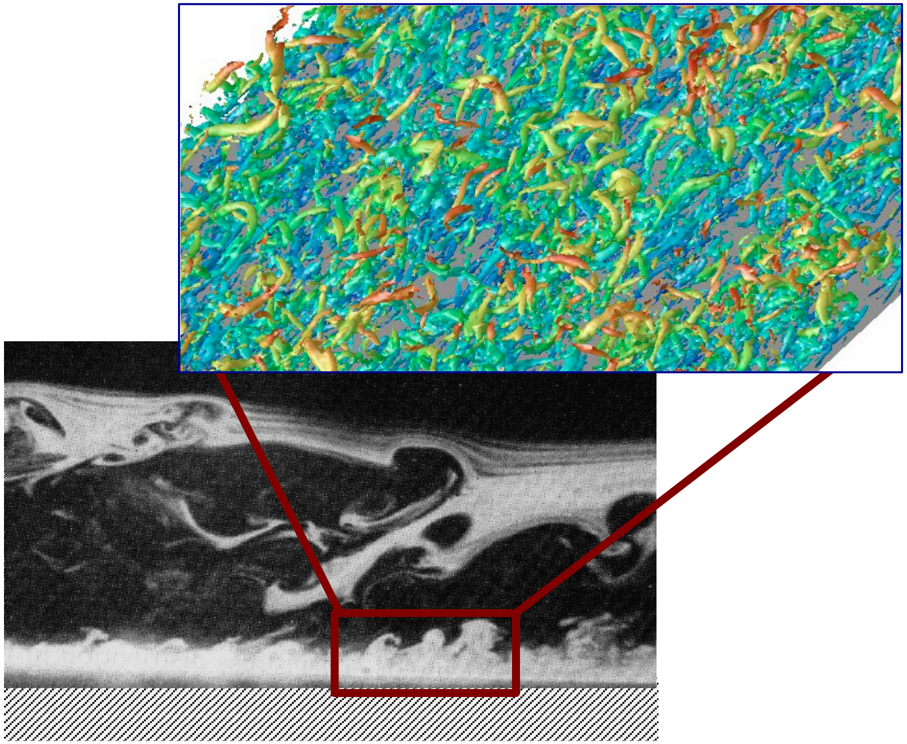
N. Gourdain, PRACE meeting 2013

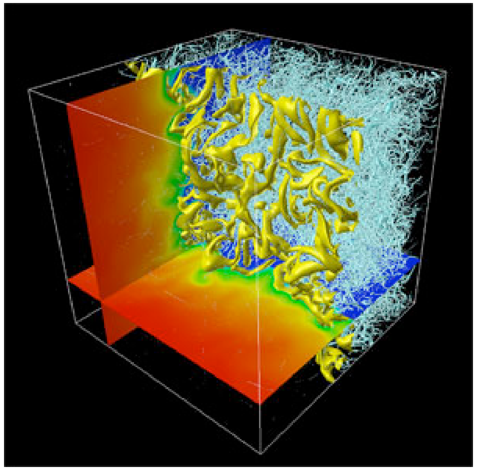
# Don’t forget: turbulence is not CFD-friendly



## Wall turbulence

**Schlatter *et al.*, PoF, 2009**





## Freestream turbulence

**M. Toshio and T. Mamoru**

**Tokyo Institute of Technology**

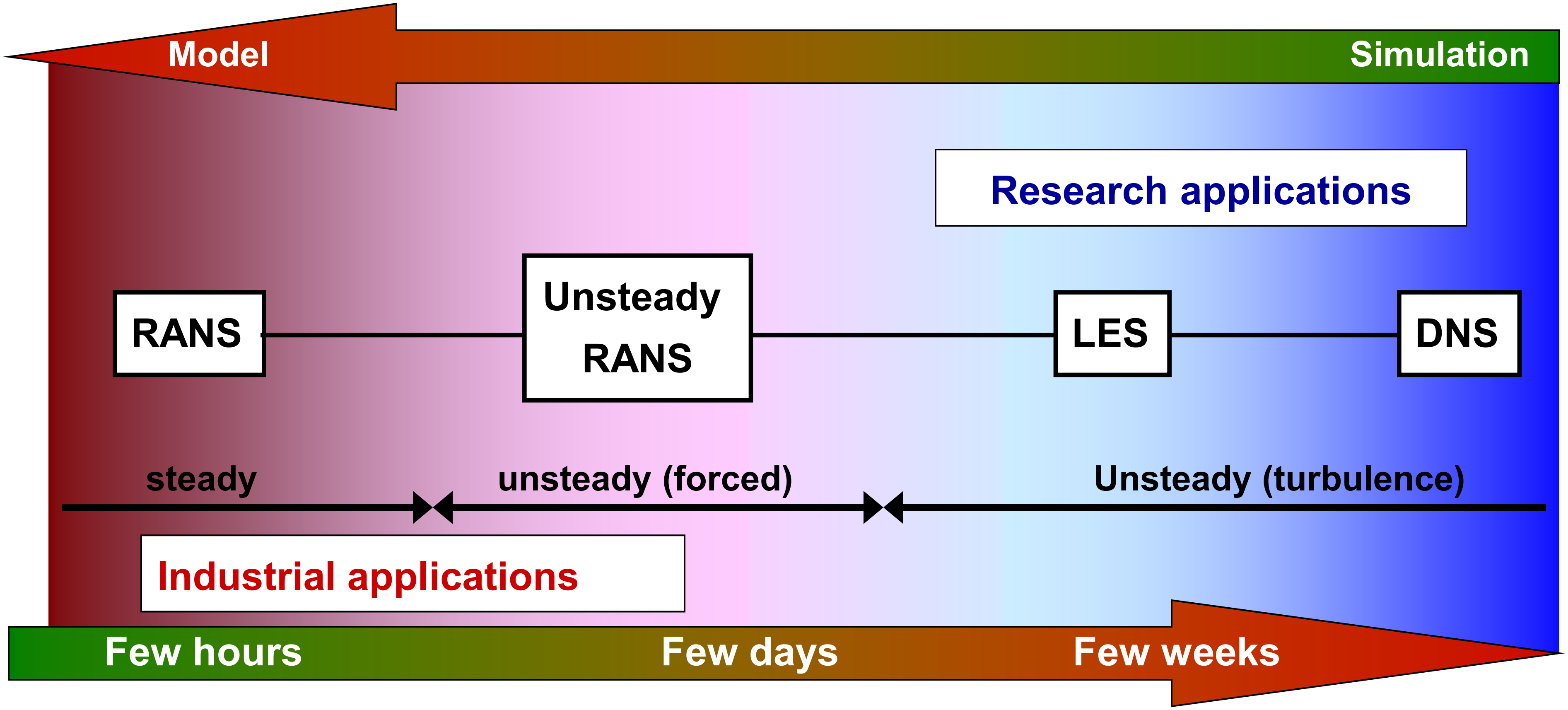
# Can supercomputers accurately predict the Future[*]?

## And the performance of my real life system…

[*] Tiffany Trader, http://www.hpcwire.com, 2013

# How to predict turbulence?

## Overview of the computational methods



**RANS: Reynolds-Averaged Navier Stokes**

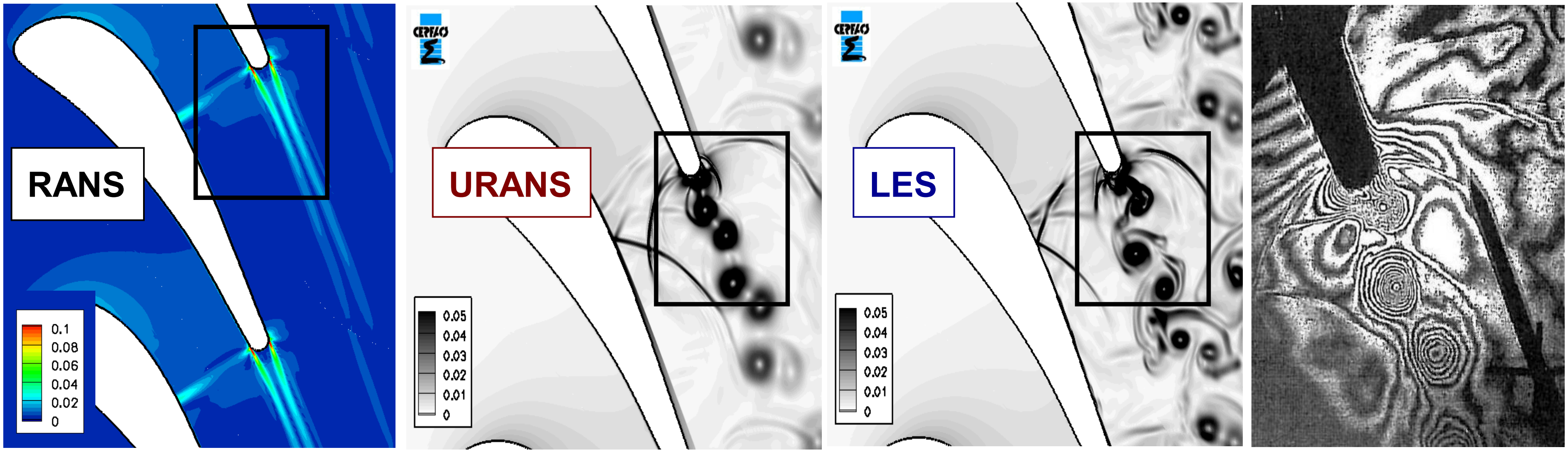
**LES: Large Eddy Simulation**

**DNS: Direct Numerical Simulation**

# Examples of numerical results

Density gradientv flow field

Experiments*

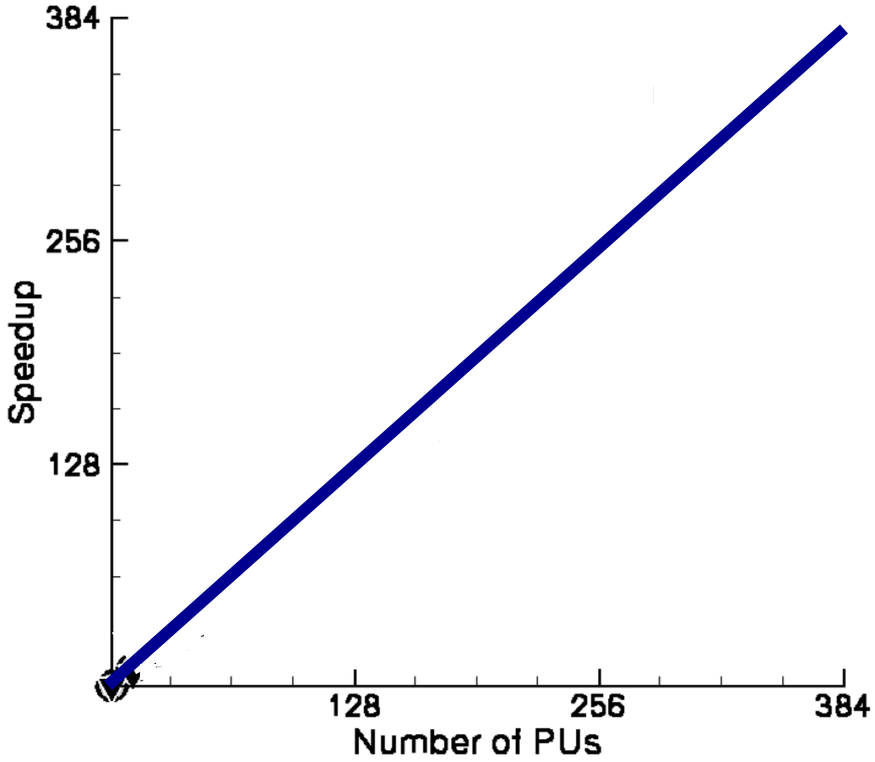


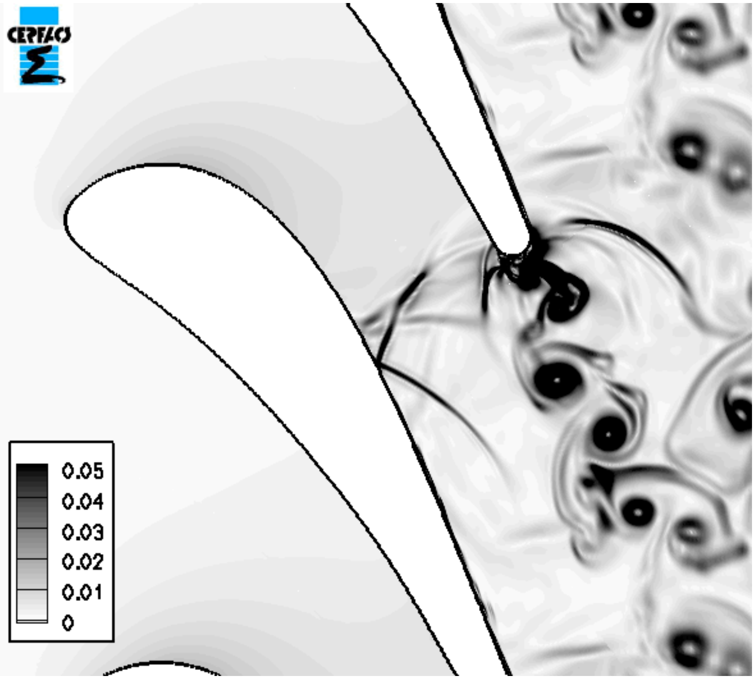
---

*C. Sieverding, H. Richard, and J.M. Desse, "Turbine Blade Trailing Edge Flow Characteristics at High Subsonic Outlet Mach Number", J. of Turbomachinery, 2003

# What we want

## The ideal speed-up… ☺



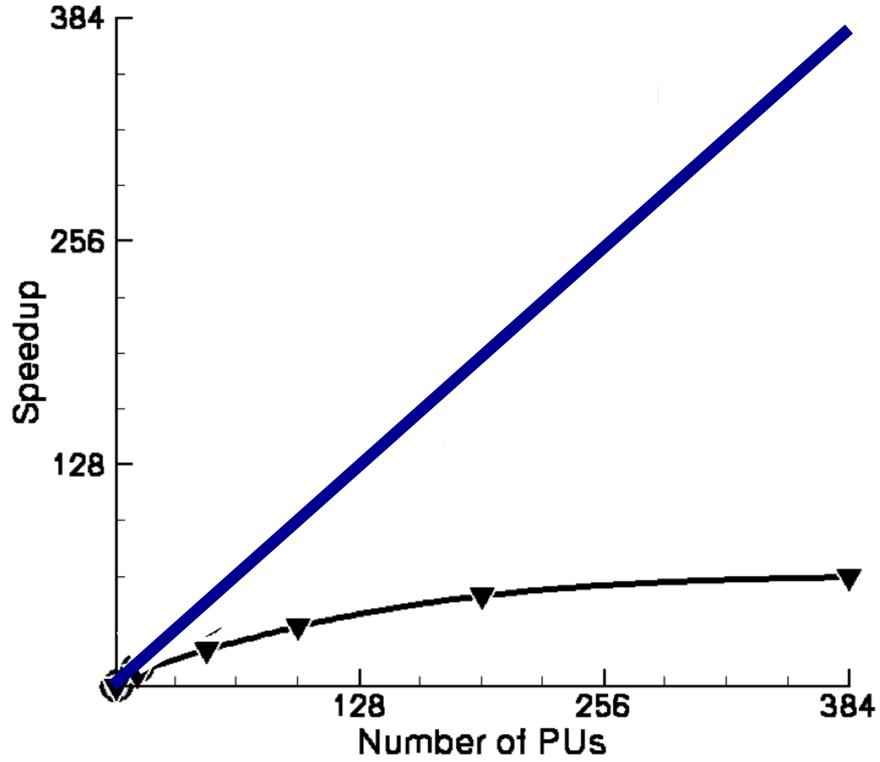


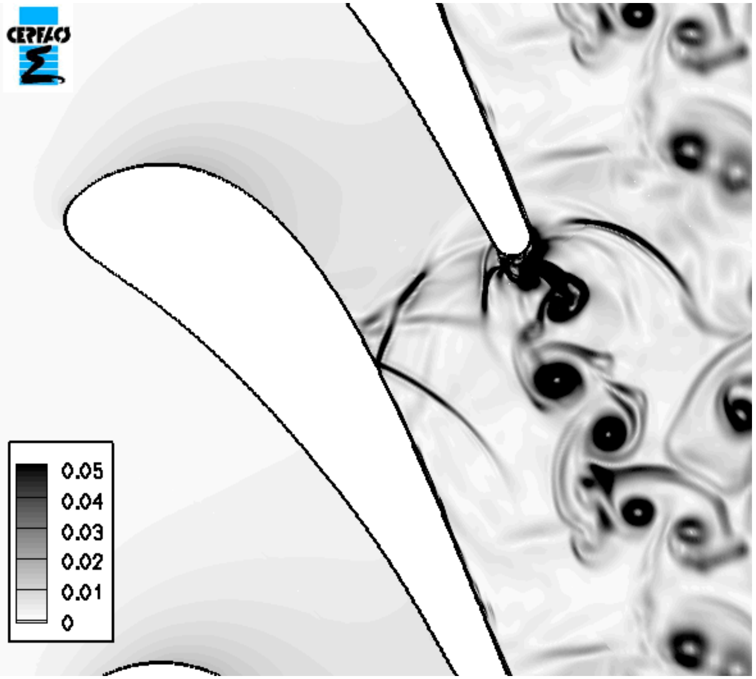
Leonard *et al.*, ASME Turbo Expo, 2010

# Parallel computing efficiency

## What we have

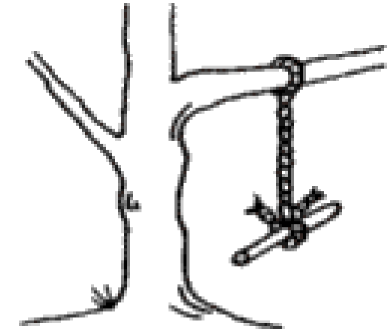
**Not really the ideal speed-up… ☹**

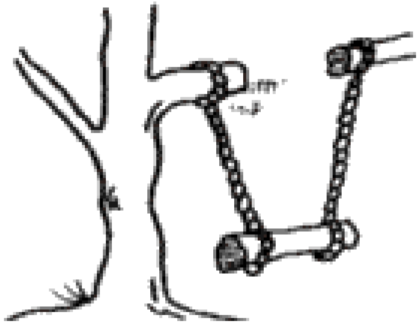


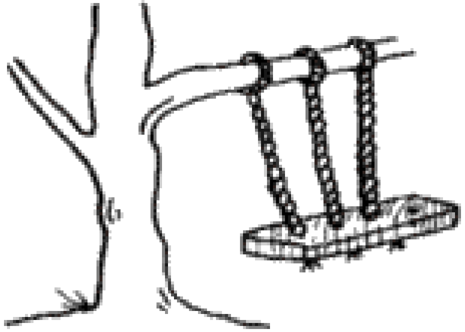


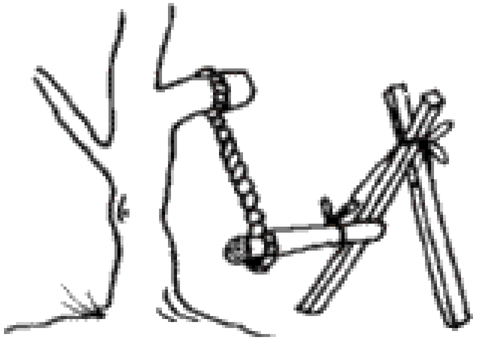
Leonard *et al.*, ASME Turbo Expo, 2010

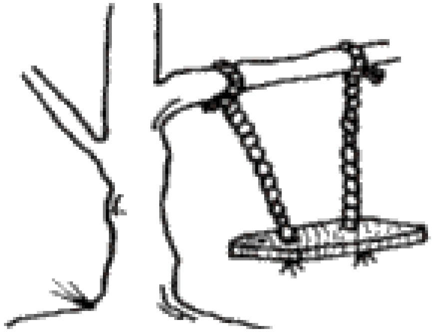
# Why my parallel efficiency can be so bad?

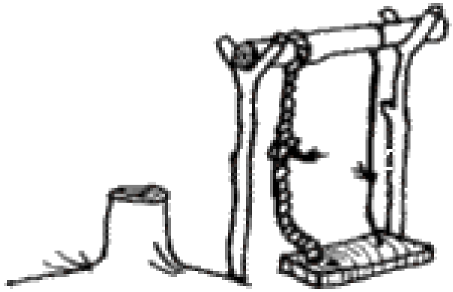

What the user asked for


How the analyst saw it


How the system was designed


As the programmer wrote it

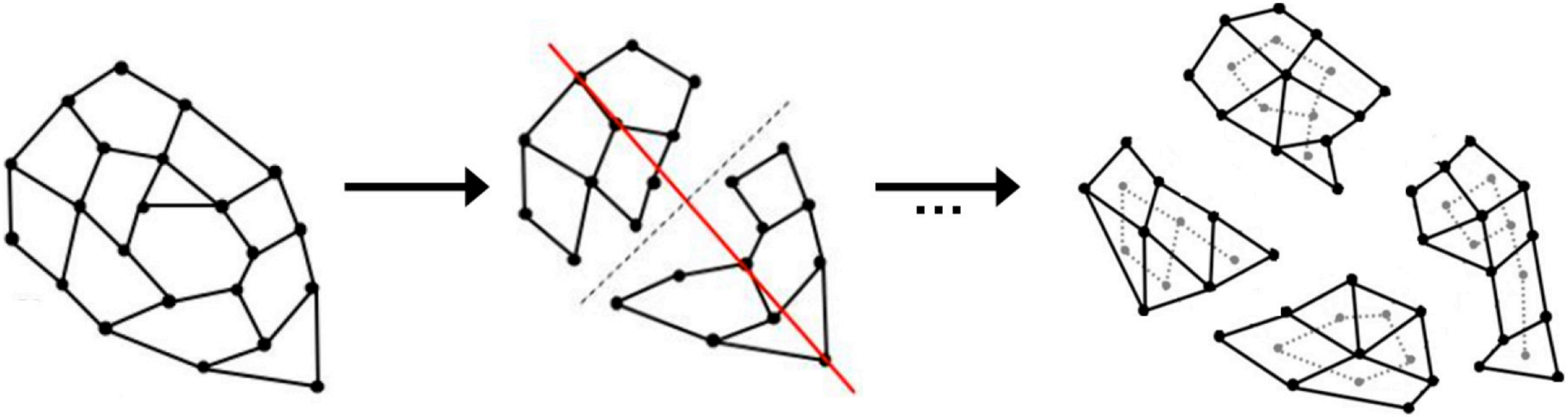

What the user really wanted


How it actually works

# Mesh-partitioning & load balancing

## Examples of algorithms

**Recursive Coordinate/Inertial Bisection (RCB/RIB): geometric based algorithms**



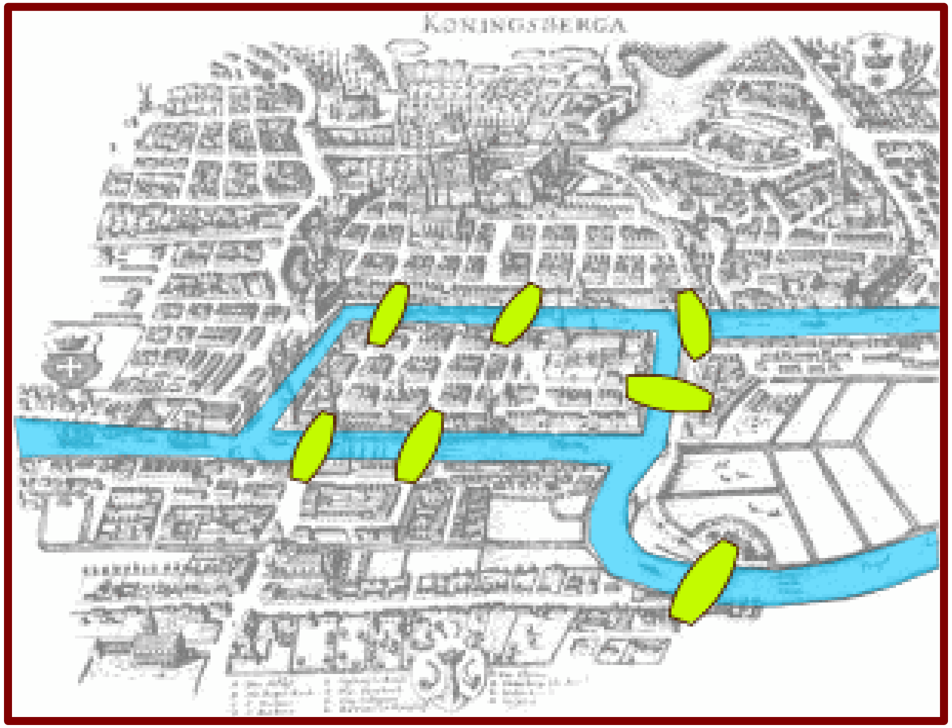
**Example of a partitioning process with an unstructured grid and the RIB algorithm**

# Mesh-partitioning & load balancing

## Examples of algorithms

**Recursive Graph Bisection (RGB): graph theory based algorithm**

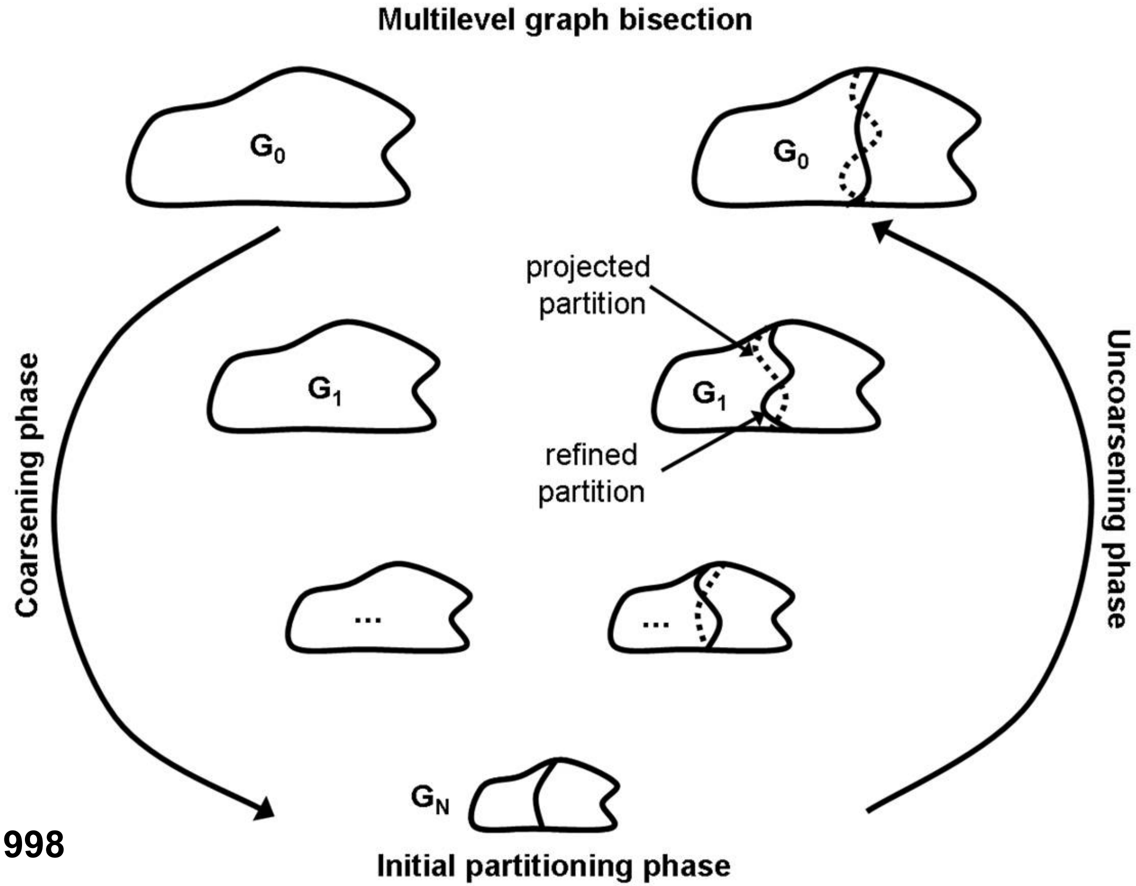
**Problem: find a walk through the city that would cross each bridge once and only once**



**The Seven Bridges of Konigsberg problem (Euler, 1736)**

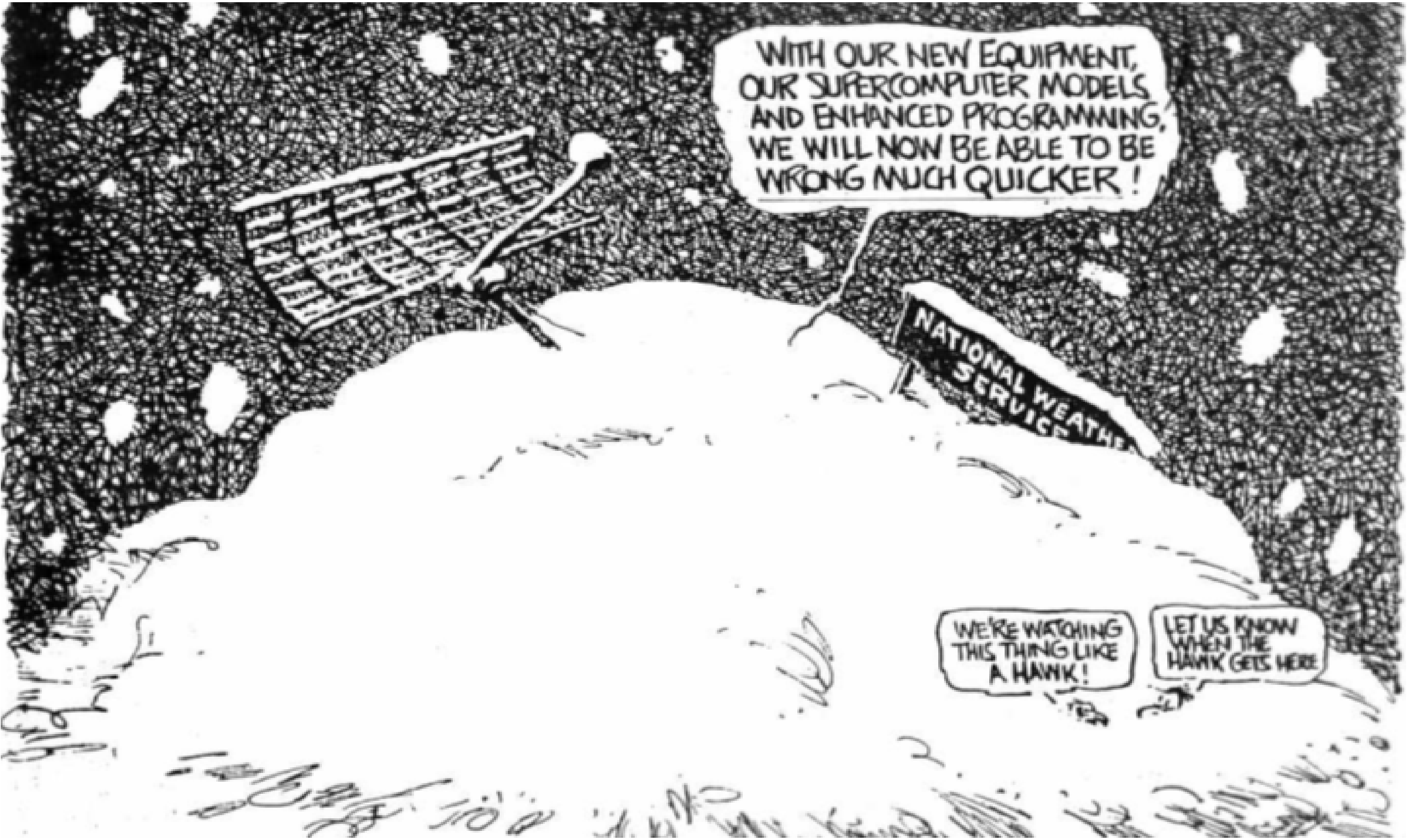
## Examples of algorithms

### METIS: multi-constraint multilevel graph partitioning



**Karypis *et al.*, 1998**

# Let’s talk about real life



# Mesh-partitioning & load balancing

## Real life is complicated for CFD…

**Adaptive mesh refinement** (AMR)
Elements/cells are added or removed during the simulation, so the refinement is modified and load balance errors can occur

**Adaptive physics models**
Computational effort associated with data points varies over time, thus points may need to be redistributed among computing cores to balance the work

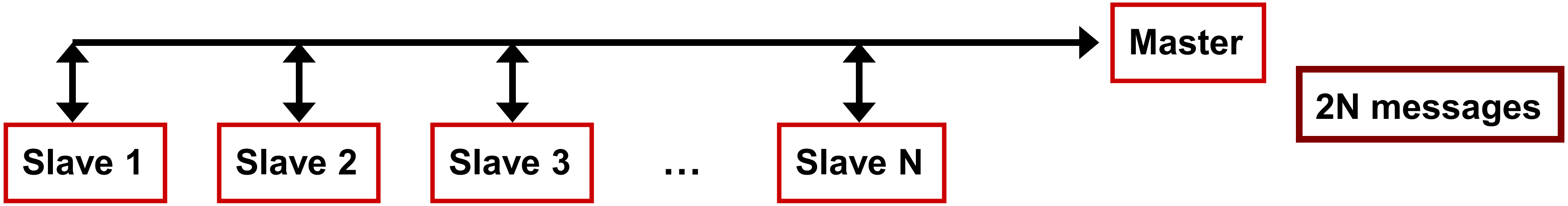
**Particle simulations**
Particles interact with (geometrically) near neighbors that can vary over time, requiring a new distribution of the particles among computing cores

**Multiphysics simulations**
Coupling of multiple physical phenomena into a single simulation required multiple mesh-partitioning

**(1) Hendrickson *et al.*, 2000**

## Communication strategy: MPI

**Point-to-point** communications (between 2 cores): MPI_*send, MPI_*recv…



**Master**

**Slave 1**

**Slave 2**

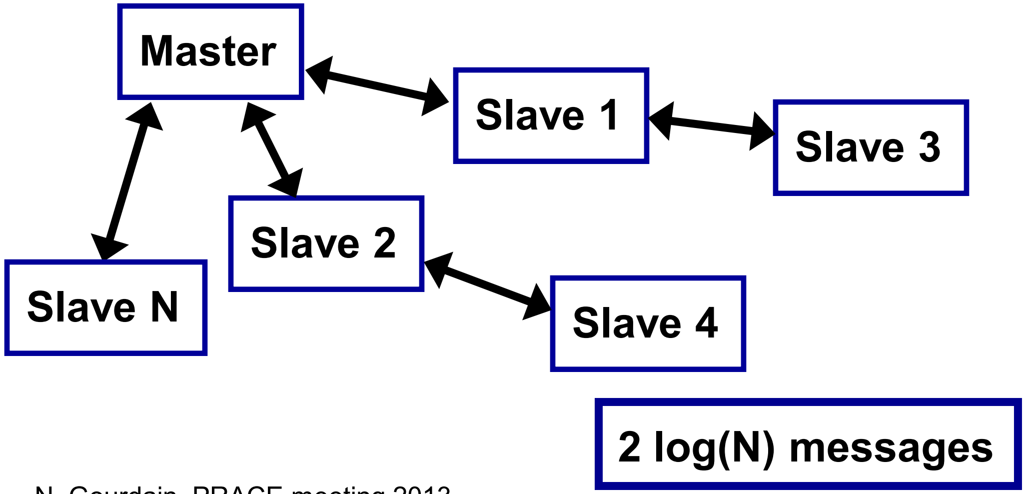
**Slave 3**

**…**

**Slave N**

**2N messages**

**Collective** communications (between a set of cores): MPI_Allreduce, MPI_Allgather, MPI_Bcast… (based on a tree graph: P ➔ In(P) message).



**Master**

**Slave 1**

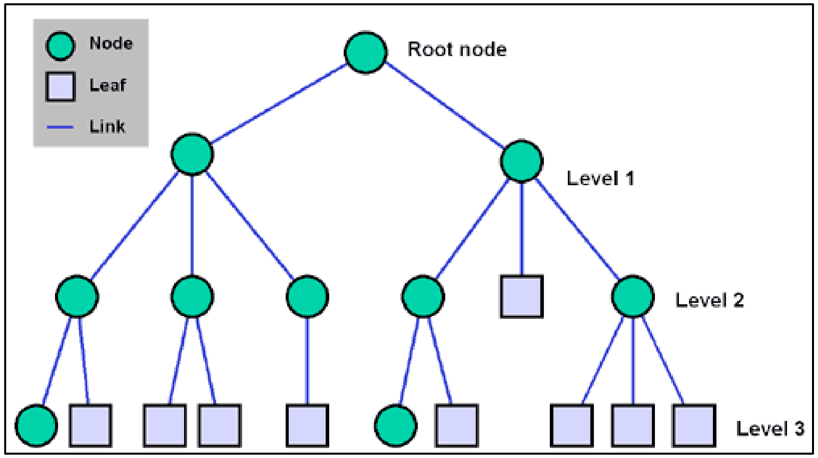
**Slave 3**

**Slave 2**

**Slave 4**

**Slave N**

**2 log(N) messages**



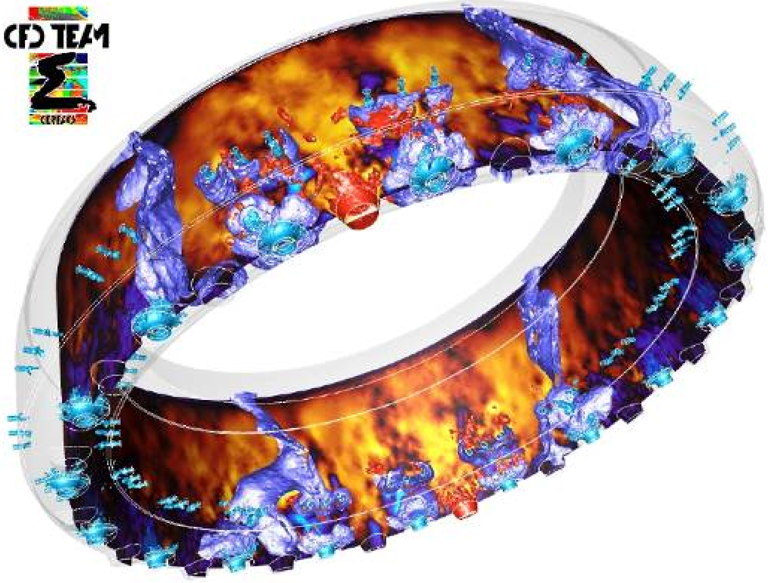
# Communication strategies

## Communication strategy (AVBP)

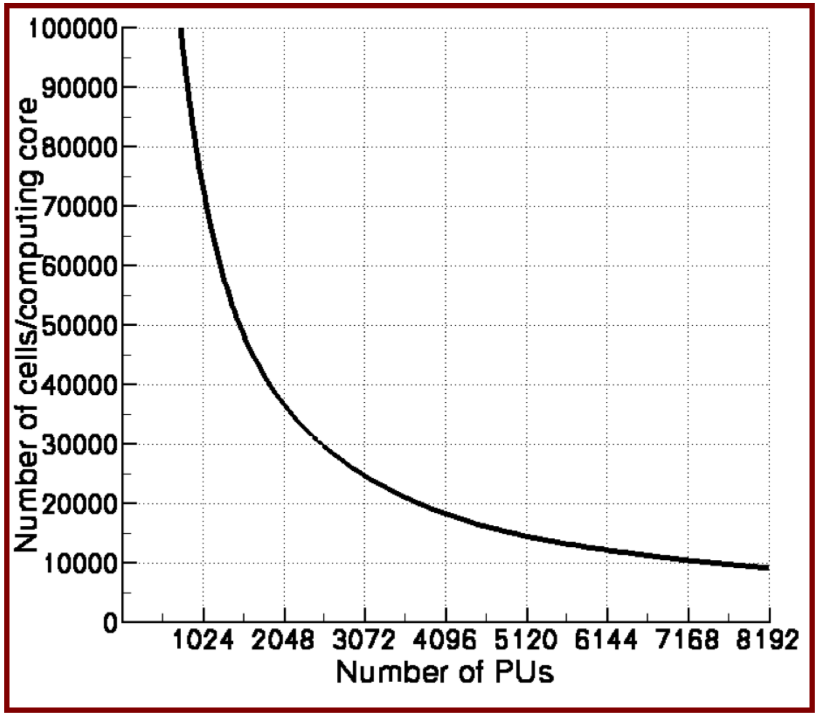
**MPI non-blocking** communications (AVBP)

All input/output are dedicated to one processor (the master process)

**Slaves have to communicate with the master for I/O**



**Application to a combustion chamber (70M cells)**



SGI Altix platform (GENCI - CINES)

# Communication strategy (AVBP)

**MPI non-blocking** communications (AVBP)

All input/output are dedicated to one processor (the master process)

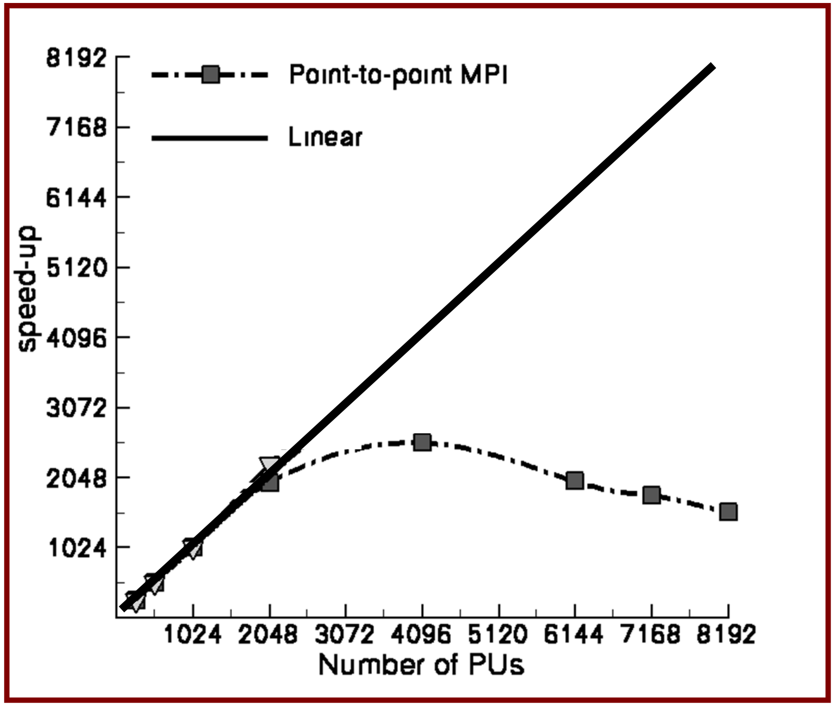
**Slaves have to communicate with the master for I/O**

↓

**Point-to-point communications: 2N messages**

☹

**Application to a combustion chamber (70M cells)**



SGI Altix platform (GENCI - CINES)

# Communication strategy (AVBP)

**MPI non-blocking** communications (AVBP)

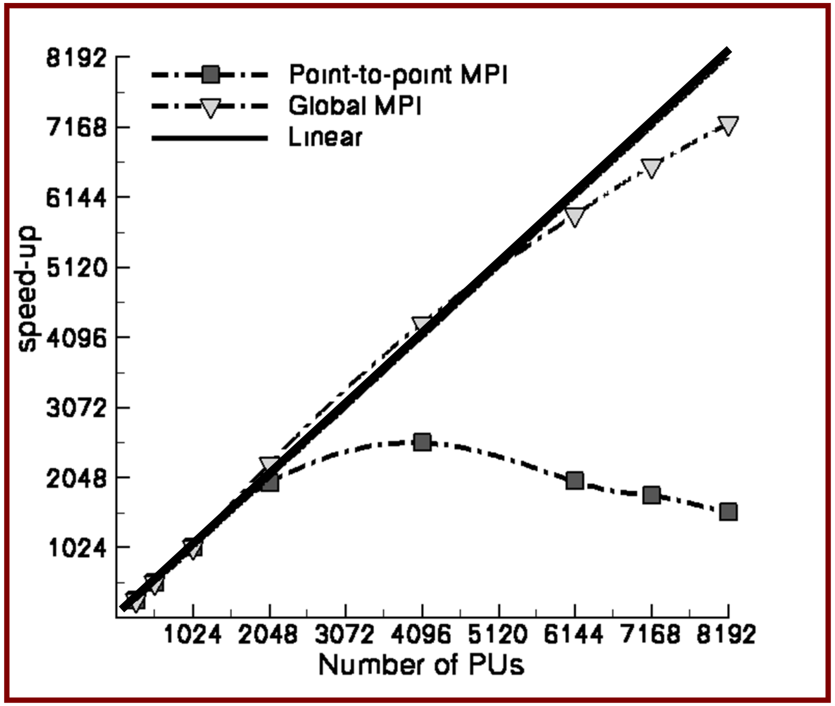
All input/output are dedicated to one processor (the master process)

**Slaves have to communicate with the master for I/O**

**Global communications: 2.log(N) messages**

☺

**Application to a combustion chamber (70M cells)**

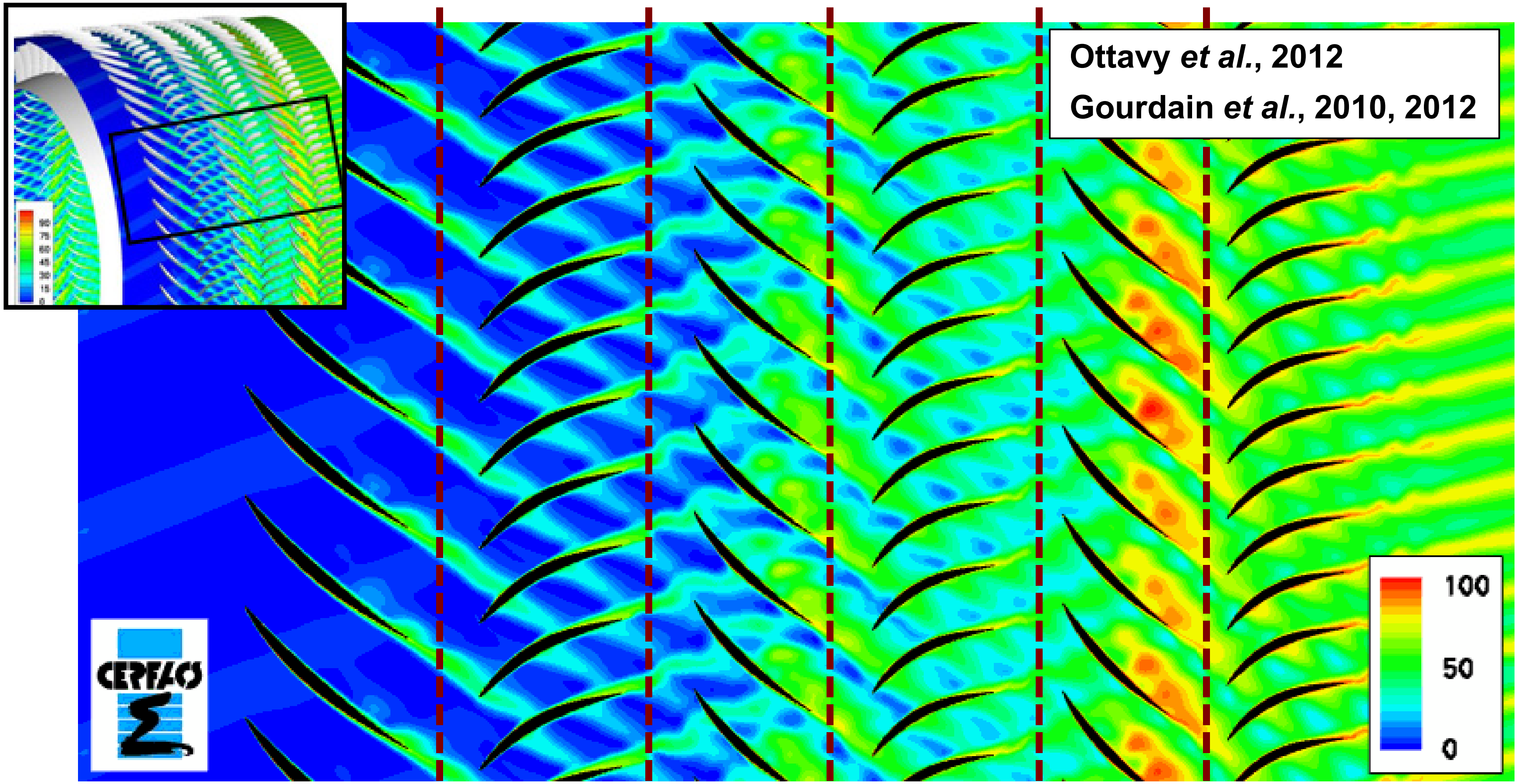


SGI Altix platform (GENCI - CINES)

# Another example…

# Communication strategies

## Communication strategy (*elsA*)



Ottavy *et al.*, 2012

Gourdain *et al.*, 2010, 2012

Entropy flow field at h/H=80%, nominal operating point

**Mesh: 134M cells, nominal operating conditions**

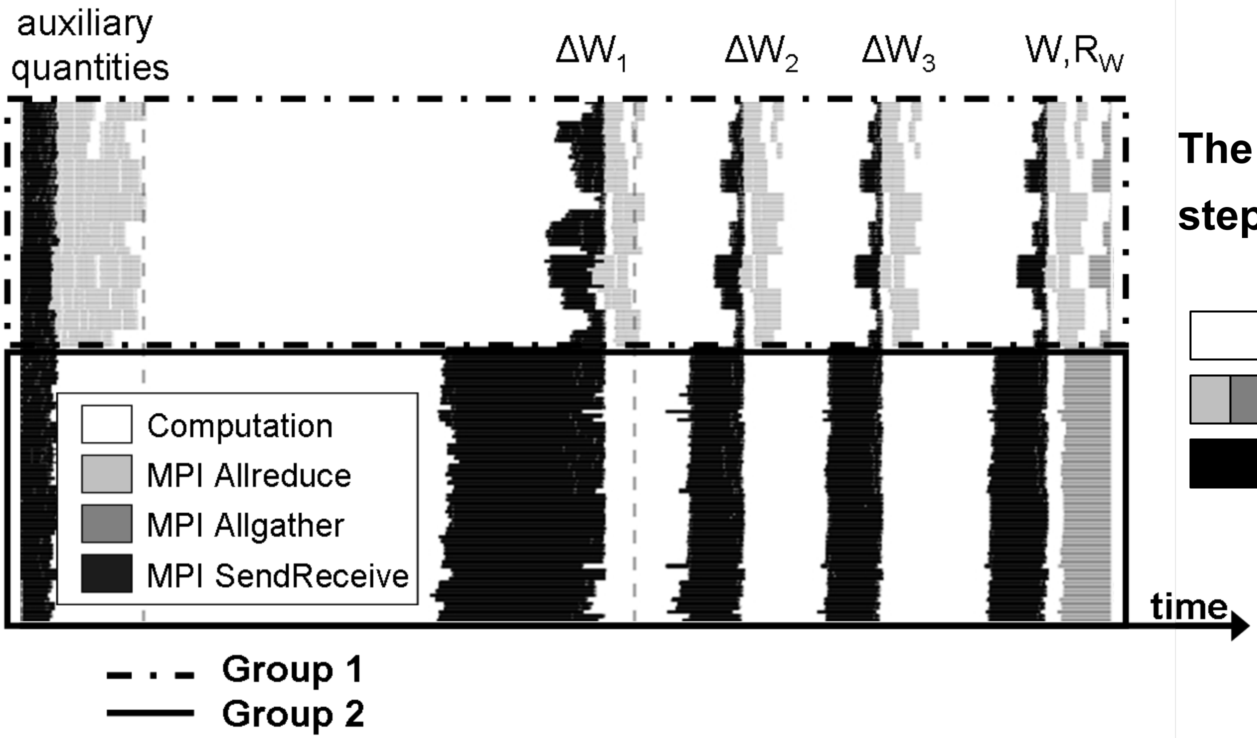
**24 days, 512 comp. cores (CERFACS IBM BG /L)**

# Communication strategies

## Communication strategy (*elsA*)

**The scheduling is done in two stages:**

- first, point-to-point communications for coincident interfaces (matching interfaces)
- then collective communications (non-matching sliding meshes)



**The graph represents one time step of the flow solver**

☐ **Computation**

▨ **Point-to-point calls**

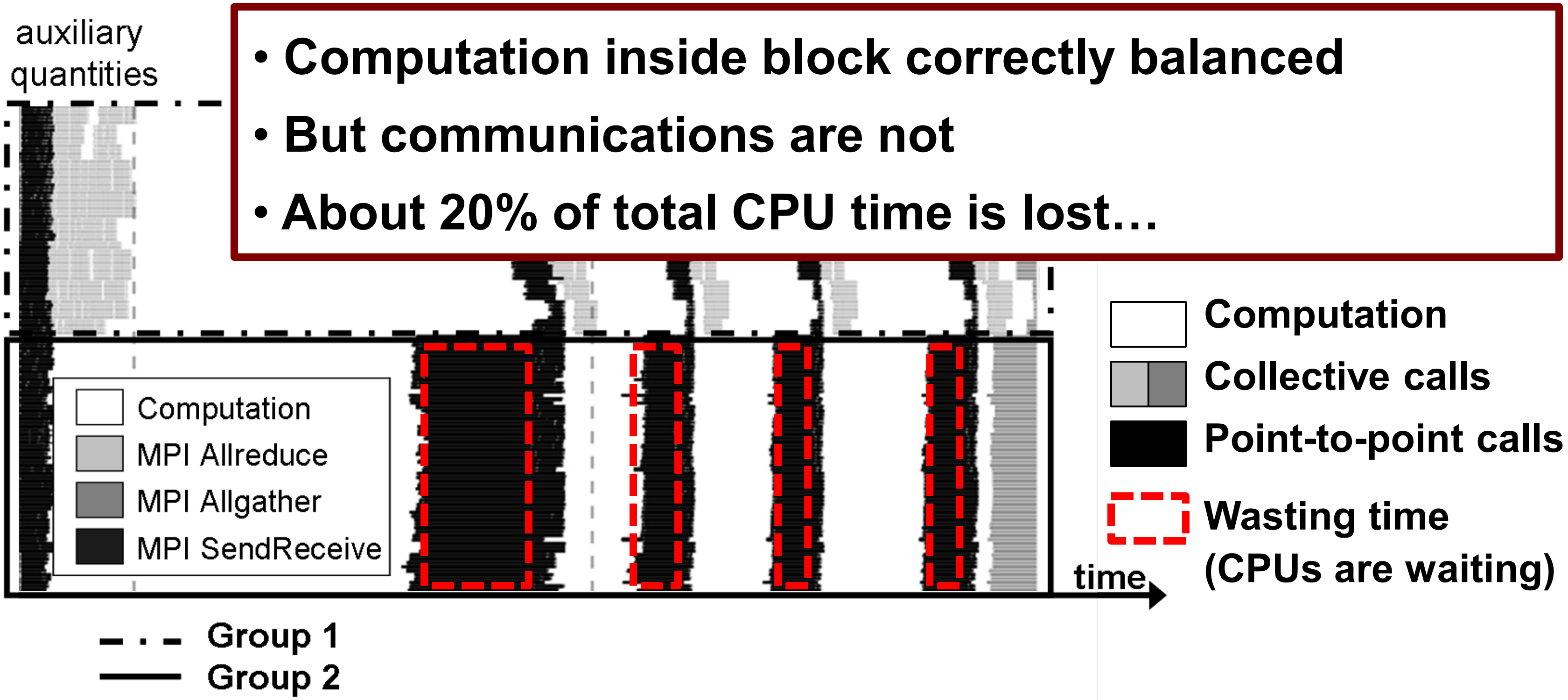
■ **Collective calls**

## Communication strategy (*elsA*)

**The scheduling is done in two stages:**

- first, point-to-point communications for coincident interfaces (matching interfaces)
- then collective communications (non-matching sliding meshes)

- **Computation inside block correctly balanced**
- **But communications are not**
- **About 20% of total CPU time is lost…**



# Communication strategies

- **It is mandatory to reduce as much as possible the size and the number of exchanged messages to achieve good scalability on massively parallel computers**

- **We didn't talk about it but the numerical method can add serious constraints (*e.g.* large stencil for high-order schemes)**
  - **Compact schemes (Spectral Volume/Difference, Discontinous Galerkin, etc.) are good candidates to combine HPC and high-order approaches**

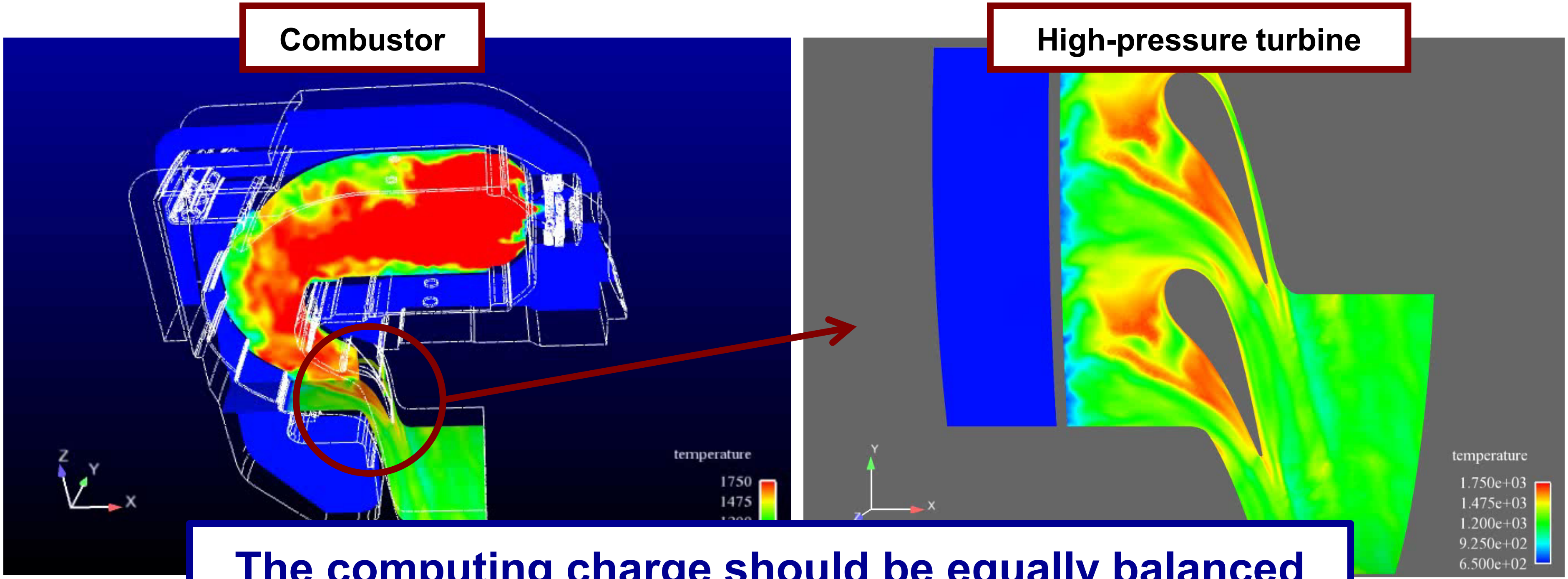
- **Load-balancing and mesh splitting will be deeply discussed during the hands-on with *elsA* this afternoon**

# Let’s start with some HPC bottlenecks…

# Some bottlenecks

## Multi-physics in a real gas turbine

**Code coupling with a code coupling tool (Open-PALM): AVBP (fluid dynamics), AVTP (heat transfer) and PRISSMA (radiation)**



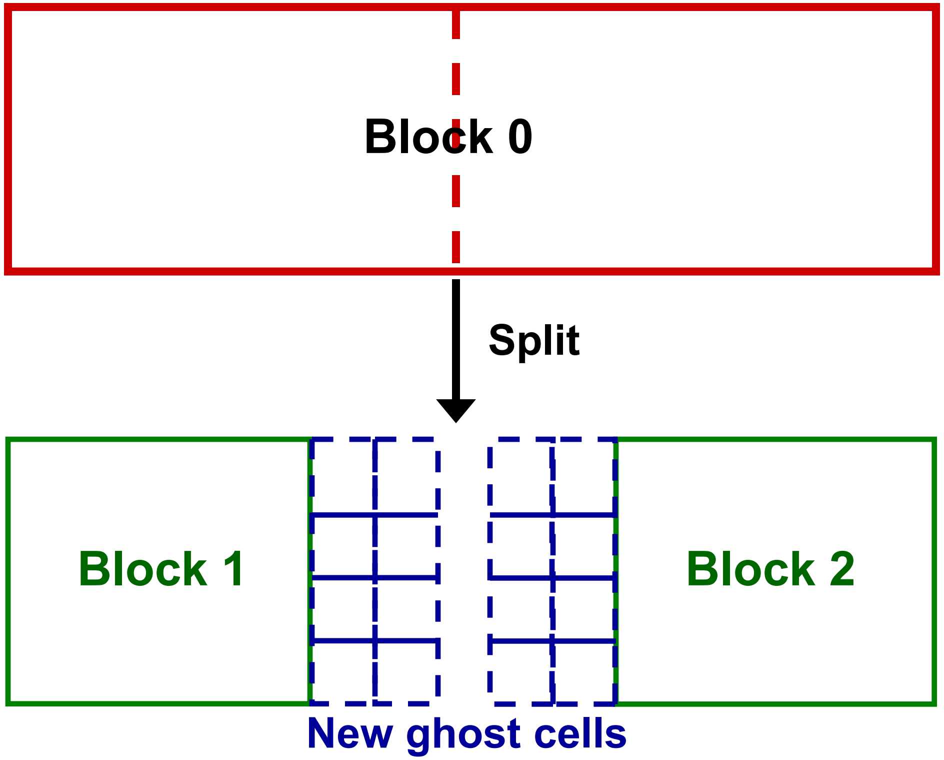
**The computing charge should be equally balanced between the three codes and the coupling tool**

*Amaya et al., 2010; PhD Collado, 2012*

## Mesh partitioning for structured mesh: impact of ghost cells

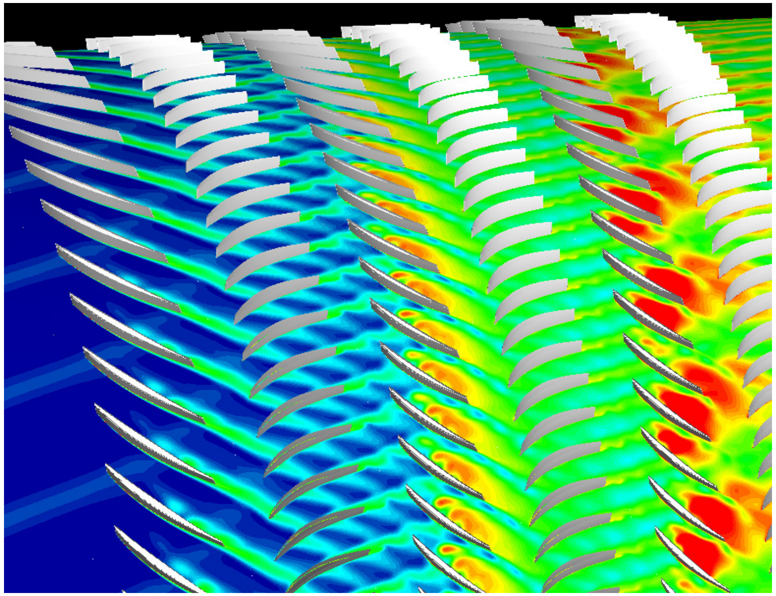
Ghost cells are used to compute fluxes at block interfaces

- ➢ block splitting adds additional ghost cells



## Some bottlenecks

### Mesh partitioning for structured mesh: impact of ghost cells

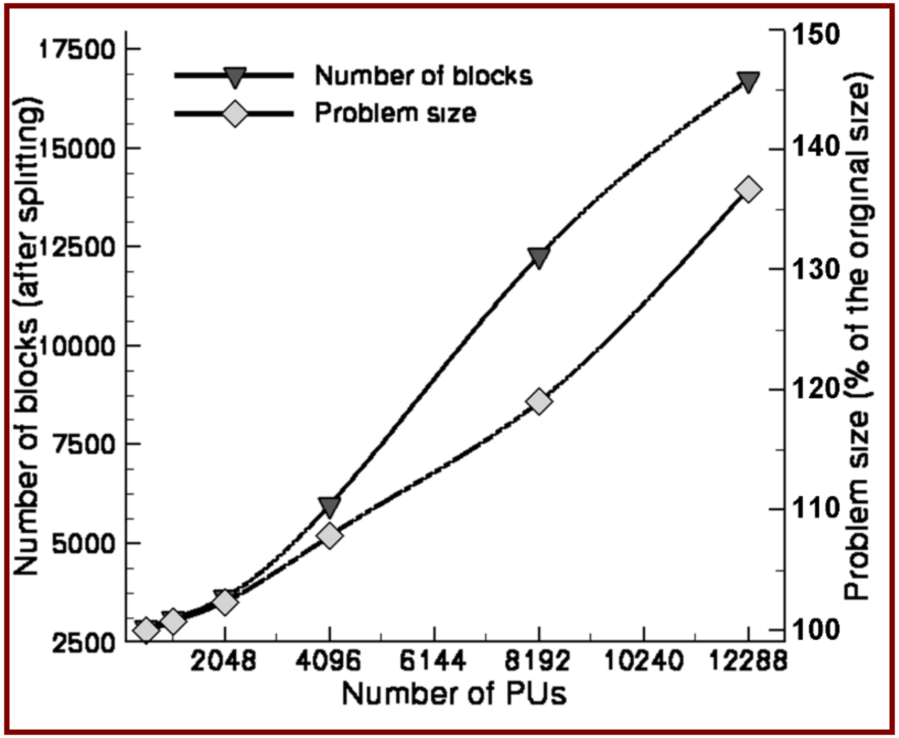


**Initial problem: 2,048 blocks**

**With 12,288 CPUs:**

- **Number of blocks x8**
- **Size of the conf. +45%**

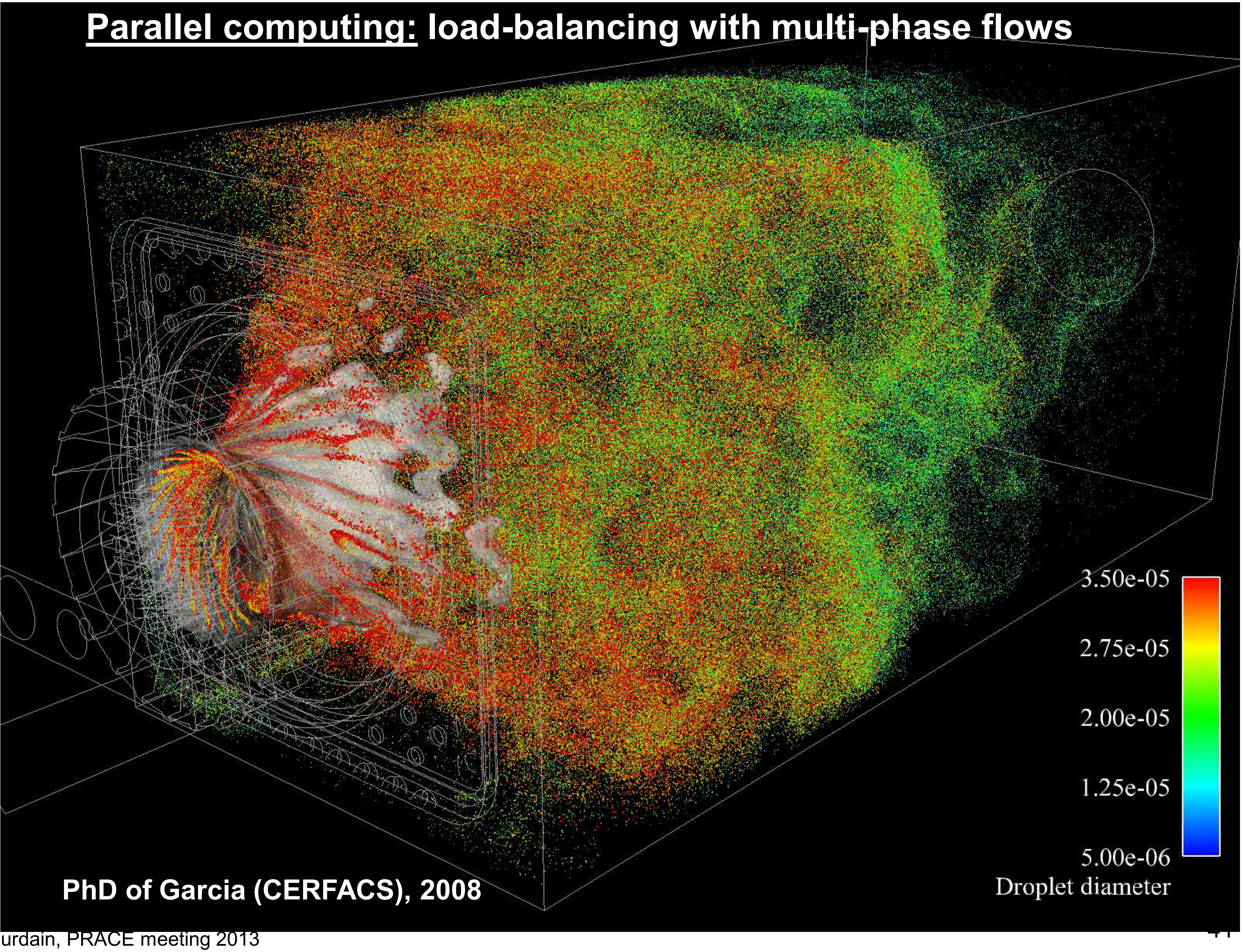
**Simulation with *elsA***



N. Gourdain, PRACE meeting 2013

# Some bottlenecks

## <u>Parallel computing:</u> load-balancing with multi-phase flows



PhD of Garcia (CERFACS), 2008

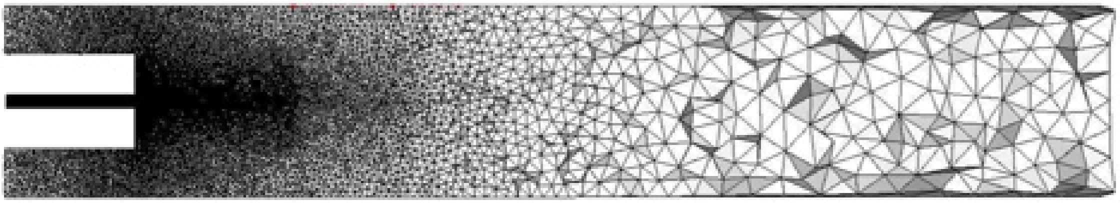
# Some bottlnecks

## Mesh partitioning: the influence of particles

**Two different partitioning algorithms are tested (AVBP):**

- RIB: geometric based algorithms
- METIS: multi-constraint multilevel graph partitioning

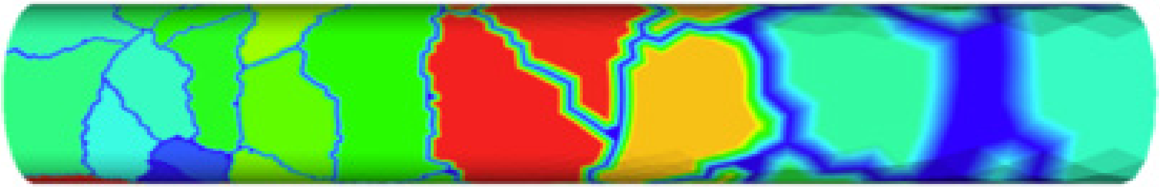
*Two-phase flow bluff body configuration (Garcia, PhD, 2009)*



**RIB**
**(32 cores)**

**One constraint**
**(geometry)**

**METIS**
**(32 cores)**

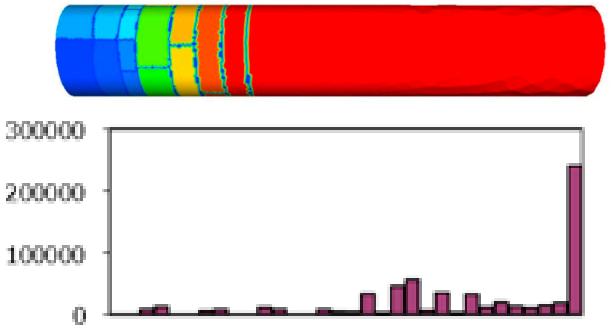


**Two constraints**
**(graph + particles)**

# Some bottlenecks

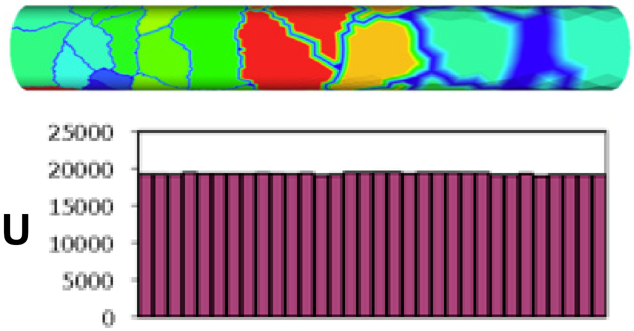
## Mesh partitioning: the influence of particles

**RIB**



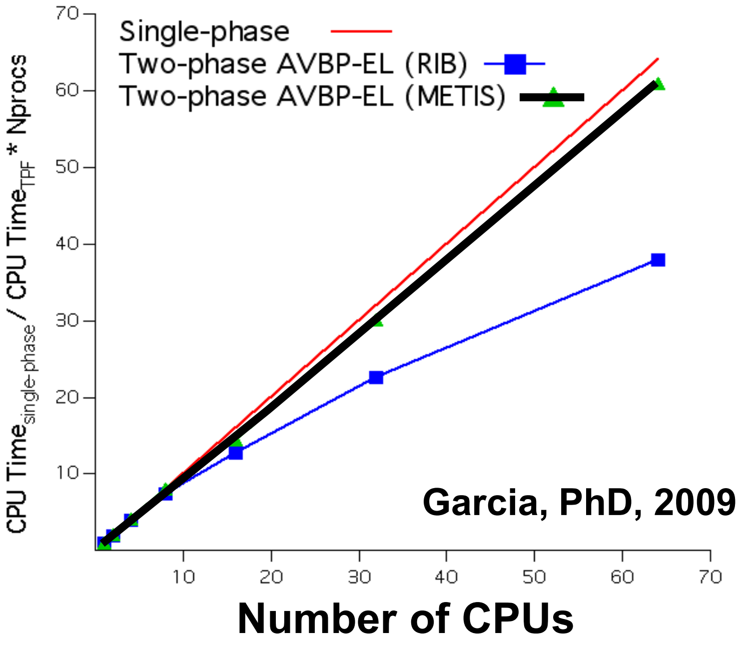
computing cores

**METIS**



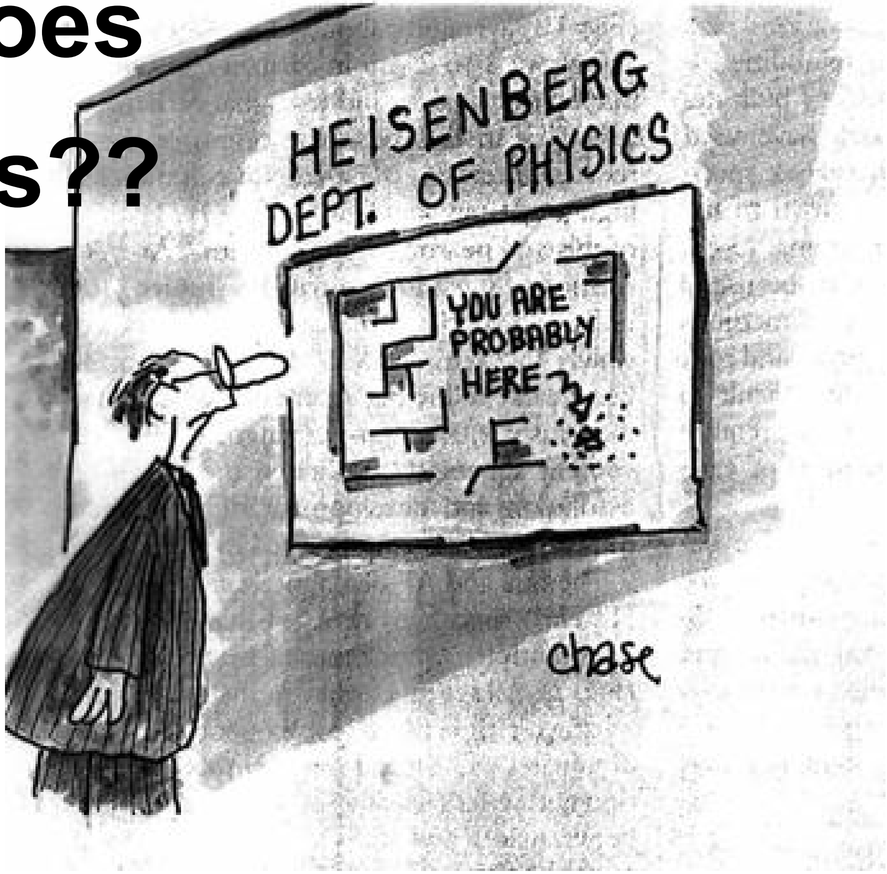
Number of particles / PU

Computing cores



Garcia, PhD, 2009

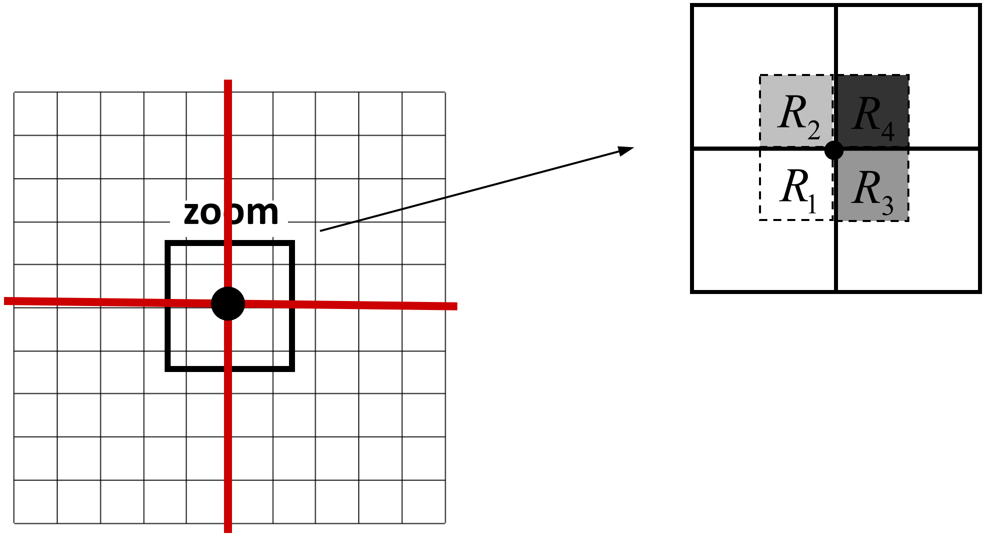
# Ok, parallel computing = application fast computed but does it mean good physics??



## Communication strategy: MPI non-blocking calls

**How to compute a residual at partition interfaces?**

**Like this…**



**4 CPUs**

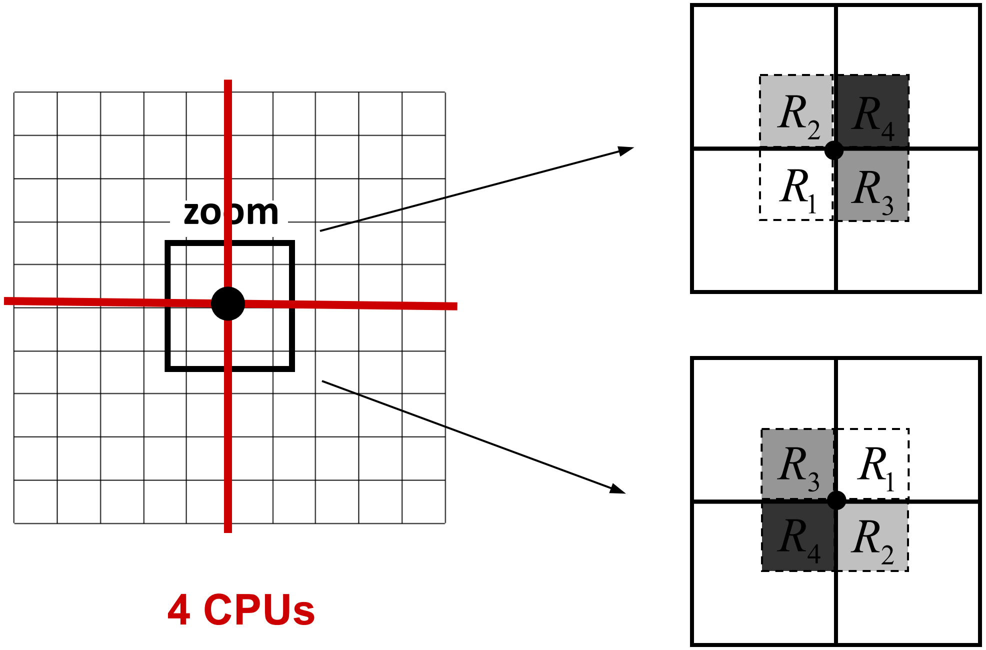
$$\frac{1}{4}\left\{R_4+\left[R_3+\left(R_2+R_1\right)\right]\right\}$$

■ D ▩ C ▨ B ⬚ A

# Quality assessment

## Communication strategy: MPI non-blocking calls

### How to compute a residual at partition interfaces?



**4 CPUs**

$$\frac{1}{4}\left\{R_4+\left[R_3+\left(R_2+R_1\right)\right]\right\}$$

| ■ | ▩ | ▨ | □ |
|---|---|---|---|
| D | C | B | A |

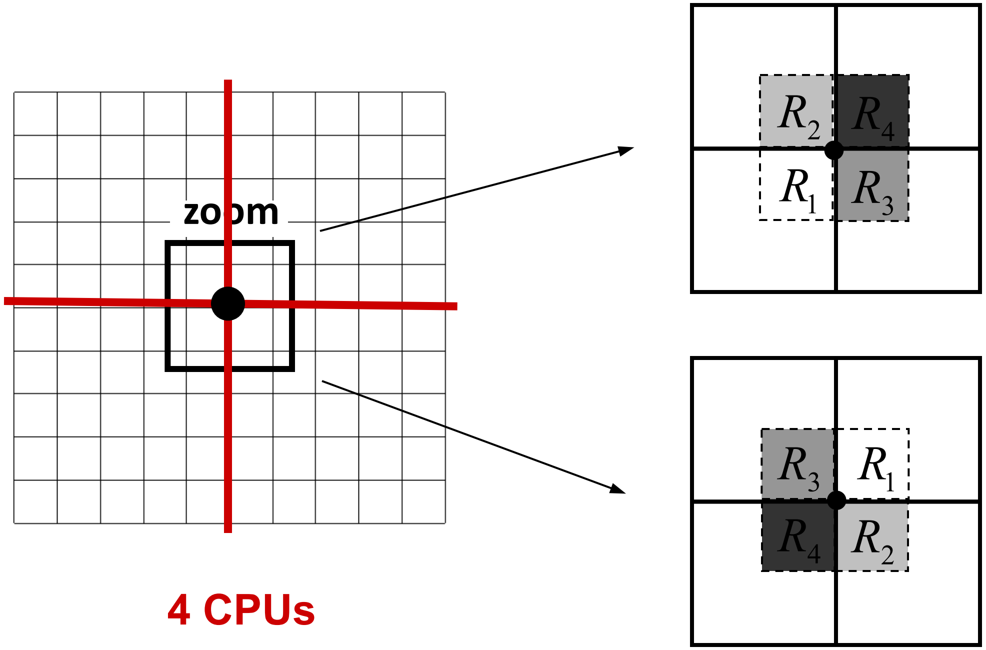
**…or like this…**

$$\frac{1}{4}\left\{R_1+\left[R_2+\left(R_3+R_4\right)\right]\right\}$$

| □ | ▨ | ▩ | ■ |
|---|---|---|---|
| A | B | C | D |

# **Communication strategy:** MPI non-blocking calls

**How to compute a residual at partition interfaces?**



**4 CPUs**

$$\frac{1}{4}\left\{R_4+\left[R_3+\left(R_2+R_1\right)\right]\right\}$$

■ D ▩ C ░ B ⬚ A

$$\neq$$

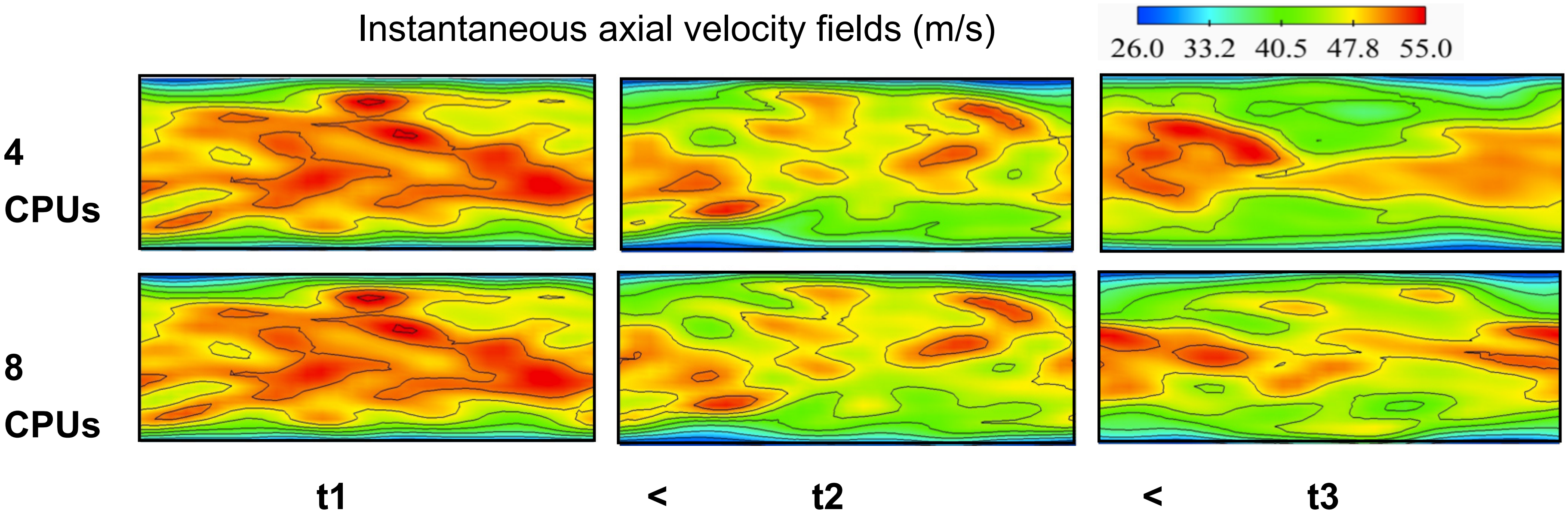
$$\frac{1}{4}\left\{R_1+\left[R_2+\left(R_3+R_4\right)\right]\right\}$$

⬚ A ░ B ▩ C ■ D

**Problem: non-blocking communications induce a non-deterministic behavior**

## Impact of rounding errors on LES

Consequence of the lack of associativity property (Floating point arithmetic) on a turbulent channel (AVBP)

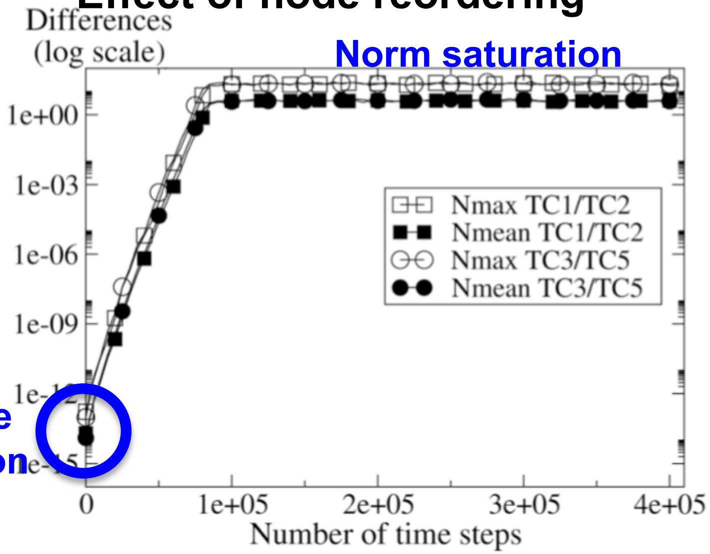


Instantaneous axial velocity fields (m/s)

**Senoner *et al.*, 2008**

## Impact of rounding errors on LES

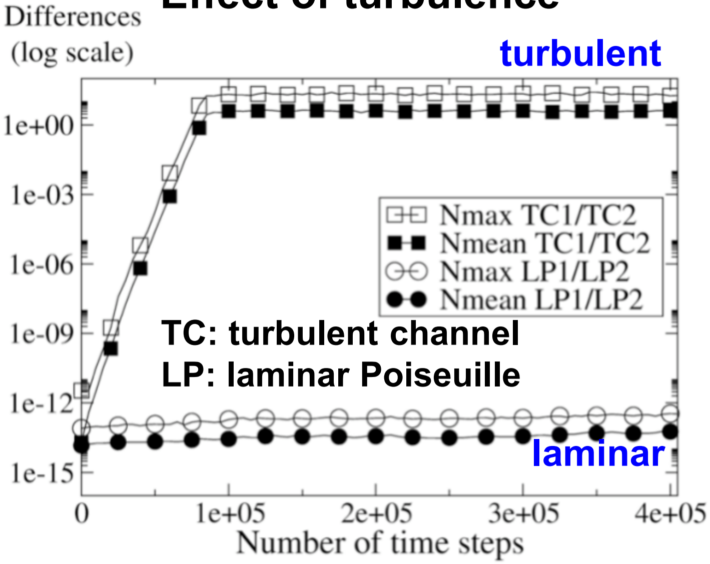
### Effect of node reordering



**Norm saturation**

**Machine precision**

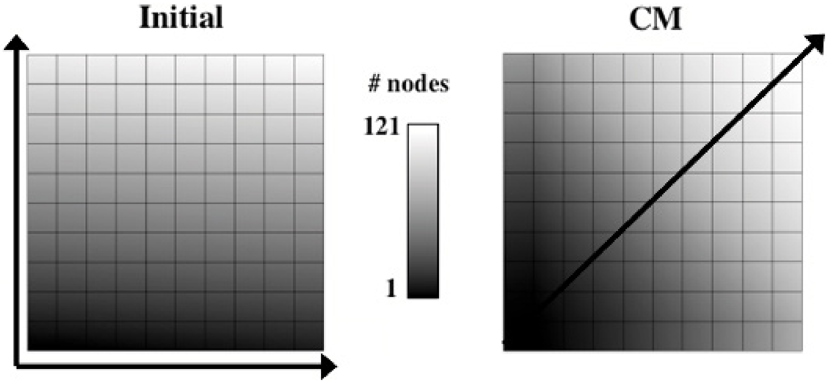
### Effect of turbulence



**turbulent**

**laminar**

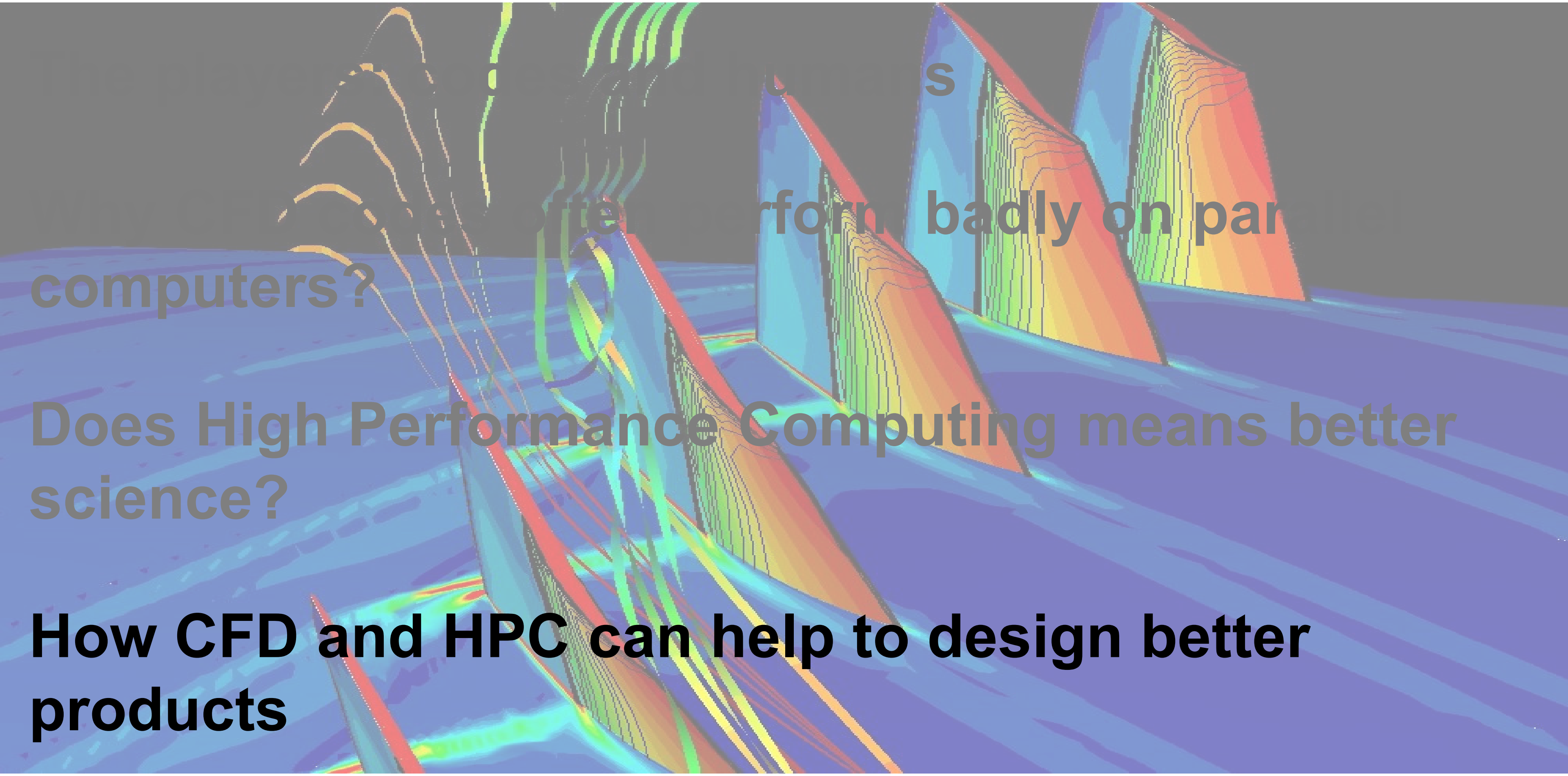
**TC: turbulent channel**
**LP: laminar Poiseuille**



**Parallel comp. induces a chaotic behavior, reflecting the true nature of the turbulent equations**

**Senoner *et al.*, 2008**

# Overview



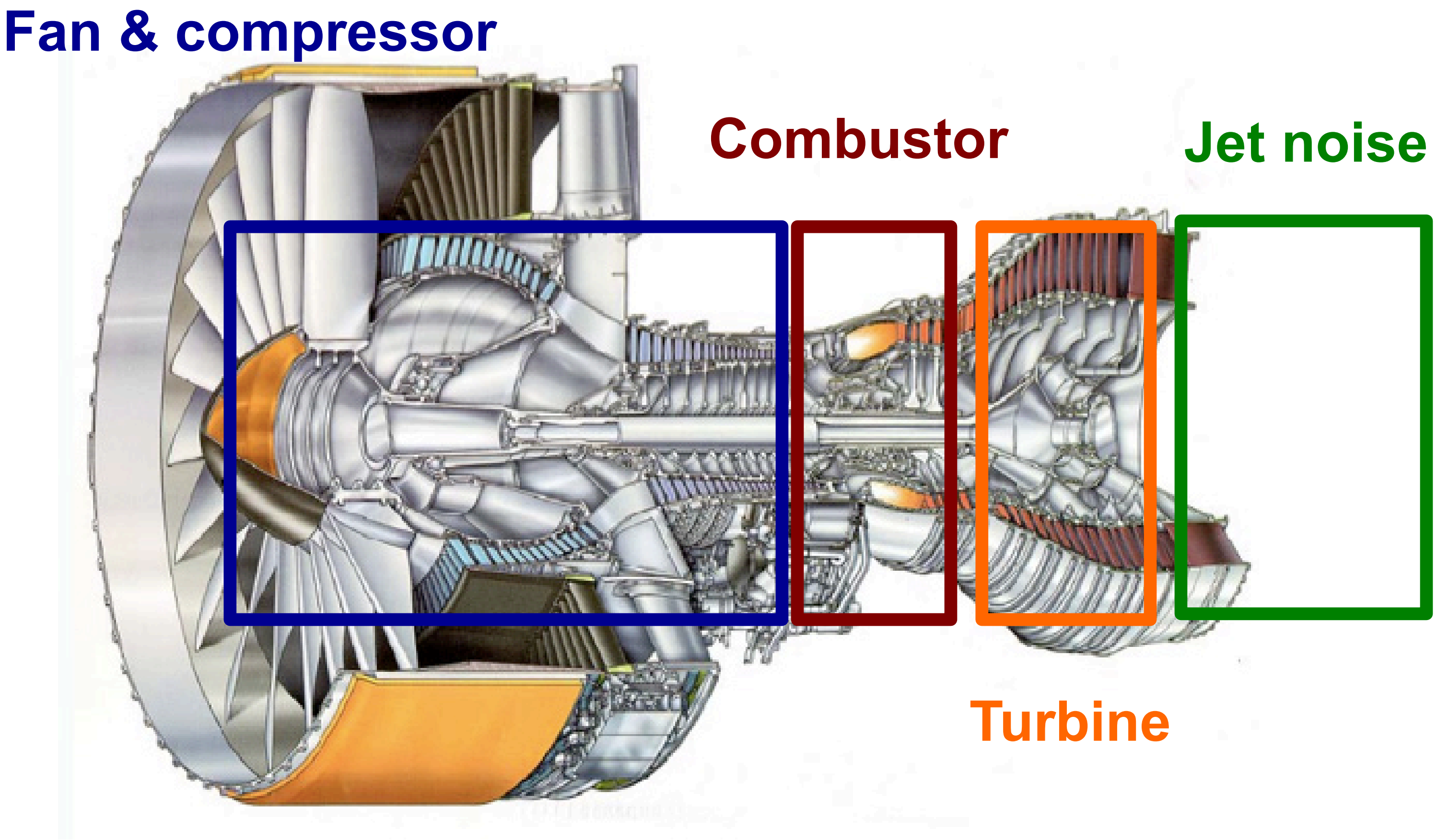
The [illegible] s

Why do CFD codes often perform badly on parallel computers?

Does High Performance Computing means better science?

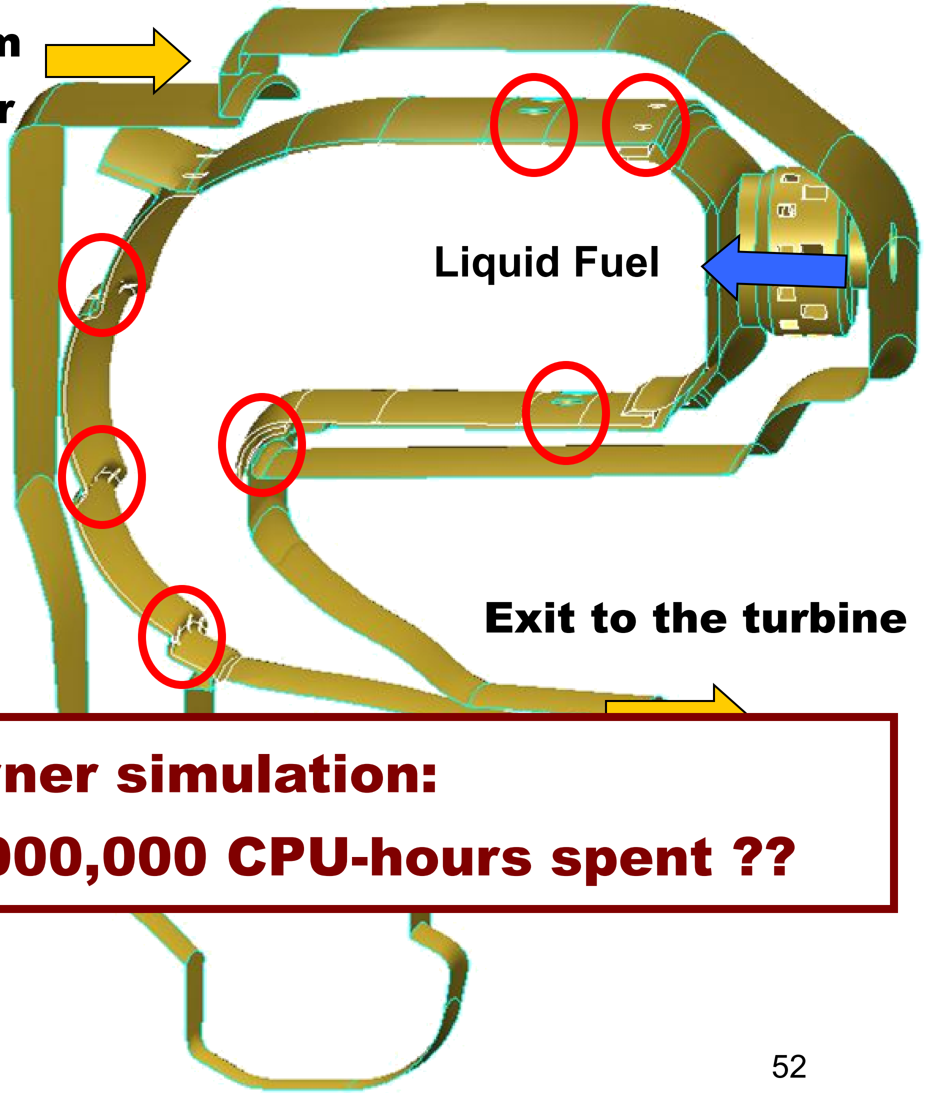
**How CFD and HPC can help to design better products**

# Overview



# Target configuration: a helicopter combustion chamber at cruise conditions

**Annular burner**



**Air coming from the compressor**

**Liquid Fuel**

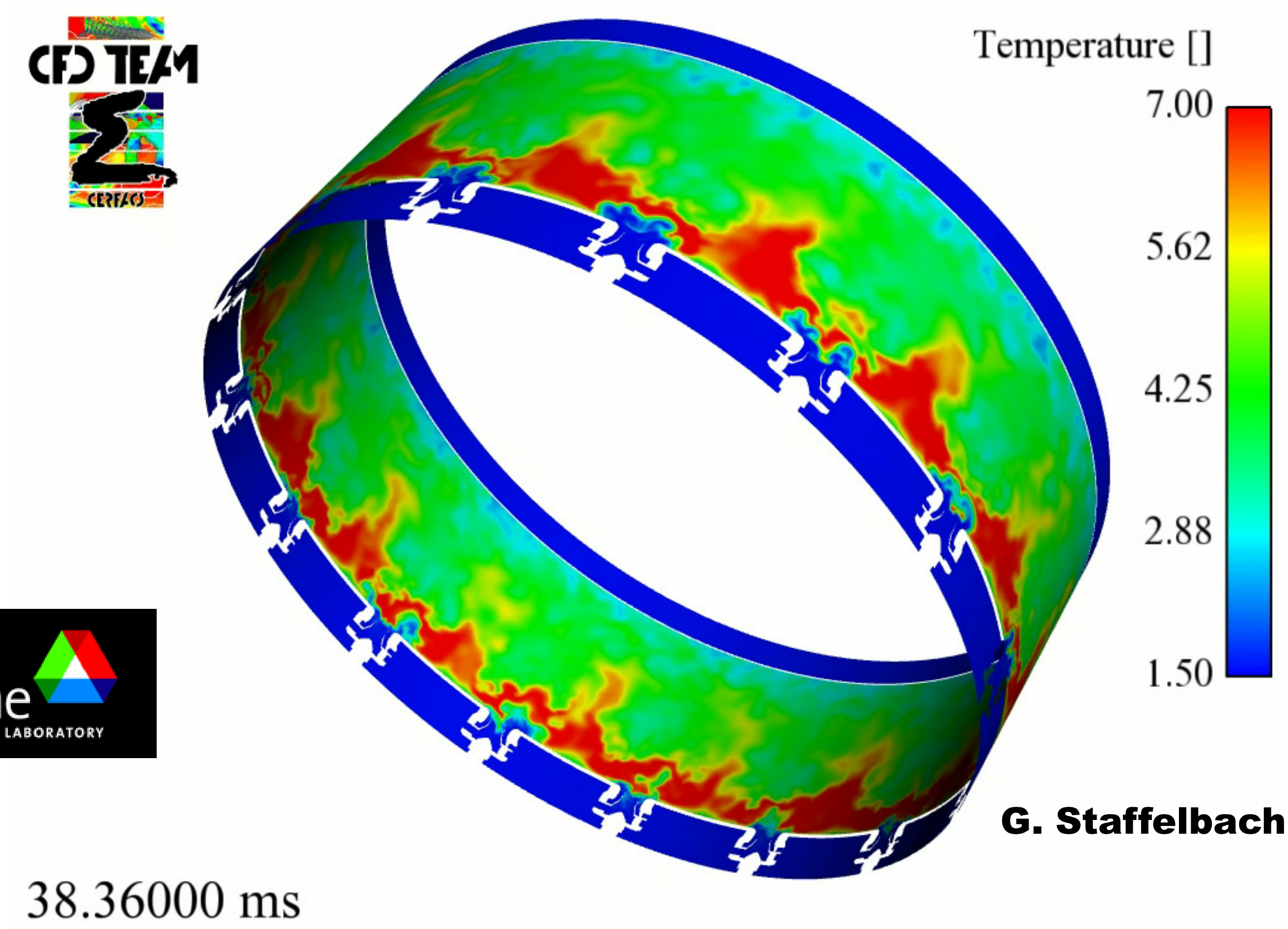
**Exit to the turbine**

**Towards a full burner simulation:**

**what do you get out of the 1,000,000 CPU-hours spent ??**

# Simulation of thermo-acoustic instabilities

## Configuration: 360º burner. Impact on temperature field. Code AVBP.



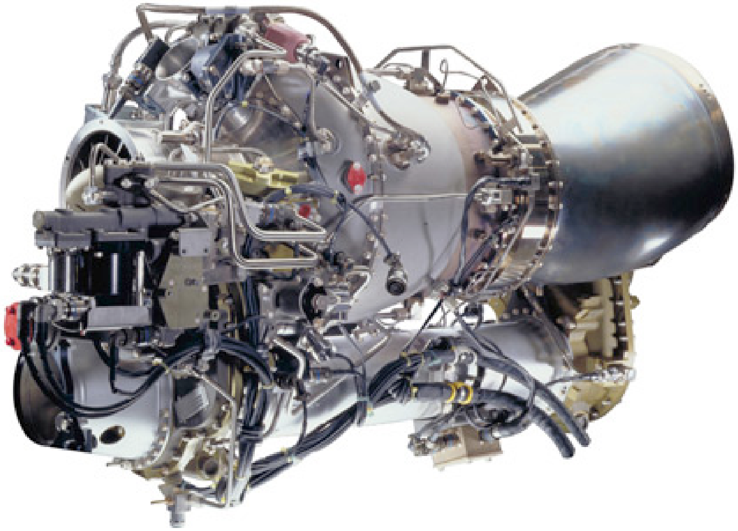
G. Staffelbach *et al.*, 2008

**Unexpected implication of a pressure instability: oscillation of the temperature field (self-sustained thermo-acoustic instability)**

# LES for combustion chambers

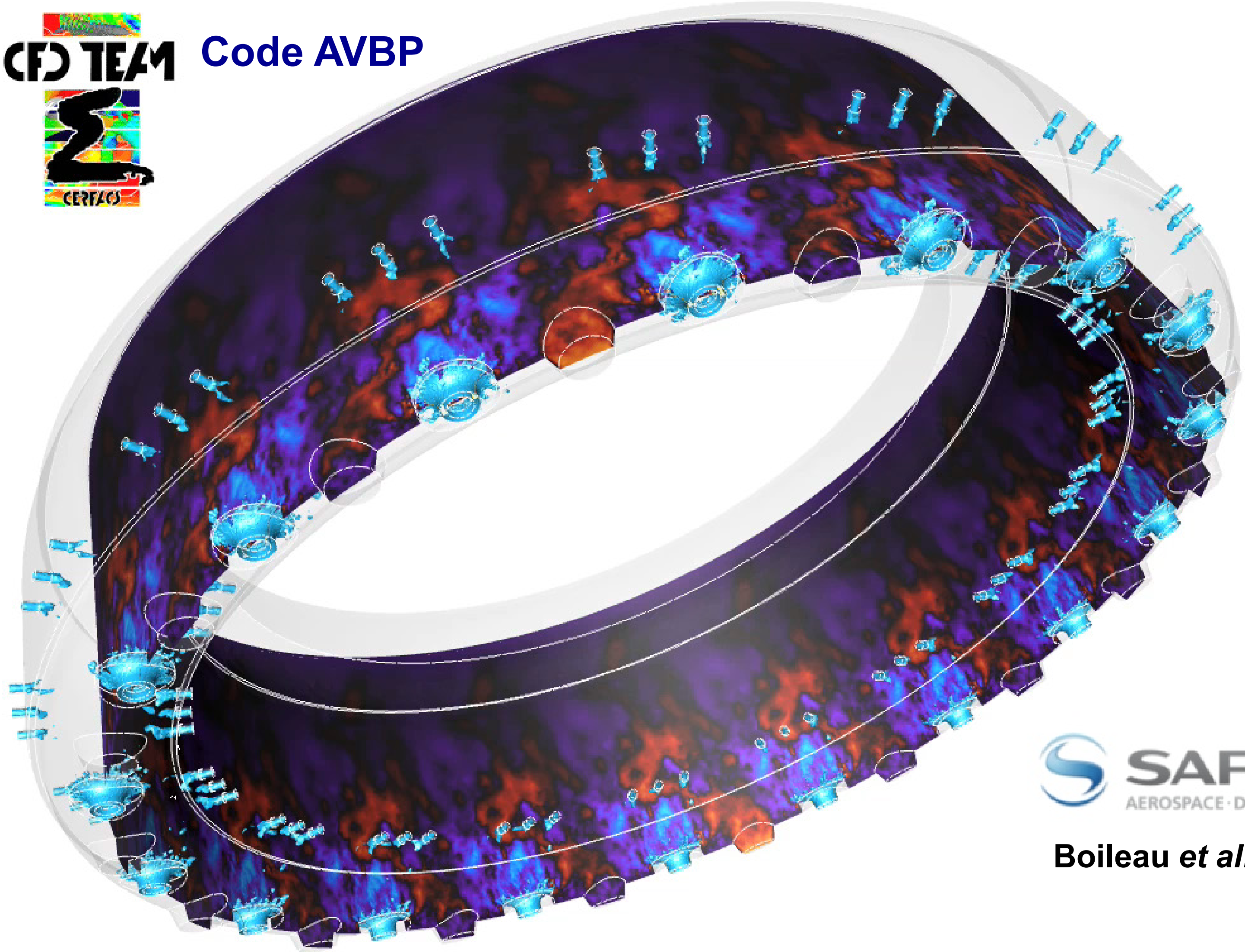
**Sometimes, our industrial partners ask us interesting questions! For instance, Turbomeca asked us:**

**“Can we certify that our engine can be restarted even in extreme conditions (*i.e.* high altitude, low temperature)?”**



# Ignition process in a combustion chamber

**Code AVBP**



Boileau *et al.*, 2008

**Actually, the answer is… maybe!**

# Simulation of compressor instabilities

**Surge is a low-frequency instability encountered in all compression systems**

**The understanding of this phenomenon is mandatory for designers since it imposes inefficient design margins**


COPYRIGHT RUBEN HOFS
AIRLINERS.NET


JETPHOTOS.NET
Image Copyright © Will Mallinson


David Cenciotti's weblog

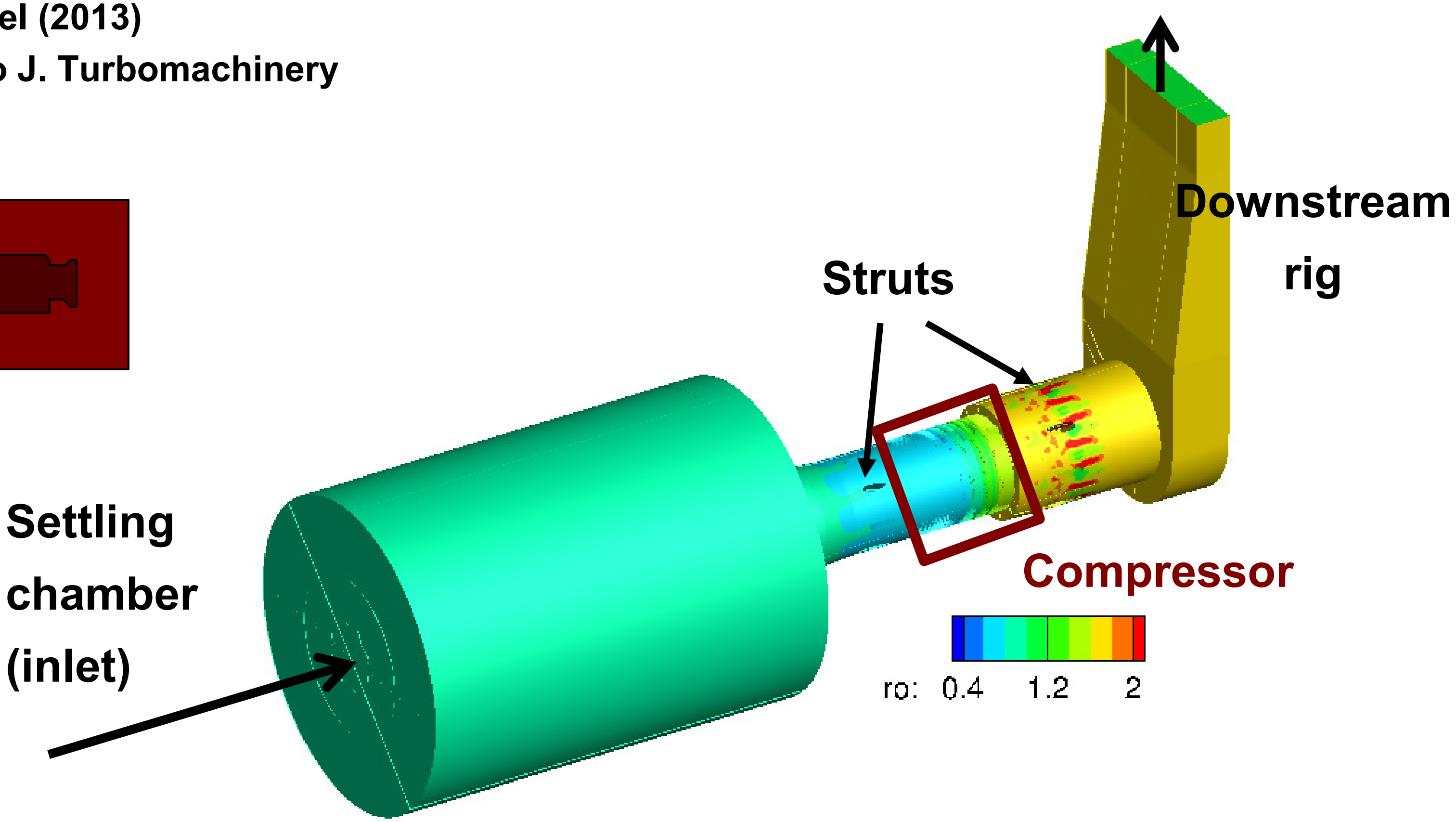
# Simulation of compressor instabilities

## Surge cycle simulation. Code *elsA*.

**Complex geometry, including all the parts of the experimental rig**

**1,024 cores (1 000 000h CPU), 30MWh (100s of a nuclear power plant)**

**PhD of Crevel (2013)**
**submitted to J. Turbomachinery**

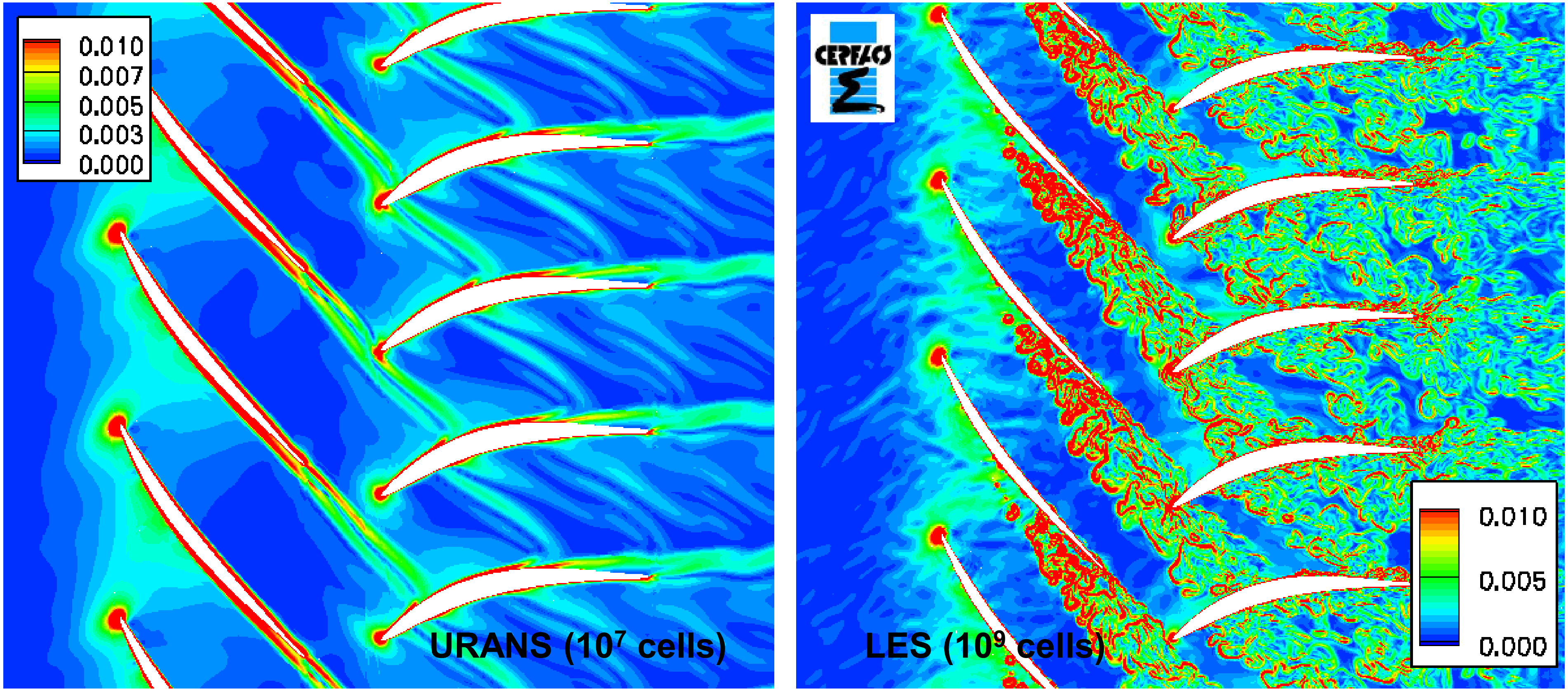


# LES for compressors

## LES in a compressor stage. Code *elsA*.

**Geometry designed by Snecma (Reynolds ~ 7x$10^5$)**

**1,024 cores (1 000 000h CPU), 30MWh (100s of a nuclear power plant)**

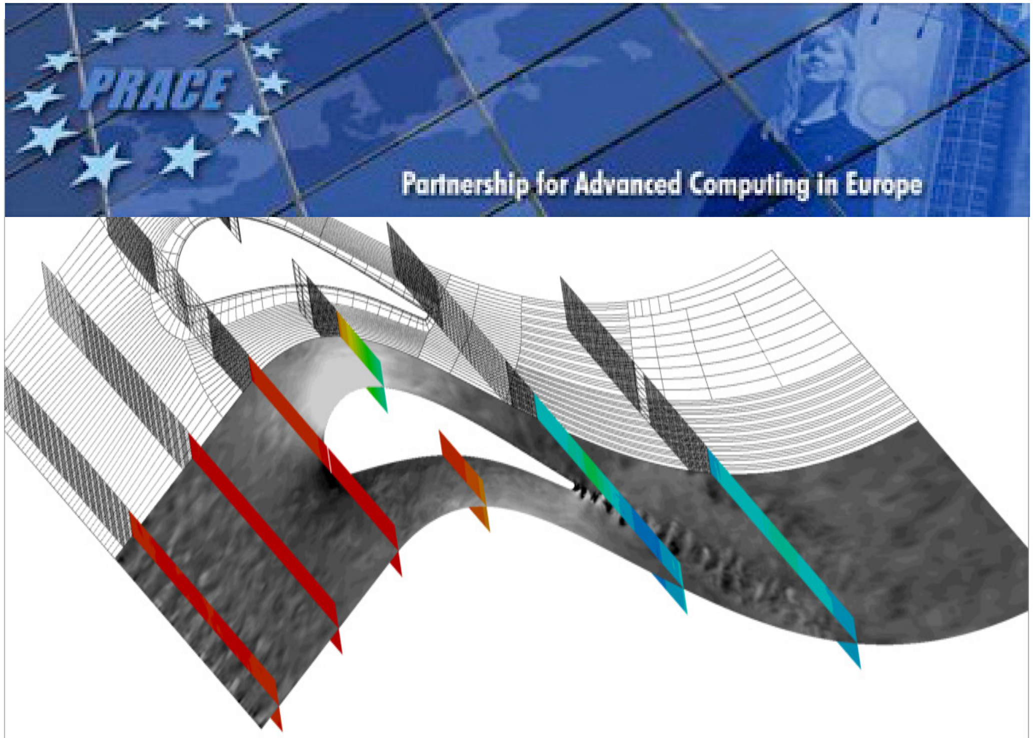


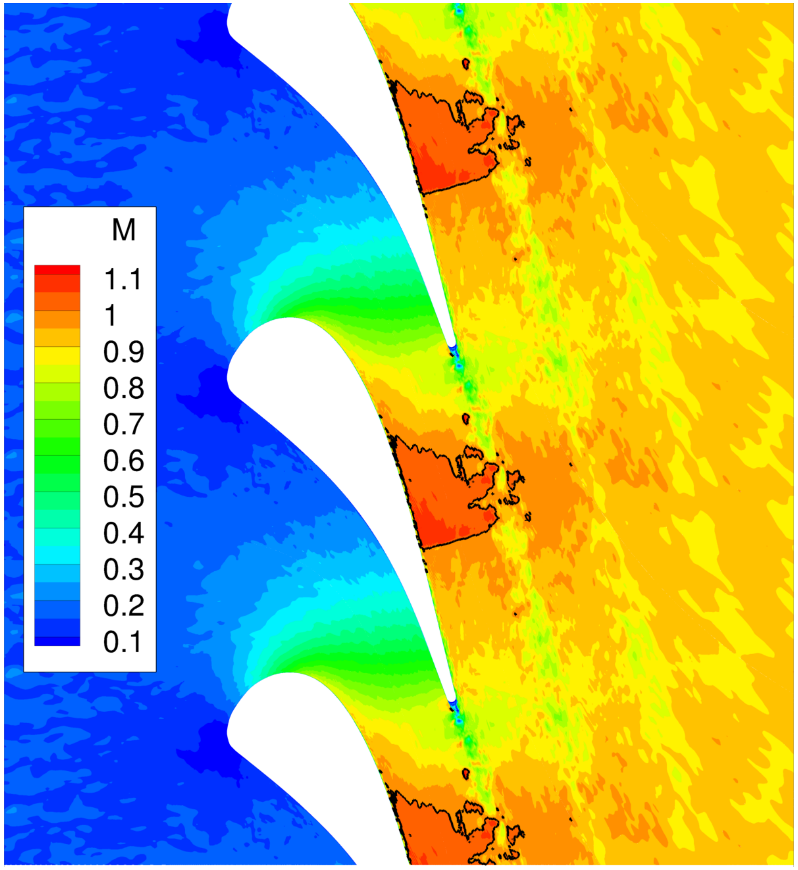
***Density gradient (tip region)***

**Gourdain, ASME Turbo Expo, 2013**

# DNS for turbines

**DENSITIES: run one of the first wall-resolved DNS of a compressible flow in a high Reynolds number turbine (Re~$10^6$). Code *elsA***





**Leader: CERFACS**

**Partners: VKI, U. Naples**

**Grid: 5.8 billions cells, 8,192 blocks on 4,096 CPUs**

**Estimated cost: 15M CPU h (1500s of a nuclear power plant)**

N. Gourdain, PRACE meeting 2013

# LES of jet noise and control

LES of an isothermal jet (M=0.9, Re=400 000). Code *elsA*.

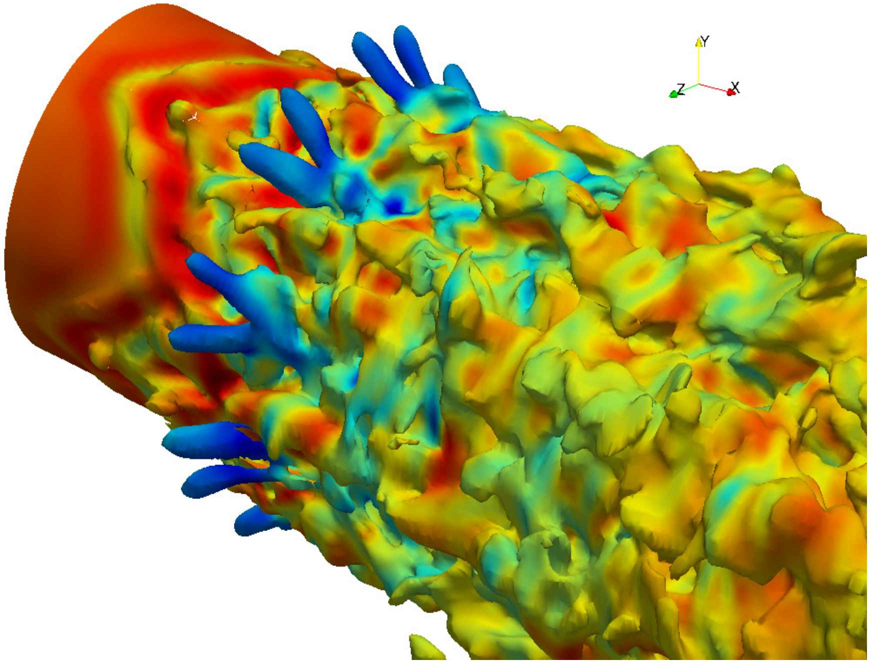
<u>Two approaches</u>:

1- Source term to model microjets effects (Shur *et al.*, JSV, 2011)

Grid ~ 24M cells, 64 CPU cores

2- Full computation with all micro jets meshed.

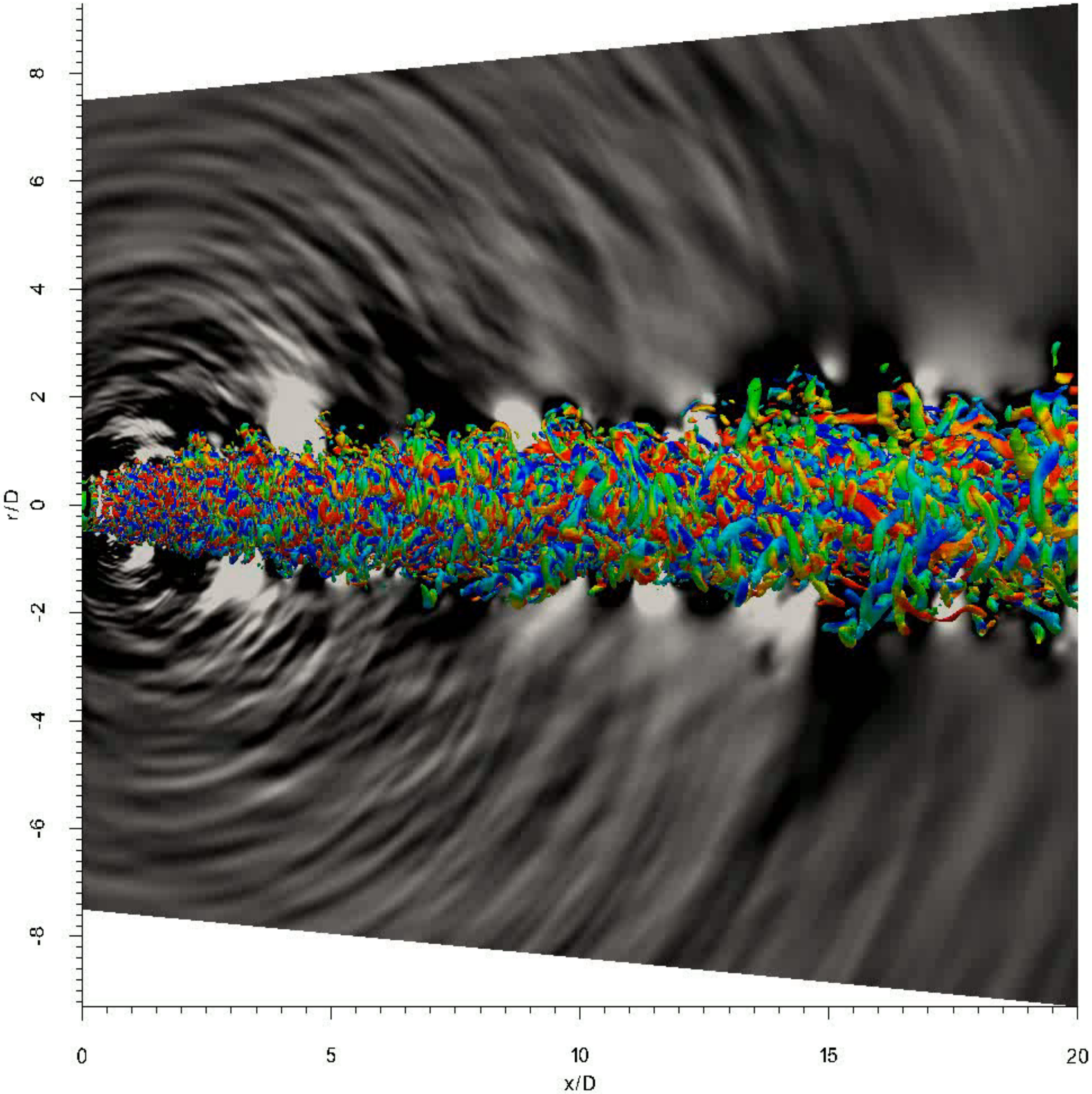
Grid ~ 2,000M cells, PRACE project on 8192 cores, 25M CPU hours (in progress)



**Numerical method**

- Spatial scheme : Compact 6th order
- Temporal scheme : RK6 3rd order

# LES of jet noise and control

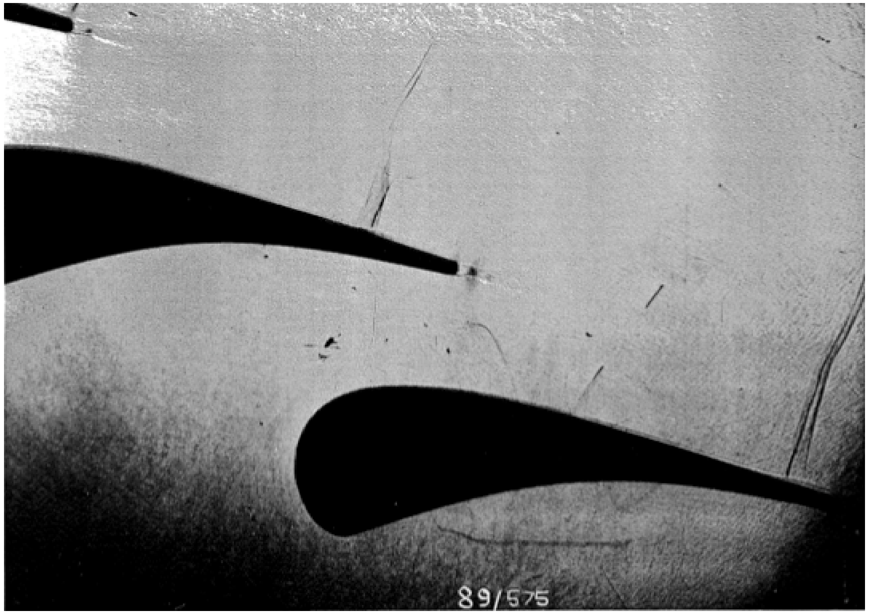


**Deniau *et al.*, 2012**

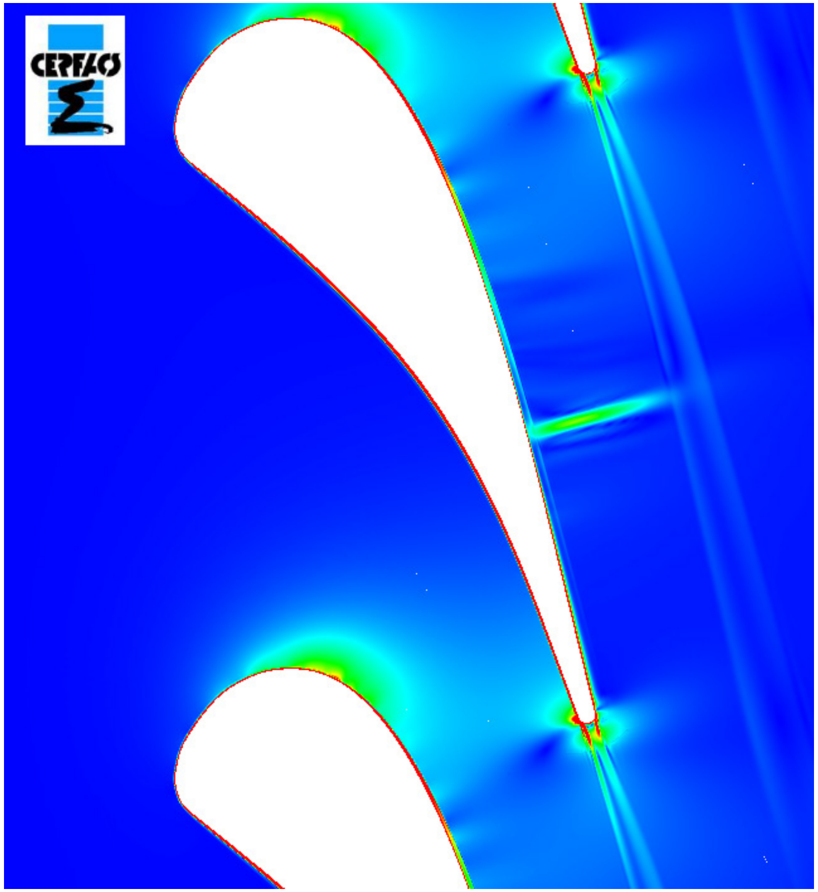
**Daviller *et al.*, 2013**

# Quantification of uncertainty

## How handle a solution that depends on parameters you don't know?



Arts *et al.*, J. Turbomachinery, 1992

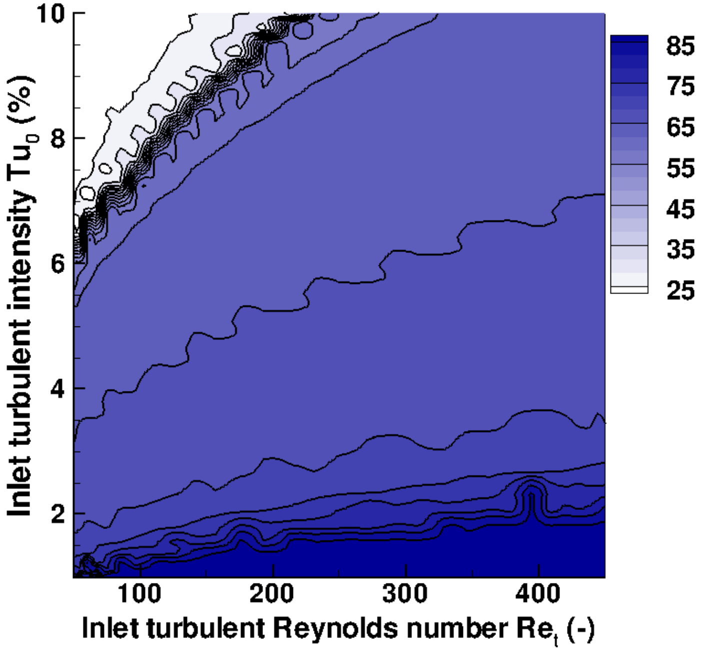


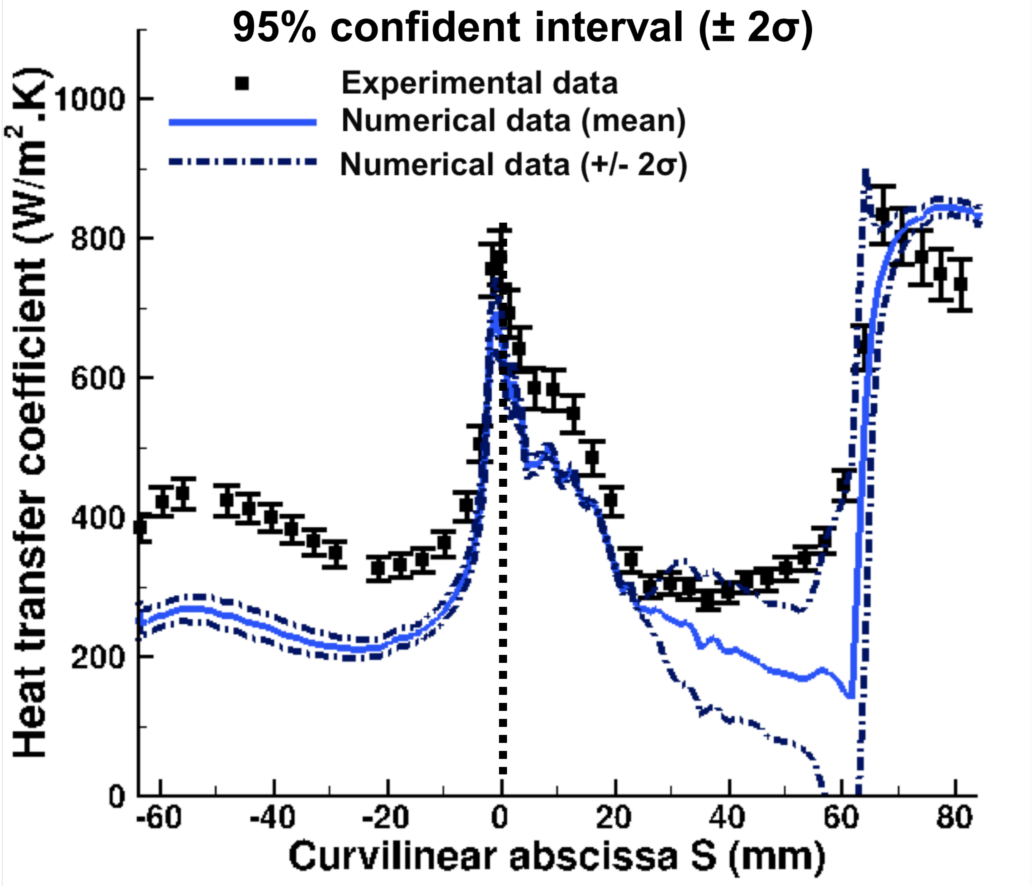
Gourdain and Dewandel, ASME, 2012

- Code *elsA*. RANS simulations
- Highly scalable task (sampling are independents)
- 2 unknown parameters: 33x33 sampling (~1000 computations)

# Quantification of uncertainty

## How handle a solution that depends on parameters you don’t know?




95% confident interval (± 2σ)

**The simulation successfully predicts the dependency of the solution to inlet parameters on the suction side but it fails on the pressure side (LES required!)**

# Conclusion

Computers are evolving very rapidly, so flow solvers need to be constantly adapted

There is currently a strong competition between the great world economic powers (USA, China, Europe) to be the first to get over the step of Exascale computing ($10^{18}$ Flops) > **decisive economic asset**

Exascale will be the reality in a couple of years but well trained scientists/engineers are still required to get the best of such computers and address societal goals

# Still one difficulty to discuss, the post-processing step…

## Problematic

- Run very fast my o($10^9$) cells problem on o($10^4$) CPUs is a interesting but it is not sufficient to make my simulation a success
  - ➢ it should also be correctly analyzed

**Some magnitude orders:**

- Usually 1M cells grid = 1Go of data/field
- Unsteady solution (LES): a priori, each field should be stored ($10^4$ x 1Go/1M cells)

## Typical (current) post-processing

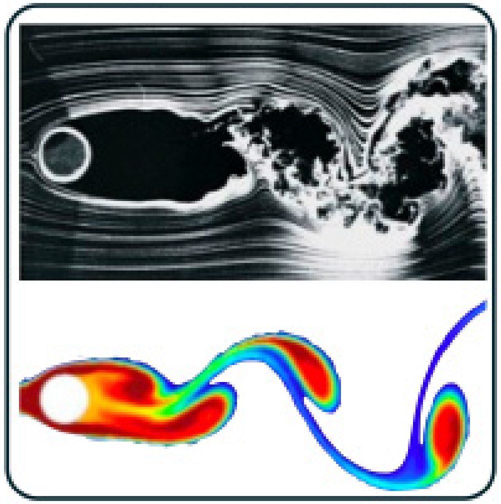
**Raw data**

```
['iteration', 'convflux_ro']
3.9000000E+04   2.0865551E+01
3.9040000E+04   2.0973377E+01
3.9080000E+04   2.1007036E+01
3.9120000E+04   2.0946099E+01
3.9160000E+04   2.0888732E+01
3.9200000E+04   2.0850831E+01
3.9240000E+04   2.0826771E+01
3.9280000E+04   2.0824366E+01
3.9320000E+04   2.0882378E+01
3.9360000E+04   2.0926301E+01
3.9400000E+04   2.0929736E+01
3.9440000E+04   2.0899944E+01
3.9480000E+04   2.0854468E+01
3.9520000E+04   2.0783363E+01
```

**My post-processing**

**No access to raw data. Limited functions**

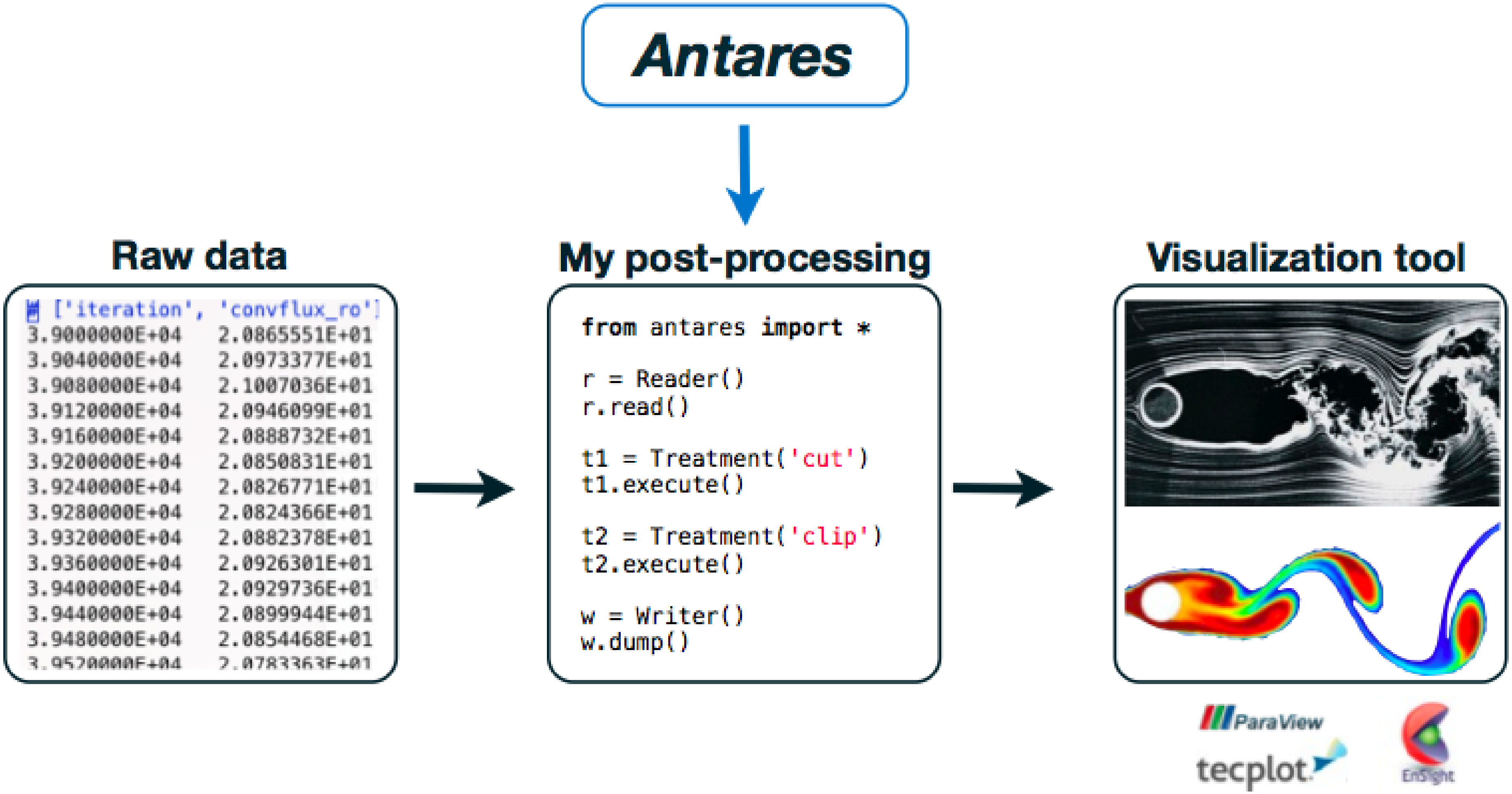
**Visualization tool**



## New way to address post-processing

**ANTARES: A Numerical Toolbox for the Analysis of RESults**

**(developed by A. Gomar & T. Leonard)**



**Antares**

**Raw data**

```
# ['iteration', 'convflux_ro']
3.9000000E+04    2.0865551E+01
3.9040000E+04    2.0973377E+01
3.9080000E+04    2.1007036E+01
3.9120000E+04    2.0946099E+01
3.9160000E+04    2.0888732E+01
3.9200000E+04    2.0850831E+01
3.9240000E+04    2.0826771E+01
3.9280000E+04    2.0824366E+01
3.9320000E+04    2.0882378E+01
3.9360000E+04    2.0926301E+01
3.9400000E+04    2.0929736E+01
3.9440000E+04    2.0899944E+01
3.9480000E+04    2.0854468E+01
3.9520000E+04    2.0783363E+01
```

**My post-processing**

```python
from antares import *

r = Reader()
r.read()

t1 = Treatment('cut')
t1.execute()

t2 = Treatment('clip')
t2.execute()

w = Writer()
w.dump()
```

**Visualization tool**

## New way to address post-processing

**Typical functions required for a “modern” post-processing, compliant with HPC:**

- **Multi-processing, adapted to multi-core node (OpenMP)**
- **Universal reader/writer (CGNS/HDF)**
- **Online post-processing:**
  - ➢ **No Terabytes to store**
  - ➢ **Direct read of data during the computation (direct access to code data)**
  - ➢ **Direct store in RAM memory (don’t use intermediate disk storage)**

**Thanks to the CERFACS teams, our partners, PRACE**
**and the University of Ljubljana (special thanks to L. Kos and M. Maffi)**

**Thanks for your attention**