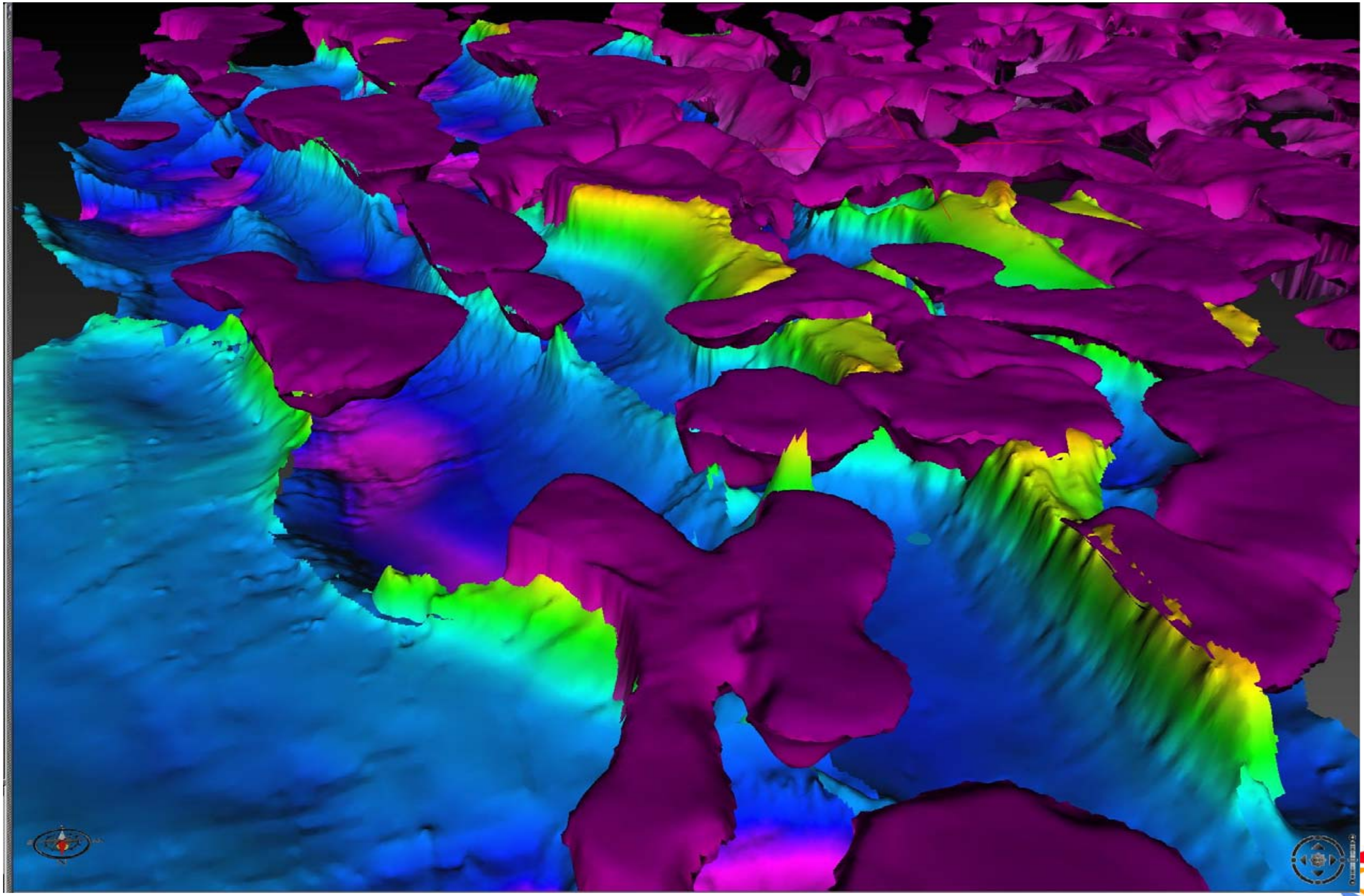
High performance Computing and O&G Challenges

Seismic exploration challenges



High Performance Computing and O&G challenges

• **Worldwide Context**

• **Seismic,sub-surface imaging**

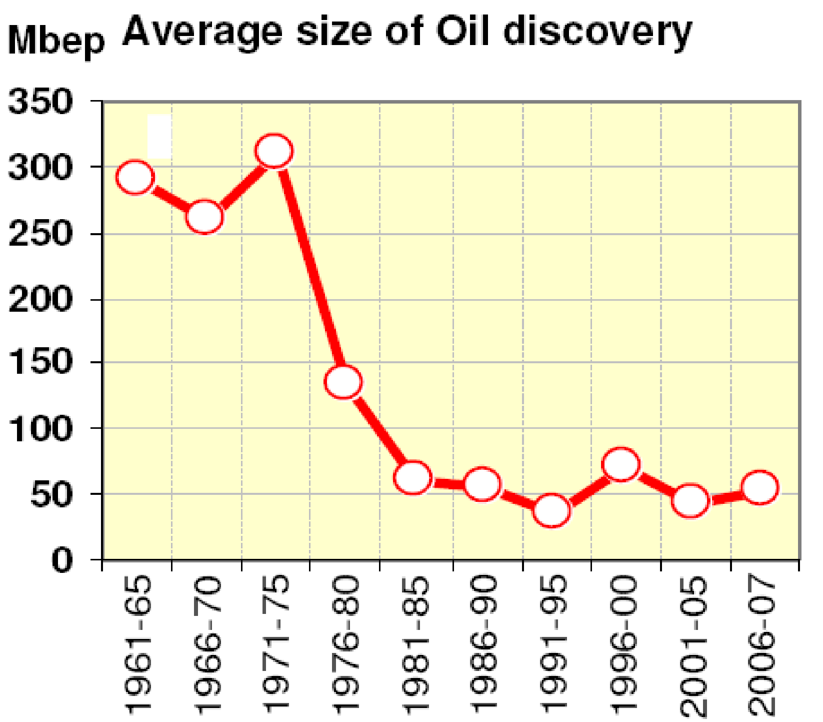
• **Computing Power needs**

• **Accelerating technology status and perspectives in TOTAL**

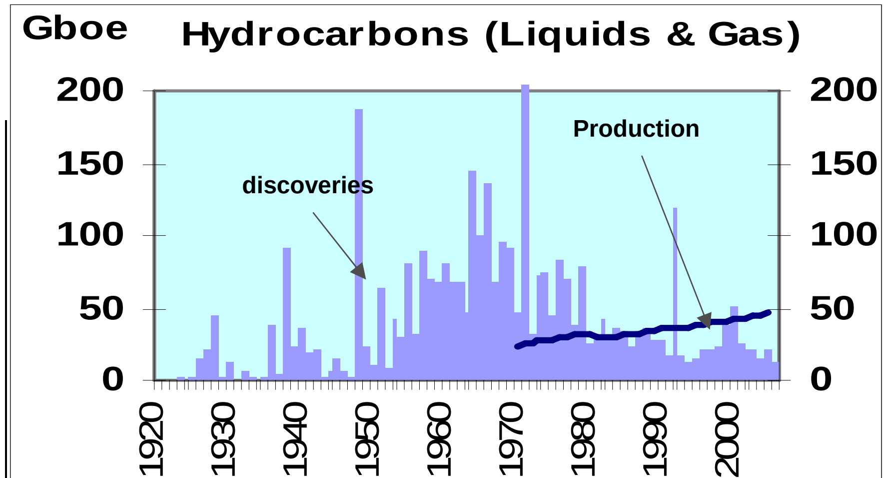
Worldwide context

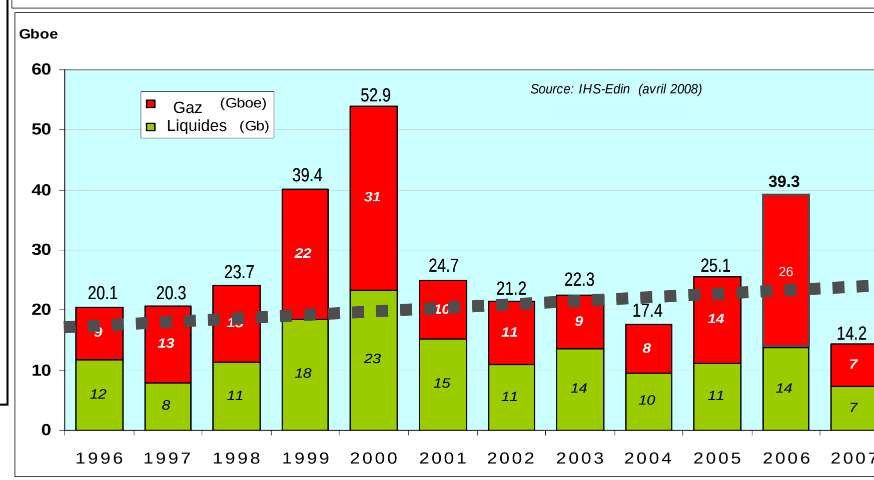
- Worldwide O&G consumption/year approx. 50 Gboe

- Average size of new Oil field discovery ~ 50 Mboe since the last 25 years



- Oil Discoveries vs. Oil Production

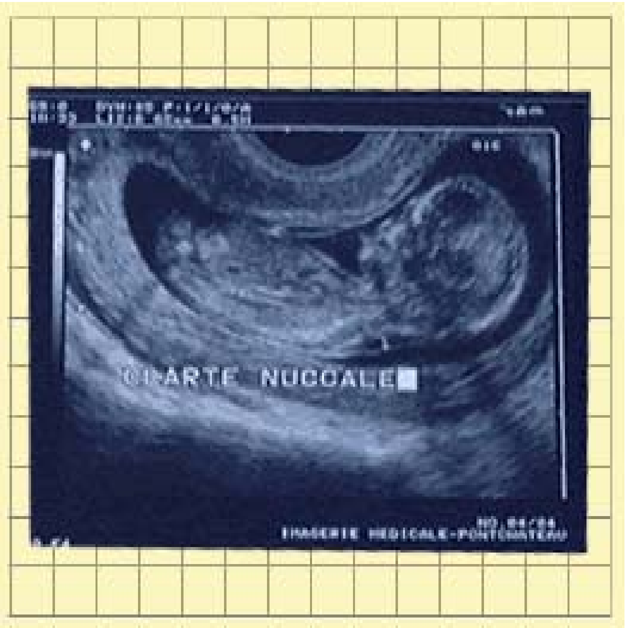


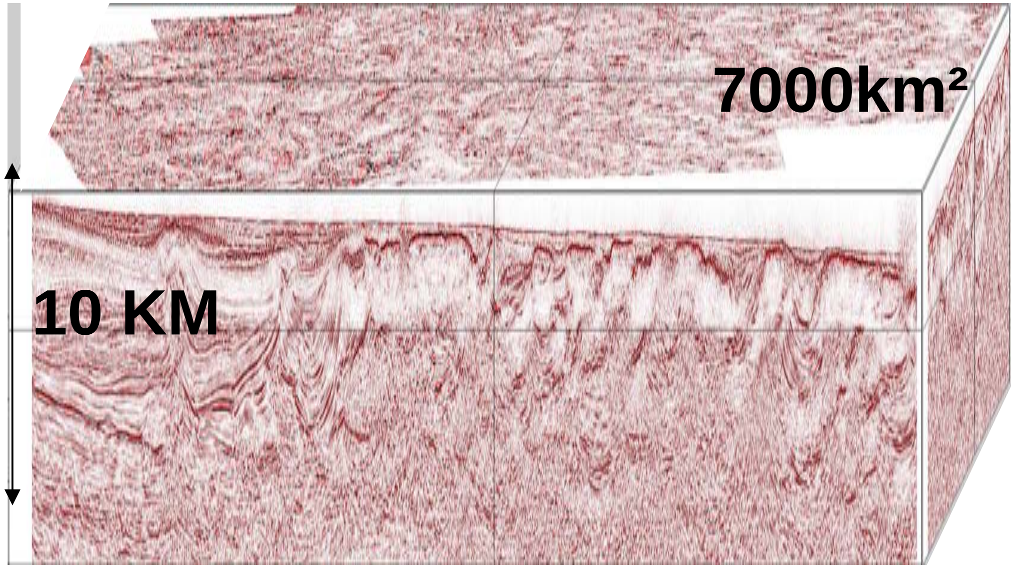


Principle: Ultrasound

Seismic imaging

Medical imaging

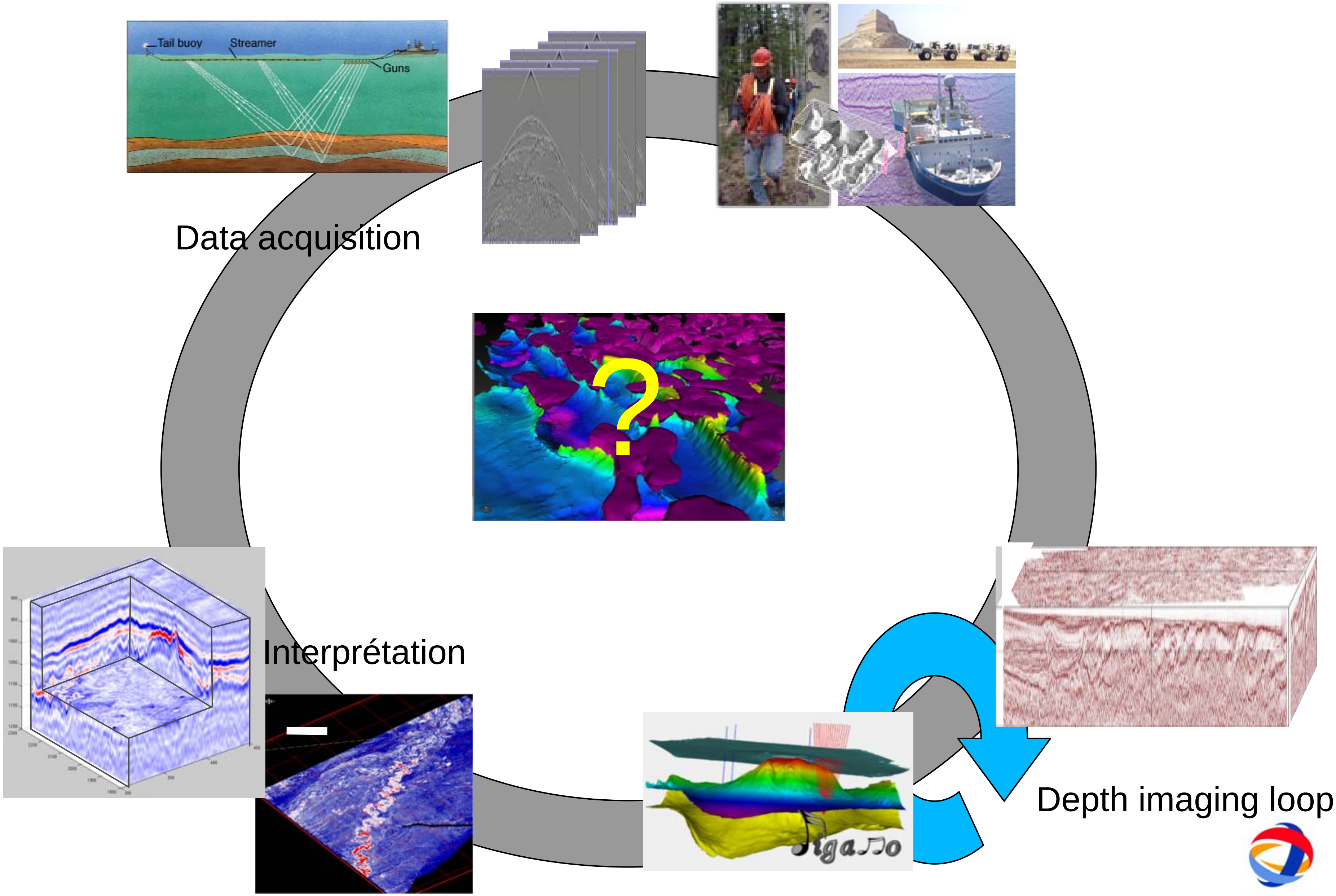




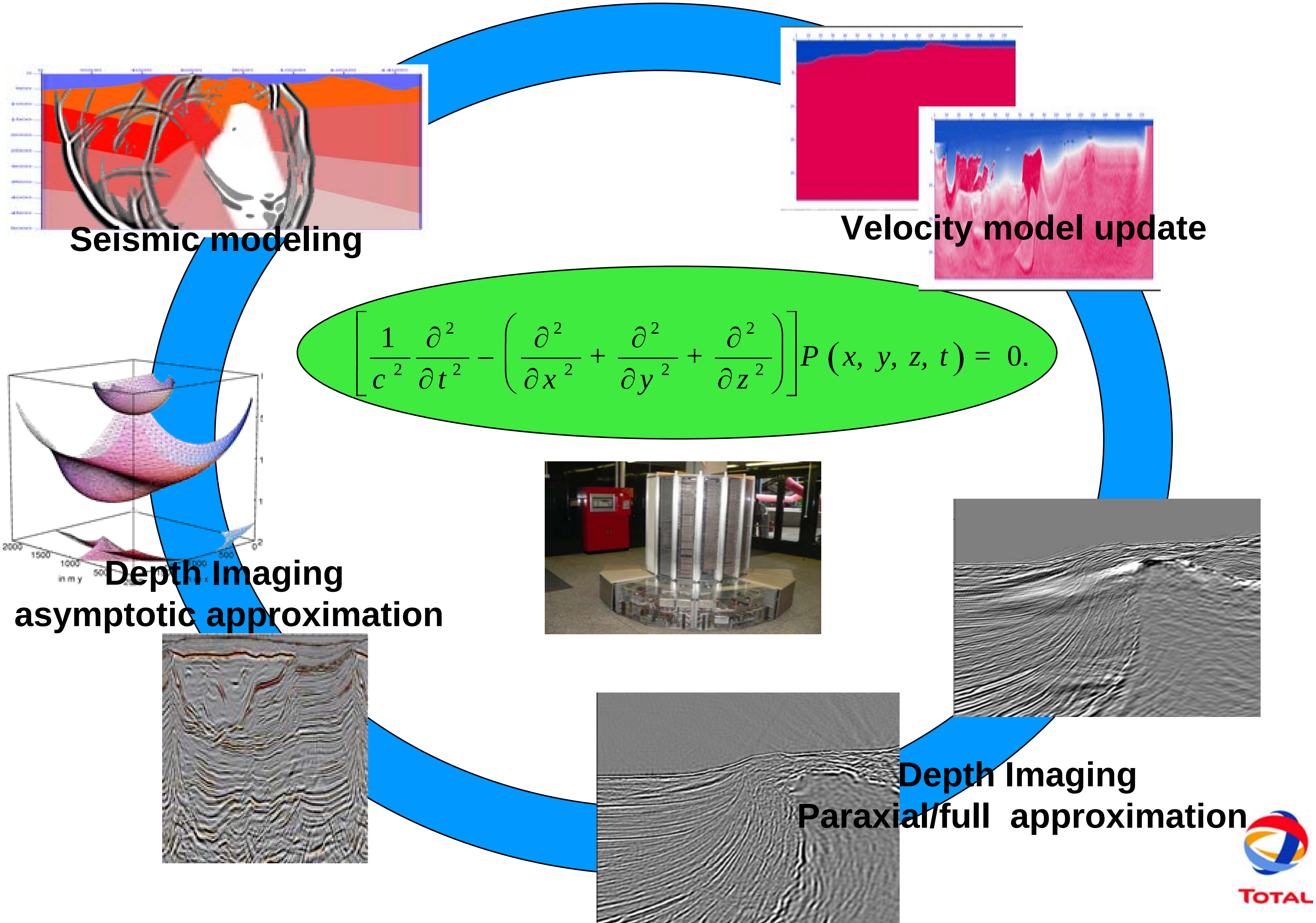
- Signal frequency: 1 MHz
  ➔ Image resolution: few mm.
- Approximately homogeneous media.

- Signal frequency: between 6 and 90 Hz
  ➔ Image resolution: some tens of m.
- Heterogeneous media (spatial variability of density and signal velocity)

Seismic imaging in a nutshell

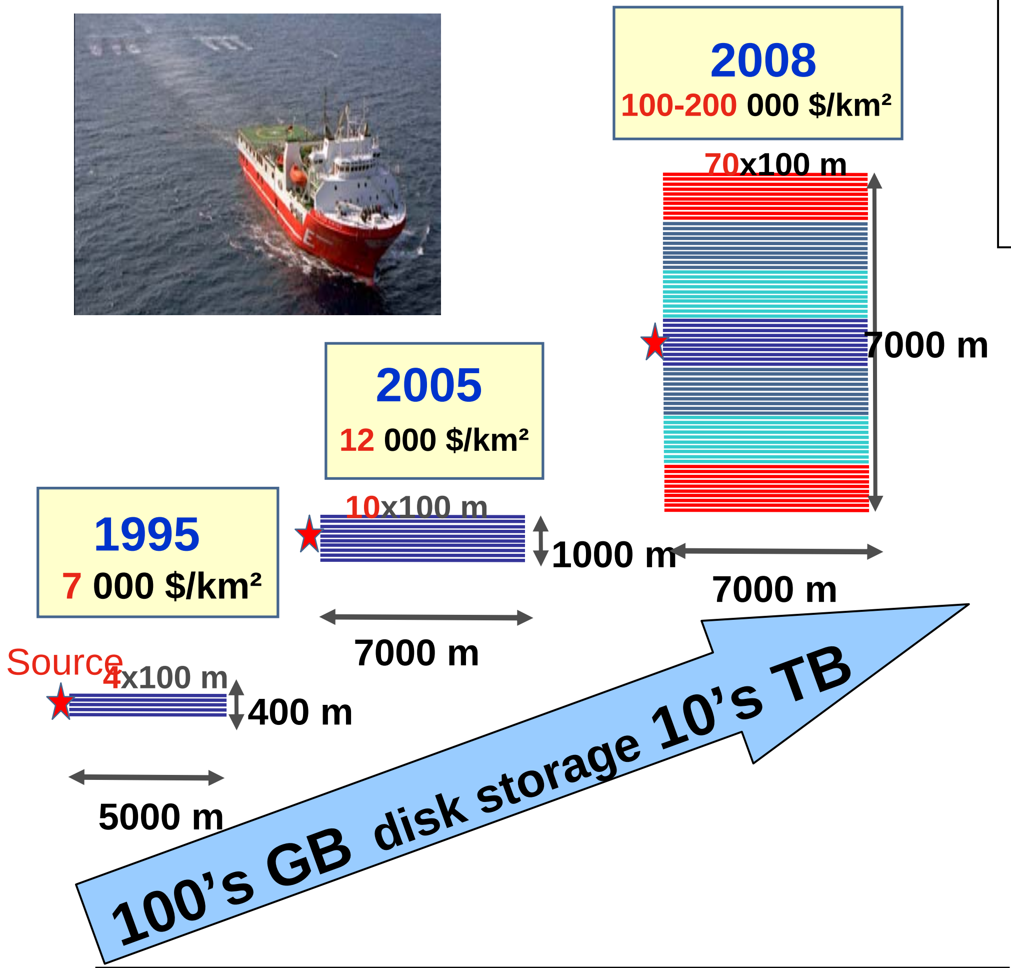


Wave Equation the basic tool



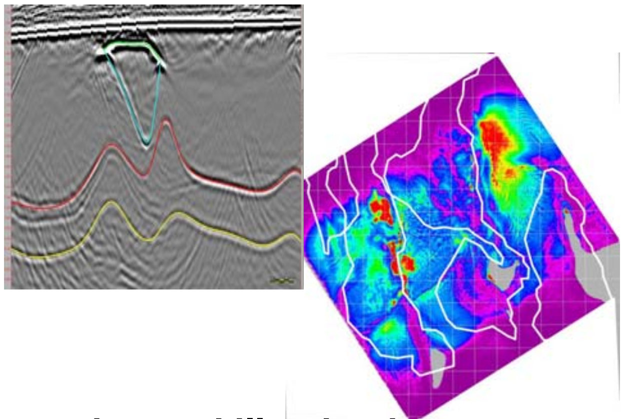
$$\left[\frac{1}{c^2}\frac{\partial^2}{\partial t^2} - \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}\right)\right] P(x,\ y,\ z,\ t) = 0.$$

Seismic acquisition breakthrough and modeling



Acquisition modeling is routinely applied to optimize survey acquisition design and reduce overall cost.

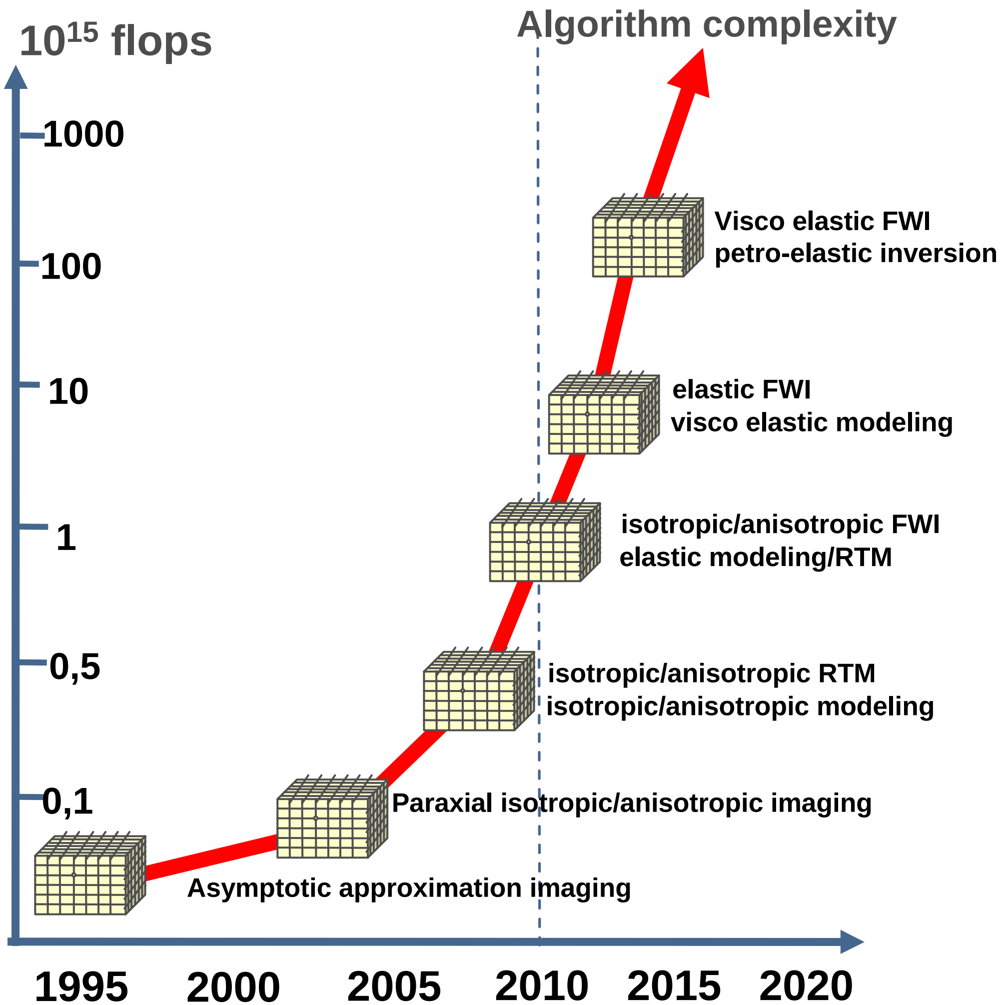
10,000 modeled shots - 4 weeks - 100TF



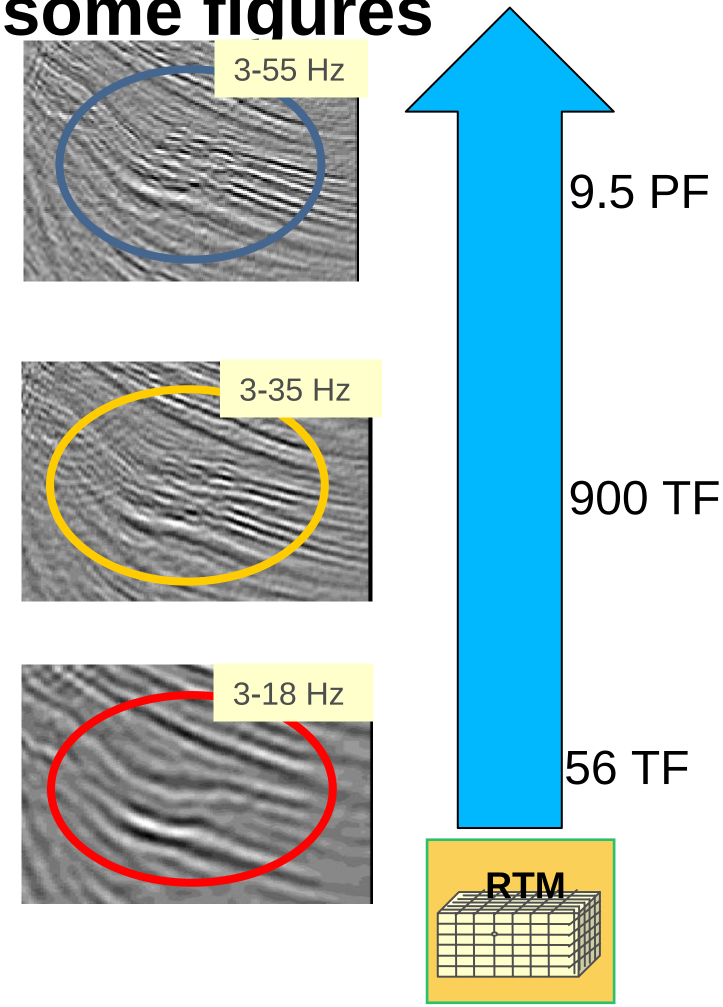
Migration section and illumination maps computed from modeled seismic acquistion provide useful information to geophysicist

Progress in acquisition seismic gives access to high seismic data quality but increase dramatically the size of storage and overall cost

Codes and computing effort some figures

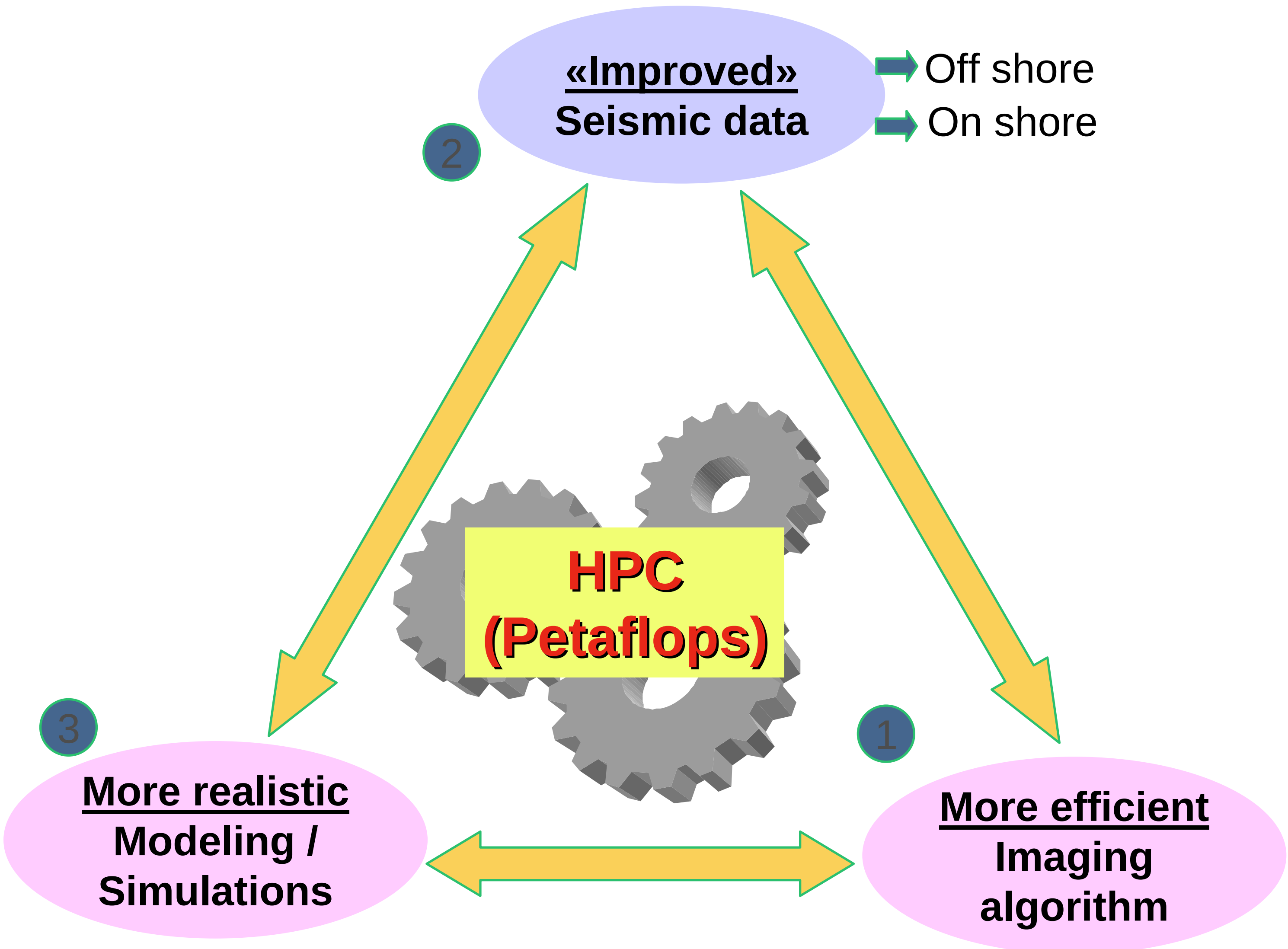


Algorithmic complexity and corresponding computing power



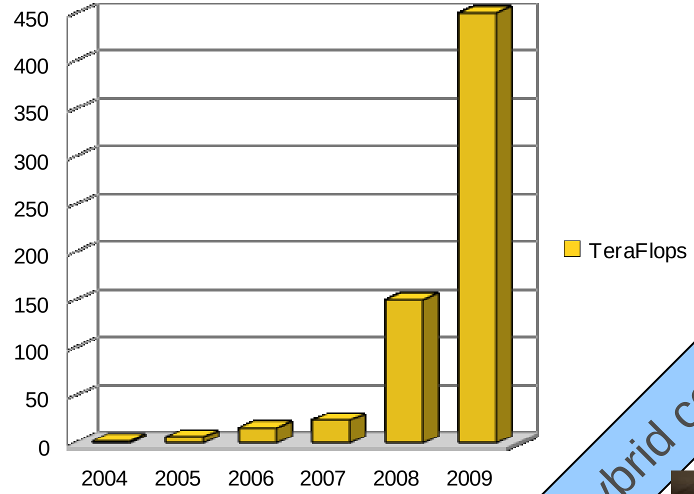
Computing ressource evolution when only one parameter is changed while using the same algorithm: In this case we request the processing to be performed in 8 days

HPC: the necessary link



TOTAL

HPC Evolution in TOTAL



From SMP cluster to MPP and Hybrid computing

SGI ICE+ MPP (16384 Intel X86-64 + 256 GPUS)

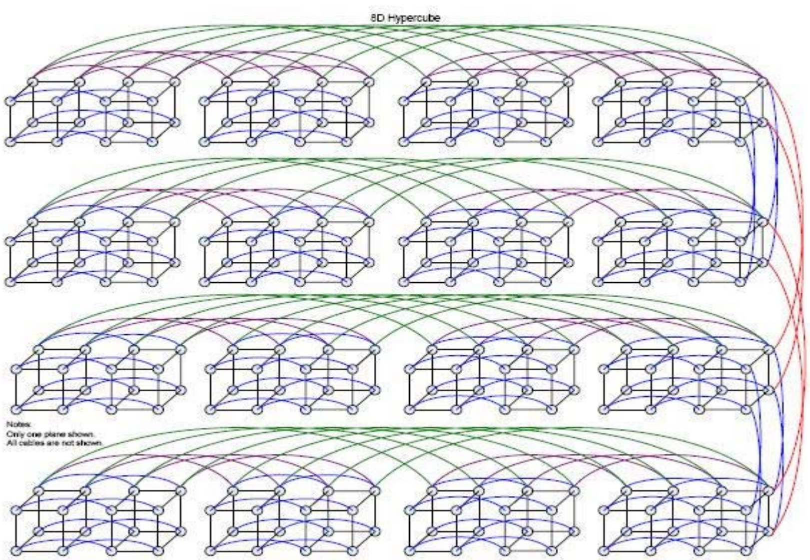
SGI ICE+ MPP (10240 Intel X86-64)

IBM SMP cluster (2048 P5)

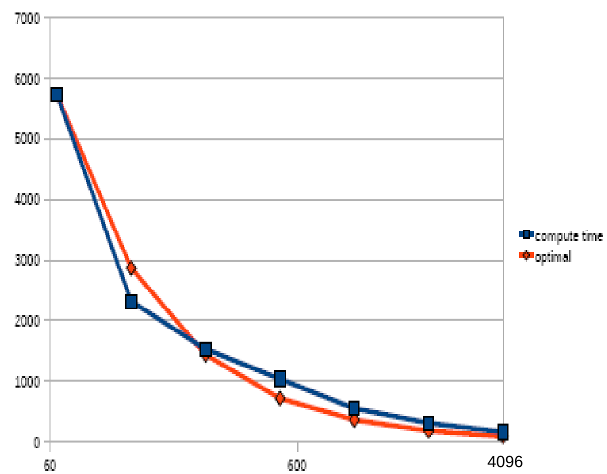
SGI Altix SMP cluster (1024 intel ia64)

Why MPP ?

- **Seismic depth is massively Parallel:**
  - Embarrassingly over the shots: ex 100000 shots can be solved in one iteration on a 100000 cpus machine
  - Explicit data distribution and data parallel model programming when processing one shot
- **Reliability and performances on very large configurations**
- **Impact on footprint and power dissipation**



SGI ICE+ Hypercube topology interconnect:
- **Scalable single system**
- **Easy to manage**

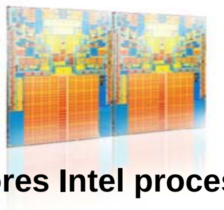


Taking advantage of Scalable interconnect
Leads to very efficient algorithm implementation

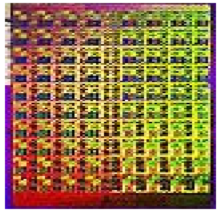
(3D Wave Equation Finite difference solver)

Why hybrid Computing ?

- Trends: multiply the number of cores
- Our codes can take advantage of many core technology
- GPUs can be seen as large multicore processors and can offer tremendous speed up.
- GPUS first step to the massive multicore technology ?

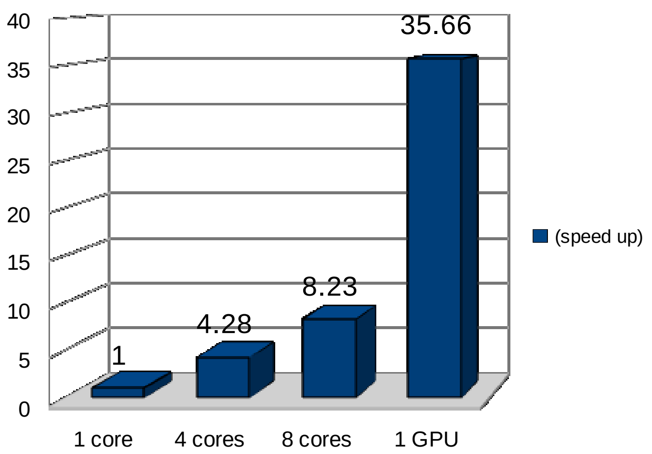


4 cores Intel processor



80 cores Intel processor (prototype)

Nvidia GPU 240 cores ~ 1TF single precision



3D Wave Equation solver speed up comparison

- Having access to different technologies through the same interconnect can offer new perspectives in terms of model programming and application development
- To reach the performances needed for solving our problems, we have no choice but combining high scalable interconnect an massive multicore technology.

Accelerating technology actual status and perspectives in TOTAL

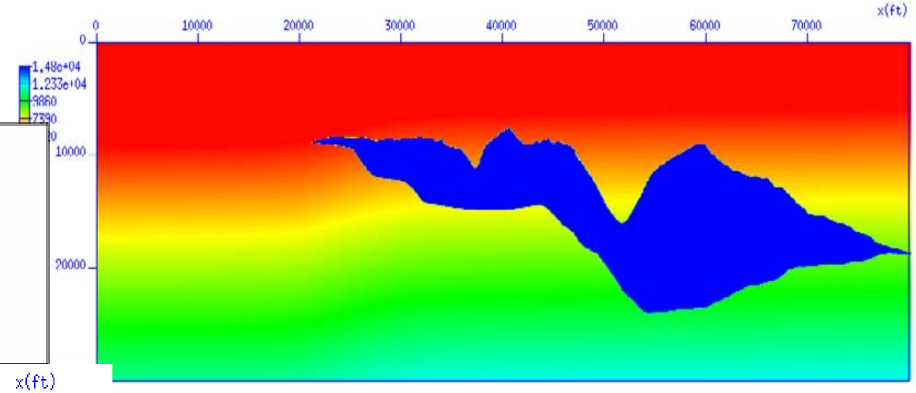
Starting in 2007 with the following questions

1. Is it possible to take advantage of GPU for seismic depth imaging
2. What programming model and what programming language?
3. What about communication between host and GPU, best configuration?
4. Can we extrapolate the performances obtained on a single GPU to a full system: Cluster ?
5. What about building industrial applications (Modeling, RTM…) on a GPU cluster based system ?

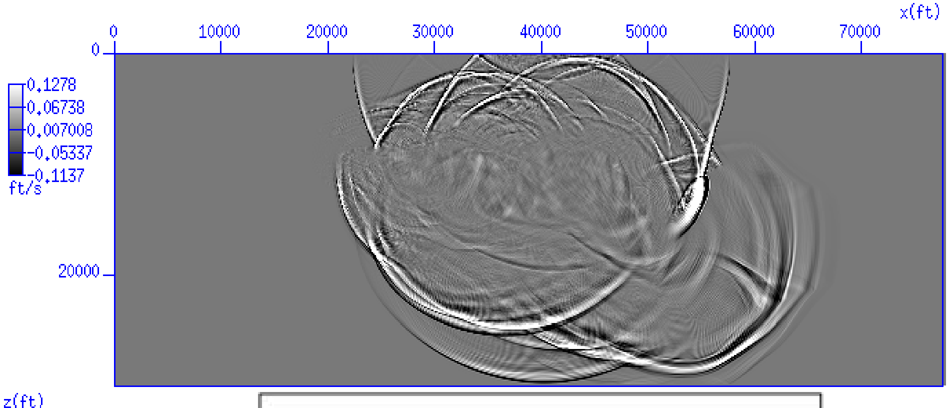
TOTAL

1. Is it possible to take advantage of GPU ?

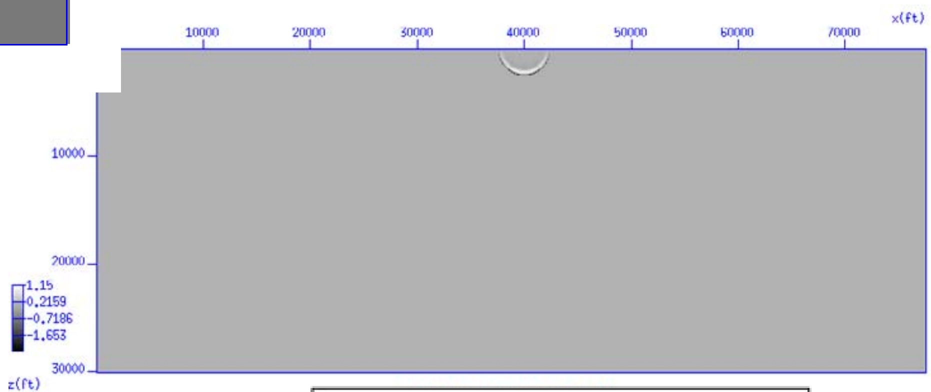
First test: 2D Wave Equation modeling
- Cuda implementation
- Real CPU time: 160s



Velocity model



GPU:12808 time steps



CPU:1778 time steps

TOTAL

Taking advantage of GPUS: actual status

❑2 main applications implemented on GPU cluster:

- Seismic modeling
- Reverse time Migration

❑ Seismic modeling

- motivation:
  - accelerate survey acquisition modeling
  - Wave Equation solver is the basic kernel for seismic depth imaging algorithm
  - Explore different implementations on GPUS: (single GPU, multi GPUs)
  - Explore model programming and programming language

❑RTM

- Motivation
  - Most accurate depth imaging method: the most expensive
  - Take advantage of progress made on Seismic modeling (same kernel)
  - Explore communication load balancing between host and device

TOTAL

Seismic modeling

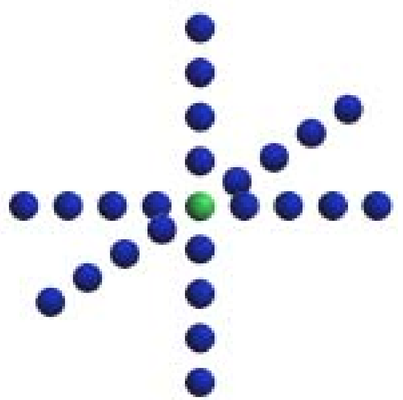
❑Based on the solution of the Wave Equation

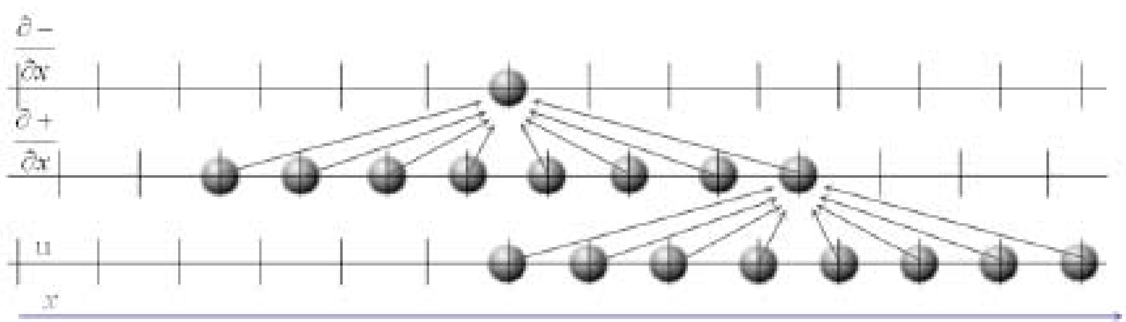
$$\frac{1}{c^2\rho}\frac{\partial^2 u}{\partial t^2} - \nabla.(\frac{1}{\rho}\nabla u) = f$$

❑ Explicit Time-space finite difference

$$\frac{\partial^2 u}{\partial t^2} = K[\frac{\partial_-}{\partial x}(\frac{1}{\rho}\frac{\partial_+ u}{\partial x}) + \frac{\partial_-}{\partial y}(\frac{1}{\rho}\frac{\partial_+ u}{\partial y}) + \frac{\partial_-}{\partial z}(\frac{1}{\rho}\frac{\partial_+ u}{\partial z})]$$

$$\frac{\partial_+}{\partial x}u((i+\frac{1}{2})\Delta x) = \sum_{n=0}^{3} a_n[u((i+1+n)\Delta x) - u((i-n)\Delta x)]$$

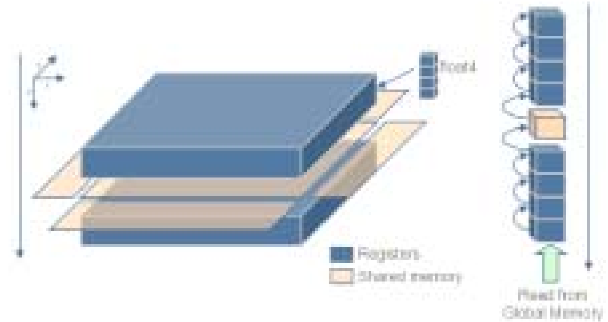




Seismic modeling

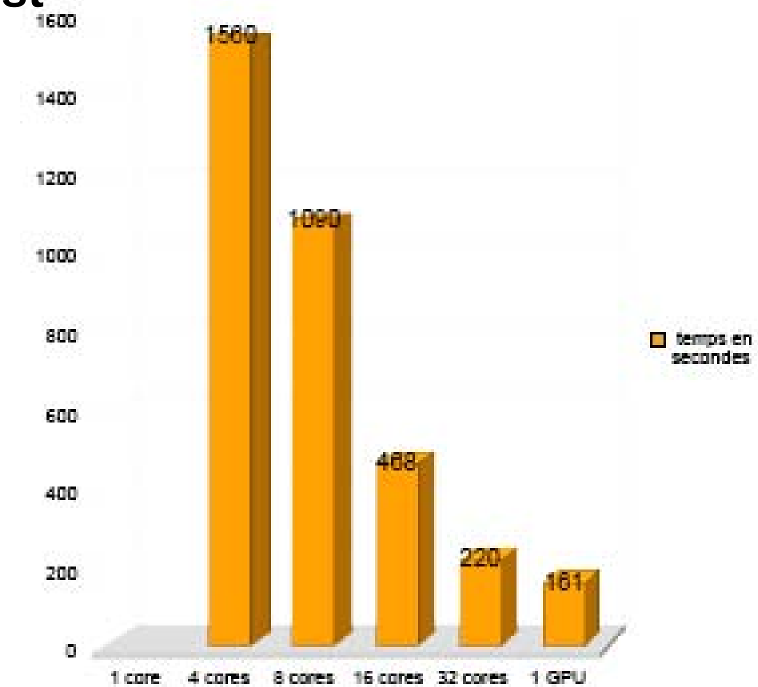
❑Implementation

- 2 passes:
  1. compute partial derivative along z axis
  2. 2D tile sliding window
- Padding to ensure global memory coalescing
- Partial transfer and page locked host memory



❑ Example

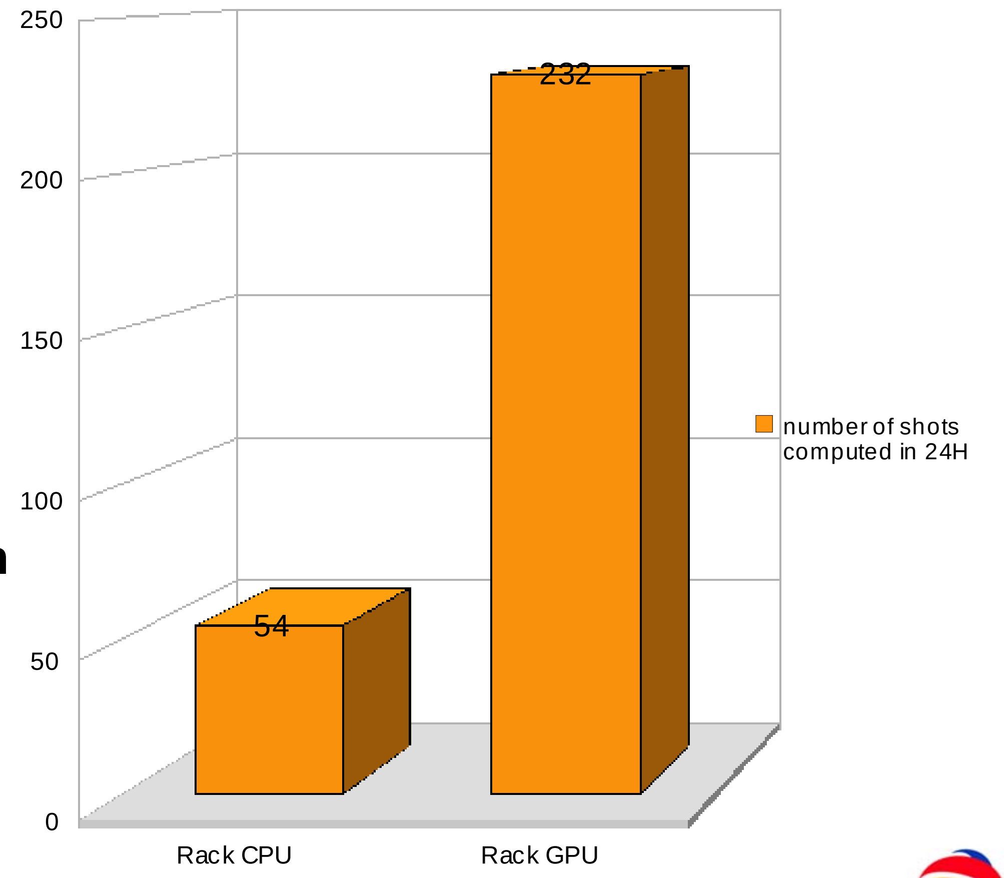
- Model size 512x512x512
- 490 time step
- SGI ICE+, Harpertown 3GHZ
- Test :
  - 1 GPU
  - 4 cores (4 MPI)
  - 8 cores (8 MPI)
  - 16 cores (16 MPI)
  - 32 cores (32 MPI)



Seismic modeling

- ❑ **Example**

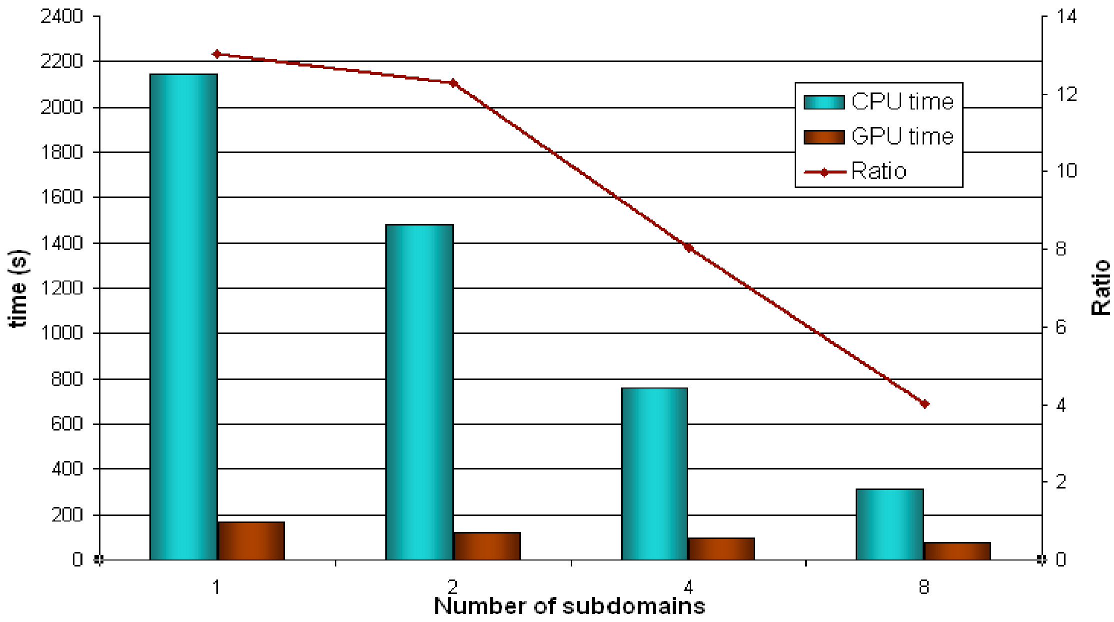
  - **Model size 560x557x905, 22760 time steps**
  - **SGI ICE+, 512 harpertown 3GHZ**
  - **SGI ICE+, 16 blades harpertown 3 GHZ, 8 Teslas**
  - **8 cores per shot (64 shots in parallel)**
  - **2 GPUs per shot (16 shots in parallel)**



Reverse Time Migration

- ❑ **Based on the solution of similar kernel used for seismic modeling**

- ❑ **RTM is solved in 2 Steps involving:**
  1. **Seismic modeling and wavefield storage**
  2. **Seismic modeling, wavefield reload and imaging condition**
  - **For efficiency reasons**
    - **Seismic modeling must recover wavefield store/load**

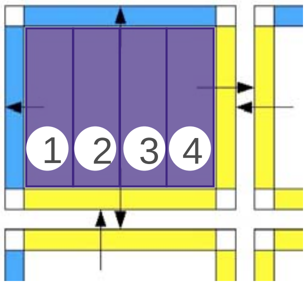


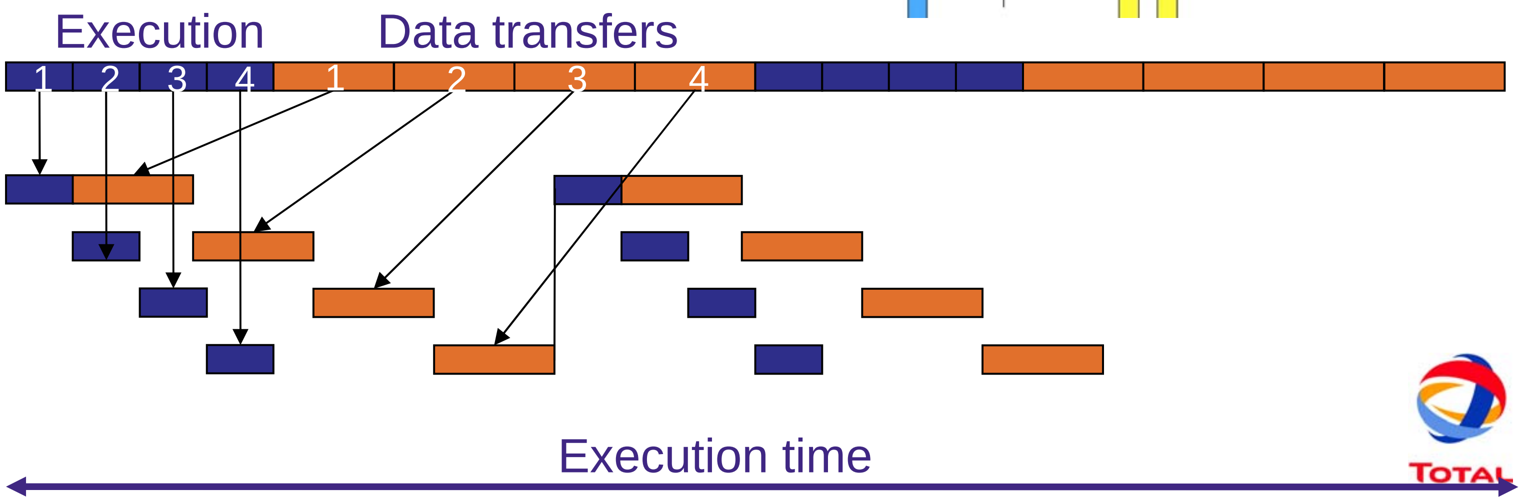
CPU-GPU Data transfer optimization

Take advantage of asynchronous communications to overlap data transfer

- **First implementation: 75% of elapsed time spent on PCIe communications**
- **Take advantage of :**
  - **PCIe Gen2 bandwidth**
  - **DMA accessible “pinned memory”**
  - **Asynchronous communications**

%	synch	asynch
FWD	32	15
BWD	41	25





2/ Programming model and programming language

- **Model programming:**
  - **Take advantage of data parallelism**
    - **Mainly based on efficiently using memory hierarchy, with different approaches:**
    - **Increase the data locality and re-use**
    - **Use simple 3D cache blocking,**
    - **Replace by a 2D cache, sliding on the 3rd dimension**

Number of read accesses per data point	
RTM 2D	4,25
RTM 3D with texture	29
RTM 3D with 3D shared memory accesses	7,5
RTM 3D with sliding 2D shared memory	4

2/ Programming model and programming language

- **Programming language:**

  - **CUDA: progess rapidely, new features such as partial copy, asynchronous communications, pined memory data access…**

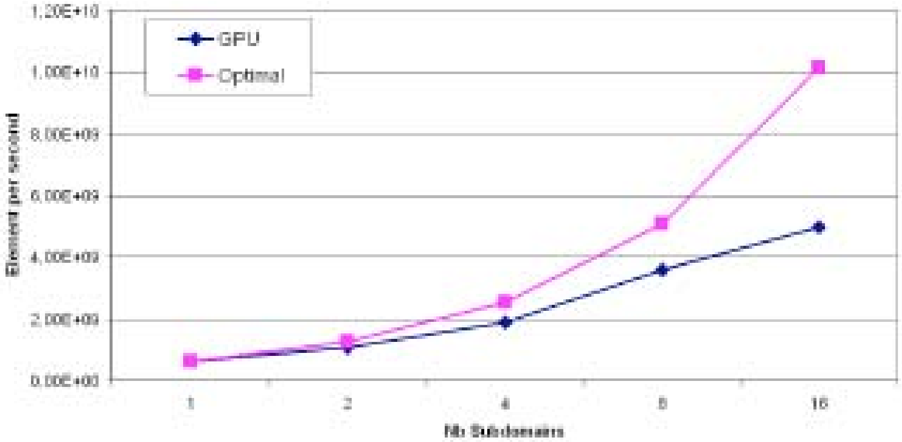
  - **HMPP:**
    - **Express parallelism and communications in C, Fortran source code:**
      - **Direct integration into Fortran Code through the use of directives “ à la OpenMP”**
    - **HMPP handles CPU-GPU communications and kernel launch.**
    - **HMPP manage automatically HW device.**
    - **Systematically used in TOTAL for developing accelerated kernel.**
    - **Work in close relation with CAPS on the automatic code generation.**
    - **Some delays between new features in CUDA and HMPP support**

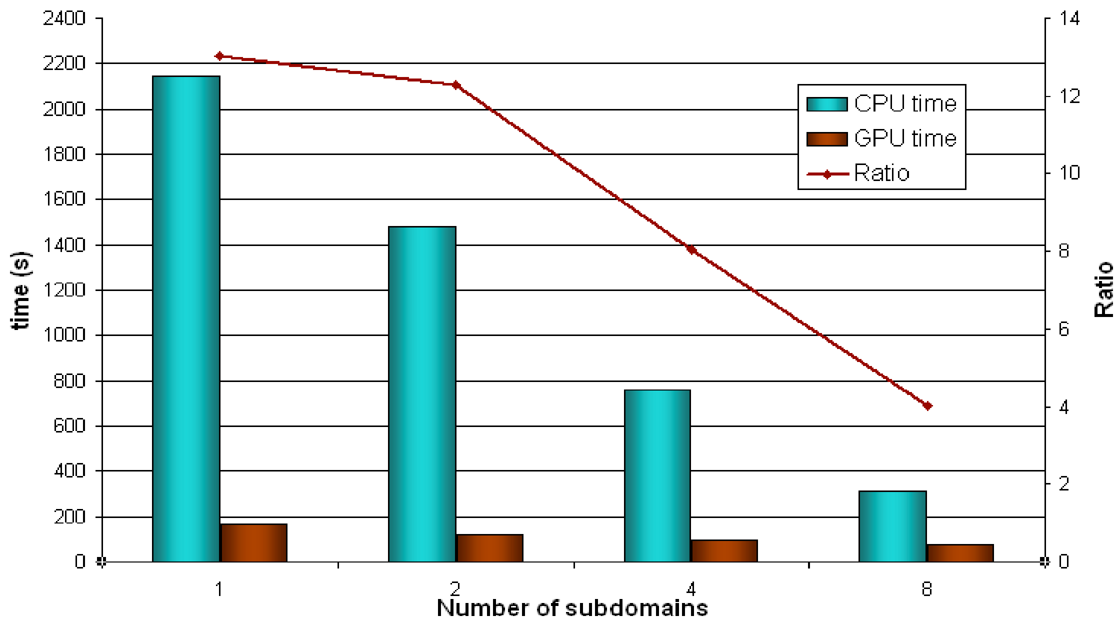
    ➔ should be anticipated: CUDA pre-released to CAPS ?

  - **OpenCL: Standard (wait and see), will be addressed through HMPP**

3/ Host Device configuration

4/ Single GPU to multi GPU : Cluster





Conclusion

❑Computation needs for seismic processing is nearly not bounded!

- Algorithm improvement taking advantage
  - Of high frequency seismic (impact the size of computational grid)
  - Seismic acquisition parameters (Azimuth width, distance between receivers, …)
- Implement more complete solution of the imaging solver to solve new challenges: foothills, Heavy Oil….

❑ Need a pro-active R&D strategy

- Algorithms (Full Waveform Inversion, new Solvers, reservoir simulation, EM…)
- New Hardware architectures and programming models
- Impact of new technology on processing:
  - Several Teraflops in a WorkStation ?
  - HPC containers: On site processing, embedded processing

❑ Accelerating solution:

- good progress in understanding the use of GPUs
- we have a clear strategy : model programming and application development
- GPUs: first step before many core technology ?
- We do not exclude hybrid technology: (many core + GPUS)
- Still need to work in close relation with industrial partners to improve Hardware and tools.

TOTAL