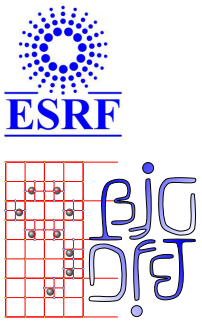


GPU and Wavelets

25e Forum ORAP

SITE EDF R& D – CLAMART

Efficient use of hybrid computing clusters for nanosciences

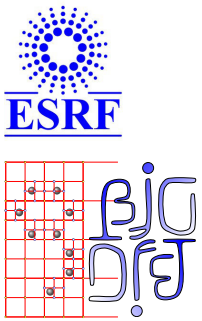
Luigi Genovese

CEA, ESRF, BULL, LIG

12 Octobre 2009

with Jean-François Méhaut, Matthieu Ospici, Thierry Deutsch

Outline



GPU and Wavelets

1. Electronic structure calculations
   - Ab initio methods
   - BigDFT code
   - Main operations, parallelisation
2. Hybrid code
   - The S_GPU library
3. Performances
4. Conclusions

Ab initio calculations with DFT

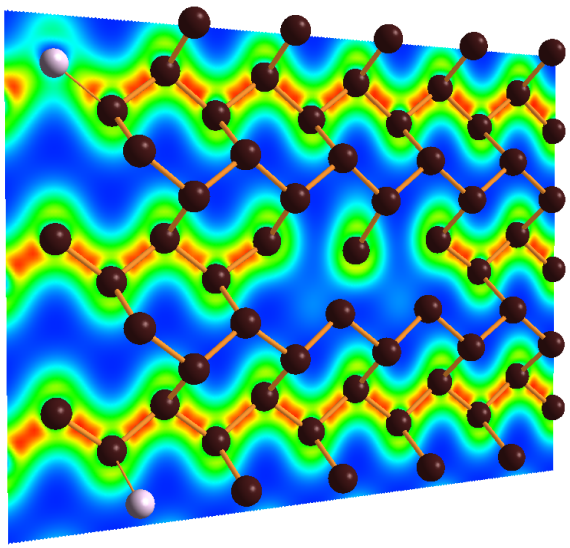
Several advantages

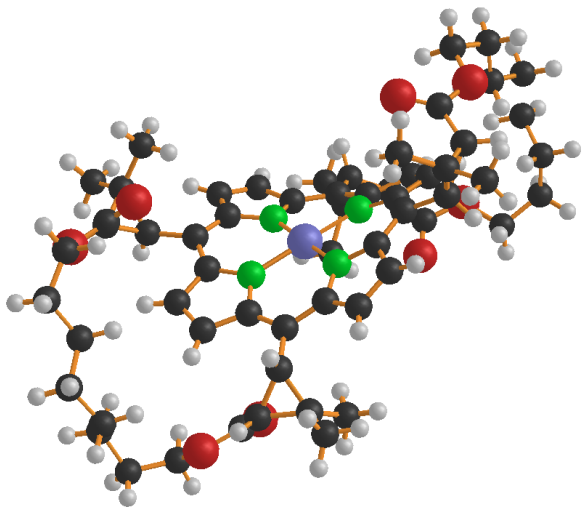
- ✔ Ab initio: No adjustable parameters
- ✔ DFT: Quantum mechanical (fundamental) treatment

Main limitations

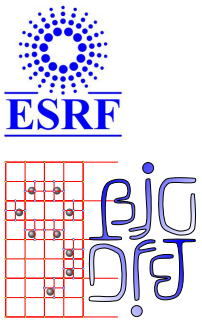
- ✘ Approximated approach
- ✘ Requires high computer power, limited to few hundreds atoms in most cases

Wide range of applications: nanoscience, biology, materials





Performing a DFT calculation (KS formalism)



GPU and Wavelets

Find a set of orthonormal orbitals $\Psi_i(\mathbf{r})$ that minimizes:

$$E = -\frac{1}{2}\sum_{i=1}^{N/2}\int \Psi_i^*(\mathbf{r})\nabla^2\Psi_i(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int\frac{\rho(\mathbf{r})\rho(\mathbf{r}')}{|\mathbf{r}-\mathbf{r}'|}d\mathbf{r}'d\mathbf{r} + E_{xc}^{LDA}[\rho(\mathbf{r})] + \int V_{ext}(\mathbf{r})\rho(\mathbf{r})d\mathbf{r}$$

with $\rho(\mathbf{r}) = \sum_i \Psi_i^*(\mathbf{r})\Psi_i(\mathbf{r})$

(Kohn-Sham) DFT "Actors"

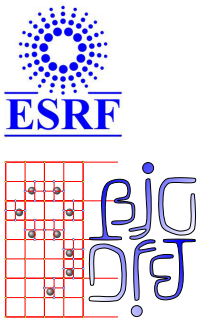
- A set of wavefunctions $|\psi_i\rangle$, one for each electron
- A computational approach on a finite basis

$\Rightarrow$ One array for each Ψ_i

$\Rightarrow$ A set of computational operations on these arrays which depend on the basis set

- A (good) computer...

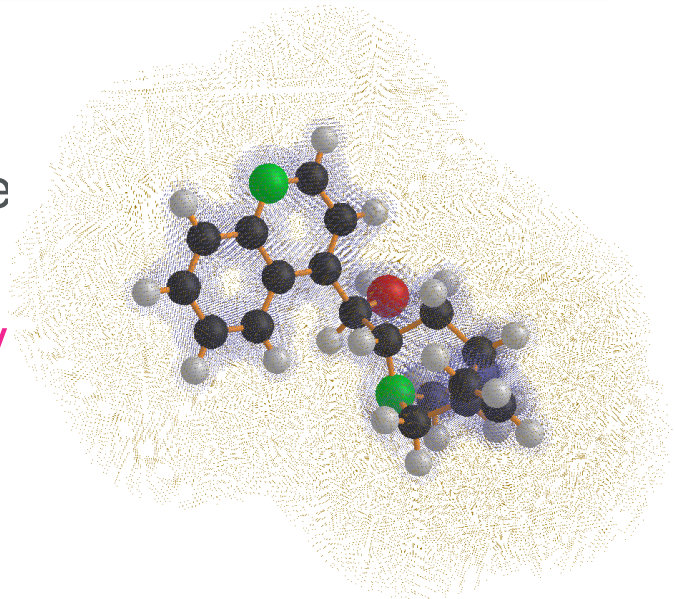
A basis for nanosciences: the BigDFT project



STREP European project: BigDFT(2005-2008)

Four partners, 15 contributors:
CEA-INAC Grenoble, U. Basel, U. Louvain-la-Neuve, U. Kiel

Aim: To develop an ab-initio DFT code based on Daubechies Wavelets, to be *integrated in ABINIT*, distributed freely (GNU-GPL license)



L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, *et al.*,

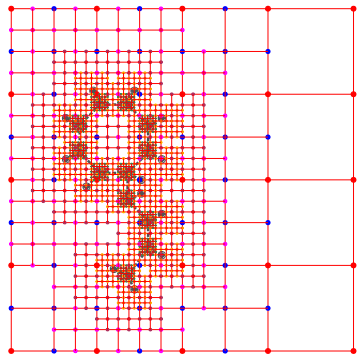
"Daubechies wavelets as a basis set for density functional pseudopotential calculations",

J. Chem. Phys. **129**, 014109 (2008)

A DFT code based on Daubechies wavelets

Wavelets

A basis with optimal properties for expanding localised information



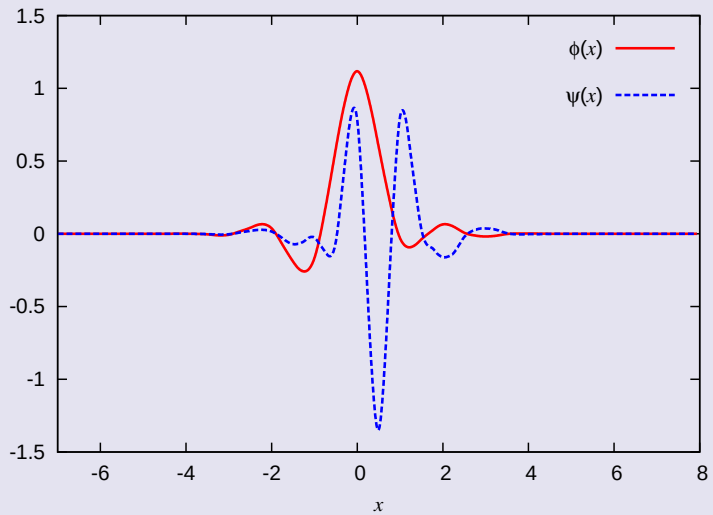
- Localised in real space
- Smooth (localised in Fourier space)
- Orthogonal basis
- Multi-resolution basis
- Adaptive
- Systematic

From early 80's

Applied in several domains

Interesting properties for DFT

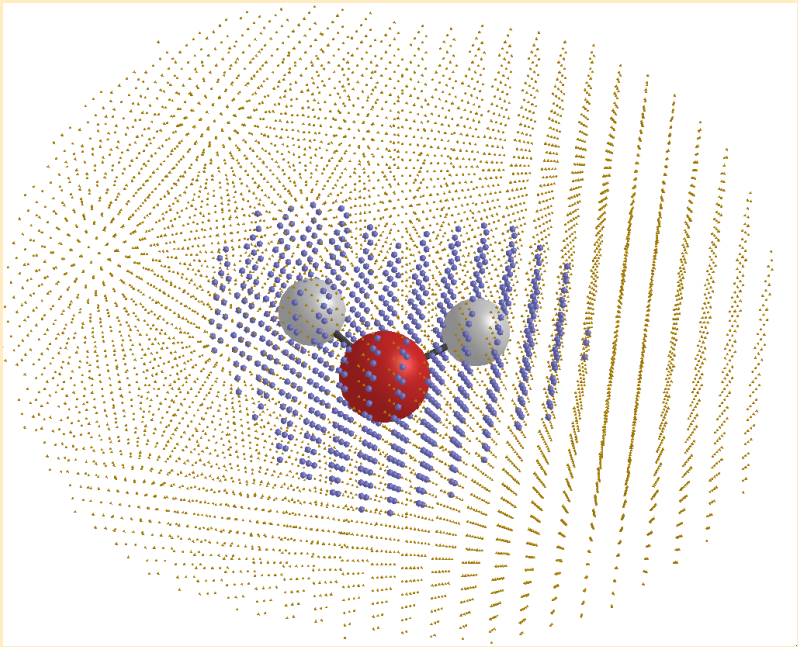
Daubechies Wavelets



Wavelet properties: adaptivity

Adaptivity

Resolution can be refined following the grid point.



The grid is divided in Low (1 DoF) and High (8 DoF) resolution points. Points of different resolution belong to the same grid. Empty regions must not be "filled" with basis functions.

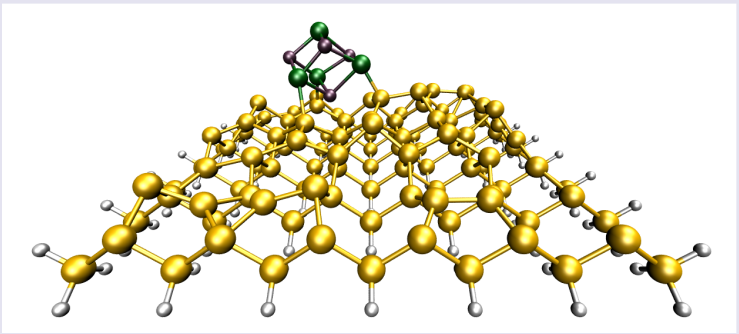
Localization property,real space description

Optimal for big & inhomogeneous systems, highly flexible

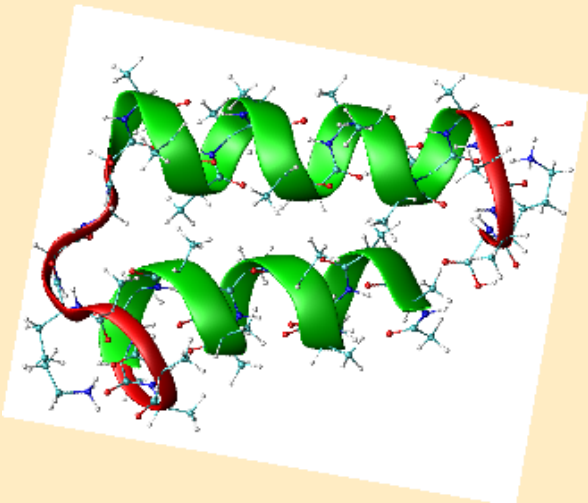
Some ongoing applications

(Group of S. Goedecker, Basel University)

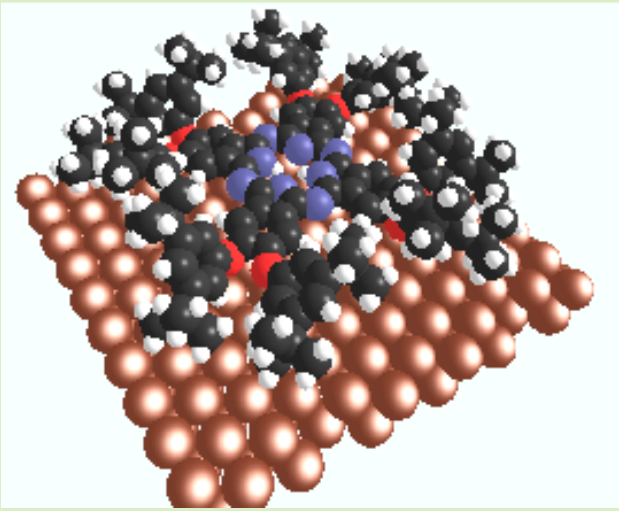
Study of favorable adsorption sites of NaCl clusters on Si tips during AFM experiments ($\sim$ 250 at.)



350 at.: DFT calculation to assess the accuracy of force fields



$\sim$ 300+500 at. : Organic molecule @ Cu surface. HOMO-LUMO chg. dens. are compared to STM expts.



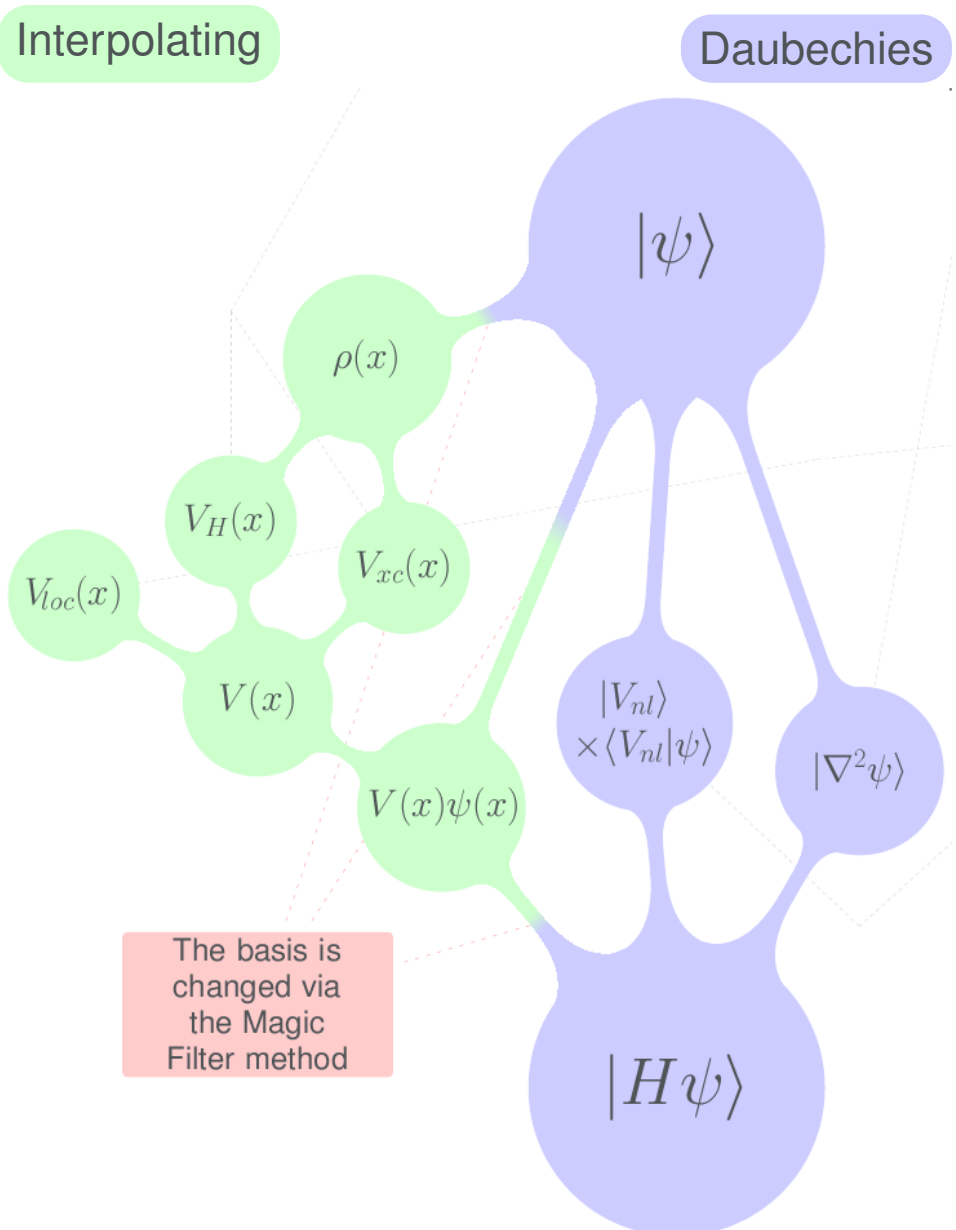
Operations performed

Different numerical operations performed:

- For each wavefunction (Hamiltonian application)
- Between wavefunctions (Linear algebra)

Comput. operations

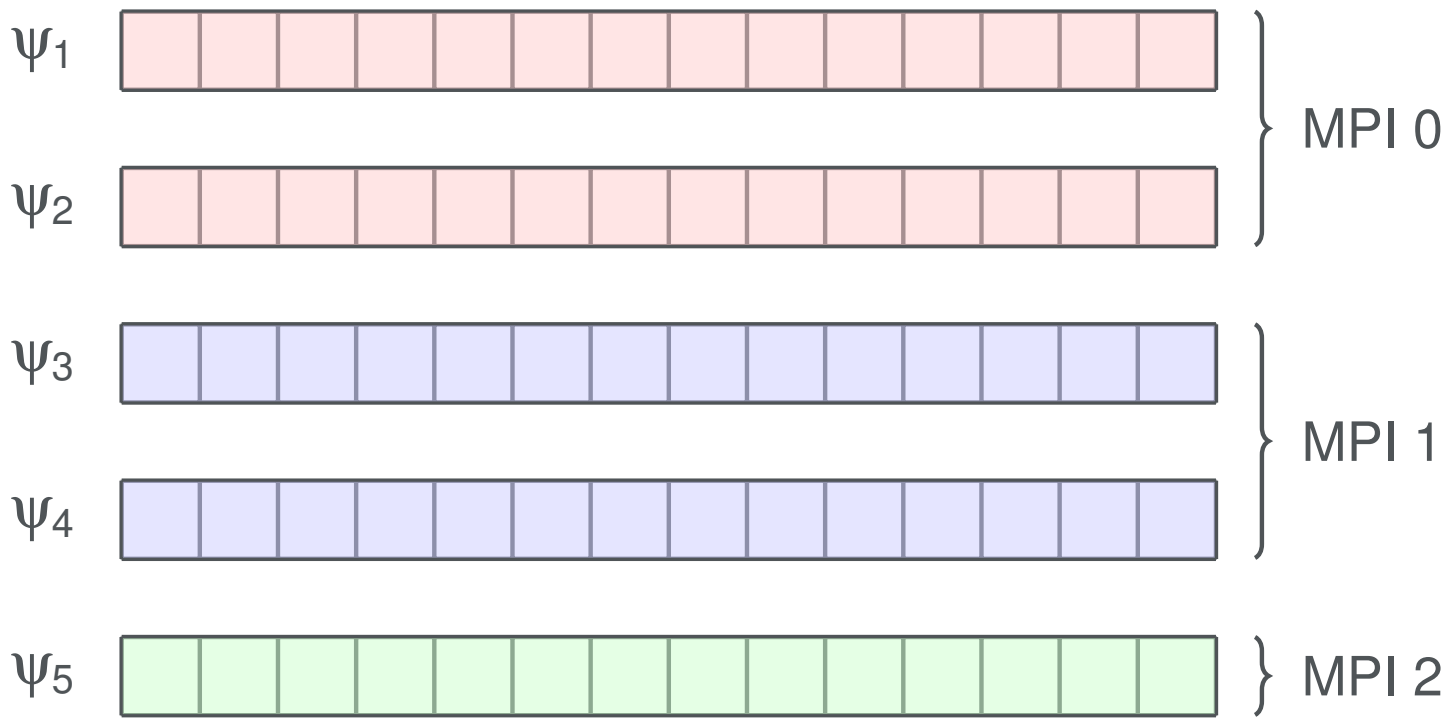
- Convolutions with short filters
- BLAS routines
- FFT (Poisson Solver)



Orbital distribution scheme

Used for the application of the hamiltonian

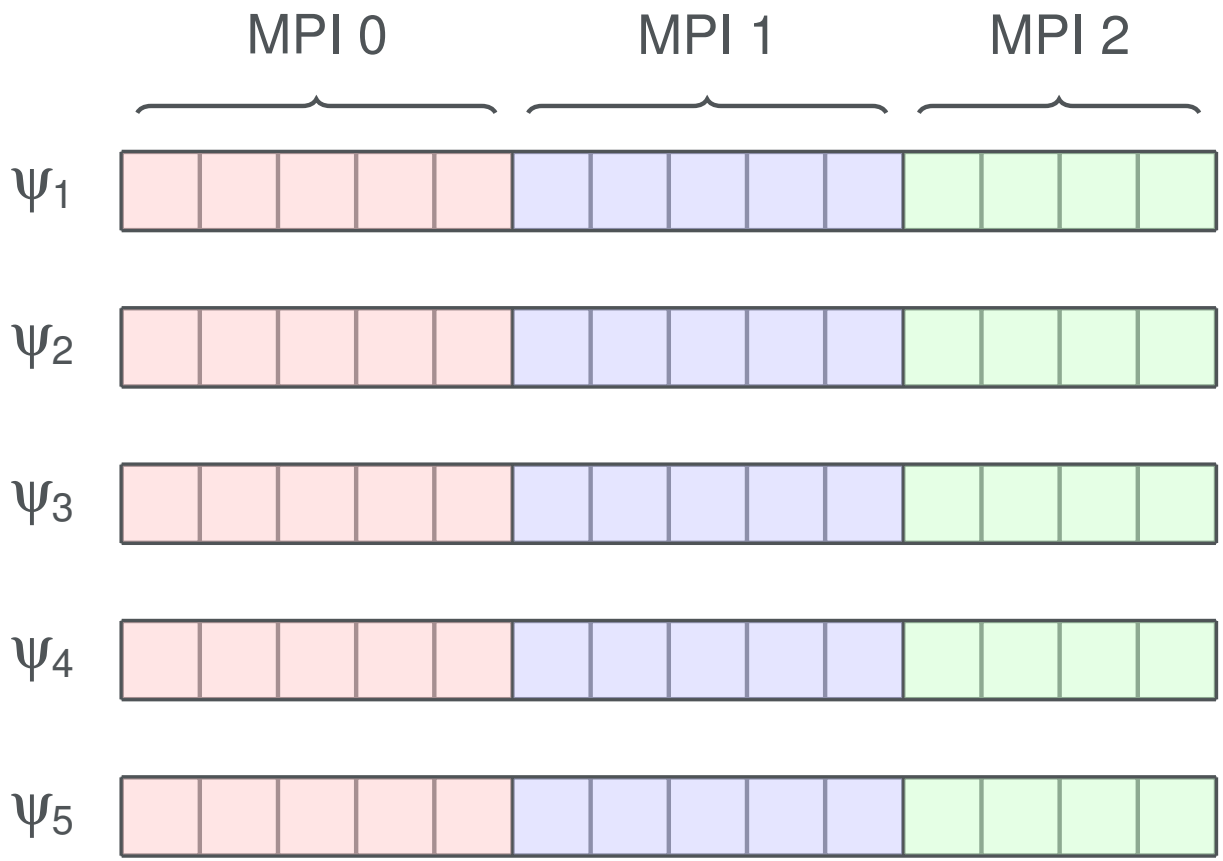
The hamiltonian (convolutions) is applied separately onto each wavefunction



Coefficient distribution scheme

Used for scalar product & orthonormalisation

BLAS routines (level 3) are called, then result is reduced



Communications are performed via MPI_ALLTOALLV

High Performance Computing

Localisation & Orthogonality $\longrightarrow$ Data locality

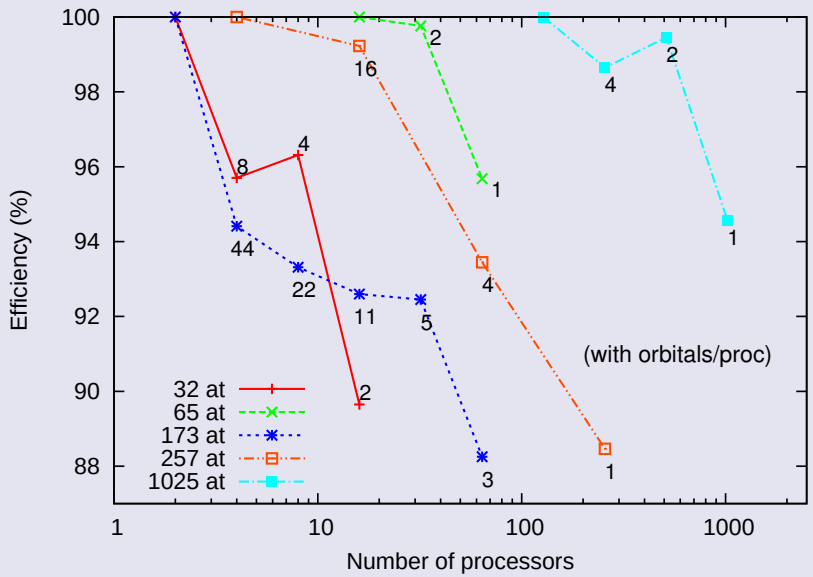
Principal code operations can be intensively optimised

Optimal for application on supercomputers

Little communication, big packets of data

- No need of fast network
- Optimal speedup

Efficiency of the order of 90%, up to thousands of processors



Data repartition optimal for material accelerators (GPU)

Graphic Processing Units can be used to speed up the computation

Separable convolutions

We must calculate

$$F(I_1, I_2, I_3) = \sum_{j_1, j_2, j_3 = 0}^{L} h_{j_1} h_{j_2} h_{j_3} G(I_1 - j_1, I_2 - j_2, I_3 - j_3)$$

$$= \sum_{j_1=0}^{L} h_{j_1} \sum_{j_2=0}^{L} h_{j_2} \sum_{j_3=0}^{L} h_{j_3} G(i_1 - j_1, i_2 - j_2, i_3 - j_3)$$

Application of three successive operations

1. $A_3(I_3, i_1, i_2) = \sum_j h_j G(i_1, i_2, I_3 - j) \qquad \forall i_1, i_2;$
2. $A_2(I_2, I_3, i_1) = \sum_j h_j A_3(I_3, i_1, I_2 - j) \qquad \forall I_3, i_1;$
3. $F(I_1, I_2, i_3) = \sum_j h_j A_2(I_2, I_3, I_1 - j) \qquad \forall I_2, I_3.$

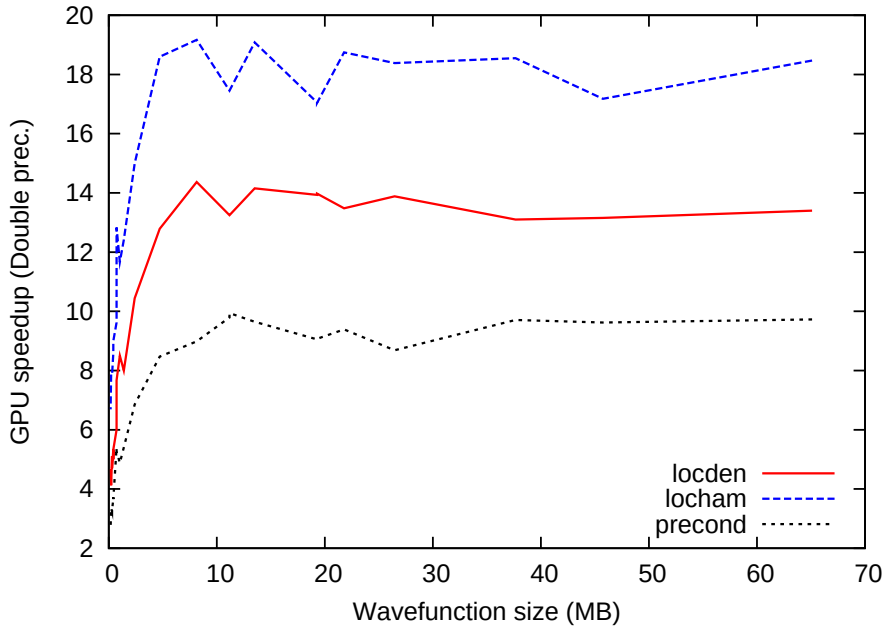
Main routine: Convolution + transposition

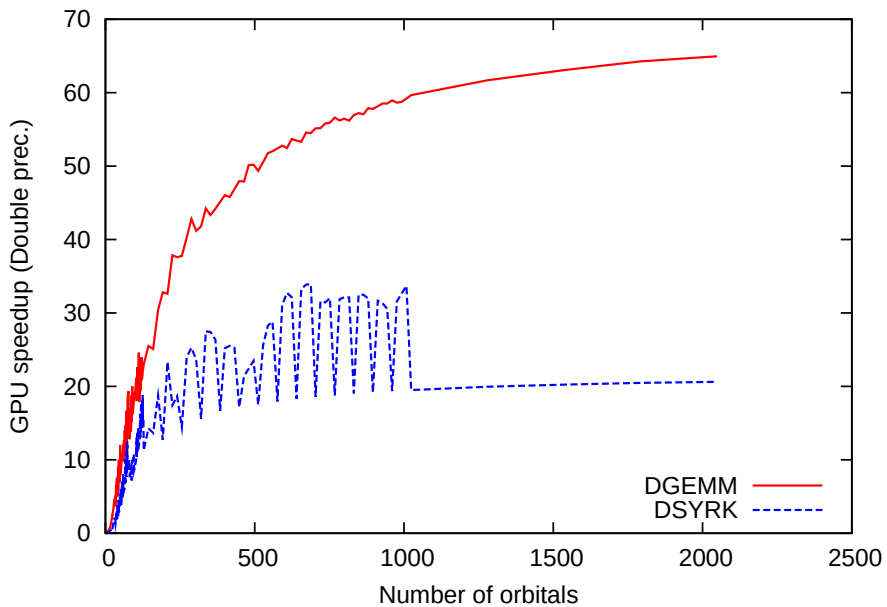
$$F(I, a) = \sum_j h_j G(a, I - j) \qquad \forall a\,;$$

GPU-ported operations in BigDFT (double precision)

Convolutions (CUDA rewritten)

GPU speedups between 10 and 20 can be obtained for different sections



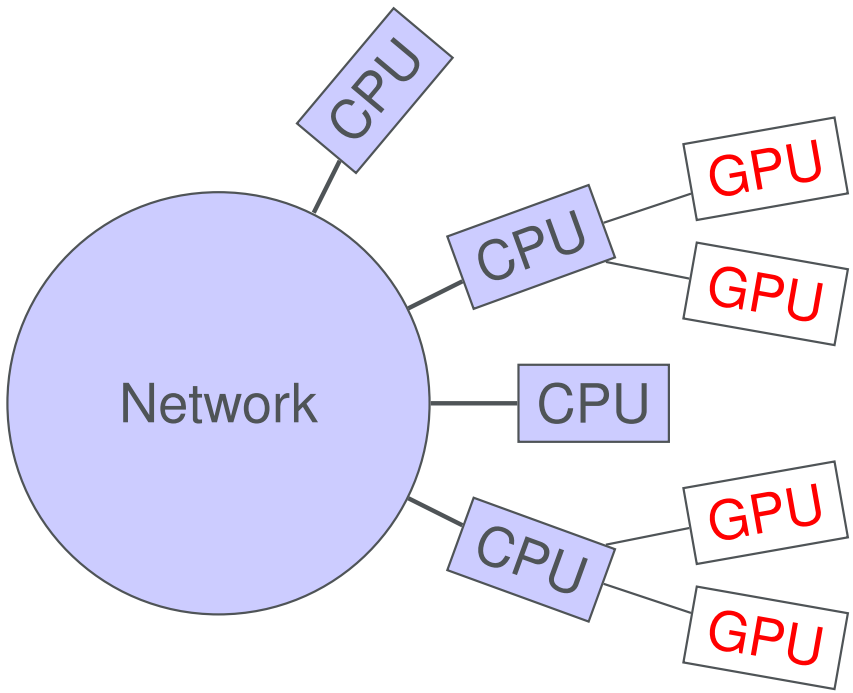


Linear algebra (CUBLAS library)

The interfacing with CUBLAS is immediate, with considerable speedups

Distribute the data on hybrid supercomputer

BigDFT should be executed on hybrid CPU-GPU architectures

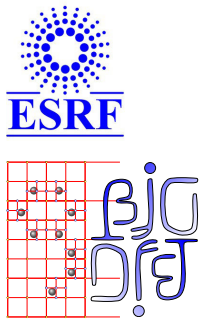


Data transfer is still MPI-based

Only internode communication between GPU and CPU

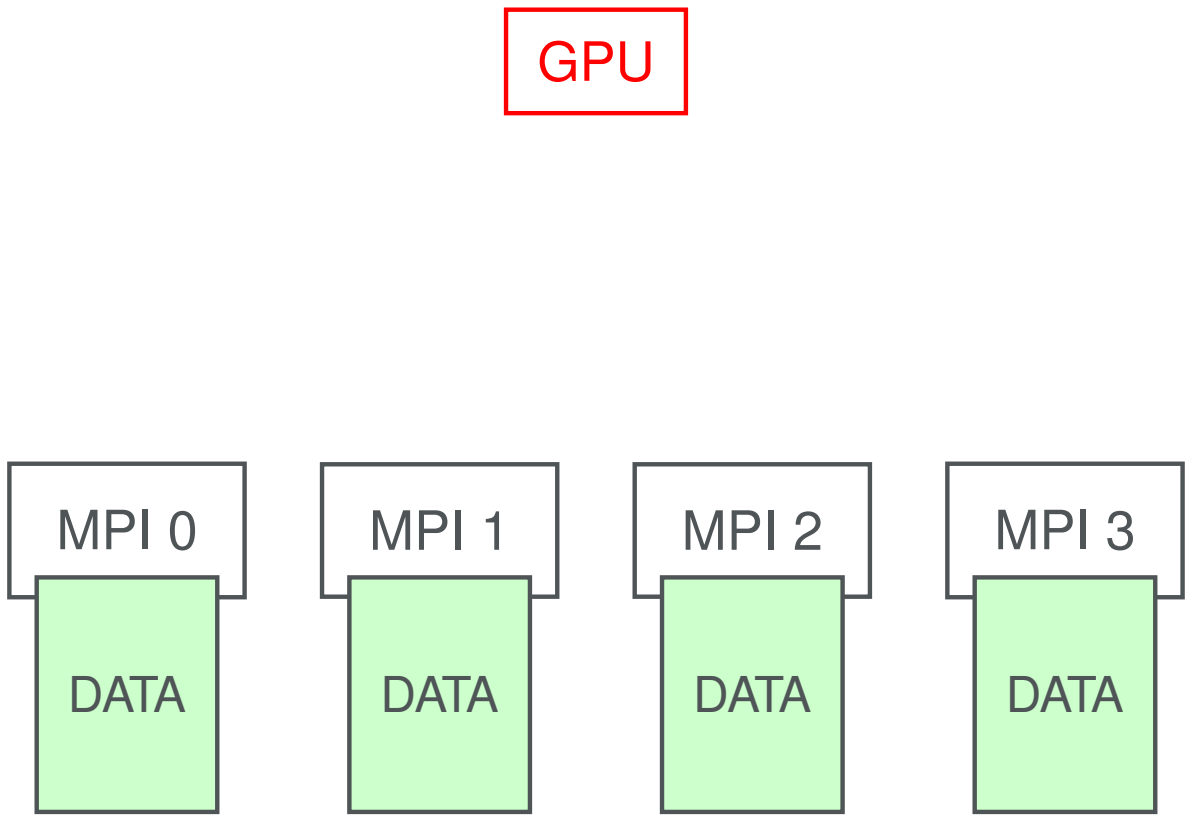
Data distribution should depend on the presence of GPUs on the nodes

Data repartition on a node

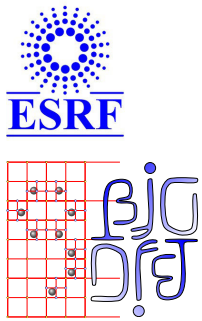


Non-hybrid case

GPU not used $\longrightarrow$ homogeneous repartition

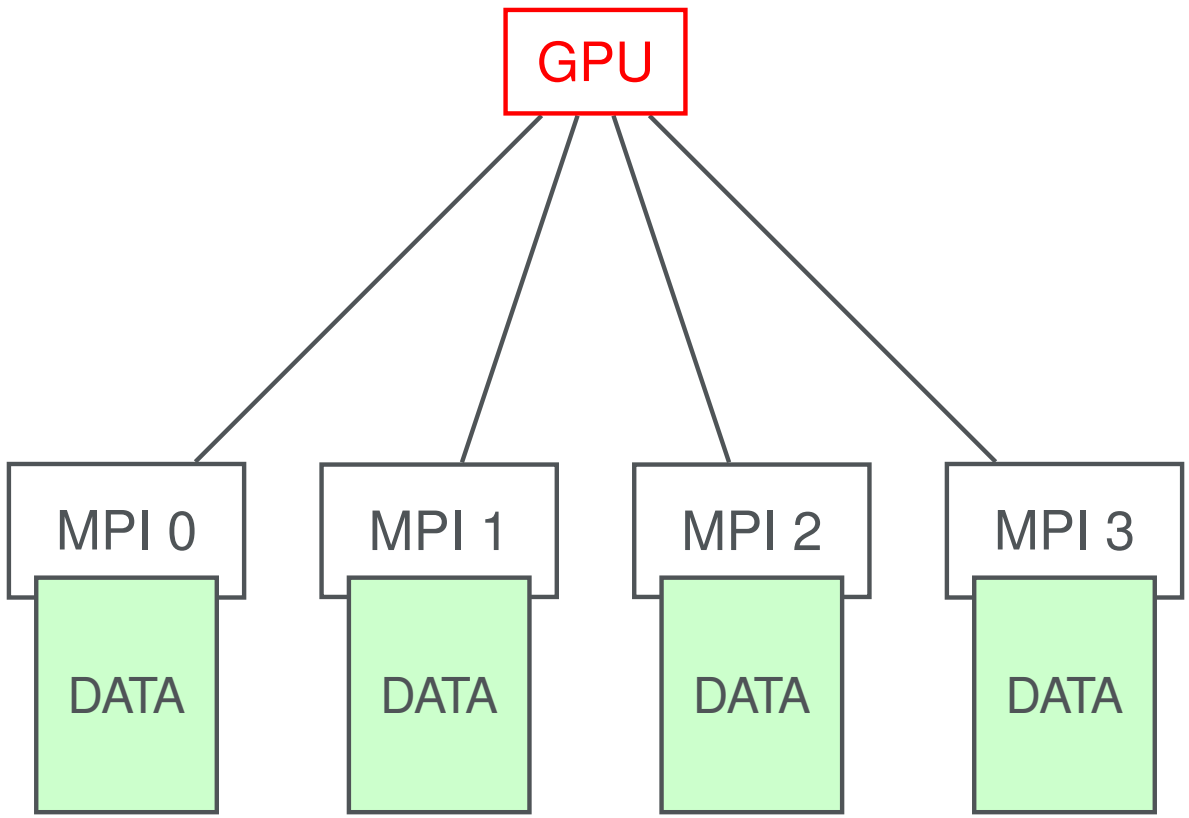


Data repartition on a node



"Naive" repartition

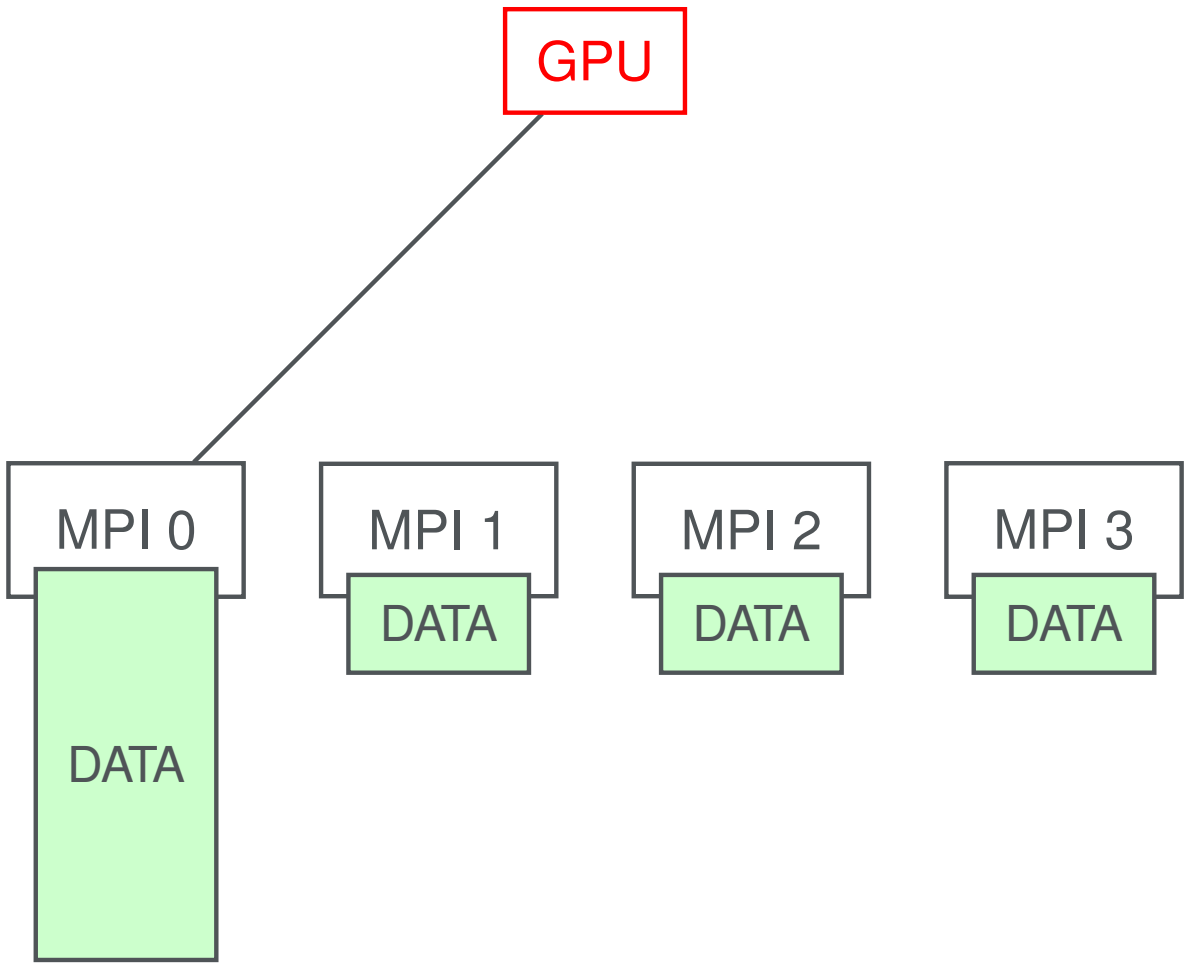
All the cores use the GPU at the same time



Data repartition on a node

Inhomogeneous repartition

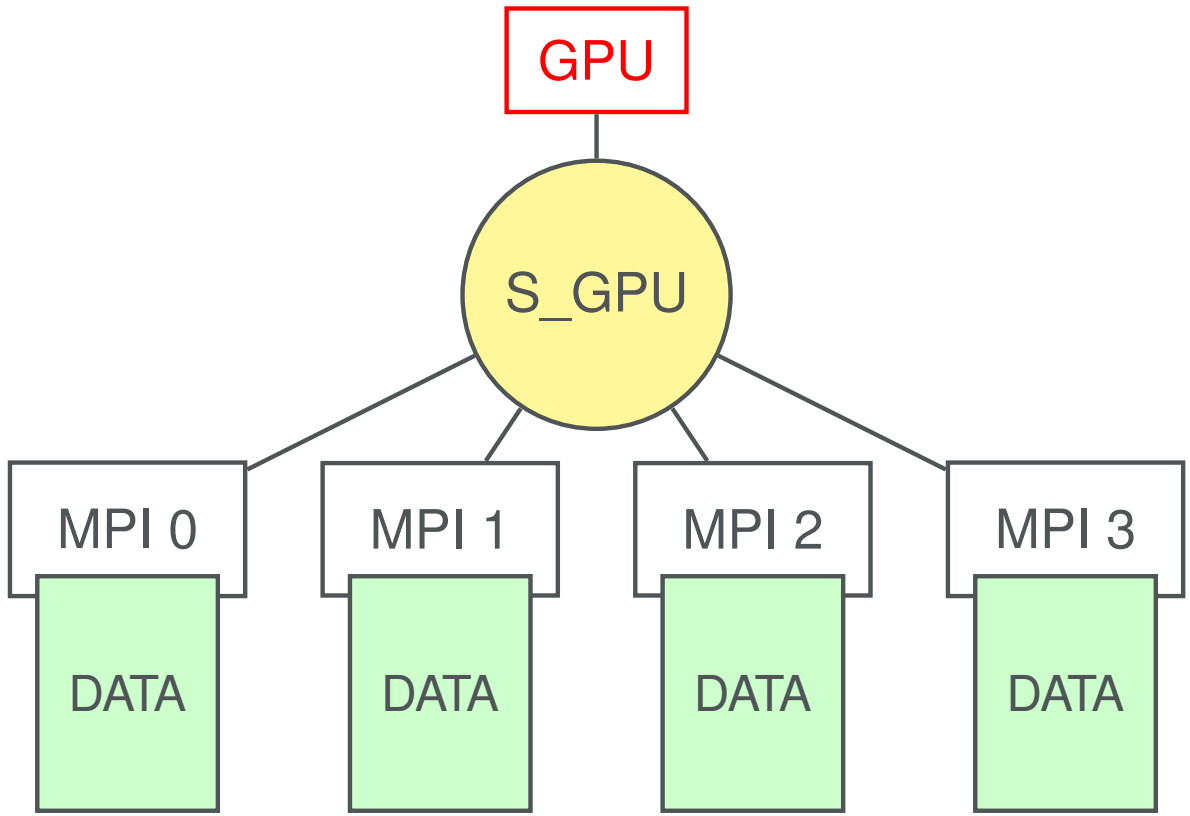
Only one node use the GPU with more data



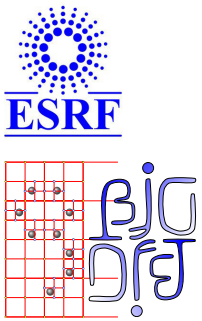
Data repartition on a node

The S_GPU approach

S_GPU library manages GPU resource within the node



S_GPU library (M. Ospici, LIG / Bull / CEA)



GPU and Wavelets

De-synchronisation of operations

Two semaphores are activated for each card on the node:

- Data transfer (CPU $\longrightarrow$ GPU and GPU $\longrightarrow$ CPU)
- Calculation on the GPU

Each operation (e.g. convolution of a wavefunction) is associated to a stream.

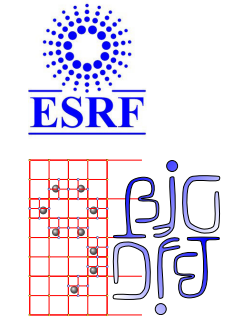
Operation overlap

Calculation and data transfer of different stream may overlap
Operation are scheduled on a first come - first served basis

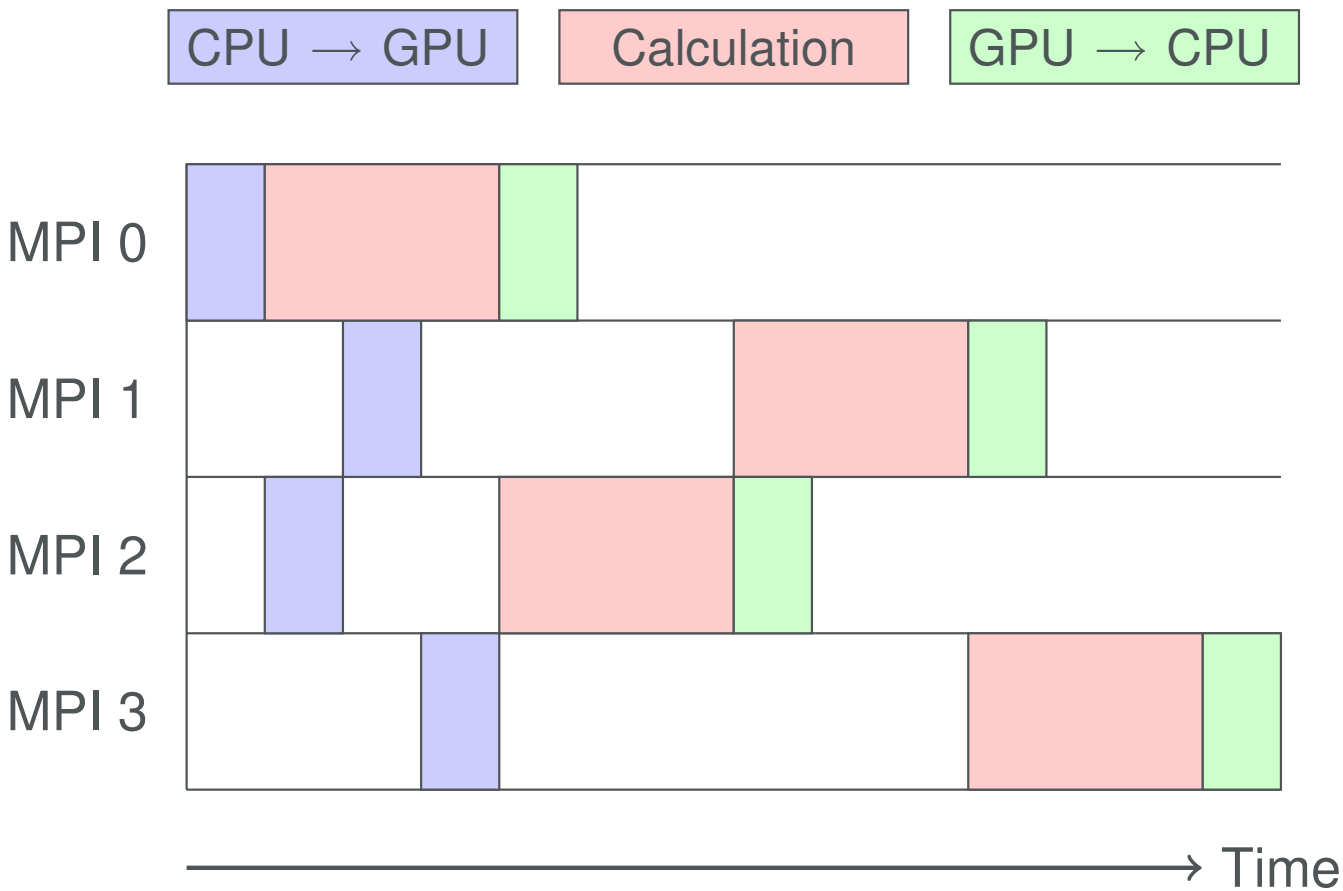
Several advantages

- The time for memory transfers is saved
- Heavy calculation can be passed to the card one - by - one, avoiding scheduling problems

Example of a time chart



The GPU can be viewed as a shared co-processor



Convenince of S_GPU approach

Different tests thanks to BigDFT flexibility

We have performed many tests, with different ratios GPU/CPU on the same node

Speedup on the full code (exemples)

S_GPU is the best compromise speedup/easiness

Examples:

CPU -GPU	8 - 1	8 - 2	4-2	2-2
S_GPU	1.96	3.69	3.73	5.09
Inhomogeneous (best)	2.08	2.64	2.32	2.40

Full code tested on Multi-GPU platforms

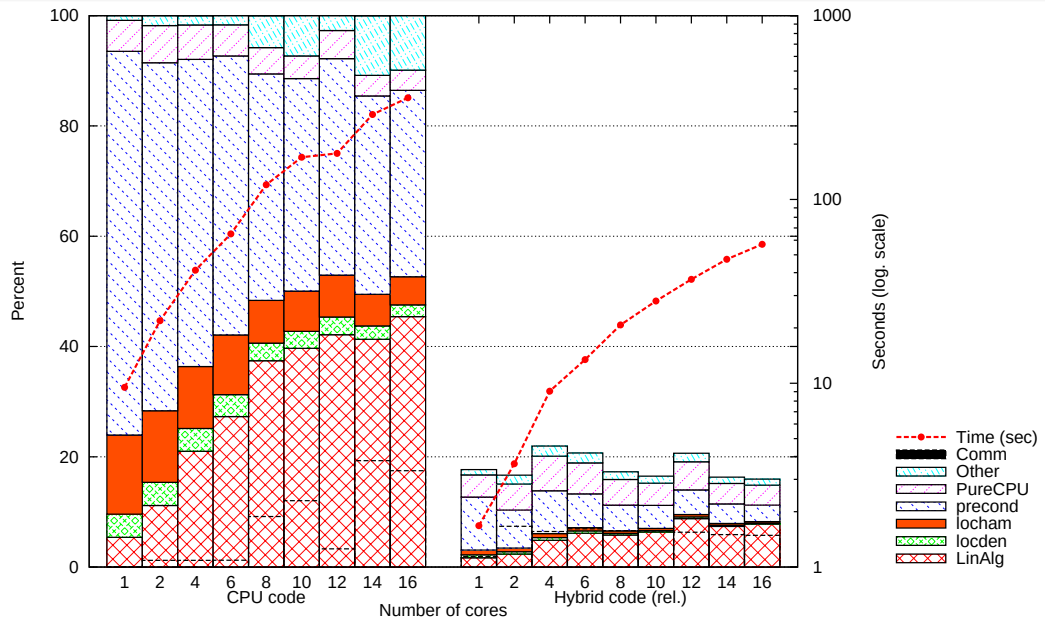
- CINES -Iblis
  48 GPU, Prototype calculations
- CCRT - Titane
  Up to 196 GPU (Grand challenge 2009)

BigDFT code on Hybrid architectures

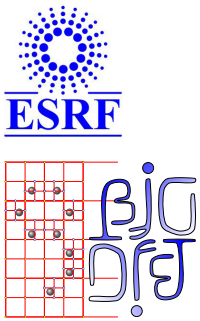
BigDFT code can run on hybrid CPU/GPU supercomputers
In multi-GPU environments, double precision calculations

No Hot-spot operations

Different code sections can be ported on GPU
up to 20x speedup for some operations,
7x for the full parallel code (under improvements)



BigDFT code on Hybrid architectures



BigDFT code can run on hybrid CPU/GPU supercomputers
In multi-GPU environments, double precision calculations

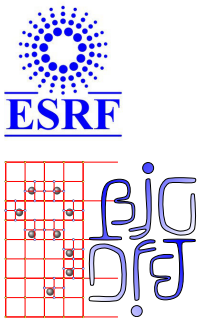
No Hot-spot operations

Different code sections can be ported on GPU
up to 20x speedup for some operations,
7x for the full parallel code (under improvements)

LG: Prix Bull-Fourier 2009 (Bull-GENCI)

Reference Paper: LG *et al.*, J. Chem. Phys. **131**, 034103 (2009)
Grand Challenge 2009 on CEA Hybrid Cluster “Titane” (192 GPU, 768 Intel Xeon Nehalem cores)

Conclusions



GPU and Wavelets

BigDFT code: a modern approach for nanosciences

- ✔ Flexible, reliable formalism (wavelet properties)
- ✔ Conceived for massive parallel architecture
- ✔ Open a path toward the diffusion of Hybrid architectures

The S_GPU library (M. Ospici)

- ✔ Share efficiently the GPU resource between the cores
- ✔ Works efficiently on massive supercomputers
- ✔ Can be generalised for other applications

BigDFT 1.3 – GNU-GPL license

Lots of applications & developments with BigDFT team:
D. Caliste, T. Deutsch (L_Sim - CEA INAC Grenoble)
S. Goedecker (U. Basel)
M. Ospici, J-F. Méhaut (LIG INRIA UJF Bull Grenoble)