Summer School CEA-EDF-INRIA

Toward petaflop numerical simulation on parallel hybrid architectures

INRIA CENTRE DE RECHERCHE SOPHIA ANTIPOLIS, FRANCE

Wavelet-Based DFT calculations on Massively Parallel Hybrid Architectures

Luigi Genovese

L_Sim – CEA Grenoble

June 9, 2011
1. Review of Atomistic Simulations
   - Density Functional Theory
   - Ab initio codes

2. The BigDFT project
   - Formalism and properties
   - The needs for hybrid DFT codes
   - Main operations, parallelisation

3. Performance evaluation
   - Evaluating GPU gain
   - Practical cases

4. Concrete examples
   - Messages
Review of Atomistic Simulations

A interdisciplinary domain
- Theory – Experiment – Simulation
- Hardware – Computers
- Algorithms

Different Atomistic Simulations
- Force fields (interatomic potentials)
- Tight Binding Methods
- Hartree-Fock
- Density Functional Theory
- Configuration interactions
- Quantum Monte-Carlo
Quantum mechanics for many particle systems

Can we do quantum mechanics on systems of many atoms?

Decoupling of the nuclei and electron dynamics

Born-Oppenheimer approximation:
The position of the nuclei can be considered as fixed, obtaining the potential “felt” by the electrons

\[ V_{\text{ext}}(r, \{R_1, \cdots, R_n\}) = - \sum_{a=1}^{n} \frac{Z_a}{|r - R_a|} \]

Electronic Schrödinger equation

The system properties are described by the ground state wavefunction \( \psi(r_1, \cdots, r_N) \), which solves Schrödinger equation

\[ H[\{R\}]\psi = E\psi \]

The quantum hamiltonian depends on the set of the atomic positions \( \{R\} \).
**Atomistic Simulations**

**Two intrinsic difficulties** for numerical atomistic simulations, related to complexity:

- **Interactions** The way that atoms interact is known:
  \[ i \hbar \frac{\partial \psi}{\partial t} = \mathcal{H} \psi \]
  \[ \mathcal{H} \psi = E_0 \psi \]

- **Exploration of the configuration space**
Atomistic Simulations

Two intrinsic difficulties for numerical atomistic simulations, related to complexity:

- **Interactions** The way that atoms interact is known:
  \[ i \hbar \frac{\partial \psi}{\partial t} = \mathcal{H} \psi \quad \mathcal{H} \psi = E_0 \psi \]

- **Exploration of the configuration space**

  ![Graph illustrating potential energy vs. configuration space](image)
Two intrinsic difficulties for numerical atomistic simulations, related to complexity:

- **Interactions** The way that atoms interact is known:
  \[ i \hbar \frac{\partial \psi}{\partial t} = \mathcal{H} \psi \quad \mathcal{H} \psi = E_0 \psi \]

- **Exploration of the configuration space**
Two intrinsic difficulties for numerical atomistic simulations, related to complexity:

- **Interactions**  The way that atoms interact is known:
  \[ i \hbar \frac{\partial \psi}{\partial t} = \mathcal{H} \psi \quad \mathcal{H}\psi = E_0 \psi \]

- **Exploration of the configuration space**
Atomistic Simulations

Two intrinsic difficulties for numerical atomistic simulations, related to complexity:

- **Interactions** The way that atoms interact is known:
  \[ i \hbar \frac{\partial \psi}{\partial t} = \mathcal{H} \psi \quad \mathcal{H} \psi = E_0 \psi \]

- **Exploration of the configuration space**
Choice of Atomistic Methods

3 criteria
1 → Generality (elements, alloys)
2 → Precision ($\Delta r$, $\Delta E$)
3 → System size ($N$, $\Delta t$)

Chemistry and Physics

- Force Fields
  - Tight Binding
  - Hartree-Fock
  - *DFT*
  - Conf. Inter.
  - Quantum Monte-Carlo

---

Laboratoire de Simulation Atomistique | http://inac.cea.fr/L_Sim | Luigi Genovese
Choice of Atomistic Methods

3 criteria
1 → Generality (elements, alloys)
2 → Precision ($\Delta r$, $\Delta E$)
3 → System size ($N$, $\Delta t$)

Chemistry and Physics
- Force Fields
- Tight Binding
- Hartree-Fock
- DFT
- Conf. Inter.
- Quantum Monte-Carlo
Choice of Atomistic Methods

3 criteria
1 → Generality (elements, alloys)
2 → Precision (\(\Delta r, \Delta E\))
3 → System size (\(N, \Delta t\))

Chemistry and Physics

- Force Fields
- Tight Binding
- Hartree-Fock
- DFT
- Conf. Inter.
- Quantum Monte-Carlo
Choice of Atomistic Methods

3 criteria
1 → Generality (elements, alloys)
2 → Precision ($\Delta r$, $\Delta E$)
3 → System size ($N$, $\Delta t$)

Chemistry and Physics

- Force Fields
- Tight Binding
- Hartree-Fock
- \textit{DFT}
- Conf. Inter.
- Quantum Monte-Carlo
Choice of Atomistic Methods

3 criteria
1 → Generality (elements, alloys)
2 → Precision ($\Delta r, \Delta E$)
3 → System size ($N, \Delta t$)

Chemistry and Physics
- Force Fields
- Tight Binding
- Hartree-Fock
- DFT
- Conf. Inter.
- Quantum Monte-Carlo
Choice of Atomistic Methods

3 criteria
1. Generality (elements, alloys)
2. Precision ($\Delta r, \Delta E$)
3. System size ($N, \Delta t$)

Chemistry and Physics
- Force Fields
- Tight Binding
- Hartree-Fock
- $DFT$
- Conf. Inter.
- Quantum Monte-Carlo
The Hohenberg-Kohn theorem

A tremendous numerical problem

\[ H = \sum_{i=1}^{N} -\frac{1}{2} \nabla_{r_i}^2 + V_{\text{ext}}(r_i, \{ R \}) + \frac{1}{2} \sum_{i \neq j} \frac{1}{|r_i - r_j|} \]

The Schrödinger is very difficult to solve for more than two electrons! Another approach is imperative

The fundamental variable of the problem is however not the wavefunction, but the electronic density

\[ \rho(r) = N \int dr_2 \cdots dr_N \psi^*(r, r_2, \cdots, r_N) \psi(r, r_2, \cdots, r_N) \]

Hohenberg-Kohn theorem (1964)

The ground state density \( \rho(r) \) of a many-electron system uniquely determines (up to a constant) the external potential. The external potential is a functional of the density

\[ V_{\text{ext}} = V_{\text{ext}}[\rho] \]
The Kohn-Sham approximation

Given the H-K theorem, it turns out that the total electronic energy is an unknown functional of the density

\[ E = E[\rho] \quad \rightarrow \quad \text{Density Functional Theory} \]

\section*{DFT (Kohn-Sham approach)}

Mapping of a interacting many-electron system into a system with independent particles moving into an effective potential.

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

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

\[
\rho(\mathbf{r}) = 2 \sum_{i=1}^{N/2} \psi_i^*(\mathbf{r}) \psi_i(\mathbf{r}) \quad \nabla^2 V_H(\mathbf{r}) = -4\pi \rho(\mathbf{r})
\]
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

Ab initio codes

Laboratoire de Simulation Atomistique  http://inac.cea.fr/L_Sim

Luigi Genovese
Performing a DFT calculation

A self-consistent equation

Then $\rho = 2 \sum_i \langle \psi_i | \psi_i \rangle$, where $| \psi_i \rangle$ satisfies

$$\left( \frac{1}{2} \nabla^2 + V_H[\rho] + V_{xc}[\rho] + V_{ext} + V_{pseudo} \right) | \psi_i \rangle = \sum_j \Lambda_{i,j} | \psi_j \rangle ,$$

Now in practice: implementing a DFT code

(Kohn-Sham) DFT “Ingredients”

- An XC potential, functional of the density
  several approximations exists (LDA, GGA, . . . )
- A choice of the pseudopotential (if not all-electrons)
  (norm conserving, ultrasoft, PAW, . . . )
- An (iterative) algorithm for finding the wavefunctions $| \psi_i \rangle$
- A basis set for expressing the $| \psi_i \rangle$
- A (good) computer . . .
KS Equations: Self-Consistent Field

Set of self-consistent equations:

\[ \psi_i = \varepsilon_i \psi_i \]

with an effective potential:

\[ V_{\text{eff}}(r) = V_{\text{ext}}(r) + \int_V dr' \frac{\rho(r')}{|r - r'|} \]

Hartree

and:

\[ \rho(r) = \sum_i f_i |\psi_i(r)|^2 \]

exchange−correlation

Hamiltonian (H)

\[ \left\{ -\frac{\hbar^2}{2 m_e} \nabla^2 + V_{\text{eff}} \right\} \psi_i = \varepsilon_i \psi_i \]
KS Equations: Self-Consistent Field

Hamiltonian (H)
\[ \begin{cases} -\frac{1}{2} \frac{\hbar^2}{m_e} \nabla^2 + V_{\text{eff}} \end{cases} \]
\[ \psi_i = \varepsilon_i \psi_i \]

Set of self-consistent equations:
\[ V_{\text{eff}}(r) = V_{\text{ext}}(r) + \int_{V} \frac{\rho(r')}{|r - r'|} \text{Hartree} \]
\[ + \frac{\delta E_{xc}}{\delta \rho(r)} \text{ exchange-correlation} \]

and:
\[ \rho(r) = \sum_i f_i |\psi_i(r)|^2 \]

Poisson Equation:
\[ \Delta V_{\text{Hartree}} = \rho \] (Laplacian: \[ \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \])

Real Mesh (100^3 = 10^6): 10^6 \times 10^6 = 10^{12} evaluations!
KS Equations: Self-Consistent Field

Set of self-consistent equations:

\[ V_{\text{eff}}(r) = V_{\text{ext}}(r) + \int_{\mathcal{V}} \frac{\rho(r')}{|r - r'|} \text{d}r' \]

with an effective potential:

\[ \psi_i = \varepsilon_i \psi_i \]

\[ \text{Hamiltonian } (H) \]

\[ \left\{ -\frac{1}{2m_e} \nabla^2 + V_{\text{eff}} \right\} \psi_i = \varepsilon_i \psi_i \]

Poisson Equation:

\[ \nabla^2 V_{\text{Hartree}} = \rho \]

(Laplacian: \( \Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \))

Real Mesh (\(100^3 = 10^6\)): \(10^6 \times 10^6 = 10^{12}\) evaluations!
KS Equations: Self-Consistent Field

Set of self-consistent equations:

\[ V_{\text{eff}}(r) = V_{\text{ext}}(r) + \int dr' \frac{\rho(r')}{|r - r'|} \]

\[ \Psi_i = \varepsilon_i \Psi_i \]

Hamiltonian \((H)\):

\[ \left\{ -\frac{\hbar^2}{2 m_e} \nabla^2 + V_{\text{eff}} \right\} \Psi_i = \varepsilon_i \Psi_i \]

Poisson Equation:

\[ \Delta V_{\text{Hartree}} = \rho \]

\(\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \)

Real Mesh \((100^3 = 10^6)\): \(10^6 \times 10^6 = 10^{12}\) evaluations!
KS Equations: Self-Consistent Field

Set of self-consistent equations:

\[ V_{\text{eff}}(r) = V_{\text{ext}}(r) + \int_{\mathcal{V}} \frac{\rho(r')}{|r-r'|} \, dr' \]

with an effective potential:

\[ \psi_i = \epsilon_i \psi_i \]

Hamiltonian (H)

\[ \left\{ -\frac{\hbar^2}{2 m_e} \nabla^2 + V_{\text{eff}} \right\} \psi_i = \epsilon_i \psi_i \]

Poisson Equation:

\[ \Delta V_{\text{Hartree}} = \rho \]

Real Mesh (100^3 = 10^6): 10^6 \times 10^6 = 10^{12} \text{ evaluations!}
KS Equations: Self-Consistent Field

Set of self-consistent equations:

$$V_{\text{eff}}(r) = V_{\text{ext}}(r) + \int_{V} \frac{\rho(r')}{|r - r'|} \text{Hartree}$$

with an effective potential:

$$\psi_i = \varepsilon_i \psi_i$$

Hamiltonian ($H$)

$$\left\{ -\frac{\hbar^2}{2m_e} \nabla^2 + V_{\text{eff}} \right\}$$

Poisson Equation:

$$\Delta V_{\text{Hartree}} = \rho$$

(Laplacian: $$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$$)

Real Mesh ($100^3 = 10^6$): $10^6 \times 10^6 = 10^{12}$ evaluations!
Performing a DFT calculation (KS formalism)

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

$$E = \sum_i \langle \Psi_i | H[\rho] | \Psi_i \rangle$$

with:

- $i = 1, \cdots, N$ (one $\Psi$ per electron)
- $\rho(r) = \sum_i \psi_i^*(r) \psi_i(r)$

(Kohn-Sham) DFT “Actors”

- A set of wavefunctions $|\psi_i\rangle$, one for each electron
- A computational approach on a finite basis
  - One array for each $\Psi_i$
  - A set of computational operations on these arrays which depend on the basis set
- A (even more) good computer...
## Basis sets for electronic structure calculation

Several basis sets exist, with different features:

<table>
<thead>
<tr>
<th><strong>Plane Waves</strong></th>
<th><strong>Gaussians, Slater Orbitals</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>ABINIT, CPMD, VASP,…</td>
<td>CP2K, Gaussian, AIMPROM,…</td>
</tr>
<tr>
<td>Systematic convergence.</td>
<td>Real space localized</td>
</tr>
<tr>
<td>✅ Accuracy increases with the number of basis elements</td>
<td>✅ Small number of basis functions for moderate accuracy</td>
</tr>
<tr>
<td>✅ Non-localised, optimal for periodic, homogeneous systems</td>
<td>✅ Well suited for molecules and other open structures</td>
</tr>
<tr>
<td>❌ Non adaptive</td>
<td>❌ Non systematic</td>
</tr>
</tbody>
</table>

### Basis Sets

- **Plane Waves**
  - **Pros**:
    - Systematic convergence.
    - Accuracy increases with the number of basis elements.
    - Non-localised, optimal for periodic, homogeneous systems.
  - **Cons**:
    - Non-adaptive.

- **Gaussians, Slater Orbitals**
  - **Pros**:
    - Small number of basis functions for moderate accuracy.
    - Well suited for molecules and other open structures.
  - **Cons**:
    - Non-systematic.

### FFT

- Robust, Easy to parallelise

### Analytic functions

- Kinetic and overlap matrices can be calculated analytically
List of *ab initio* Codes

- **Plane Waves**
  - ABINIT — Louvain-la-Neuve — http://www.abinit.org
  - CPMD — Zurich, Lugano — http://www.cpmd.org
  - PWSCF — Italy — http://www.pwscf.org
  - VASP — Vienna — http://cms.mpi.univie.ac.at/vasp

- **Gaussian**
  - Gaussian — http://www.gaussian.com
  - DeMon — http://www.demon-software.com
  - CP2K — http://cp2k.berlios.de

- **Numerical-like basis sets**
  - Wien2K — Vienna — http://www.wien2k.at (FPLAPW, all electrons)

- **Real space basis set**
  - ONETEP — http://www.onetep.soton.ac.uk
  - BigDFT — http://inac.cea.fr/L_Sim/BigDFT
A basis for nanosciences: the BigDFT project

STREP European project: BigDFT(2005-2008)

Four partners, 15 contributors:
CEA-INAC Grenoble (T.Deutsch), U. Basel (S.Goedecker),
U. Louvain-la-Neuve (X.Gonze), U. Kiel (R.Schneider)

Aim: To develop an ab-initio DFT code
based on Daubechies Wavelets, to be
integrated in ABINIT.
BigDFT 1.0 → January 2008

...why have we done this? Was it worth it?

- Test the potential advantages of a new formalism
- A lot of outcomes and interesting results
- A lot can be done starting from present know-how
A DFT code based on Daubechies wavelets

**BigDFT: a PSP Kohn-Sham code**

- Systematic, Orthogonal
- Localised, Adaptive
- Kohn-Sham operators are analytic

**Properties**

- A Daubechies wavelets basis has unique properties for DFT usage
- Short, Separable convolutions
  \[ \tilde{c}_\ell = \sum_j a_j c_{\ell - j} \]
- Peculiar numerical properties

**Real space based, highly flexible**

- Big & inhomogeneous systems

**Daubechies Wavelets**

- [Graph showing Daubechies wavelets φ(x) and ψ(x)]

---

Laboratoire de Simulation Atomistique
http://inac.cea.fr/L_Sim

Luigi Genovese
Wavelet properties: multi-resolution

Example of two resolution levels: step function (Haar Wavelet)

**Scaling Functions**: Multi-Resolution basis

Low and High resolution functions related each other

\[
\begin{align*}
\text{Low resolution functions} & = \text{Scaling Functions} + \text{Wavelets} \\
& = \frac{1}{2} \sum \text{Low resolution functions} + \text{Wavelets}
\end{align*}
\]

**Wavelets**: complete the low resolution description

Defined on the same grid as the low resolution functions

\[
\begin{align*}
\text{Scaling Function} + \text{Wavelet} & = \text{High resolution} \\
& = \frac{1}{2} \sum \text{Low resolution functions} + \text{Wavelets}
\end{align*}
\]

We increase the resolution without modifying the grid spacing
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
## Basis set features

<table>
<thead>
<tr>
<th>BigDFT features in a nutshell</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓ Arbitrary absolute <strong>precision</strong> can be achieved</td>
</tr>
<tr>
<td>✓ Good convergence ratio for real-space approach ( O(h^{14}) )</td>
</tr>
</tbody>
</table>
| ✓ Optimal usage of the degrees of freedom (**adaptivity**)
  Optimal speed for a systematic approach (**less memory**)
| ✓ Hartree potential accurate for **various boundary conditions**
  Free and Surfaces BC Poisson Solver
  (present also in CP2K, ABINIT, OCTOPUS)
| ⚠ Data repartition is suitable for optimal scalability
  Simple communications paradigm, **multi-level parallelisation** possible (and implemented)

## Improve and develop know-how

Optimal for *advanced* DFT functionalities in **HPC framework**
### BigDFT version 1.5.2: (ABINIT-related) capabilities

**http://inac.cea.fr/L_Sim/BigDFT**

- Isolated, surfaces and 3D-periodic boundary conditions (k-points, symmetries)
- All XC functionals of the ABINIT package (libXC library)
- Hybrid functionals, Fock exchange operator
- Direct Minimisation and Mixing routines (metals)
- Local geometry optimizations (with constraints)
- External electric fields (surfaces BC)
- Born-Oppenheimer MD, ESTF-IO
- Vibrations
- Unoccupied states
- Empirical van der Waals interactions
- Saddle point searches (NEB, Granot & Bear)

All these functionalities are GPU-compatible.
Operations performed

The SCF cycle

Orbital scheme:
- Hamiltonian
- Preconditioner

Coefficient Scheme:
- Overlap matrices
- Orthogonalisation

Comput. operations
- Convolutions
- BLAS routines
- FFT (Poisson Solver)

Why not GPUs?

---

BigDFT and GPU

Atomistic Simulations
DFT
Ab initio codes

BigDFT Properties

BigDFT and HPC
GPU
Practical cases

Discussion
Messages

BigDFT and GPUs
Code details

Laboratoire de Simulation Atomistique
http://inac.cea.fr/L_Sim

Luigi Genovese
Hybrid Supercomputing nowadays

GPGPU on Supercomputers
- Traditional architectures are somehow saturating
  More cores/node, memories (slightly) larger but not faster
- Architectures of Supercomputers are becoming hybrid
  3 out to 4 Top Supercomputers are hybrid machines
- Extrapolation: In 2015, No. 500 will become petaflopic
  Likely it will be a hybrid machine

Codes should be conceived differently
- # MPI processes is limited for a fixed problem size
- Performances increase only by enhancing parallelism
- Further parallelisation levels should be added (OpenMP, GPU)

Does electronic structure calculations codes are suitable?
How far is petaflop (for DFT)?

At present, with traditional architectures

Routinely used DFT calculations are:

- Few dozens (hundreds) of processors
- Parallel intensive operations (blocking communications, 60-70 percent efficiency)
- Not freshly optimised (legacy codes, monster codes)

Optimistic estimation: 5 GFlop/s per core \times 2000 cores \times 0.9 = 9 TFlop/s = 200 times less than Top 500’s #1!

It is such as

Distance Earth-Moon = 384 Mm
Distance Earth-Mars = 78.4 Gm = 200 times more

Are we able to go to Mars? (…in 2015?)
The vessel: Ambitions from BigDFT experience

Reliable formalism
- Systematic convergence properties
- Explicit environments, analytic operator expressions

State-of-the-art computational technology
- Data locality optimal for operator applications
- Massive parallel environments
- Material accelerators (GPU)

New physics can be approached
- Enhanced functionalities can be applied relatively easily
- Limitation of DFT approximations can be evidenced
- A formalism of interest for Post-DFT treatments

Laboratoire de Simulation Atomistique  http://inac.cea.fr/L_Sim  Luigi Genovese
Separable convolutions

We must calculate

\[ F(l_1, l_2, l_3) = \sum_{j_1, j_2, j_3=0}^{L} h_{j_1} h_{j_2} h_{j_3} G(l_1 - j_1, l_2 - j_2, l_3 - j_3) \]

= \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. \( A_3(I_3, i_1, i_2) = \sum_j h_j G(i_1, i_2, I_3 - j) \) \( \forall i_1, i_2 \);
2. \( A_2(I_2, I_3, i_1) = \sum_j h_j A_3(I_3, i_1, I_2 - j) \) \( \forall I_3, i_1 \);
3. \( F(l_1, l_2, l_3) = \sum_j h_j A_2(I_2, I_3, l_1 - j) \) \( \forall l_2, l_3 \).

Main routine: Convolution + transposition

\[ F(l, a) = \sum_j h_j G(a, l - j) \) \( \forall a \);
Basic Input-Output operation

From sequential to GPU

Same operation schedule for monocore, multithread, GPU

Can be treated as a library
Initially, naive FORTRAN routines

\[ y(j, l) = \sum_{\ell=L}^{U} h_{\ell} x(l + \ell, j) \]

- Easy to write and debug
- Test the formalism
- Define reference results

Optimisation can then start (Ex. X5550, 2.67 GHz)

<table>
<thead>
<tr>
<th>Method</th>
<th>GFlop/s</th>
<th>% of peak</th>
<th>SpeedUp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive (FORTRAN)</td>
<td>0.54</td>
<td>5.1</td>
<td>1/(6.25)</td>
</tr>
<tr>
<td>Current (FORTRAN)</td>
<td>3.3</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>Best (C, SSE)</td>
<td>7.8</td>
<td>73</td>
<td>2.3</td>
</tr>
</tbody>
</table>

Laboratoire de Simulation Atomistique  http://inac.cea.fr/L_Sim  Luigi Genovese
How to optimize?

A trade-off between benefit and effort

**FORTRAN based**

- Relatively accessible (loop unrolling)
- Moderate optimisation can be achieved relatively fast
- Compilers fail to use vector engine efficiently

**Push optimisation at the best**

- Only one out of 3 convolution type has been implemented
- About 20 different patterns have been studied for one 1D convolution
- Tedious work, huge code → Maintainability?

👉 Automatic code generation under study
MPI parallelization I: Orbital distribution scheme

Used for the application of the Hamiltonian

Operator approach: The Hamiltonian (convolutions) is applied separately onto each wavefunction

\[ \psi_1 \]
\[ \psi_2 \]
\[ \psi_3 \]
\[ \psi_4 \]
\[ \psi_5 \]
MPI parallelization II: Coefficient distribution scheme

**Used for scalar products & orthonormalisation**

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

<table>
<thead>
<tr>
<th></th>
<th>MPI 0</th>
<th>MPI 1</th>
<th>MPI 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\psi_1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\psi_2)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\psi_3)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\psi_4)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\psi_5)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

At present, MPI\_ALLTOALL(V) is used to switch
OpenMP parallelisation

Innermost parallelisation level

(AAlmost) Any BigDFT operation is parallelised via OpenMP

- Useful for memory demanding calculations
- Allows further increase of speedups
- Saves MPI processes and intra-node Message Passing

- Less efficient than MPI
- Compiler and system dependent
- OMP sections should be regularly maintained

![Graph showing OMP speedup vs. number of MPI processes]

No. of MPI procs

OMP Speedup

0 20 40 60 80 100 120

2OMP

3OMP

6OMP

Laboratoire de Simulation Atomistique  http://inac.cea.fr/L_Sim  Luigi Genovese
What are the ideal conditions for GPU

GPU-ported routines should take the majority of the time

What happens to parallel efficiency?
Parallelisation and architectures

Same code, same runs. Which is the best?

CCRT Titane (Nehalem, Infiniband)  CSCS Rosa (Opteron, Cray XT5)

Titane is 2.3 to 1.6 times faster than Rosa!

Degradation of parallel performances: why?

1. Calculation power has increased more than networking
2. Better libraries (MKL)

Walltime reduced, but lower parallel efficiency

This will always happen while using GPU!
Architectures, libraries, networking

BigDFT and GPU

Atomistic Simulations
DFT
Ab initio codes

BigDFT
Properties
BigDFT and GPUs
Code details

BigDFT and HPC
GPU
Practical cases
Discussion
Messages

A case-by-case study

Consideration are often system-dependent, a thumb rule not always exists.

-know your code!
## Nature of the operations

- Operators approach via convolutions
- Linear Algebra due to orthogonality of the basis
- Communications and calculations do not interfere
- A number of operations which can be accelerated

## Evaluating GPU convenience

### Three levels of evaluation

1. **Bare speedups**: GPU kernels vs. CPU routines
   - Does the operations are suitable for GPU?

2. **Full code speedup on one process**
   - Amdahl’s law: are there hot-spot operations?

3. **Speedup in a (massively?) parallel environment**
   - The MPI layer adds an extra level of complexity

---

Laboratoire de Simulation Atomistique  
[http://inac.cea.fr/L_Sim](http://inac.cea.fr/L_Sim)  
Luigi Genovese
Convolutions and Transposition in GPU

Blocking operations
A work group performs convolutions plus transpositions
Different boundary conditions can be implemented
OpenCL data management

Example of $4 \times 4$ block with a filter of length 5

```
<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0,0</td>
<td>1,0</td>
<td>2,0</td>
<td>3,0</td>
</tr>
<tr>
<td>0,1</td>
<td>1,1</td>
<td>2,1</td>
<td>3,1</td>
</tr>
<tr>
<td>0,2</td>
<td>1,2</td>
<td>2,2</td>
<td>3,2</td>
</tr>
<tr>
<td>0,3</td>
<td>1,3</td>
<td>2,3</td>
<td>3,3</td>
</tr>
<tr>
<td>0,0</td>
<td>1,0</td>
<td>2,0</td>
<td>3,0</td>
</tr>
<tr>
<td>0,1</td>
<td>1,1</td>
<td>2,1</td>
<td>3,1</td>
</tr>
<tr>
<td>0,2</td>
<td>1,2</td>
<td>2,2</td>
<td>3,2</td>
</tr>
<tr>
<td>0,3</td>
<td>1,3</td>
<td>2,3</td>
<td>3,3</td>
</tr>
</tbody>
</table>
```

SIMD implementation

Each work item is associated to a single output element

Convolution buffers $\rightarrow$ data reuse
Convolution kernel with OpenCL

### Comparison with CPU work

(Ex. X5550, 2.67 GHz)

<table>
<thead>
<tr>
<th>Method</th>
<th>GFlop/s</th>
<th>% of peak</th>
<th>SpeedUp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naive (FORTRAN)</td>
<td>0.54</td>
<td>5.1</td>
<td>1/(6.25)</td>
</tr>
<tr>
<td>Current (FORTRAN)</td>
<td>3.3</td>
<td>31</td>
<td>1</td>
</tr>
<tr>
<td>Best (C, SSE)</td>
<td>7.8</td>
<td>73</td>
<td>2.3</td>
</tr>
<tr>
<td>OpenCL (Fermi)</td>
<td>97</td>
<td>20</td>
<td>29 (12.4)</td>
</tr>
</tbody>
</table>

### Very good and promising results

- No need of data transfer for 3D case (chain of 1D kernels)
- Deeper optimisation still to be done (20% of peak)
- Require less manpower than deep CPU optimisation
- Automatic generation will be considered
GPU-ported operations in BigDFT (double precision)

**Convolutions Kernels**
- (OpenCL (re)written)
- Fully functional (all BC)
- Based on the former CUDA version
- **A 10 to 50 speedup**

GPU BigDFT sections
- GPU speedups between 5 and 20, depending of:
  - Wavefunction size
  - **CPU-GPU Architecture**
BigDFT in hybrid codes

Acceleration of the **full** BigDFT code

- Considerable gain may be achieved for suitable systems
- Amdahl’s law should always be considered
- Resources can be used concurrently (OpenCL queues)
  - More MPI processes may share the same card!

![Graph showing speedup for different architectures]

- **Badiane, X5550 + Fermi S2070, ZnO 64 at.**
  - CPU vs. Hybrid

**Laboratoire de Simulation Atomistique**  
[http://inac.cea.fr/L_Sim](http://inac.cea.fr/L_Sim)  
Luigi Genovese
The time-to-solution problem I: Efficiency

Good example: 4 C at, surface BC, 113 Kpts

Parallel efficiency of 98%, convolutions largely dominate.

Node:
2 × Fermi + 8 × Westmere
8 MPI processes

<table>
<thead>
<tr>
<th># GPU added</th>
<th>2</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SpeedUp (SU)</td>
<td>5.3</td>
<td>9.8</td>
<td>11.6</td>
</tr>
<tr>
<td># MPI equiv.</td>
<td>44</td>
<td>80</td>
<td>96</td>
</tr>
<tr>
<td>Acceler. Eff.</td>
<td>1</td>
<td>.94</td>
<td>.56</td>
</tr>
</tbody>
</table>

![Graph showing speedup vs. number of MPI processes]
The time-to-solution problem II: Robustness

Not so good example: A too small system

- CPU efficiency is poor (calculation is too fast)
- Amdahl’s law not favorable (5x SU at most)
- GPU SU is almost independent of the size
- The hybrid code always goes faster
Hybrid and Heterogeneous runs with OpenCL

**NVidia S2070**
Connected each to a Nehalem Workstation
BigDFT may run on both

**ATI HD 6970**

---

**Sample BigDFT run: Graphene, 4 C atoms, 52 kpts**

<table>
<thead>
<tr>
<th></th>
<th>MPI 1</th>
<th>1</th>
<th>4</th>
<th>1</th>
<th>4</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GPU</strong></td>
<td>NO</td>
<td>NV</td>
<td>NV</td>
<td>ATI</td>
<td>ATI</td>
<td>NV + ATI</td>
</tr>
<tr>
<td><strong>Time (s)</strong></td>
<td>6020</td>
<td>300</td>
<td>160</td>
<td>347</td>
<td>197</td>
<td>109</td>
</tr>
<tr>
<td><strong>Speedup</strong></td>
<td>1</td>
<td>20.07</td>
<td>37.62</td>
<td>17.35</td>
<td>30.55</td>
<td>55.23</td>
</tr>
<tr>
<td><strong>GFlop/s</strong></td>
<td>1.34</td>
<td>26.84</td>
<td>50.33</td>
<td>23.2</td>
<td>40.87</td>
<td>73.87</td>
</tr>
</tbody>
</table>

Next Step: handling of Load (un)balancing

---

**BigDFT and GPU**

Atomistic Simulations
DFT
Ab initio codes

**BigDFT Properties**

**BigDFT and GPUs**

**Code details**

**BigDFT and HPC**

GPU

**Practical cases**

**Discussion**

**Messages**

---

Laboratoire de Simulation Atomistique  [http://inac.cea.fr/L_Sim] Luigi Genovese
Concrete examples with BigDFT code

**MD simulation, 32 water molecules, 0.5 fs/step**

<table>
<thead>
<tr>
<th>Mixed MPI/OpenMP BigDFT parallelisation vs. GPU case</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MPI•OMP</strong></td>
</tr>
<tr>
<td><strong>s/SCF</strong></td>
</tr>
<tr>
<td><strong>MD ps/day</strong></td>
</tr>
</tbody>
</table>

**An example: challenging DFT for catalysis**

Multi-scale study for OR mechanism on PEM fuel cells

- Explicit model of $H_2O/Pt$ interface
- Absorption properties, reaction mechanisms

Outcomes from the understanding of catalytic mechanism at atomic scale:

- Conception of new active and selective materials
- Fuels cell ageing, more efficient and durable devices
Scientific directions: Energy conversion

A scientific domain which embodies a number of challenges:

- The quantities are “complicated” (many body effects)
  Study/prediction of fundamental properties: band gaps, band offsets, excited state quantities,…
- Objects are “big” and the environment matters
  Systems react with the surroundings
- Building new modelisation paradigms

How to achieve these objectives?

Short and medium term objectives

- State-of-the art functionalities of DFT for complex environments
- Explore new formalisms for Post DFT treatments
A look in near future: science with HPC DFT codes

A concerted set of actions

- Improve codes functionalities for present-day and next generation supercomputers
- Test and develop new formalisms
- Insert ab-initio codes in new scientific workflows (Multiscale Modelling)

The Mars mission

Is Petaflop performance possible?

- GPU acceleration → one order of magnitude
- Bigger systems, heavier methods → (more than) one order of magnitude bigger

BigDFT experience makes this feasible

An opportunity to achieve important outcomes and know-how
General considerations

Optimisation effort

- Know the code behaviour and features
  Careful performance study of the complete algorithm
- Identify and *make modular* critical sections
  Fundamental for maintainability and architecture evolution
- Optimisation cost: consider *end-user* running conditions
  Robustness is more important than best performance

Performance evaluation know-how

- No general thumb-rule: what means High Performance?
  A multi-criterion evaluation process
- Multi-level parallelisation always to be used
  Your code will not (anymore) become faster via hardware
Conclusions

BigDFT code: a modern approach for nanosciences
- Flexible, reliable formalism (wavelet properties)
- Easily fit with massively parallel architecture
- Open a path toward the diffusion of Hybrid architectures

Messages from GPU experience with BigDFT
- GPU allow a significant reduction of the time-to-solution
- Require a well-structured underlying code which makes multi-level parallelisation possible
- To be taken into account while evaluating performances
  Parallel efficiency ⇐ dimensioning of system wrt architecture

CECAM BigDFT tutorial next October
A tutorial on BigDFT code is scheduled!
Grenoble, 19-21 October 2011
Acknowledgments

CEA Grenoble – Group of Thierry Deutsch
LG, D. Caliste, B. Videau, M. Ospici, I. Duchemin, P. Boulanger, E. Machado-Charry, F. Cinquini, B. Natarajan

Basel University – Group of Stefan Goedecker
S. A. Ghazemi, A. Willand, M. Amsler, S. Mohr, A. Sadeghi, N. Dugan, H. Tran

And other groups
- Montreal University:
  L. Béland, N. Mousseau
- European Synchrotron Radiation Facility:
  Y. Kvashnin, A. Mirone