Introduction to the Qbox code

François Gygi
University of California, Davis
fgygi@ucdavis.edu
http://eslab.ucdavis.edu

Joint NNIN/C & NCN Fall Workshop
Cornell University, Nov 14-16, 2010
Qbox code: main features

- C++/MPI implementation of First-Principles Molecular Dynamics
- DFT/GGA exchange-correlation
- Plane-wave, norm-conserving pseudopotentials
- Designed for large-scale parallel platforms
- Main design constraint: small memory footprint (< 512MB per task)
- Built on optimized parallel libs: PBLAS, ScaLAPACK
- XML interface
- Installed on TeraGrid platforms (Kraken, Ranger, etc.)
Qbox code architecture

- Qbox
  - ScaLAPACK/PBLAS
    - BLAS/MASSV
      - D/ZGEMM lib
    - BLACS
      - MPI
  - Apache XercesC XML lib
  - FFTW lib
Examples of large-scale simulations

- The structure of elemental boron is a complex defected lattice
- DFT calculations coupled with lattice Monte Carlo simulations show that boron is disordered at low temperature
- 1280 atoms, 3840 e⁻


1280-atom supercell of β-rhombohedral boron
Water confined at the nanoscale

- DFT MD simulations of H$_2$O confined in carbon nanotubes and between graphene sheets
- Study the effect of the interface on the hydrogen bond network

Nanoscience application: properties of a $\text{Au}_{102}(p\text{-MBA})_{44}$

- Crystal of thiol-coated Au particles, recently synthesized and characterized by X-Ray diffraction [1]
  - monodisperse Au clusters: $\text{Au}_{102}$
  - Nanoparticles interact through thiol adsorbates via hydrogen bonds


Qbox data flow

Qbox cmd script → Qbox → my_simulation.xml
Qbox basic operation

- Start Qbox in interactive mode (serial version)
  - $ qb
  - Qbox prompt: [qbox]
- Start Qbox reading from an input script
  - $ qb input.i
- Start Qbox using an input script, writing on an output file
  - $ qb input.i > output.r
- output.r is a well-formed XML document
Qbox commands

- Qbox reads commands from input and executes them sequentially
- Examples
  - define the plane wave energy cutoff (Ry)
    [qbox] set ecut 35
  - define an atom at a given position
    [qbox] atom C carbon 0.123 0.456 0.789
    - position in atomic units (Bohr)
### Qbox commands

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>angle</td>
<td>compute the angle formed by three atoms</td>
</tr>
<tr>
<td>atom</td>
<td>define an atom</td>
</tr>
<tr>
<td>compute_mlwf</td>
<td>compute maximally localized Wannier functions</td>
</tr>
<tr>
<td>constraint</td>
<td>manage constraints on atomic positions</td>
</tr>
<tr>
<td>distance</td>
<td>compute the distance between two atoms</td>
</tr>
<tr>
<td>fold_in_ws</td>
<td>fold atoms into the Wigner-Seitz cell</td>
</tr>
<tr>
<td>help</td>
<td>print a short message about the use of a command</td>
</tr>
<tr>
<td>kpoint</td>
<td>add or remove k-points</td>
</tr>
<tr>
<td>list_atoms</td>
<td>print a list of currently defined atoms</td>
</tr>
<tr>
<td>list_species</td>
<td>print a list of currently defined atomic species</td>
</tr>
<tr>
<td>load</td>
<td>load a sample from a file previously saved</td>
</tr>
<tr>
<td>move</td>
<td>move atoms</td>
</tr>
</tbody>
</table>
**Qbox commands**

- **print**: print the value of a Qbox variable
- **quit**: exit Qbox
- **randomize_wf**: add random noise to the wavefunction coefficients
- **reset_vcm**: set the velocity of the center of mass to zero
- **run**: run MD or electronic optimization steps
- **save**: save a sample on a file for later use
- **set**: assign a value to a Qbox variable
- **species**: define a new atomic species
- **status**: print a summary of the current state
- **strain**: impose a strain on the sample
- **torsion**: compute the dihedral angle defined by four atoms
- **!(shell escape)**: execute a shell command
Qbox commands

- List all commands using the “help” command

```
[qbox] help
valid commands are:

<table>
<thead>
<tr>
<th>angle</th>
<th>atom</th>
<th>compute_mlwf</th>
<th>constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>distance</td>
<td>fold_in_ws</td>
<td>help</td>
<td>kpoint</td>
</tr>
<tr>
<td>list_atoms</td>
<td>list_species</td>
<td>load</td>
<td>move</td>
</tr>
<tr>
<td>plot</td>
<td>print</td>
<td>quit</td>
<td>randomize_wf</td>
</tr>
<tr>
<td>reset_vcm</td>
<td>run</td>
<td>save</td>
<td>set</td>
</tr>
<tr>
<td>species</td>
<td>status</td>
<td>strain</td>
<td>torsion</td>
</tr>
</tbody>
</table>
```

[qbox]
Qbox commands

• Get more details using “help <command>”

```bash
[qbox] help move
move

syntax: move atom_name {to|by} x y z

The move command displaces an atom to a new position. The new position is defined by absolute coordinates (to) or by a relative displacement (by).

• A detailed description of all commands is given in the user guide: QboxUserGuide.pdf
Qbox variables

- Qbox variables can be set using the “set” command.
- Variable values are printed using the “print” command
- Examples
  - set the ecut variable
    [qbox] set ecut 35
  - print the value of the ecut variable
    [qbox] print ecut
Qbox variables

- **atoms_dyn**: ionic dynamics control variable
- **cell**: dimensions of the unit cell
- **cell_dyn**: unit cell dynamics control variable
- **cell_lock**: control of allowed unit cell motions
- **cell_mass**: fictitious mass of the unit cell
- **charge_mix_coeff**: mixing coefficient for charge density update
- **charge_mix_ndim**: Anderson dimension for charge density mixing
- **charge_mix_rcut**: Kerker screening for charge density update
- **debug**: debug parameters (not for normal use)
- **dt**: time step (a.u.)
- **ecut**: plane wave energy cutoff (Ry)
- **ecutprec**: energy cutoff of the preconditioner (Ry)
- **ecuts**: energy cutoff of the confinement potential
- **emass**: fictitious electronic mass (for CP dynamics)
Qbox variables

- ext_stress: externally applied stress (GPa)
- fermi_temp: Fermi temperature (K)
- nempty: number of empty states
- net_charge: net charge of the system
- nrowmax: maximum size of process grid columns
- ref_cell: dimensions of the reference unit cell
- stress: stress control variable
- thermostat: thermostat control variable
- th_temp: thermostat temperature (K)
- th_time: thermostat time constant (a.u.)
- th_width: thermostat width (K)
- wf_diag: wavefunction diagonalization control variable
- wf_dyn: wavefunction dynamics control variable
- xc: exchange-correlation control variable
Qbox output: an XML document

```xml
<?xml version="1.0" encoding="UTF-8"?>
<fpmd:simulation xmlns:fpmd="http://www.quantum-simulation.org/ns/fpmd/fpmd-1.0">
  =============================
  I qbox 1.52.2              I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I                          I
  I http://eslab.ucdavis.edu I
  =============================

  <release> 1.52.2 pavane </release>
  <user> fgygi </user>
```
Qbox output: an XML document

...<sysname> Linux </sysname><nodename> pavane.das.ucdavis.edu </nodename><start_time> 2009-07-28T16:42:54Z </start_time><mpi_processes count="1"><process id="0"> pavane.das.ucdavis.edu </process></mpi_processes>[qbox]
Extracting elements from Qbox output: XML parsers

- The “xml_grep” command can be used to extract elements from Qbox output

  $ xml_grep sysname output.r

  <?xml version="1.0" ?>
  <xml_grep version="0.7" date="Tue Jul 28 10:59:52 2009">
  <file filename="output.r">
    <sysname> Linux </sysname>
  </file>
  </xml_grep>

  $ xml_grep --nowrap sysname output.r
  <sysname> Linux </sysname>

- Using XML and XML parsers is safer than using plain text and grep
Extracting elements from Qbox output: XPath syntax

- XPath is a WWW standard for referring to fragments of XML documents

```
$ xml_grep 'atom[@name="Si2"]/position' cg1.r

<?xml version="1.0" ?>
<xml_grep version="0.7" date="Tue Jul 28 11:09:34 2009">
<file filename="cg1.r">
  <position> 0.00000000 2.00000000 0.00000000 </position>
  <position> 0.00000000 2.10021981 0.00000002 </position>
  <position> 0.00000000 2.12916202 0.00000002 </position>
  <position> 0.00000000 2.16991837 0.00000003 </position>
  <position> 0.00000000 2.19074911 0.00000004 </position>
  ...
```

- World Wide Web Consortium (W3C) [http://www.w3.org](http://www.w3.org)
- XPath: [http://www.w3.org/TR/xpath](http://www.w3.org/TR/xpath)
A simple Qbox input script

# optimization of the geometry of Si4
set cell 20 0 0  0 20 0  0 0 20
species silicon http://fpmd.ucdavis.edu/potentials/Si/Si_HSCV_LDA-1.0.xml
atom Si1 silicon  3.500  0.000  0.000
atom Si2 silicon  0.000  2.000  0.000
atom Si3 silicon -3.500  0.000  0.000
atom Si4 silicon  0.000 -2.000  0.000
set ecut 6

# compute the electronic ground state
# select wavefunction optimization algorithm
set wf_dyn PSDA
# randomize wavefunctions to avoid high symmetry saddle points
randomize_wf
# run 200 SCF iterations
run 0 200

# optimize the geometry, running 50 ionic steps with 5 SCF steps
set atoms_dyn CG
run 50 5

...
A simple Qbox input script

. . .

# displace one atom
move Si2 by 0.2 0 0

# recalculate the ground state
run 0 20

# run MD simulation
set atoms_dyn MD
set dt 40
run 100 5

# save sample for later use
save md.xml
The Qbox sample file

- Qbox saves its current state in a sample file (restart file) using the `save` command.
- Sample files can be reloaded later using the `load` command.
- Qbox sample files conform to the XML schema specified at 
- Sample files are portable across platforms
Part of the information in restart files consists of large arrays of floating point data.

Could be saved in binary form in a separate file (but would not be portable).

Keeping track of multiple files lead to confusion and errors.

We use base64 little-endian encoding:
- inflates data by 30%
- portable

Keep a single-file model: One sample, one file.
<?xml version="1.0" encoding="UTF-8"?>
<fpmd:sample xmlns:fpmd="http://www.quantum-simulation.org/ns/fpmd/fpmd-1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.quantum-simulation.org/ns/fpmd/fpmd-1.0 sample.xsd">
  <description>Created 2009-07-28T18:58:50Z by qbox-1.48.0</description>
  <atomset>
    <unit_cell>
      a="20.00000000 0.00000000 0.00000000"
      b="0.00000000 20.00000000 0.00000000"
      c="0.00000000 0.00000000 20.00000000" />
    <species name="silicon" href="silicon.xml"/>
    <atom name="Si1" species="silicon">
      <position>3.78993440 -0.05782115 -0.00028353</position>
      <velocity>6.481116e-05 -7.347506e-05 -5.581231e-08</velocity>
    </atom>
    <atom name="Si2" species="silicon">
      <position>0.05373812 2.25793417 0.00133574</position>
      ...
  </atomset>
</fpmd:sample>
The Qbox sample file (cont’d)

... 
<wavefunction ecut="3" nspin="1" nel="16" nempty="0"> 
  <domain a="20 0 0" 
     b="0 20 0" 
     c="0 0 20"/> 
<grid nx="18" ny="18" nz="18"/> 
<slater_determinant kpoint="0 0 0" 
  weight="1" size="8"> 
  <density_matrix form="diagonal" size="8"> 
    2.00000000 2.00000000 2.00000000 2.00000000 2.00000000 2.00000000 2.00000000 2.00000000 
  </density_matrix> 
  <grid_function type="double" nx="18" ny="18" nz="18" encoding="base64"> 
    JOKp5mfU7j81TFHumHAZwGswARq/giHAgZbowT3hFMBArxDWNAnWv2OdQWQ4qO8/hnpXb9Ys 
        4z/iCe2p/wLaP7CA2Fs148g/ldYPLTVj0tz+YesKvbXq3Pzz795WF18o/1h9MnNXF8D9PE2Sa 
        omoEQDsgdtEK5xBAjyqjXY/IEBcCDauHK80QBGbSjefjSNAjfwGjxuC/D86W50+3NoQwKJW 
        TL6svxbAXBarLa3BAsgTTCugPRPxGpsQzuu4/vtv7o4r5D/G3kkZWFLaPyBgoXndC8Y/ 
        9TVNRhx/tT8IdDgiFui0P17fmNFDdMM/PhR12ITj6T9ScaluessCQHW7UqHpLhPApFUza8LG 
        H0BKfH+Z26QnQGo7C1jS7yJAftATNO8dCECLXX2o0h3zvwLD7E54sgDAqE0t93M58b+m9D5g 
  </grid_function> 
</slater_determinant> 
</wavefunction>
Qbox I/O

Qbox

XMLPreprocessor

MPI-IO

fread

MPI

Apache Xerces-C

write

read

network
Coupling Qbox with other codes: a client-server model

- In some applications, it is useful to have Quantum simulations “driven” by another simulation code
- Example: Path Integral Monte Carlo simulations
  - Monte Carlo code generates atomic configurations
  - FPMD code computes corresponding energy
- Requires two-way communication between the driver and the FPMD code
- File-based Client-server model
Qbox client-server implementation

Client and server may run on separate platforms (must share an NFS file system)
Post-processing

- Users build post-processing pipelines
- `xsltproc` XSLT processor
  - namespace-aware
  - web-aware (can post-process web-based samples)
- `xml_grep` (Twig toolkit)
  - easy to use in scripts (uses the XPath language)
- Some applications based on libxml2
Post-processing pipelines

Qbox cmd script

my_sample.xml → Qbox → my_simulation.xml

my_species.xml → Qbox

my_new_sample.xml

visualization tools

XSLT processor

my_simulation.xml → XSLT script

gnuplot
Data analysis using mixed scripts

- XSLT and gnuplot scripts embedded in a bash shell script

```
XSLT script

sample.xml

xsltproc

gnuplot

Correlation time (ps)

r (Å)
```

UC Davis
University of California
Using Qbox on nanolab

$ cd
$ cp /home/fgygi/public/qbox_tutorial.tgz .
$ tar xzf qbox_tutorial.tgz
$ cd qbox_tutorial
$ . ./qbox_setup.sh
$ ls

- Open the Qbox User Guide:
Running the examples

- Examples are in the following directories:
  - si4: compute the electronic ground state of the Si4 cluster
  - si4_optim: optimize the geometry of Si4
  - si4_md: run molecular dynamics for Si4
  - ch4: ground state of methane
  - ch4_mlwf: compute Maximally Localized Wannier Functions for methane
  - ch4_www: use www features of Qbox
  - al_stress: compute the stress tensor of FCC Al
  - sidot: molecular dynamics of a Si26H30 quantum dot
Example 1: si4

$ cd si4
$ cat README
$ cat gs.i

$ qb gs.i > gs.r &