Welcome!

Virtual tutorial starts at 15:00 GMT

Please leave feedback afterwards at:
www.archer.ac.uk/training/feedback/online-course-feedback.php
CP2K: Recent performance improvements and new TD-DFT functionality

ARCHER Virtual Tutorial, 23rd Nov 2016
Iain Bethune ibethune@epcc.ed.ac.uk
Matt Watkins mwatkins@lincoln.ac.uk

EPSRC | NERC | SCIENCE OF THE ENVIRONMENT
archer | CRAY | THE SUPERCOMPUTER COMPANY | epcc

THE UNIVERSITY OF EDINBURGH
Reusing this material

This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License.

http://creativecommons.org/licenses/by-nc-sa/4.0/deed.en_US

This means you are free to copy and redistribute the material and adapt and build on the material under the following terms: You must give appropriate credit, provide a link to the license and indicate if changes were made. If you adapt or build on the material you must distribute your work under the same license as the original.

Note that this presentation contains images owned by others. Please seek their permission before reusing these images.
Outline

• CP2K 4.1 Release

• Performance Improvements
  • eCSE 06-06 (Iain Bethune & Mark Tucker, EPCC)

• Time Dependent DFT functionality
  • eCSE 03-11 (Matt Watkins & Sergey Chulkov, University of Lincoln)
CP2K Overview

“CP2K is a program to perform atomistic and molecular simulations of solid state, liquid, molecular, and biological systems. It provides a general framework for different methods such as e.g., density functional theory (DFT) using a mixed Gaussian and plane waves approach (GPW) and classical pair and many-body potentials.”

From [www.cp2k.org](http://www.cp2k.org) (2004!)

2nd most heavily used code on ARCHER, >2000 MAU since 2014
CP2K Overview

- Many force models:
  - Classical
  - DFT (GPW, GAPW + vDW)
  - Hybrid Hartree-Fock
  - LS-DFT
  - post-HF (MP2, RPA)
  - Combinations (QM/MM, mixed)

- Simulation tools
  - MD (various ensembles)
  - Monte Carlo
  - Minimisation (GEO/CELL_OPT)
  - Properties (Spectra, excitations …)

- Open Source
  - GPL, [www.cp2k.org](http://www.cp2k.org)
  - 1m loc, ~2 commits per day
  - ~20 core developers
CP2K 4.1

- Released 5th Oct 2016
  - Installed on ARCHER 19th October

- Default version:
  - module load cp2k

- Previous releases also available:
  - module load cp2k/3.0.16521
  - module load cp2k/2.7.15791

- New features
  - Modified Atomic Orbitals analysis (Heinzmann & Alrichs, 1976)
  - Interface to OMEN for NEGF transport calculations
  - Linear Scaling DFT
    - Polarized Atomic Orbitals (Berghold et al, 2002)
    - Curvy Steps (Shao et al, 2003)
  - O(N^3) RPA implementation
  - k-Points improvements (https://www.cp2k.org/faq:kpoints)
  - TD-DFT with Hybrid Functionals (more later)

https://www.archer.ac.uk/documentation/software/cp2k/ for documentation, benchmarks, hints and tips...

Complete list: https://www.cp2k.org/version_history
CP2K Training

• CP2K User Group Meeting
  • Monday 9th Jan 2017
  • In Edinburgh (travel funding available for UK users)
  • Keynote speaker - Prof. Jürg Hutter, University of Zurich
  • Method & application talks
  • Lightning talks

• Info from last year: https://www.epcc.ed.ac.uk/content/cp2k-uk-workshop-2016

• Hands-on CP2K training / development support
  • Contact ibethune@epcc.ed.ac.uk
Performance Improvements

• Load balancing for disordered / inhomogeneous systems

• Existing algorithm:
  • Serial, $O(p^2)$ memory and time
  • Disabled for $p > 1024$

• New algorithm:
  • MPI parallel, $O(p)$ memory and time
  • Uses MPI_Scan()
  • Enabled by default

From P. Shivadasan, MSc Thesis, 2014
Performance Improvements

<table>
<thead>
<tr>
<th>Nodes of ARCHER</th>
<th>2</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>32</th>
<th>48</th>
<th>64</th>
<th>96</th>
</tr>
</thead>
<tbody>
<tr>
<td>Old Algorithm (millsec)</td>
<td>26</td>
<td>32</td>
<td>51</td>
<td>153</td>
<td>389</td>
<td>1140</td>
<td>1864</td>
<td>5406</td>
</tr>
<tr>
<td>New Algorithm (millsec)</td>
<td>17</td>
<td>20</td>
<td>34</td>
<td>69</td>
<td>115</td>
<td>171</td>
<td>305</td>
<td>607</td>
</tr>
<tr>
<td>Speedup</td>
<td>1.53</td>
<td>1.60</td>
<td>1.50</td>
<td>2.22</td>
<td>3.38</td>
<td>6.67</td>
<td>6.11</td>
<td>8.91</td>
</tr>
</tbody>
</table>

Table 1: Time in optimize_load_list

Saving 3.3GB memory per node

- Charged cluster of 216 water molecules in 34Å³ box
- TZV2P MOLOPT basis set
- PBC off
- ~10% speedup
Performance Improvements

- Gaussian and Augmented Plane Waves (GAPW) Method

- Represent core electronic density on spherical grids around each atom

- Avoids frozen-core approximation

- Extra computation not present in GPW:
  - Compute the ‘hard’ density around each atom - `calculate_rho_atom_coeff()`
  - Compute updates to the KS matrix elements for each GAPW atom pair - `update_ks_atom()`

- Not OpenMP parallelised
Performance Improvements

<table>
<thead>
<tr>
<th>Version</th>
<th>Processes</th>
<th>Threads</th>
<th>Run Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original Code</td>
<td>24</td>
<td>1</td>
<td>273.17</td>
</tr>
<tr>
<td>Original Code</td>
<td>6</td>
<td>4</td>
<td>171.25</td>
</tr>
<tr>
<td>Implement OpenMP in update_ks_atom</td>
<td>6</td>
<td>4</td>
<td>172.30</td>
</tr>
<tr>
<td>Implement OpenMP in calc_rho_atom_coeff</td>
<td>6</td>
<td>4</td>
<td>73.77</td>
</tr>
<tr>
<td>Use hash table in OpenMP in update_ks_atom</td>
<td>6</td>
<td>4</td>
<td>61.63</td>
</tr>
<tr>
<td>Improve efficiency of OpenMP use &amp; automatic arrays in calc_rho_atom_coeff</td>
<td>6</td>
<td>4</td>
<td>47.27</td>
</tr>
</tbody>
</table>

Table 3: Times (seconds) as work on update_ks_atom and calculate_rho_atom_coefficient progressed

- 32 water molecules in a periodic box
- TZV2P basis set
- GAPW default settings
- 3.6x speedup for whole code
Performance Improvements

- Dispersion-corrected functionals
  - Important for a wide range of systems:
    - molecular, MOFs, surfaces...
  - Pair-potential type (Grimme D2, D3)
  - Non-local type (vdW-DF, rVV10 ...)

- Small but measurable overhead
  
  vdW_energy()  
  get_potential()  
  calculate_dispersion_pairpot()  
  calculate_dispersion_nonloc()  

- Not OpenMP parallelised

Recent overview of vDW corrections in CP2K:  

Performance Improvements

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Final</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pure MPI</td>
<td>360 MPI</td>
<td>360 MPI</td>
</tr>
<tr>
<td>Entire Program</td>
<td>414.7</td>
<td>425.4</td>
<td>402.4</td>
</tr>
<tr>
<td></td>
<td>1.057x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>calculate_dispersion_nonloc</td>
<td>9.27</td>
<td>9.61</td>
<td>5.49</td>
</tr>
<tr>
<td></td>
<td>1.750x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vDW_energy</td>
<td>18.20</td>
<td>34.34</td>
<td>21.61</td>
</tr>
<tr>
<td></td>
<td>1.589x</td>
<td></td>
<td></td>
</tr>
<tr>
<td>get_potential</td>
<td>6.09</td>
<td>6.12</td>
<td>2.63</td>
</tr>
<tr>
<td></td>
<td>2.327x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Run times (seconds) for routines involved in WP3.

- Water / BN interface
- 400 H₂O, 120 BN
- 1440 atoms, 26x25x40Å cell
- DZVP MOLOPT basis
- optB88-vDW functional (Libxc)
- ~5% speedup
TD-DFT
http://www.archer.ac.uk/training/

- Face-to-face courses
  - timetable, information and registration
  - material from all past courses

- Virtual tutorials & webinars
  - https://www.archer.ac.uk/training/virtual/
  - timetable plus slides and recordings
  - please leave feedback on previous tutorials after viewing material
Goodbye!

Thanks for attending

Please leave feedback at:
www.archer.ac.uk/training/feedback/online-course-feedback.php