Weile Jia\textsuperscript{1}, Long Wang\textsuperscript{1}, Xuebin Chi\textsuperscript{1}, Lin-Wang Wang\textsuperscript{2}

Supercomputing Center of Chinese Academy of Science
Lawrence Berkeley National Lab
2015-06-23

A CPU/GPU Linear Scaling Three Dimensional Fragment Method for Large Scale Electronic Structure Calculations on Titan Supercomputer
Outline

Motivation

LS3DF Algorithm

One Fragment on GPU

Testing results

Future work
# Software in Supercomputing center of CAS

<table>
<thead>
<tr>
<th>Software</th>
<th>Fields</th>
<th>USER Num</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>VASP</td>
<td>First principle, commercial</td>
<td>65</td>
<td>25.2%</td>
</tr>
<tr>
<td>NAMD</td>
<td>MD, open source</td>
<td>14</td>
<td>14.5%</td>
</tr>
<tr>
<td>Gromacs</td>
<td>MD, open source</td>
<td>36</td>
<td>13.7%</td>
</tr>
<tr>
<td>Lammps</td>
<td>MD, open source</td>
<td>63</td>
<td>5.8%</td>
</tr>
<tr>
<td>NWchem</td>
<td>First principle, open source</td>
<td>17</td>
<td>3.9%</td>
</tr>
<tr>
<td>Amber</td>
<td>MD, Commercial</td>
<td>18</td>
<td>3.0%</td>
</tr>
<tr>
<td>Siesta</td>
<td>Material simu, open source</td>
<td>6</td>
<td>2.4%</td>
</tr>
<tr>
<td>Flutter</td>
<td>Force simulation</td>
<td>1</td>
<td>2.0%</td>
</tr>
<tr>
<td>Dovis dock</td>
<td>Medicine</td>
<td>1</td>
<td>1.7%</td>
</tr>
<tr>
<td>Autodock4</td>
<td>Molecular simulation, open source</td>
<td>2</td>
<td>1.6%</td>
</tr>
<tr>
<td>Espresso</td>
<td>DFT, open source</td>
<td>12</td>
<td>1.2%</td>
</tr>
<tr>
<td>Match</td>
<td>User developed code</td>
<td>1</td>
<td>1.2%</td>
</tr>
<tr>
<td>Gaussian</td>
<td>First principle, commercial</td>
<td>59</td>
<td>0.7%</td>
</tr>
<tr>
<td>Materials Studio</td>
<td>First principle, commercial</td>
<td>15</td>
<td>0.4%</td>
</tr>
</tbody>
</table>

**First principle: 31%**
DFT in NERSC community

A survey of computational material science algorithm in NERSC community (2007)
DFT on Titan

18,688 nodes

16-core AMD Opteron 6274 CPU
1 Nvidia Tesla K20X GPU - 1.31Tflops
Peak Performance 27 Pflops
GPU contributes 24Pflops

Running CPU application only use 12% of the Titan computing power.
Three main challenges for ab initio material simulations

(1) Accuracy
(climb Jacob’s ladder)

(2) Temporal scale (from fs to seconds)
(new algorithms, like the accelerated MD)

(3) Size scale (mesoscale problems)
(Divide & Conquer methods)

All can be helped by exascale computing

What is LS3DF?

- A novel divide and conquer scheme with a new approach for patching the fragments together
- No spatial partition functions needed
- Uses overlapping positive and negative fragments
- New approach minimizes artificial boundary effects
Total = \sum_{F} \{ F \rightarrow F \} 

LS3DF in 2D and 3D

Total = $\Sigma_F \{ \}

Boundary effects are (nearly) cancelled out between the fragments

$System = \sum_{i,j,k} \{ F_{222} + F_{211} + F_{121} + F_{112} - F_{221} - F_{212} - F_{122} - F_{111} \}$
Schematics for LS3DF calculation
Flow chart of LS3DF

Based on the plane wave PEtot code: http://hpcrd.lbl.gov/~linwang/PEtot/PEtot.html
Operation counts and convergence

- Cross over with direct LDA method [PEtot] is 500 atoms.
- Similar to other O(N) methods.

Convergence of the LS3DF code – electronic structure SCF
LS3DF CPU profiling

gen_vr_fragment  solve Fragment  occupy
gen_total_density  gen_potential

Potential mixing: \( V_{in}(r) \rightarrow V_{tot}(r) \)

Gen_VF: \( V_{in}(r) \rightarrow V_{F}(r) \)

PEtot_F: solve for \( F \) and \( i=1,m \)

\[
\left(-\frac{1}{2}\nabla^2 + V_F(r)\right)\psi_i^F(r) = E_i^F\psi_i^F(r)
\]

\( \rho_F(r) = \sum_i |\psi_i^F(r)|^2 \)

Gen_dens: \( \rho_{tot}(r) = \sum_{F} \text{Sign}_F \rho_F(r) \cdot \text{Gen_total_density} \)

Poisson: \( \rho_{tot}(r) \rightarrow V_{out}(r) \)
One Fragment on GPU
The DFT formula (e.g., local density approximation) is used to calculate \( V(r) \) from \( \rho(r) \). There are \( N \) electron wave functions \( \Psi_i \), where \( 2N \) is the number of total valence electrons in the system.
The all-band CG (AB-CG) method for $H\psi_i = \varepsilon_i \psi_i$. The time consuming steps are indicated by the asterisk sign. The other parts will be called collectively as the Fortran-do-loops.
3-levels of parallelization:

- K-point parallel
- Band-index parallel
- G-space parallel
Hybrid GPU parallelization

For GPU:
Original parallelization are too fragmented for FFT
Too much communication
Memory copy between CPU-GPU

\[ h(i, j) = \langle \psi_i | H | \psi_j \rangle \]

\[ P_i = H\psi_i - \epsilon_i \psi_i \]

\[ P_i = A(P_i - \frac{\lambda_i}{\lambda_i^0} P_i^o) \]

Precond. CG step

\[ P_i = P_i - \sum_{j=1,i} \langle P_i | \psi_j \rangle \]

Projection, *

\[ \psi_i = \psi_i \cos \theta_i + P_i \sin \theta_i \]

Line minimiz.

\[ \psi_i = \psi_i - \sum_{j<i} \langle \psi_i | \psi_j \rangle \]

Orth., *

\[ h(i, j) = \langle \psi_i | H | \psi_j \rangle \]

Sub.diag, *
Mix-precision for the wave function residual $\Pi$

Convergence of the PEtot code after utilizing the $\Pi$ mix-precision calculation
Hpsi calculation (single precision)

Hpsi calculation for one wave function.
Hψ calculation for a single wave function.

<table>
<thead>
<tr>
<th></th>
<th>Percentage of one Hψsi calculation</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU-GPU</td>
<td>3.5%</td>
</tr>
<tr>
<td>Computing</td>
<td>62%</td>
</tr>
<tr>
<td>GPU-CPU</td>
<td>6.5%</td>
</tr>
<tr>
<td>Latency</td>
<td>28%</td>
</tr>
</tbody>
</table>

- CUDA computing
- CPU
- GPU
- cudaHostToDevice
- cudaDeviceToDevice
Convert latency problem to bandwidth problem.
GPU Library

ELPA: A consortium lead by Fritz-Haber-Inst. Max-Planck-Inst

CULA: Single GPU

MAGMA: Single GPU
The speedup of GPU CG_AllBand over CPU PEtot code on Titan.
Testing results
PEtot testing result - one fragment

<table>
<thead>
<tr>
<th>GPU No.</th>
<th>32</th>
<th>64</th>
<th>128</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>PEtot GPU</td>
<td>31.6</td>
<td>20.8</td>
<td>13.2</td>
<td><strong>11.4</strong></td>
</tr>
<tr>
<td>CPU No.</td>
<td>32x16</td>
<td>64x16</td>
<td>128x16</td>
<td>256x16</td>
</tr>
<tr>
<td>PEtot CPU</td>
<td>277</td>
<td>223</td>
<td>203</td>
<td>216</td>
</tr>
</tbody>
</table>

First phase of Titan:
- GPU: Tesla C2090
- CPU: AMD 16core
Testing system is a 3877 atom system. LS3DF runs 18 step of SCF.
LS3DF Testing result

LS3DF computational time comparison CPU/GPU

- CPU
- GPU
We implemented a GPU LS3DF on hybrid CPU/GPU supercomputer Titan, currently it has 4x speedup compared with CPU code.

For the single fragment calculation, Petot code, we have 10x-20x times speedup.

Our results show that the data locality and local MPI communication makes divide-and-conquer algorithms ideal in utilizing the heterogeneous architecture computing power.
Future work

- Optimize charge gathering/patching from fragment charge to global charge (MPI/CPU)
- We estimate that 10x times of overall speedup could be achieved by using GPU
Acknowledgement

- INCITE program
- China Scholarship Council
- US Department Of Energy, BES, Office of Science
Thanks!