Accelerating Computational Chemistry Applications using Emerging Architectures

Akila Gothandaraman

Ph. D. Candidate
Department of Electrical Engineering and Computer Science
University of Tennessee, Knoxville

Innovative Computing Laboratory Seminar
March 13, 2009
Bio

- Ph.D., Computer Engineering, Minor in Computational Science
  (anticipated: August 2009)

- NSF funded computational chemistry project
  - Dr. Gregory Peterson (Major advisor, EECS, UTK)
  - Dr. Robert Harrison and Dr. Robert Hinde (Chemistry, UTK)
  - Team members: Students from {EECS, Chemistry} at UTK and faculty/students from NCSA/UIUC and University of Pennsylvania

- Summer Internships
  - Advanced Computing Laboratory, GE Global Research
    - Acceleration of a CT Image Reconstruction Algorithm using NVIDIA GPUs
  - Signal Synthesis Group, Analog Devices Inc.
    - Design of Frequency Lock Algorithms for Phase Locked Loops
Cornell Theory Center defines Computational Science as

“A field that concentrates on the effective use of computer software, hardware and mathematics to solve real problems. It is a term used when it is desirable to distinguish the more pragmatic aspects of computing from

(1) computer science, which often deals with the more theoretical aspects of computing; and from

(2) computing engineering, which deals primarily with the design and construction of computers themselves.

“Computational science is often thought of as the third leg of science along with experimental and theoretical science.”
Computer Simulations

- Computer simulation methods
  - Indispensable tools to solve problems
  - Bridge between model and theoretical predictions or between the model and experimental results
  - Physics and Physical Chemistry:
    - Molecular Dynamics (MD), Quantum MC (QMC)
    - Underlying kernels are computationally intensive

Ideal candidates for hardware acceleration
Emerging Architectures

- Ever-increasing performance demands

- Current computing trends
  - Chip makers are turning to **multi-core**
  - Reconfigurable computing - **field programmable gate arrays**
  - Graphics processors for general-purpose computing - **GP-GPUs**
  - Sony-Toshiba-IBM **Cell Processor**
  - Hybrid-computing platforms
Explore emerging architectures such as Field-Programmable Gate Arrays (FPGAs), Graphics Processing Units (GPUs) and hybrid-computing platforms combining CPUs, FPGAs, and GPUs to exploit the best features of these platforms for two widely used simulation methods - Molecular Dynamics and Quantum Monte Carlo methods.

RC platforms and GPUs

- Reconfigurable Computing (RC)
  - High Performance Reconfigurable Computing (HPRC)
  - Exploit “polygranular” parallelism
  - FPGA-based solutions - Cray (XD1, XT5), Nallatech, SRC Computers, DRC Computers, XtremeData, Maxwell

- Graphics Processing Units (GPUs)
  - Improvement in hardware performance and programmability - General-Purpose GP-GPU computing
  - GPU solutions - nVidia, AMD/ATI
Molecular Dynamics

- Molecular Dynamics (MD)
  - Computational {chemistry, biology}, material science
  - Deterministic method - Newtonian mechanics

- Iterative process:
  - Force Computation
    - Force on each particle due to other particles in the system, acceleration computed using Newton’s Law
  - Integration
    - Update the particle positions

- Contributions to forces in an MD simulation
  - Bonded
  - Non-bonded forces - computationally intensive, $O(N^2)$
    - Lennard-Jones (van der Waals)
    - Coulombic forces
Monte Carlo

- Monte Carlo
  - Stochastic - science, engineering and finance
  - Studying many-particle systems and calculating high-dimensional integrals

- Quantum Monte Carlo (QMC)
  - Solving the many-body Schrödinger equation
  - Structural and energetic properties of clusters of atoms or molecules
  - Variational Monte Carlo - used in this work
    - Random walks to calculate the multi-dimensional integrals of expectation values, such as total energy
    - Trial wavefunction that when optimized best approximates the exact wavefunction
Variational Monte Carlo (VMC) algorithm

REPEAT (for \( N \) iterations)

Step 1: Select a reference configuration, \( R(x, y, z) \) at random

Step 2: Obtain a new configuration, \( R' \) by adding a small displacement, to all the particles in the system

Step 3: Compute the ground-state properties (energy, wavefunction) of the particles in the current configuration, \( R' \)

Step 4: Accept or reject the present configuration using the ratio of the wave function values,

\[
p = \frac{\psi_T(R')}{\psi_T(R)}
\]

If \( p \geq 1 \), \( R' \) is accepted, else if \( p < 1 \), compare \( p \) with a URN, \( R' \) is rejected, \( R \) is retained

UNTIL finished
Block diagram (VMC algorithm)

O(N) co-ordinate positions

- QMC on host processor
- Distance Calculation
- Potential Energy (PE) Calculation
- Wave Function (WF) Calculation
- Derivatives of Wave Function

O(N) results

O(N^2) squared distances

- On GPU/FPGA
- Presently on host processor

Software
Description of Kernel – Potential Energy (PE)

$N$-particle potential energy, $V_{total} \approx \sum_{i<j}^{N} V(r_{ij})$

1) Rewrite pair-wise potential as function of $r_{ij}^2$

2) Transform the potential to overcome domain and range problems

$V_{total} = V_I + V_{II} = \sum_{(i<j)\in I} V_I(r_{ij}) + \sum_{(i<j)\in II} V_{II}(r_{ij})$

- Exponential transform in Region I
- Logarithmic binning scheme in Region II

- Potential rescaled to take values less than or equal to one
- Quadratic interpolation on the transformed potential

Region I: $0 \leq r_{ij}^2 < \sigma^2$
Region II: $r_{ij}^2 \geq \sigma^2$

Potential vs. Distance

Infinite range
Infinite domain

TENNESSEE ADVANCED COMPUTING LABORATORY
Top-level design on Cray XD1

- Potential Energy (PE) and Wave Function (WF) calculated on FPGA
- Rest of the application (random number generation [SPRNG], accept/reject ratio, accumulation of results) in software

- QMC interface connects design modules with Cray platform-related interfaces [Cray XD1]
- RT Client Block - processes read and write requests to and from processor
- Block RAM interface and Register Interface
  - User design to read status signals and Block RAM contents
  - Write results back to registers
Results

- **Target platform**
  - Pacific Cray XD1 HPRC platform - Single chassis system with 6 nodes
  - Each node - 2.2 GHz AMD Opteron dual-core dual-processor, Xilinx Virtex-II Pro (XC2VP50) or Xilinx Virtex-4 (XC4VLX160) FPGA
  - Processor and FPGA communication - RapidArray interconnect

- **Development language and tools**
  - VHDL, Synplicity Synplify tools, Xilinx 8.1 ISE/EDK

<table>
<thead>
<tr>
<th>Kernel / Resource type</th>
<th>PE (26%)</th>
<th>PE and WF (45%)</th>
<th>PE - Potential Energy</th>
<th>WF - Wave Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLICEs (67,854)</td>
<td>17,984</td>
<td>30,432</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Block RAMs (288)</td>
<td>79 (27%)</td>
<td>147 (51%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DSP48s (96)</td>
<td>25 (26%)</td>
<td>50 (52%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**PE and WF on the same FPGA**
Results

Execution times (in seconds) (4000 particle simulation)

<table>
<thead>
<tr>
<th></th>
<th>QMC</th>
<th>PE + WF</th>
<th>Rest of the application</th>
<th>Overhead</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>HW accelerated</td>
<td>1084.21</td>
<td>482.12</td>
<td></td>
<td>1.61</td>
<td>1567.94</td>
</tr>
<tr>
<td>Software only</td>
<td>39733.37</td>
<td>482.11</td>
<td></td>
<td>--</td>
<td>40215.48</td>
</tr>
</tbody>
</table>

Software QMC, C++ (double-precision) [Warren05]
One node of dual-core dual-processor AMD Opteron 2.2 GHz
gcc, -03 optimization

Hardware accelerated QMC (fixed-point)
One Xilinx Virtex-4 FPGA
Deeply pipelined design (58 pipe stages)
Design operates at 66 MHz

Speedup vs number of particles (Pacific Cray XD1)

≈ 25x speedup
GPU Implementation

- GPU used as a co-processor
- Compute Unified Device Architecture (CUDA) implementation
- Brute force calculation of $N^2$ interactions [GPUGems3]

- Multiple blocks ($N/p$): Multiple looping threads in each block
- Shared memory (16KB/block) to store particle positions in one step

Target GPUs: Tesla C870 GPU and Tesla C1060 NVIDIA GPU

Preliminary results show a speedup of 100x over the CPU
Ongoing Research

- Hybrid-Computing Platform - FPGAs and GPUs
  - DRC System with Xilinx Virtex-4 FPGA in the processor socket and NVIDIA GPU through PCI-Express interface

- Performance Models
  - Tools to analyze and exploit available computational resources
  - Analytic Performance Model (shared, HPRC systems) [Peterson94], [Smith03]
Contributions

- **First effort to explore HPRC architectures for QMC**
  - Significant performance improvement using one computing node on Cray XD1 HPRC platform
  - Open source, user-friendly hardware accelerated QMC framework
  - Useful tool for studying many-particle systems

- **First effort to explore hybrid-architectures for MD and QMC**
  - Significant impact in computational science
  - “Starting point” for hybrid-computing platforms in related scientific applications

- **Analytical Performance Modeling**
  - Extend existing models for HPC and HPRC
  - How to “best” map applications on architectures
As an ICL team member

- **Current Research Focus:**
  - Computer Architecture, High Performance Reconfigurable Computing (HPRC), Graphics Processors for General-Purpose Computing (GPGPUs), Computational Science

- **Research Interests:**
  - Multi-core Platforms, Hybrid-computing Platforms, Performance Models

- **Interesting Projects at ICL:**
  - PLASMA
  - MAGMA
  - Exploiting single precision arithmetic

- **Collaboration/Mentoring graduate students**
Thank You
Relevant Publications

Journals:


Conferences:


Publications

Posters:


References


Backup slides
Squared Distance Calculation

Datapath of CalcDist module

- CalcDist - Squared distance between pairs of atoms
- On Cray XD1, atom positions loaded into Position Memory by the host processor
- 32-bit fixed-point representation for (x, y, z) positions and 52-bit squared distance per clock cycle
- Resulting distances classified as region I or region II distances
Function \{PE, WF\} calculation

- Pipelined modules that output a final potential/wavefunction every clock cycle
- Potential is accumulated as
  - running product in region I and running sum in region II
  - Note: This is due to the transformation schemes employed
- Wavefunction is accumulated as
  - running products in region I and II
  - Note: This is due to the functional form of wavefunction
- Host processor reconstructs the floating-point values of potential energy and wavefunction

**Datapath of CalcFunc module**

```
Datapath of CalcFunc module

Diagram showing the flow of data through the modules with labels for 'fa', 'fb', 'delta', 'fc', and the resulting output 'V' or 'ψ'. The latency is indicated as 7 clock cycles and 1 clock cycle.
```
Binning and Memory Platform

Binning schemes

- Two regions - non-identical numerical behavior
- Quadratic interpolation on the transformed PE and WF
- Region I divided into 256 bins
- Logarithmic binning scheme
  - Region II divided into 21 sub-regions
  - Each sub-region is divided into 64 bins
- Total of \([256 + 21 \times 64] \times 3\) interpolation coefficients

Memory Platform

- On-chip Block RAMs on Xilinx FPGAs
  - Position Memory
  - PE Coefficient Memory
  - WF Coefficient Memory
Error analysis

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Formats</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x, y, z) co-ordinate positions</td>
<td>s12.20</td>
</tr>
<tr>
<td>Squared distances</td>
<td>u27.26</td>
</tr>
<tr>
<td>Potential energy, wavefunction</td>
<td>s0.51</td>
</tr>
<tr>
<td>Interpolation coefficients</td>
<td>s0.51</td>
</tr>
</tbody>
</table>

- Fixed-point representation accurately reproduces the floating-point results almost over the entire range.
- Worst error performance when $r^2 > 2^{16}$
- Distances are confined to be within $2^{16}$ for our application.

No compromise in accuracy for our chemistry application.
Description of Kernels – Wave Function (WF)

- Rewrite wave function as function of $r_{ij}^2$
- A similar region classification approach ($\sigma$ obtained by setting first derivative of wave function to zero)
- Quadratic interpolation on the rescaled wave function

Wavefunction vs. Squared distance
Potential energy equation

\[ E = \sum_{\text{bonds}} \frac{a_i}{2} (l_i - l_{i0})^2 + \sum_{\text{angles}} \frac{b_i}{2} (\theta_i - \theta_{i0})^2 + \sum_{\text{torsions}} \frac{V_n}{2} (1 + \cos(n\omega - \gamma)) + \frac{1}{2} \sum_{i=1}^{N} \sum_{j\neq i}^{N} 4\epsilon_{ij} \left[ \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left( \frac{\sigma_{ij}}{r_{ij}} \right)^{6} \right] + \frac{1}{2} \sum_{i=1}^{N} \sum_{j\neq i}^{N} \frac{q_i q_j}{r_{ij}}, \]

**A, B, C – Bonded interactions**

A: Harmonic potential between bonded atoms
B: Harmonic potential in the valence angles
C: Torsional potential due to bond rotations

**D, E – Non-bonded interactions**

D: Lennard-Jones potential
E: Coulomb's electrostatic potential
Bonded and non-bonded interactions

- CPU time – evaluation of the potential as well as timescale
- Bonded terms: $O(N)$
- Non-bonded: $O(N^2)$

- Lennard Jones potential - Fast convergence allows truncation using a cut-off distance
- Coulomb interactions do not decay fast enough to provide an absolutely convergent series

Cut-off method not appropriate for electrostatic force calculation. Calculation of long range electrostatic force calculation is a problem!
Binning schemes

Region I – Bin lookup

- Region I divided into 256 bins
- Region II divided into 21 sub-regions or regimes
- Each sub-region accommodates 64 bins
- Total of $[256 + 21 \times 64] \times 3$ interpolation coefficients $\rightarrow f_a, f_b, f_c$

Region II – pseudo logarithmic binning

Region II – 1st stage Bin lookup

Region II - 2nd stage Bin lookup

**Equations**

- $\theta$ Region I divided into 256 bins
- $\theta$ Region II divided into 21 sub-regions or regimes
- $\theta$ Each sub-region accommodates 64 bins
- $\theta$ Total of $[256 + 21 \times 64] \times 3$ interpolation coefficients $\rightarrow f_a, f_b, f_c$