Streaming supercomputing for Molecular Modeling

Eric Darve

Ian Buck, Mattan Erez, Vidya Rangasayee
VLSI capabilities are currently not fully exploited

Exploiting data parallelism is necessary to get maximum performance
Streaming scientific computation exploits the capabilities of VLSI

• Modern VLSI technology makes arithmetic cheap
  – 100s of GFLOPS/chip today TFLOPS in 2010
• But bandwidth is expensive
• Streams change the ratio of arithmetic to bandwidth
  – By exposing producer-consumer locality
    • Cannot be exploited by caches – no reuse, no spatial locality
• Streams also expose parallelism
  – To keep 100s of FPUs per processor busy
• High-radix networks reduce cost of bandwidth when it’s needed
  – Simplifies programming
• Net result
  – 10x to 100x performance per unit cost
  – Scalable from workstation to supercomputer
Streaming technology has already been adopted in several processors

- **MD-Grape**: specialized processor for pair wise force calculations.
- **Playstation**: shading, rendering, character motion
- **Cell processor**: new generation of processors for games and more
- **NVIDIA and ATI graphics cards**: shading, rendering and more
  - More in Ian’s talk
- **Imagine**: graphics applications
- **Merrimac**: scientific calculations
- **Cray**: Cascade’s HWP – may adopt some features of Merrimac
- ...
Stream Architecture exploits Parallelism and Locality

16GB/s

64GB/s

512GB/s

3,840GB/s
Producer-Consumer Locality: optimal usage of BW hierarchy

Memory

Grid of Cells

Table

Stream Cache

Table

Stream Reg File

Cells

Indices

Results 1

Results 2

Results 2

Results 3

Local Registers

K1
50 Ops

K2
100 Ops

K3
70 Ops

K4
80 Ops

Grid of Cells

9.5Words

12Words

58Words

900Words
Main applications

- **streamSPAS**
  Sparse Matrix/Vector Linear Algebra

- **streamFEM-3D**
  Finite-element Method

- **streamFLO**
  2D Computational Fluid Dynamics

- **streamMD**
  Molecular Dynamics
How does one program a streaming processor?

- 100s of ALUs need to be kept busy.
- Idle time is minimized by achieving high ops-to-load ratio (7 to 50):
  - Streams of records (e.g. atoms)
  - Kernels (e.g. force calculation):
    - arguments can be IN, OUT, VOUT or REDUCE.
    - No external side effect (no static or global variables).
  - Producer Consumer Locality
  - Strip-mining
  - Software pipeline
- Almost optimal performance on single node for standard MD calculations.
- Would work even better for calculations requiring high-fidelity models (i.e. polarization effects...).
  - See Vijay’s presentation.
### What are typical ops-to-loads ratios?

<table>
<thead>
<tr>
<th>Application</th>
<th>Sustained GFLOPS</th>
<th>FP Ops / Mem Ref</th>
<th>LRF Refs</th>
<th>SRF Refs</th>
<th>Mem Refs</th>
</tr>
</thead>
<tbody>
<tr>
<td>StreamFEM3D¹ (Euler, quadratic)</td>
<td>31.6</td>
<td>17.1</td>
<td>153.0M (95.0%)</td>
<td>6.3M (3.9%)</td>
<td>1.8M (1.1%)</td>
</tr>
<tr>
<td>StreamFEM3D¹ (MHD, constant)</td>
<td>39.2</td>
<td>13.8</td>
<td>186.5M (99.4%)</td>
<td>7.7M (0.4%)</td>
<td>2.8M (0.2%)</td>
</tr>
<tr>
<td>StreamMD¹ (grid algorithm)</td>
<td>14.2²</td>
<td>12.1²</td>
<td>90.2M (97.5%)</td>
<td>1.6M (1.7%)</td>
<td>0.7M (0.8%)</td>
</tr>
<tr>
<td>GROMACS¹</td>
<td>22.0²</td>
<td>7.1²</td>
<td>181M (95.4%)</td>
<td>5.3M (2.8%)</td>
<td>3.4M (1.8%)</td>
</tr>
<tr>
<td>StreamFLO</td>
<td>12.9²</td>
<td>7.4²</td>
<td>234.3M (95.7%)</td>
<td>7.2M (2.9%)</td>
<td>3.4M (1.4%)</td>
</tr>
</tbody>
</table>

1. Simulated on a machine with 64GFLOPS peak performance
2. The low numbers are a result of many divide and square-root operations
streamMD (GROMACS)

- GROMACS: fastest MD code available.
- Can simulate complicated bio-molecules: proteins, polymers, etc.
- Ported to Brook. Optimization and performance analysis.
Kernel is fully packed

Latency fully covered

Schedule of kernel

Schedule of entire application

Computation Memory access

Merrimac vs. 1.7GHz P4
x56 times faster
x95 times less cycles
Hand-optimizations

• Loop unrolling and software pipelining: kernel almost fully packed.
• Manual strip-mining: no idle time between kernels. Memory access completely hidden.
• Data structure (neighbor list) was re-organized for better efficiency: duplication of atoms and padding with dummy atoms.
• Future approach: dynamic rate streaming with conditional streams.

R-stream (Reservoir Labs Inc.): process will become automatic thanks to streaming compiler
Current: pad + duplicate (only option currently supported by Brook)

Dynamic rate streaming (currently supported by hardware)
Arithmetic performance is extremely high: 50% of peak performance (64 Gflops)

Bandwidth hierarchy is fully utilized: 2 orders of magnitude drop between LRF and Memory
Multi-scale molecular modeling

- **I**
  - General Modified Embedded Atom Method
  - Molecular Dynamics specifically suited for Carbon

- **II**
  - Harrison Universal parameterization scheme
  - Self-consistent Tight Binding
  - Transport Module

- **III**
  - Refined geometry optimization
  - Detailed Electronic Structure Analysis
Quantum Transport Modeling

Transport simulation based on Non-Equilibrium Green’s Function Formalism

\[ i = \frac{e}{h} Tr[\Sigma^{in} G^{R} \Gamma G^{A} - \Gamma G^{R} \Sigma^{in} G^{A}] \]

Preliminary result from a Featureless Contact

![Diagram of a nanotube with Au leads and DOS and Transmission plots]
ab initio Modeling of CNT

**Work Function of Different CNTs**

<table>
<thead>
<tr>
<th>CNT</th>
<th>Diam (Å)</th>
<th>Work Function(eV)</th>
<th>Band Gap</th>
</tr>
</thead>
<tbody>
<tr>
<td>(8,0)</td>
<td>6.35</td>
<td>4.77</td>
<td>~ 4.8eV</td>
</tr>
<tr>
<td>(10,0)</td>
<td>7.88</td>
<td>4.70</td>
<td>0.87</td>
</tr>
<tr>
<td>(16,0)</td>
<td>12.70</td>
<td>4.69</td>
<td>0.56</td>
</tr>
<tr>
<td>exp</td>
<td>1~3nm</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

From the image, we can observe the following:

- The calculated work function of the (8,0) CNT using LDA is -2.545 eV.
- The local potential profile of the CNT shows a band gap of approximately 0.56 eV.
- The work function values for different CNTs are compared with experimental (Exp) and theoretical (Theo) values.
- The table includes the diameter in Å, work function, and band gap for each CNT type.

The theoretical work function for (8,0) CNT is 4.77 eV, which is close to the experimental value of ~4.8 eV.
Nanotechnology applied to bio-sensing

Nanostructures

Bio molecules

Coupling mechanisms
Scale compatibility
Process compatibility

Biosensing and labeling
Biological functionality enhancers
Bio-artificial interfaces
Biologically inspired fabrication techniques
Quantum calculations
Density Functional Theory

- Every observable for a quantum mechanical system can be calculated from the ground-state electron density alone.
- Kohn-Sham equation of density functional theory:

\[
\rho(r) = \sum_{i=1}^{N} |\psi_i(r)|^2
\]

\[
\left\{ -\frac{\nabla^2}{2} - \left( \sum_{A=1}^{M} \frac{Z_A}{r_{1A}} \right) + \int \frac{\rho(r_2)}{r_{12}} dr_2 + V_{XC}[r_1] \right\} \psi_i(r_1) = \epsilon_i \psi_i(r_1)
\]

- Different basis sets can be chosen:
  - Gaussian orbitals
    \[
    \psi_n(r) = x^{k_n} y^{l_n} z^{m_n} \exp(-\zeta_n r^2)
    \]
  - Slater type orbitals
    \[
    \psi_n(r) = x^{k_n} y^{l_n} z^{m_n} \exp(-\zeta_n r)
    \]
  - Plane waves
    \[
    \psi_n(r) = \exp(ik_n r)
    \]

- With plane waves, numerical solution of Kohn-Sham requires:
  - 3D FFTs
  - Solution of large eigenvalue problems
- Preliminary results on Merrimac for 2D FFTs.
Conclusion

• Merrimac provides high performance platform at the workstation and supercomputer level at a fraction of the cost.
• Achieved by using the computing power of 100s of ALUs while requiring low global bandwidth.
• High performance demonstrated on all applications including molecular dynamics.
• Will be extended to:
  – more complicated molecular force fields (polarization, etc.)
  – quantum mechanics (DFT)
  – blood flow