Integration: Computer Science/Applications

streamSPAS

streamFLO

streamFEM-3D

streamMD
Merrimac Applications Goals

Drive hardware, compiler and language groups to meet the performance needs of scientific simulation codes

Simulation Group

CSL Merrimac

Ensure high-performance of ASCI applications on Merrimac

Key to achieve those goals:

• Team of researchers interacting both with the Merrimac group and Simulation group.
• Integration of computer science and computational engineering.
Highlights of FY03 accomplishments

- **Toy Problem** Full 3D applications:
  - streamFEM-3D: full application + 3D
  - streamFLO: full application
  - streamMD: GROMACS (world fastest MD code)

- Irregular data structure:
  - streamSPAS: sparse Vector and Matrix toolkit

- Real Hardware validation on Imagine (streamMD)

- Multi-node analysis: SORT algorithm
Objective: stress the Merrimac architecture.
streamFEM 2D (FY02) → 3D (FY03)
Kernels are more complicated. Data set is larger.
Demonstrated: large fraction of peak is sustained for irregular mesh finite element code.
• Discontinuous Galerkin finite element method for systems of nonlinear conservation laws.
\[ u_t + \text{div}(\vec{f}) = 0 \]

• PDEs of increasing complexity:
  – Scalar advection equations (1 PDE)
  – Euler equations (5 PDEs)
  – Magnetohydrodynamics (MHD) equations (8 PDEs)

• Piecewise polynomial approximations with increasing order:
  – Constant (1), linear (4), quadratic (10), cubic (20)

• Irregular memory accesses on unstructured 3D meshes
• Variable arithmetic intensity
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.

Future work:
1. Use of Stream Cache and SRF indexing to reduce memory bandwidth requirements (better re-use)
2. Multi-node implementation.
3. Comparison with a C/OpenMP implementation on a shared memory computer.
streamFLO
M. Fatica – M. Erez

• Full application:
  – 2D Euler in conservative form.
  – Non linear multigrid acceleration based on Runge-Kutta smoother of A. Jameson.
  – H-CUSP scheme for artificial dissipation.

• Code is typical of a computational fluid dynamics application.

• Only application ported from FORTRAN.
- **Kernel for diffusive flux:** 32 Gflops = 50% peak
  - Low Gflops due to divide and square-root operations
  - 60 Gflops if counting operations used for iterations
- **Set up phase is slow (20%):** issue will be addressed by the Stream Cache and Indexable SRF.
- **Overall performance for a single grid level:** 13 Gflops
  - 26 Gflops if counting operations used for iterations

**Computation of the flux:** 90% utilization

**Snapshot from application:** >60% cluster occupation

<table>
<thead>
<tr>
<th>clusters</th>
<th>memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>650000</td>
<td></td>
</tr>
<tr>
<td>670000</td>
<td></td>
</tr>
<tr>
<td>690000</td>
<td></td>
</tr>
<tr>
<td>710000</td>
<td></td>
</tr>
<tr>
<td>730000</td>
<td></td>
</tr>
<tr>
<td>750000</td>
<td></td>
</tr>
<tr>
<td>770000</td>
<td></td>
</tr>
<tr>
<td>790000</td>
<td></td>
</tr>
<tr>
<td>810000</td>
<td></td>
</tr>
<tr>
<td>830000</td>
<td></td>
</tr>
<tr>
<td>850000</td>
<td></td>
</tr>
<tr>
<td>870000</td>
<td></td>
</tr>
<tr>
<td>890000</td>
<td></td>
</tr>
<tr>
<td>910000</td>
<td></td>
</tr>
<tr>
<td>930000</td>
<td></td>
</tr>
<tr>
<td>950000</td>
<td></td>
</tr>
<tr>
<td>970000</td>
<td></td>
</tr>
<tr>
<td>990000</td>
<td></td>
</tr>
<tr>
<td>1010000</td>
<td></td>
</tr>
<tr>
<td>1030000</td>
<td></td>
</tr>
<tr>
<td>1050000</td>
<td></td>
</tr>
<tr>
<td>1070000</td>
<td></td>
</tr>
</tbody>
</table>
Future work

- **3D Navier-Stokes.** Will contain a large portion of 3D RANS solver TFLO.
- **CDP\(\alpha\) (streamLES):** multigrid Poisson solver with unstructured highly anisotropic 3D grids:
  - Multi-coloring for Gauss-Seidel
  - Anisotropic coarsening
- Performance benchmark for realistic, industrial-strength flow applications.
- Will be used to test Brooktran.
streamSPAS
T. Barth, J.H. Ahn

- StreamSPAS: Stream SParse Algebra Suite
- Implements sparse matrix-vector products.
- Irregular data structure
- 4 algorithms have been tested:
  - Compressed sparse column: ScatterOP
  - Compressed sparse row: GatherOP
  - Hypergraph edge storage: Mix
  - Element-by-element storage (FEM): extra operations but element-wise dense 3X3 matrix-vector operations.
• Contrary to the other applications, streamSPAS is **bandwidth limited** (not arithmetic).

• Comparison with P4 shows that Merrimac chip is **20 times faster** (Poly 3)
  – Merrimac: 0.24ms (EBES) vs. P4: 5ms (CSR)

• Memory bandwidth is **18 times higher** on Merrimac:
  – Merrimac: 38.4Gbytes/sec vs. P4: 2.1 Gbytes/sec

• Merrimac:
  • CSC > CSR (no scatter-add for CSR)
  • EBES best for large cases (because of smaller memory access)
Performance for irregular data structure

- streamFEM + streamSPAS: both involve irregular data structure.
- Most severe test cases.
- Merrimac performs well because of:
  - High memory bandwidth
  - Full support for gather/scatter operations.
  - Indexable SRF + Stream cache
- Future work:
  - Preconditioned Conjugate gradient
  - Incomplete factorization
  - superLU (uses BLAS 1, 2 and 3).
streamMD
Y. Zhao, A. Garg, E. Darve

- GROMACS: fastest MD code available.
- Can simulate complicated bio-molecules: proteins, polymers, etc.
- Ported to Brook. Optimization and performance analysis.
• Several optimization were made:
  – Data structure (neighbor list) was re-organized for better efficiency.
  – Loop unrolling and software pipelining: kernel almost fully packed.
  – Manual strip-mining + number of Memory Address Registers increased to 22.
    No idle time between kernels. Memory access completely hidden.

• Comparison with P4. Merrimac is:
  – 56 times faster.
  – 95 times less cycles.
Merrimac vs. Protein Explorer

- “Protein Explorer:” special-purpose computer.
- Simulates huge biomolecules.
- RIKEN institute (Japan).
- Based on MDGRAPE-3 (special purpose chip for molecular simulations)
- 1 chip = 20 pipelines = 200 Gigaflops.
- Protein Explorer = 128 nodes x 40 chips = 1 Peta flops
  (NEC Earth Simulator = 30+ Tera flops)

<table>
<thead>
<tr>
<th></th>
<th>Merrimac</th>
<th>Protein Explorer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gflops peak</td>
<td>64*/128</td>
<td>200</td>
</tr>
<tr>
<td>Sustained (GROMACS)</td>
<td>22*</td>
<td>?</td>
</tr>
<tr>
<td>Cycles / atomic interaction</td>
<td>8</td>
<td>1</td>
</tr>
</tbody>
</table>
GROMACS on hardware: Imagine

- Run-time = 18 ms (1.1 ms on Merrimac)
- Number of Gflops = 1.62
- Percent of peak performance = 20.25%
- Cycles / interaction = 14
- Bandwidth from Mem/SRF/LRF = 0.8/1.264/31.92 (GB/s)
Conclusion

• Merrimac performance demonstrated
  – Full applications
  – 3D
  – Irregular data structure
• Influenced the design of the Hardware:
  – Stream Cache
  – Indexable SRF
• Influenced design choices for Brook.

Future work:
• ASCI Codes:
  – CDP\(\alpha\) (streamLES)
  – TFLO (large portions of NS 3D RANS)
• Peta flops supercomputer:
  – Multi-node
• Legacy codes:
  – Real-world FORTRAN codes ported using Brooktran