L13: Application Case Study III: Molecular Visualization and Material Point Method

Administrative

- Proposal due TODAY
- STRSM due March 15
- Midterm coming
  - In class March 30 or April 4, open notes
  - Review notes, readings and review lecture (before break)
  - Will post prior exams
- Design Review
  - Intermediate assessment of progress on project, oral and short
  - Tentatively April 11 and 13
- Final projects
  - Poster session, April 27 (dry run April 25)
  - Final report, May 4

Project Proposal (due TODAY)

- Proposal Logistics:
  – Significant implementation, worth 55% of grade
  – Each person turns in the proposal (should be same as other team members)
- Proposal:
  – 3-4 page document (11pt, single-spaced)
  – Submit with handin program:
    "handin cs6963 prop <pdf-file>"

Content of Proposal

I. Team members: Name and a sentence on expertise for each member

II. Problem description
- What is the computation and why is it important?
- Abstraction of computation: equations, graphic or pseudo-code, no more than 1 page

III. Suitability for GPU acceleration
- Amdahl's Law: describe the inherent parallelism. Argue that it is close to 100% of computation. Use measurements from CPU execution of computation if possible.
- Synchronization and Communication: Discuss what data structures may need to be protected by synchronization, or communication through host.
- Copy Overhead: Discuss the data footprint and anticipated cost of copying to/from host memory.

IV. Intellectual Challenges
- Generally, what makes this computation worthy of a project?
- Point to any difficulties you anticipate at present in achieving high speedup

Triangular Solve (STRSM)

```
for (j = 0; j < n; j++)
   for (k = 0; k < n; k++)
      if (B[j*n+k] != 0.0f) {
           for (i = k+1; i < n; i++)
             B[j*n+i] -= A[k * n + i] * B[j * n + k];
      }
```

Equivalent to:

```
cublasStrsm('l' /* left operator */, 'l' /* lower triangular */,
            'N' /* not transposed */, 'u' /* unit triangular */,
             N, N, alpha, d_A, N, d_B, N);
```

See: http://www.netlib.org/blas/strsm.f

Approaching Projects/STRSM/Case Studies

1. Parallelism?
   - How do dependences constrain partitioning strategies?
2. Analyze data accesses for different partitioning strategies
   - Start with global memory: coalesced?
   - Consider reuse: within a thread? Within a block? Across blocks?
3. Data Placement (adjust partitioning strategy?)
   - Registers, shared memory, constant memory, texture memory or just leave in global memory
4. Tuning
   - Unrolling, fine-tune partitioning, floating point, control flow, …

Step 1. Simple Partition for STRSM

```
__global__ void strsm1( int n, float *A, float *B )
{
 int bx = blockIdx.x;
 int tx = threadIdx.x;
 int j = bx*THREADSPERBLOCK + tx; // // one thread per column, columns work
  independently
 int JN = j * n;
 int i, k;

 for (k = 0; k < n; ++k) {    // ROW
   int KN = k * n;
   for (i = k+1; i < n; ++i) {  // ALSO row
     // B[i][j] -= A[i][k] * B[k][j] element depends on elts in ROWS above it in same col
     B[ JN + i ] -= A[ KN + i ] * B[ JN + k ];
   }
 }
}
```

Slide source: Mark Hall

Outline

- Two application case studies
- Molecular Dynamics Visualization
  - Read Kirk and Hwu, Ch. 9
  - Slide source: John Stone (excerpted)
  - Link: http://www.ks.uiuc.edu/Research/gpu/files/ece498lec21-22.pdf
- Material Point Method
  - Class project in 2009
  - Read "GPU Acceleration of the Generalized Interpolation Material Point Method," Wei-Fan Chiang, Michael DeLisi, Todd Hummel, Tyler Prete, Kevin Tew, Mary Hall, Phil Wallstedt, and James Guilkey, Symposium on Application Accelerators for High Performance Computing, July 2009.
  - Slides from SAAHPC 2009