CS 560: Take-home Midterm
Spring 2013

S. Rajopadhye, Colorado State University

The midterm consists of four questions. The total is 100, and there are three extra credit questions. The exam is open book/notes. You may look up information on the web, but you are expected to cite references and to work alone—no discussion with anyone else.

Part I: CUDA & GPU Computing [25 pts] In HW1 & HW3, you explored a couple of variants of the matrix multiplication program with a view to analyze performance bottlenecks. This question tests your ability to do similar analyses, but more importantly to think beyond problems that are posed for you. I'm using the same notation/terminlogy as I used in HW3.

We explored parallelizations where the footprint was 64×16 and each thread computed a 4×1 block. There was one parameter that was unspecified (the width of the block of A that was fetched, that most of you assumed was also 16). Moreover, you assumed that the mapping that decided which elements of C were computed by a given thread was the block strategy.

Problem I.1: Analyze the case when this is changed to a *quadrant* mapping where thread number $\langle x, y \rangle$ (x-th row and y-th column) is responsible for the elements in "four quadrants:" $\langle x, y \rangle$, $\langle x+16, y \rangle$, $\langle x+32, y \rangle$ and $\langle x+48, y \rangle$. What is the footprint of C that is computed by each threadblock? [1 pt]

How many ops does each thread perform in the innermost loop? [1 pt]

How many loads from *shared* memory does each thread perform? [1 pt]

Are these balanced? [1 pt]

Are there any bank conflicts (explain in detail)? [2 pts]

Problem I.2 Consider a variant where the block of A is 16×16 and the block of B is 16×64. Now, what is the footprint of C computed by a threadblock? [1 pt]

Again consider the two possibilities of allocating elements of C to threads (quadrants and blocks). For each of them, what is the set of elements of C that thread $\langle x, y \rangle$ computes: [2 pts]

Now, how many ops does each thread perform for each of the two strategies? [2 pts]

- **Quadrants:**
- **Blocks:**

Now how many loads from *shared* memory does each thread perform? [2 pts]

- **Quadrants:**
- **Blocks:**

Are these balanced? [2 pts]

- **Quadrants:**
- **Blocks:**

Are there any bank conflicts (explain)? [4 pts]

- **Quadrants:**
- **Blocks:**

Problem I.3 This is like the second problem in HW3 (bank conflicts). Consider the following loop from `matmultKernel00.cu` of HW1.

```
#pragma unroll
    for(int e=0; e<BLOCK_SIZE; ++e)
       Cvalue += shared_A[thread_row][e] * shared_B[e][thread_col];
```

We propose to change this to

```
#pragma unroll
    for(int e=0; e<BLOCK_SIZE; ++e)
       Cvalue += shared_A[thread_col][e] * shared_B[e][thread_row];
```

We also propose to change the final update to the `C` array from

```
Csub[thread_row * C.stride + thread_col] = Cvalue;
```

to

```
Csub[thread_col * C.stride + thread_row] = Cvalue;
```

No other changes are made. Does this code correctly compute matrix multiplication (justify your answer)? [3 pts]

Regardless of correctness, what happens vis-à-vis performance, especially in light of shared memory bank conflicts? [3 pts]

Problem I.4 (extra credit) Now for the challenge question (warning: this one's really dificult, don't worry if you don't get it). Recall that in HW3, the 64×16 footprint case *did not have* perfect balance. You weren't asked to implement this, but if you had, you would have discovered that it gives much better performance than even the 32×32 case (that one has perfect balance). The key question I want you to answer is the following. *Why is this so?* In order to better understand the issues, you are free to implement this (and the ones in Problem I.1 and I.2), but this is a thinking problem first and foremost, and you don't have too much time. [5 pts]

Part II: Equational Programming & Complexity Analysis [10 pts] This question revisits the problems that you saw in HW4 (writing, compiling and executing Alpha programs to measure the asymptotic complexity (degree and coefficient). If you like, you may reuse that code.

Problem II.1 Start with a simple one-liner program for square matrix multiplication that returns C=AB. Execute it and deduce the polynomial giving its complexity. Submit a brief report as a separate file, but in the line below, simply write the polynomial. [3 pts]

Problem II.2 Now modify it (or reuse the program from HW4) and report its complexity (be careful that you run experiments on the same machine as before; you may want to rerun experiments just to be safe). Write the polynomial in the line below. [3 pts]

Problem II.4 What is the ratio of the two, and explain why this is so. [4 pts]

Part III: Domains, Functions, Polyhedral Operations [25 pts] **Problem III.1** This question asks you to find the vertex repsesentation of a specific polyhedron.Revisit the Alpha code you wrote in HW4 (Problem2 part2) to multiply two triangular matrices but where the first one is upper triangular and the second one is lower triangular (computing C=UL). What is the context domain of the reduction body expression in the constraint form? You may either compute it by hand using

the bottom up rules, or simply report the value that AlphaZ gives when you do printAST [5 pts]

Problem III.2 Determine the vertices of this polyhedron. [15 pts]

The equation `X=(i,j,k->i-1,k,k)@A * (i,j,k->i-k,N-i+k)@B` occurs somewhare in an Alpha program that has a single parameter `N`. The domain of A is a cube $\{i,j,k|0 \leq (i,j,k) < N\}$, and that of B is a square $\{i,j|0 \leq (i,j) < N\}$.

Problem III.3 Write the function $f_1 =$`(i,j,k->i-1,k,k)` in matrix notation (`N` is the last row). Is it one-to-one (bijective) or many-to-one? In the first case, what is its inverse, in the latter case, what is its kernel (null space or share space). [10 pts]

Problem III.4 Write the function $f_2 =$`(i,j,k->i-k,N-i+k)` in matrix notation. Is it bijective or many-to-one? In the first case, what is its inverse, in the latter case, what is its kernel? [10 pts]

Problem III.5 (extra credit) Find new functions (not the identity) g, $f_1' =$ and $f_2' =$ such that $f_1 = f_1' \circ g$ and $f_2 = f_2' \circ g$ (i.e., find a common right factor). What does this factorization enable you to deduce about the variable `X`. [5 pts]

Part IV: Equations as Programs/Algorithms [25 pts]

Problem IV.1 Using mathematical analysis similar to the foundation notes, systematically derive an Alpha program to factorize a square matrix A into the product of an upper and a lower triangular matrix (A=UL, solve for U and L). This is a paper and pencil exercise, I want to see your math. [15 pts]

Problem IV.2 A banded matrix is one in which the non-zero elements are only in a (typically small) band near the main diagonal (i.e., when $-w \leq i - j \leq w$, for some $w \ll n$). Mathematically prove that the LU factorization of a banded matrix is also banded and determine the size of bands of L and U. [10 pts]

Problem IV.3 In the Foundations notes, the rule for determining the image of a polyhedron by an affine function was simply stated without any proof. Prove this claim for *bounded polyhedra* (polytopes) by using by using the definition of the generator representation (Definition 5). [5 pts]