CS 6230/5960

Programming Assignment 3

Due midnight Monday, 4/13/2020

You are to implement a CUDA kernel for transposed matrix multiplication $C=A^TB$. For the purpose of this assignment, you can assume that the problem size is a perfect power of 2 (i.e., boundary check for threads is not necessary).

```
#define N (1024)
float c[N][N, b[N][N], a[N][N];

for(i=0; i<N; i++)
 for(j=0; j<N; j++)
  for(k=0; k<N; k++)
     c[i][j] = c[i][j] + a[k][i]*b[k][j];
```

1. (20 points) First implement a base parallel version that directly accesses the elements of array A from global memory. Use a square 2D shape for thread blocks.

2. (20 points) Next improve performance by using shared memory.

3. (20 points) Perform 2-way loop unrolling along i, for the shared-memory version.

4. (20 points) Perform 2-way loop unrolling along j instead, for the shared-memory version.

5. (20 points) Perform 2-way loop unrolling along both i and j, for the shared-memory version.

6. (Optional extra credit: 50 points) Use any transformations to achieve even higher performance.

Template code will be provided, along with target performance numbers. Since CHPC has several different GPUs on nodes of their clusters, and the wait time to get an allocation on a node with a specific GPU type can be very high, target performance will be provided for each possible CHPC GPU type. Thus, you can ask for any GPU node and work with whichever type of GPU you happen to get.

Submit code as well as a report documenting achieved performance.