CS6023: GPU Programming

Assignment 2 (7 marks)

Due `August 27 / September 3, 2017 by 23:55` on Moodle

Problem Specification

Implement a `CUDA` kernel for sorting an array of integers using *parallel merge-sort*.

Sorting an array of integers: You have an array of integers $A[0..n-1]$ which you want to re-order such that the following property holds $\forall i \in \{0, \cdots, n-1\}$, $n \in \mathbb{N}$, and $A[i] \in \mathbb{Z}$:

$$A[0] \leqslant A[1] \leqslant \cdots \leqslant A[i-1] \leqslant A[i] \leqslant A[i+1] \leqslant \cdots \leqslant A[n-1]$$

Merge operation: You have to implement the *merge* operation in *parallel*, the steps of which are illustrated through an example in Figure 1.

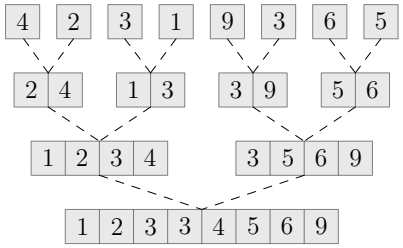

Figure 1: Merge operation

Merging is performed level-wise starting from individual elements placed in sorted lists of size 1. Each level reduces the number of lists by half by merging two adjacent lists of the preceding level. The process terminates at the last level when only one list remains.

You are provided a tarball containing:

1. **kernel.cu**: Contains only the kernel signature. Your task is to implement the kernel here.

```
__global__ void msort(int *d_input, int* d_temp, int N){
     // Implement your kernel here
}
```

d_input will initially contain the unsorted input array of size `N`.
d_temp is an auxiliary array of size `N`.
`The final sorted array should be in` d_input.

Assume the kernel `msort` to be invoked with `N` threads using 1-D thread mapping.

If your program needs any extra memory (apart from kernel parameters: `int *d_input, int* d_temp, int N`) inside the kernel, you may do so by declaring global-scope device variables outside the kernel, in `kernel.cu`. For example, the following code declares an array `my_array` of 50 integers in the host code and space is allocated in the global memory of the device. This array may be used in the device code, in `kernel.cu`.

```
__device__ int my_array[50]; // This is part of the host code

__global__ void msort(int *d_input, int* d_temp, int N){
    ...
    _ = my_array[i]; // A sample use of my_array in the device code
    ...
}
```

Note: Do not change the kernel's signature or its name. For this assignment, we will not provide any other code. For testing your code, you need to write your own `main.cu`. We will use our `main.cu` for evaluations.

Submission Instructions

When ready to submit,

1. Rename the file kernel.cu to ROLL_NUMBER.cu.
   For example, if your roll number is `CS14D406`, your file should be called `CS14D406.cu`
2. Upload ROLL_NUMBER.cu on moodle: `https://courses.iitm.ac.in/course/view.php?id=837`
3. Download your file, and make sure it was the one you intended to submit.