

Project #4 - Counting Sort using CUDA

SWE3021 Multicore Computing - Fall 2019

Due date: December 13 (Fri) 17:00pm

1 Counting Sort

In this assignment, you will implement a parallel counting sort algorithm in CUDA. Counting sort is a non-comparative sorting algorithm, which can be easily implemented in C function as follows.

```
void counting_sort(int arr[], int size, int MAX)
{
    int histogram[MAX];

    for (i=0; i<MAX; i++){
        histogram[i] = 0;
    }

    for (i=0; i<size; i++){
        histogram[arr[i]]++;
    }

    for (i=0; i<MAX; i++){
        for (j=0; j<histogram[i]; j++){
            cout << i << " ";
        }
    }
}
```

Skeleton codes are located in /home/swe3021/project4. Note that you need to implement all CUDA functions in counting_sort.cu file since it is the only file that you will submit. Driver files will be changed to measure the performance of your code with different inputs. You are free to change the driver code for testing purpose.

```
yourid@titan1:~$ cp -r ~swe3021/project4 .
yourid@titan1:~$ cd project4
yourid@titan1:~/project4$ ls
counting_sort.cu  driver1.cpp  Makefile
```

In order to compile, simply run make command as follows.

```
yourid@titan1$ make
```

2 Grading

Again, your project score will be given as follows.

1. If you do not submit, you will get 0 point.
2. If your code does not compile, you will get only 1 point.
3. If your code compiles and runs CUDA kernel functions but outputs are incorrect, you will get only 2 points.

4. The remaining 8 points are the performance score, which will be prorated by the following equation.

$$\frac{(\text{the slowest code's execution time}) - (\text{your code's execution time})}{(\text{the slowest code's execution time}) - (\text{the fastest code's execution time})}$$

This project will not require a significant amount of your time to implement. Note that you can even find some counting sort implementations in CUDA from github. But please do not look at those implementations. If you do, your code is likely to be similar to that and my "cheating detection program" may catch it. Again, this project is easy and I believe this will be a good practice for those who never wrote a CUDA program before.

3 How to Submit

To submit your project, you must make sure that your file is `counting_sort.cu` and run `multicore_submit` command in your project directory in `swin.skku.edu` server.

```
$ multicore_submit project4 counting_sort.cu
```

For any questions, please post them in Piazza so that we can share your questions and answers with other students. Please feel free to raise any issues and post any questions. Also, if you can answer other students' questions, please don't hesitate to post your answer. You would get some credits for posting questions and answers in Piazza.