

Due: 29-12-2017 by 23.59

Homework 3

The purpose of this project is to get you familiar with GPGPU and CUDA.

The main problem in the project is very similar to homework 2, where decoding a message from a given encoded file (encodedfile.txt). The decoding process in this homework will be a little bit different as follows; specific character is comma (,) and in each line the next character after commas are the secret message characters.

For example:

Ern,basd,b3.,msd
324,4!...;4erwer,2

Should return "bbm442"

Write a program that processes the decoding stage both in serial on CPU and parallel on CUDA.

For parallel version the file resides in the CPU side and CPU is responsible for reading the file and sending the lines and inputs to GPU.

After all process done, the final formed hidden message should be available in CPU side and write to the output file named decoded.txt as a one line of message.

For simplicity, in the input file, each line will be 100 characters long, there will be exactly 4 commas per line, commas will not be the last character and there will be 15360 lines in the encodedfile.txt. You will be given the input file before the deadline to test.

Report: You also need to write a serial code of the program that runs on CPU as well along with the CUDA version, and compare their timing. In CPU timing include only the decoding part, in GPU part include memcpy, kernel run and memcpy back. Do not include memory allocation, file reading and file writing to your timing.

For timing you can use time.h and gettimeofday() function.

For usage see example below

```
#include <time.h>
#include <sys/time.h>

struct timeval stop, start;
gettimeofday(&start, NULL);

//do work to time

gettimeofday(&stop, NULL);
float elapsed = (stop.tv_sec - start.tv_sec) * 1000.0f + (stop.tv_usec - start.tv_usec) / 1000.0f;
printf("Code executed in %f milliseconds.\n", elapsed);
```

Include the time results in your report for both serial and GPU version and print on the screen when the program finishes.

Notes

- What to submit:
 - Source code (.cu file that includes both serial and parallel code)
 - Makefile
 - decoded.txt
 - README.pdf file
 - README file should include the information about compiling and running, timing info, as well as the design choices you make such as how CUDA threads divide the job
 - compress all as a zip file
- Turn in the compressed file via **submit system**.
- Don't forget to comment your code for readability. Code without any documentation will lose 5 points
- This is an individual homework