CS550 Design Document (PA#4)

Cyril Trosset A20316209 Section 1

CS 550

Introduction

In this assignment, we have to design a program that reproduces the grep UNIX function. But we have to do this using GPU capabilities with the CUDA framework.
The grep function consists of printing lines which contain a particular word.
For example, if you input test.txt hello, the program has to print all the lines in test.txt file that contain the word hello.

Design

Overall design

My program consists of a main loop which loads the file into an array (host memory). When the number of lines reaches a maximum integer, it sends the array to the device memory and start the kernel (massive parallel job).

The kernel part treats the data and returns the index of lines which contain the word. It handles 4 regular expressions.

Then the host prints the results and goes on.

Host part

In the host part there are few major steps :

1. First it copies the searched word into the device memory.
2. It allocates a big array which will contain a big number of lines.
3. It concatenates the lines in a single array and fills an array stocking the index of each lines with the following loop :

LOOP :

Each line reading, I insert the characters of the new line in the main array, and store in the Index array the index for which the line starts. If the number of lines reaches a maximum value it sends the work to the GPU with the following routine :

I first used strcat/strlen to copy the lines into the big array, but I figured out that it runs much faster by copying characters by characters.

SEND TO GPU :

First, I copy the main array to the device memory. And I copy the index array too.

Next I allocate a result array which consists of bool value for each line (true if the word is matched, false if not).

I run 1024 blocks of 1024 threads maximum. (Kernel described below)

Then I wait for the kernel job to finish, free the memory, copy the results array from the device and print the results.

When this is over the program get back to the main loop.

Kernel part

In the kernel part, the row index corresponds to the x value of thread ID plus the block ID multiplied by the grid dimension. Then I go through the array and if the word is matched it sets the Result array to true and stop.

Moreover the kernel part is able to handle 4 regular expressions :

1. ^ : match the beginning of a line. For example **^Hello** will return only the lines which start by the word Hello.
2. * : match any characters between the letters without number constraints. For example **H*o** will return Hello, Hallo, Heo, …
3. . : match any characters just one time. For example **H.o** will match Heo, Hao, Huo but not Hello.
4. $: match the end of a line. For example **Hello$** will return only the lines which end by the word Hello.

Enhancements

Future enhancements can be the following :

- Add more regex recognition.
- Read the file with a lower level method than fopen().
- Implement a better algorithm to match the word more efficiently (for example divide and conquer algorithm).
- Better CUDA gestion (block/grid/thread optimization)/(memory copy optimization)