GPU Computing Assignment 2

Siddharth Yadav | 2016268

Introduction

In this assignment, we do pattern matching on GPU. Given a list of keywords and a text file, we find the frequency of each keyword in the text file.

Strategy

Implementation 1

NWORDS -> number of keywords we need to find.
Each block is 1-D with the size of the number of keywords that we need to find. This means that BLOCK_DIM = (NWORDS,1,1). Each thread in the block is responsible for finding the number of the number of occurrences of one keyword(i.e assigned to the thread) in the text loaded into the shared memory of that block. In the end, the occurrence number is updated atomically to global memory. Each thread loads text from global memory using titling. That's each thread loads TILE_SIZE+TILE_SIZE(to handle corner cases) elements from the global memory of text.

- Launch kernel with following config:
  - Grid Size: (Length Of Text)/(TILE_SIZE*NWORDS)
  - Block Size: NWORDS
- load `text` into shared memory using titling
- assign one keyword to each thread in a block
- iterate through all memory location in the shared memory corresponding to the block to find the word count in that block. In other words, each thread will go through TILE_SIZE*NWORDS words to match a keyword and update a local sum counter
- Atomically update add the sum counter to a global array

Implementation 1

We use an 8-way stream for copying data with a title_size = 1. Note other n-way streams won't work. Note: Code uploaded later.

Result

Execution Time(In Seconds)	Small	Medium	Large
cpu	0.463011	1.208597	2.394089
kernel only	0.000241	0.000566	0.001103
kernel + memcpy	0.001623	0.003839	0.007573
(kernel+memcpy) stream with 1 title_size	0.001060s	0.002685s	0.005204s

	Small	Medium	Large
cpu	1	1	1
kernel only	1921.207469	2135.330389	2170.524932
kernel+memcpy	285.2809612	314.8207867	316.1348211
(kernel+memcpy) stream with 1 title_size	~450	~450	~450