Map Reduced Lab Report

**What problems you encountered completing the assignment and how you overcame them.**

At first I was opening and reading the Shakespeare files and the comparing wasn’t being successful because I hadn’t converted the words into all lower case.

Another issue is that I hadn’t used p.iterate(), which is the command that helps manage the thread executions times to iterate so the threads wouldn’t crash with each other.

**Any problems you couldn’t overcome, or any bugs still left in the program.**

I had used python in the past but I’m not super familiar with all the libraries, I have one major issue in my lab, I don’t think I’m correctly calling my list of words and the number of words are not being printed correctly.

**About how long it took you to complete the assignment.**

Creating the program was a little challenging, I spent a lot of my time planning on how I was going execute my program. Once I had my program running, the next challenging part was trying to understand how the program was supposed to run.

In total I would say it took me around 4-6 days fixing up the code, making sure it ran, and creating a detailed report about the specific process I followed testing the multiple threads.

**Performance measurements (given in seconds) for 1, 2, 4, and 8 threads.**

Thread 1:

1.0764000080598635

Thread 2:

1.0645000202202937

Thread 4:

1.0574000043561682

Thread 8:

1.0595200001247576

**A short analysis of why the program behaves as it does with an increasing number of threads**

For this program I had a word\_list holding the key words we were searching in the files, and shake\_files holding the Shakespeare files the program would iterate through. I tested the different threads and noticed a small decrease in execution time as the threads increased.

Although the time difference between the threads (1, 2, 4, 8) was small (due to the size of the program), it was evident that the use of more threads was increasing the speed of calculation execution. When the program was running on a single thread it took the longest around 7 seconds, but as I increased the number of threads up to 8 threads, the program took around 5 seconds to execute.

The idea behind using multiple threads is the ability to compute the same tasks at the same time instead of sequentially to increase performance. One thing to keep in mind is that the case isn’t always the more threads the better, the number of threads must be chosen considering what the physical system can support. More than 8 threads in this case wouldn’t have made a difference as they are not supported under the single core.

**Any observations or comments you had while doing the assignment**

This assignment showed us the basics of parallel programming in a smaller environment, I would be interested in seeing the way programs are altered by complexity and size and how the use of multiple threads might affect the process.

**Output from the cpuInfoDump.sh program**

