# **Cristobal Chinolla**

# **How to Run:**

mpirun -n <# process> python3 mapreduce\_MPI.py

# **Problems Encountered:**

The main problem I encountered was testing. Because I am working on two different machines (one to write the code and the

other to test in the VM (cuz i have m1 mac))

I could not test as frequently as I would have liked.

I encountered a lot of errors with testing, I had errors with the print statements, then with importing the MPI library.

# **Time Spent:**

because this lab built off the last lab, it did not take as long as I thought it would.

The examples provided were really helpful. Total time spend ~ 4 hours

# **Performance Measurements:**

16 threads - 63.72s

Text

Description automatically generated

8 threads - 31.82s

Text

Description automatically generated

4 threads - 30.25s

Text

Description automatically generated

3 threads - 28.64s

2 threads - 24.49s

Text

Description automatically generated

1 thread - 37.08s

# **Analysis:**

My implentation using MPI over OpenMP took significantly longer to finish the same task. I am not sure if it is because of

MPI or if my implementation was just inefficient (probably the second one)

As far as the analysis for MPI, I noticed that if the number of threads was greater than the number of files (8), this actually

increased the total time to complete. I assume this is because the excess amount of threads have nothing to do and increase the

overhead time.

I also noticed a pattern. When the threads:amount of files reached a specific ratio, (for example 2 threads), where there were

less threads and each thread had an even amount of files. The run time was lower.

# **CPU Info**

Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz

4 36 216