**Assignment 5 – Part 1. Concurrency**

*By Divit Koradia, Ethan Griffee*

Introduction

In the context of computer science, concurrency, is the ability for a program to be decomposed into parts such that they can run independently of each other. This means that tasks can be executed out of order and the results would still be the same if they are executed in order. The idea is similar to parallel processing, but with the possibility of many independents’ jobs doing different things at once rather than executing the same job. Thus, one of the key benefits of implementing concurrency is to reduce the time it takes for running and executing a program.

In order to test this claim, we designed an experiment to measure the performance increase of **pmap** over **map** – using the performance on map as a baseline. Specifically, we are trying to test if the **pmap** function, in general, is faster at executing simple calculations on the rows of a data frame in comparison to the **map** function.

Implementation

Analysis performed

In order to test performance differences between **map** and **pmap,** we did the following:

1. Used a for loop to create a synthetic datafile.txt with approximately 3.5 million rows and 11 columns.
2. Implemented 4500NE teams’ SoReR by replacing their columns with the columns in our data frame so that we could read the datafile.txt and populate our data frame.
3. Once we had populated the data frame with information, we created two different Rower subclasses to test the behavior and implementation of our **map** and **pmap** function.
   1. Rower1 – is a dumb rower which counts up to 10,000 for every row it parses and then returns the sum of counts for each row in the data frame.
   2. Rower2 –
4. For testing purposes, we choose to run each of the rower subclasses with our **map** and **pmap** function for three different values of length (number of rows we read from the datafile.txt) to record the CPU time it took to execute the code. We repeat this process three times for accuracy and also because we plot the averages in our graphs. Lastly, we run the same test on a different computer for a holistic viewpoint of how our functions perform across different software and hardware specifications.
5. We used our personal laptops as test benches. Both of us had the latest version of macOS Catalina (v 10.15.3), however, our hardware configurations were different.
   1. Ethan’s laptop is running a 2015 2.2 GHz Quad-Core Intel Core i7 with 16GB of ram.
   2. Divit’s laptop is running a 2019 2.4 GHz Quad-Core Intel Core i5 with 8GB of ram.