Brandon D'Anna
108470479
AMS 598 Project 1 Report

In this assignment, I implemented the Map/Reduce concept to count frequencies of integers from 0-100 across 20 data files using 4 processes on the SeaWulf HPC and provide a final output of the top 5 highest frequency integers with their corresponding frequencies. To do this, I read in all of the file names as a list using a wildcard. That list was then sectioned into 4 sublists.

The one mapper function, named CountFrequency, takes as input a list with file names and the rank of the process the function is being run on. It opens each file, imports the integers into a list, and a data frame is created using the integer list and a column of static ones. The data frame is then aggregated to sum the counts of the integers. Lastly, the data frame is subsetted and the subsets are written to 400 text files. 100 files per process, each with one individual key in the format "m"[mapper number]"k"[integer key value].txt.

The MPI functions send and receive are then used to send and receive the sublists of data to the respective nodes. Since only 4 nodes were used, the master node was utilized to process the first sublist and the remaining 3 lists were sent to the 3 respective worker nodes. The Barrier.() function was used to ensure all mappers completed before the reducers read in the results from the hard disk.

The one reducer function, named ReduceNumbers, takes as input a start integer, a stop integer, and the rank of the process the function is being run on. The function loops through each of the mapper output files and reads in only those containing a key found in the range of the [start:stop] input variables. For the files containing a valid key, each is read in and appended onto an empty data frame. Again, the data frame is aggregated on the count column. The top 5 highest frequency values are then outputted in the format "r"[reducer number].txt, for a total of 4 files.

Lastly, only the root node is used to read in the 4 reducer output files and append the 4 data frames. The top 5 values are then outputted, producing a data frame of the overall top 5 integer values and the corresponding frequencies. This data frame is saved to the res.txt file.