**Micro-Processors Report Milestone 3**

in this milestone we started using broadcast and reduce rather than using send and receive,

the percentage of the parallelized code on the source code level is around 20%

the use of parallelism in our code is just the matrix multiplication part of the code

hence the remaining part of the code which serves the rest of the functionalities of the code is not parallelized

however out of the matrix multiplications we can say around 70% of the work if not more can be parallelized since we give every process a pointer to work with and then send the results back to the root process to do combine the answers together.

The main difference between using broadcast and reduce rather than send and receive is the fact that with send and receive you need to do it separately for every process rather than sending once from the root to every other or reducing from all the processes at the root

so we don’t have to use a for loop to receive from/send to all processes we can simply use broadcast/reduce to reduce the lines of code used

both broadcast and reduce can be used in the code

broadcast to send to all processes stuff like when to start the input received from the user and anything needed to be initialized at all nodes

reduce can be used when doing the dot product as concatenating the output array into an array of array for the input, simply as if we’re doing gather. However it’s a hassle to try to create the function ourselves since we need to know who’s array is sent first and define the operation for all processes

so we used broadcast in our implementation however we chose not to implement reduce since the operations supported doesn’t help in concatenating arrays into an array of arrays