WSB- Wyższa Szkoła Bankowa w Poznaniu

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Parallel Programming- Algorithms and Techniques**

**Matrix Multiplication**

**Project Report**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Benjamin Ngarambe**

**Introduction:**

The principal purpose of this project is to perform various operations on multiprocessing and record the time taken for computation with serial computation on the same functions. For this work, I used matrix multiplication as it requires NxN operations to compute if N is the matrix size.

**Results:**

A screenshot of a computer

Description automatically generated

**Working:**

In the code, initially, it randomly fills the arrays, then it breaks the array evenly according to the number of processes, that is row = array size/number of processes. It will also set an offset(cancel) variable which will tell from which index the values of matrix A will be sent. We then send the offset, rows, array A, and the entire array B among all the processes. Then it will compute the multiplication and store it in array C. It will finally perform the send function, which will send back the resultant matrix C to get the result.

**Conclusion:**

In the beginning serial execution was quicker than MPI but as the size of the matrix increases the performance of serial execution decreases while MPI performance increases. Hence for greater data and computation multiprocessing becomes faster than serial computation.