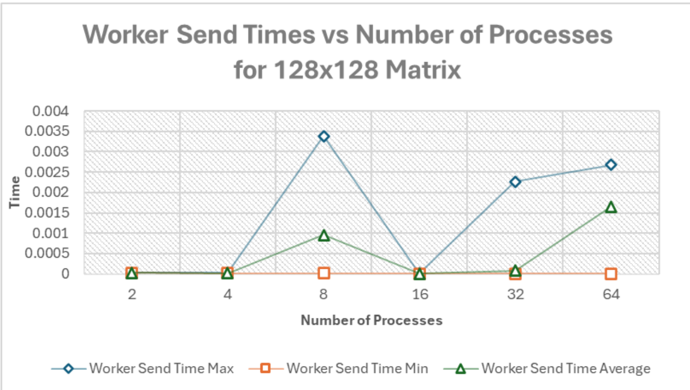
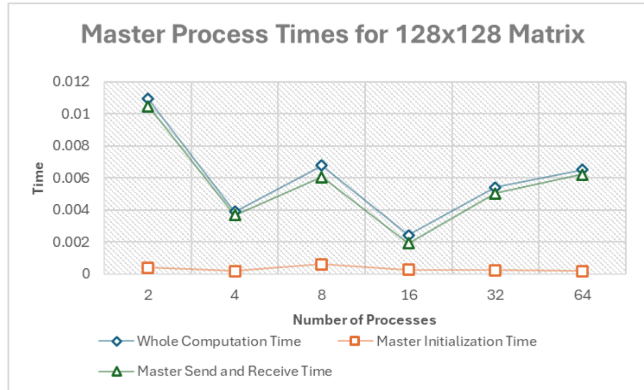
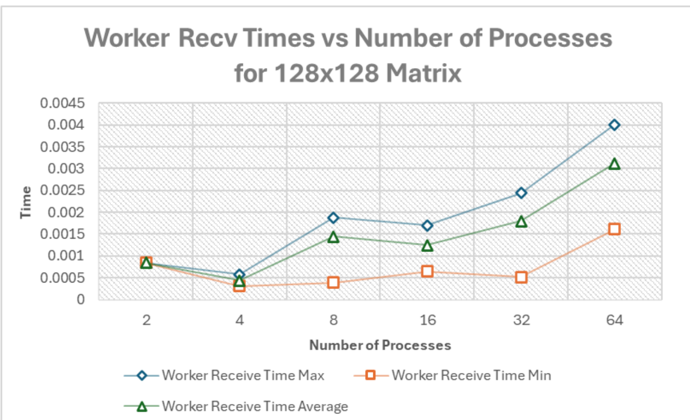
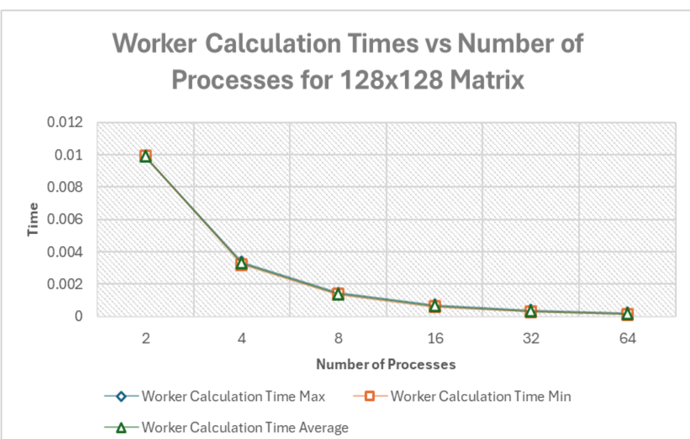
Rahul Singh - 132002363

Plots for 128x128 Matrices Multiplication





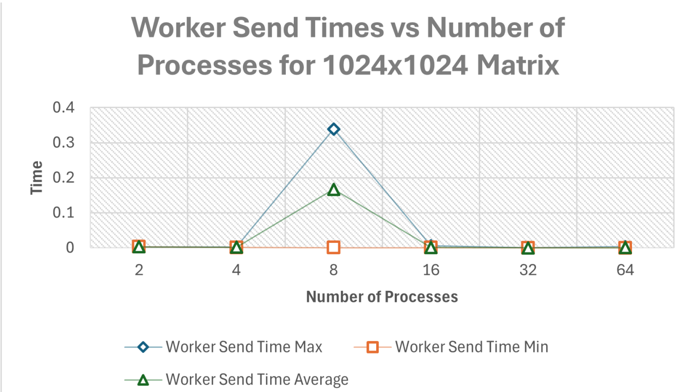


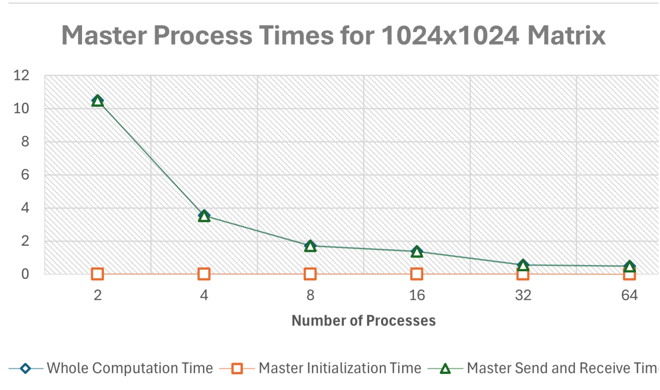


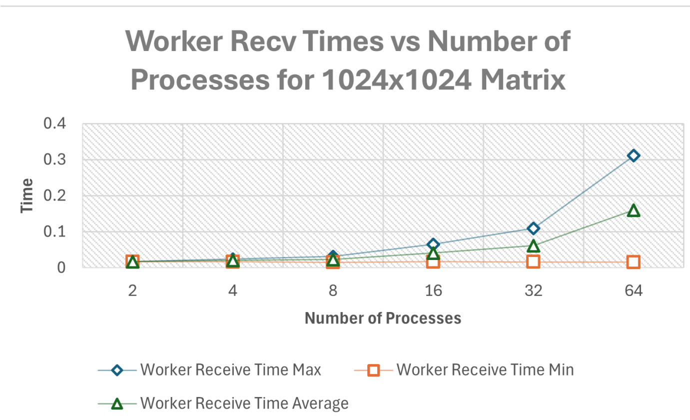
Observations: The 128x128 matrix calculation initially shows improved performance when increasing from 2 to 4 workers, indicating successful parallel processing. However, as the number of workers grows beyond 16, the benefits of parallelization diminish and overall job completion time actually increases. This performance decline likely stems from rising communication costs between workers and the master thread, which eventually outweigh the advantages of distributed computing for this relatively small matrix size. The observed pattern suggests that the overhead of coordinating numerous processors becomes counterproductive past a certain threshold for this particular problem scale.

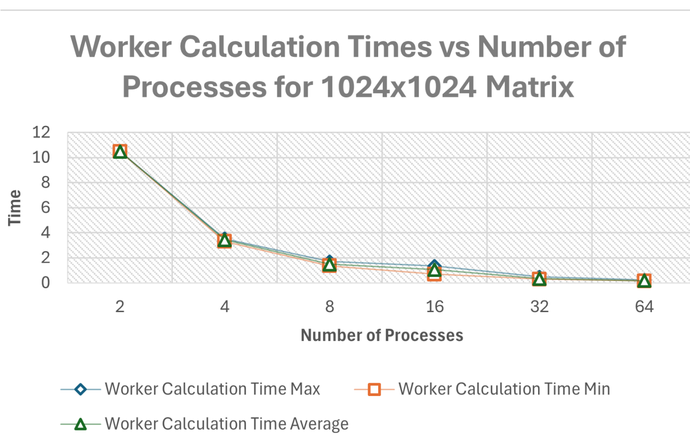
Rahul Singh - 132002363

Plots for 1024x1024 Matrices Multiplication





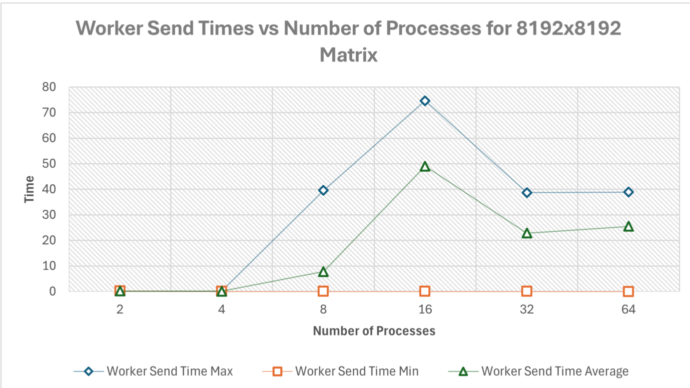


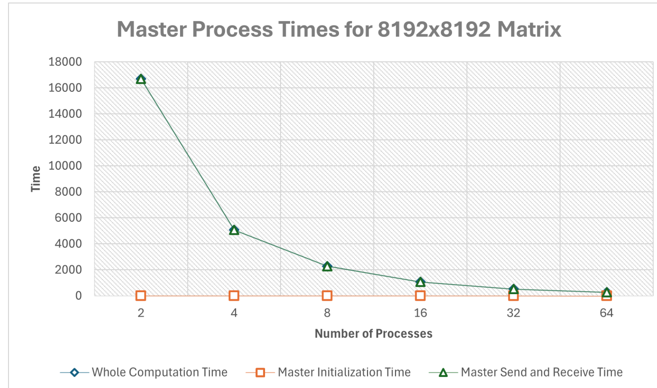


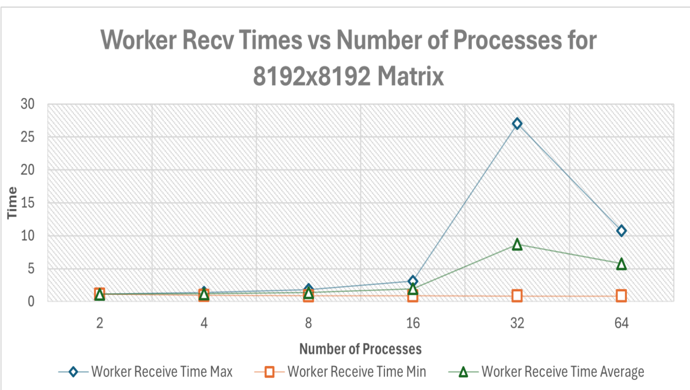
Observations: The 1024x1024 matrix exhibits a more typical scaling pattern, demonstrating substantial efficiency improvements as the worker count rises, particularly in the transition from 2 to 4 workers. Performance continues to enhance as more processors are added, albeit with diminishing returns, following an asymptotic trend. This behavior indicates that the parallelization approach is well-suited for medium-sized matrices. The gradual leveling off of performance gains at higher worker counts illustrates the inherent limitations of parallelization for a problem of this specific size. It highlights the point at which adding more processors yields progressively smaller benefits due to the limits of parallelization for a given problem size.

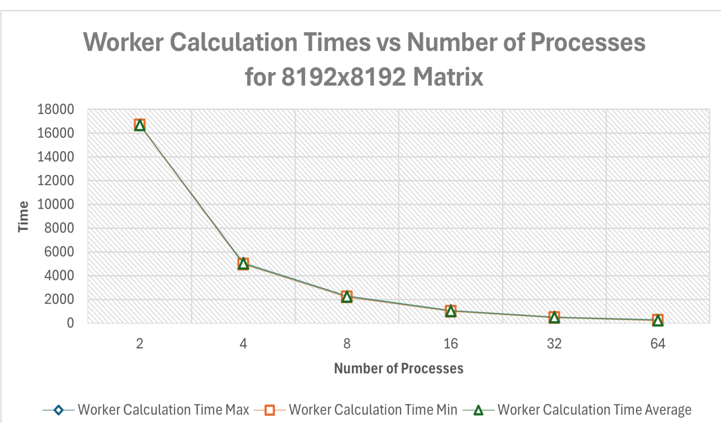
Rahul Singh - 132002363

Plots for 8192x8192 Matrices Multiplication









Observations: The 8192x8192 matrix multiplication shows excellent scaling, with performance improving significantly as the number of processes increases. Total computation time drops from about 16,690 seconds with 2 processes to just 272 seconds with 64, indicating highly effective parallelization. While there is some variability in worker send and receive times, suggesting minor communication overheads at higher process counts, these do not significantly impede overall performance gains. The consistently low master initialization time across all process counts further emphasizes that for this large matrix, the benefits of parallel processing far outweigh any associated overheads. This pattern illustrates the effectiveness of the parallelization strategy for large-scale matrix operations, where the problem size is sufficient to fully utilize the increased computational resources.