**Non-Blocking MPI**

**Final Project**

**CPEN/CPSC 435**

# Introduction:

# Parallel computing allows for partitioning and disturbing data sets to multiple processes, allowing for increased processing performance. Compared to a traditional single program, single process platform a single program multiple process reduces latency wait times. Instead of having to wait for sequential computations to finish, independent functions can run concurrently.

# Using a master slave model, this project will implement a message passing interface utilizing a non-blocking send and receive structure.

# Implementation:

# Performance Analysis:

* Fixed matrix of N512, scalability of the program? (# of nodes speedup)
* Change matrix “Size”….. \*note different performance changes.
* Compare to Assignment #6 describe how blocking or non-blocking affect performance, \* try MPI\_sendrecv( ) vs MPI\_send() & MPI\_recv() compare performance

GIVE TABLES FOR ALL DATA w/ supporting paragraphs explaining what was changed, and how the change affected the performance.

# Discussion and Conclusion:

* Relevant issue or difficulties, as encountered in context with parallel computing.
* Make a conclusion as to why we would chose a blocking/ non- blocking to be better, explain use data comparisons.

# Contributions:

* Mathew Lee,
* Trevor Greenside,
* Thomas Hughes’s contribution to the project came from the project write up, presentation power point slide.