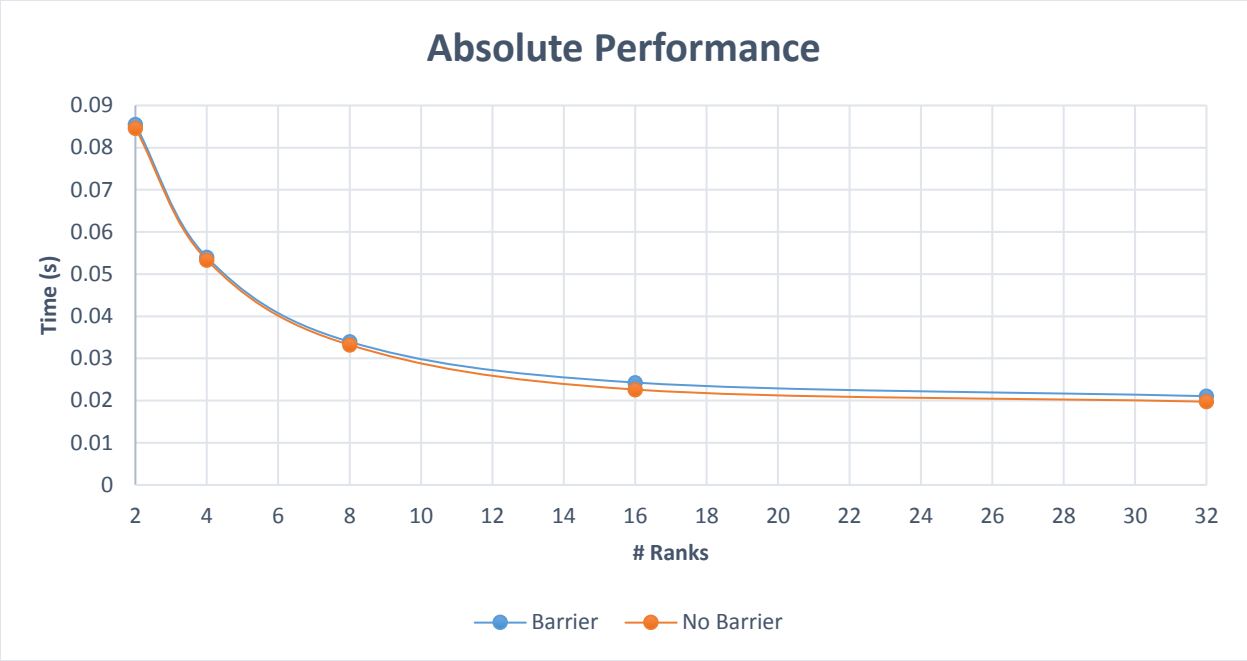
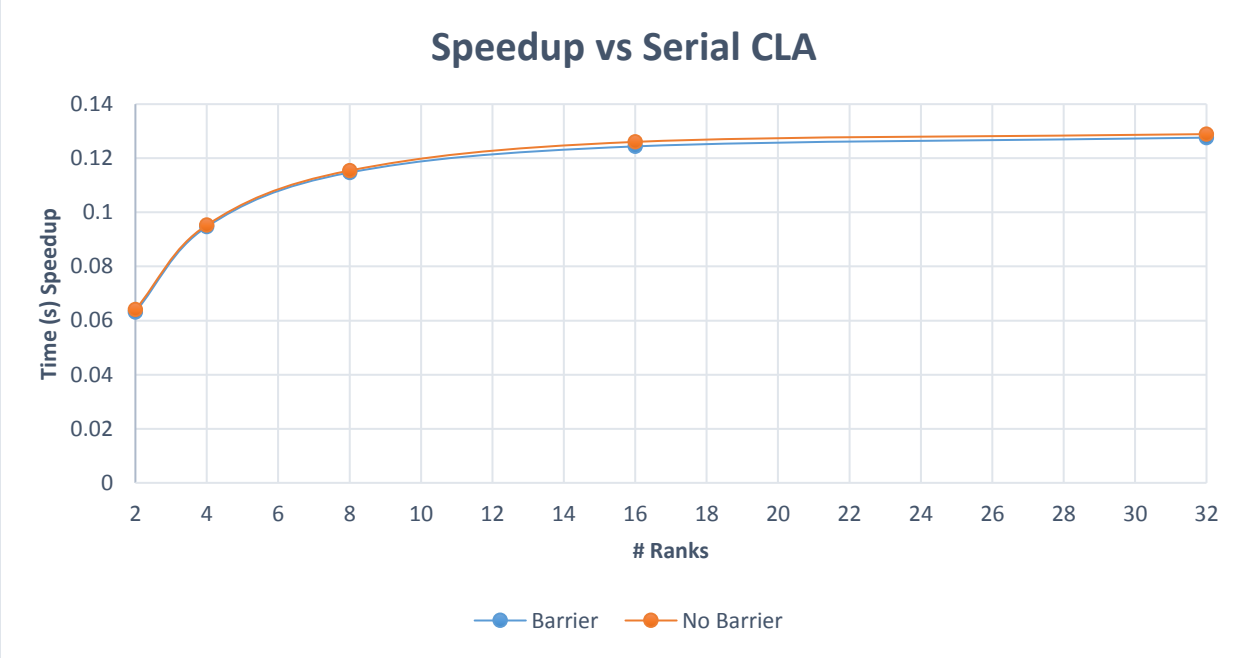
Ryan Stillings

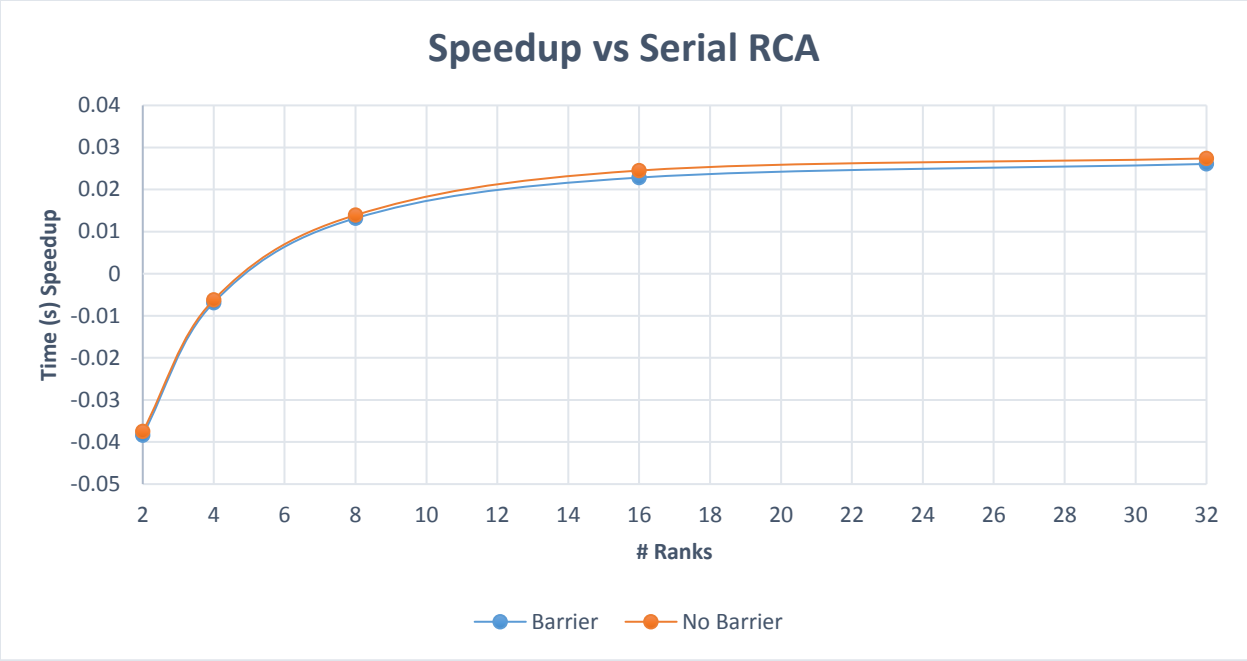
Parallel Programming and Computing

Assignment 2 Performance Report

Performance Graphs







Raw Data

Run Configuration	Execution Time
RCA 1 rank (Serial)	0.047140s
CLA 1 rank (Serial) + barrier	0.148674s
CLA 1 rank (Serial)	0.149581s
CLA 2 ranks + barrier	0.085501s
CLA 2 ranks	0.086583s
CLA 4 ranks + barrier	0.054008s
CLA 4 ranks	0.053359s
CLA 8 ranks + barrier	0.033950s
CLA 8 ranks	0.034215s
CLA 16 ranks + barrier	0.024287s
CLA 16 ranks	0.022622s
CLA 32 ranks + barrier	0.021053s
CLA 32 ranks	0.019752s

Analysis

The results that I observed are in keeping with the fundamental performance implications behind supercomputer technology that we have discussed in class thus far. The first striking result is the fact that across all ranks, executing the program with a call to MPI_Barrier between each step of the algorithm resulted in very slightly longer runtimes. This is unsurprising, since each call adds a bit of overhead with no real benefit in this instance. After all, the ranks will largely catch back up to each other when distributing the sscl carry bit regardless, and will catch back up entirely when the full result array is gathered by rank 0.

The curve of all 3 performance graphs indicates that performance improves each time the number of ranks used is doubled, but by a smaller percentage margin each time. This is unsurprising, as there are bits of overhead (the speed of function calls in C, the inherent delay in MPI message passing, the cost of gathering from MPI_COMM_WORLD, etc..) that pose theoretical limits to the maximum benefit achievable via parallelization.

The speedup vs a single rank (serial) CLA run shows us just how impactful the addition of the first few cores is, with improvement beginning to really stagnate after 8 cores. Finally, the speedup vs a serial RCA run does not show an immediate benefit. Rather, the performance of CLA is worse than RCA at just a few cores, intercepting the performance of RCA at around 5 cores, with additional cores pushing CLA to perform more quickly than RCA. This too is unsurprising, as the serial RCA does not suffer from any of the overhead from multicore processing and message passing in MPI, or from the increased complexity of the CLA algorithm, allowing it to perform better than its parallel equivalent running on just a few cores.