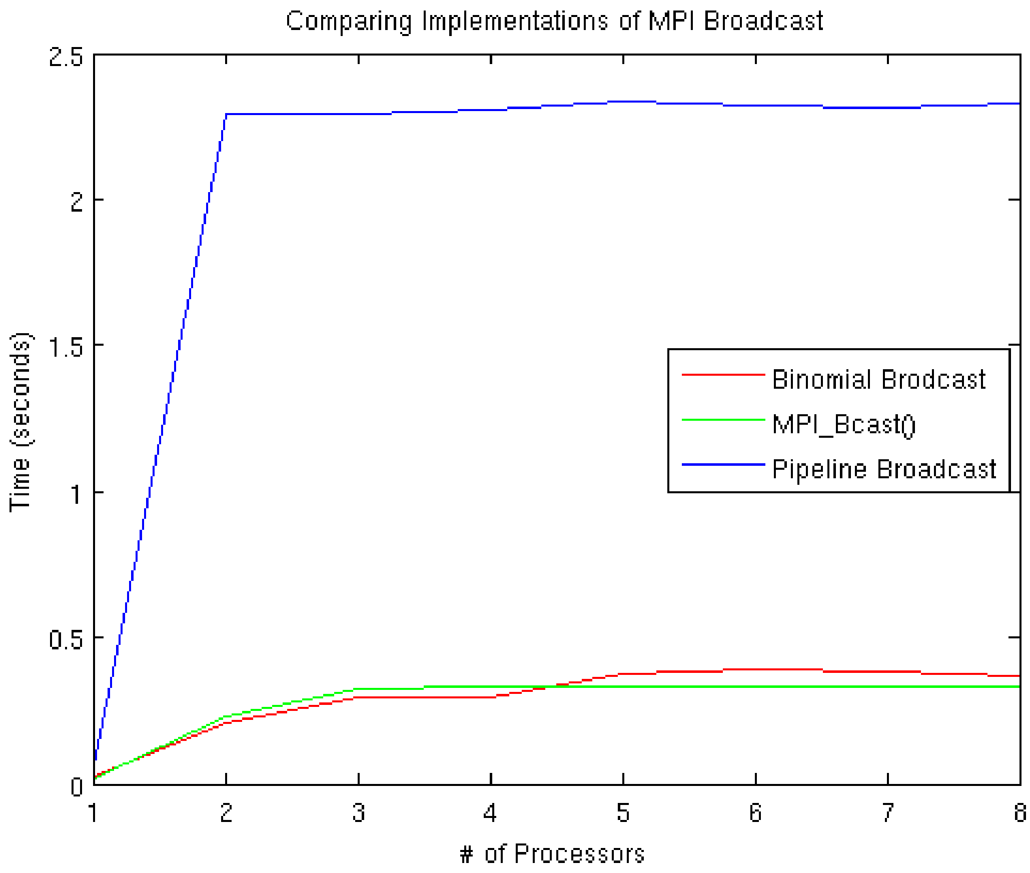
Trevor Whitney **CSCI-563**

Homework #3



1. The performance of my version of MPI Broadcast, using the Binomial Tree algorithm, is represented by the red line in the above graph. It performed comparably to the built in MPI_Bcast() algorithm, which is represented by the green line. As the number of processors was increased, the performance of my algorithm became worse than that of MPI_Bcast(). One reason for this could be optimizations built into MPI_Bcast() to slightly reduce communications. I suspect this since when less communications were required because fewer processors were used, my algorithm performed better. All three algorithms were run on 1-8 processors, and broadcast an array of 2,621,440 arbitrary integers.

2. My version of MPI Broadcast using the Pipe-lining algorithm is represented by the blue line in the above graph.

   a) $p-1$

   b) $n+p-1$

   c) One step: $\frac{p(\alpha+\beta m)}{n}$ All steps: $p(\alpha+\beta m)$

   d) The optimal number for n depends on the operating system. Ideally you want to minimize communications, so you want to break up m into the largest datatype you can. For example, since chars are small, instead of breaking m up in n chars, it would be better to break up m in n ints. This would allow you to pass 4 bytes at a time opposed to 2. Using this ideal number for n, the broadcast time would be: $p(\alpha+\beta m)$

   e) There is no e

   f) See the above graph and included source code. The Pipeline algorithm is represented in blue.