CS596: Assignment2

code

```
#include "mpi.h"
#include <stdio.h>

int nprocs;  /* Number of processes */
int myid;    /* My rank */

double global_sum(double partial) {
  /* Write your hypercube algorithm here */
  double mydone, hisdone, partner;
  int bitValue;
  MPI_Status status;

  mydone = partial;
  for(bitValue=1; bitValue<nprocs; bitValue*=2){
    partner = myid ^ bitValue;
    //send mydone to partner;
    MPI_Send(&mydone,1,MPI_DOUBLE,partner,bitValue,MPI_COMM_WORLD);
    //receive hisdone from partner;
    MPI_Recv(&hisdone,1,MPI_DOUBLE,partner,bitValue,MPI_COMM_WORLD, &status);
    mydone += hisdone;
  }
  return mydone;
}

int main(int argc, char *argv[]) {
  double partial, sum, avg;
  double cpu1, cpu2;

  MPI_Init(&argc, &argv);
  MPI_Comm_rank(MPI_COMM_WORLD, &myid);
  MPI_Comm_size(MPI_COMM_WORLD, &nprocs);

  partial = (double) myid;
  printf("Node %d has partial value %le\n", myid, partial);

  cpu1 = MPI_Wtime();
  sum = global_sum(partial);
  cpu2 = MPI_Wtime();

  if (myid == 0) {
    avg = sum/nprocs;
```

```
    printf("Global average = %le\n", avg);
    printf("Execution time (s) = %le\n",cpu2-cpu1);
 }

  MPI_Finalize();
  return 0;
}
```

Figure: Results for running the global summation program on 4 and 8 processors