Name:

Suyanpeng Zhang (suyanpen@usc.edu)

1964926975

Source 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;

MPI\_Status status;

mydone = partial;

for (int bitvalue = 1 ; bitvalue<nprocs;bitvalue\*=2){

int partner = myid ^ bitvalue;

MPI\_Send(&mydone,1,MPI\_DOUBLE,partner,bitvalue,MPI\_COMM\_WORLD);

MPI\_Recv(&hisdone,1,MPI\_DOUBLE,partner,bitvalue,MPI\_COMM\_WORLD,&status);

mydone = 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("Rank %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;

}

Results:

A screenshot of a computer

Description automatically generated