CS5500 HW2

Chase Mortensen A01535275

August 2019

1 Description

Write an MPI program that performs an integer sort. Make your program in the "master/slave" style. Have the master process send data segments to the slave processes, where the lists are sorted individually. The lists should then be merged back together on the master process.

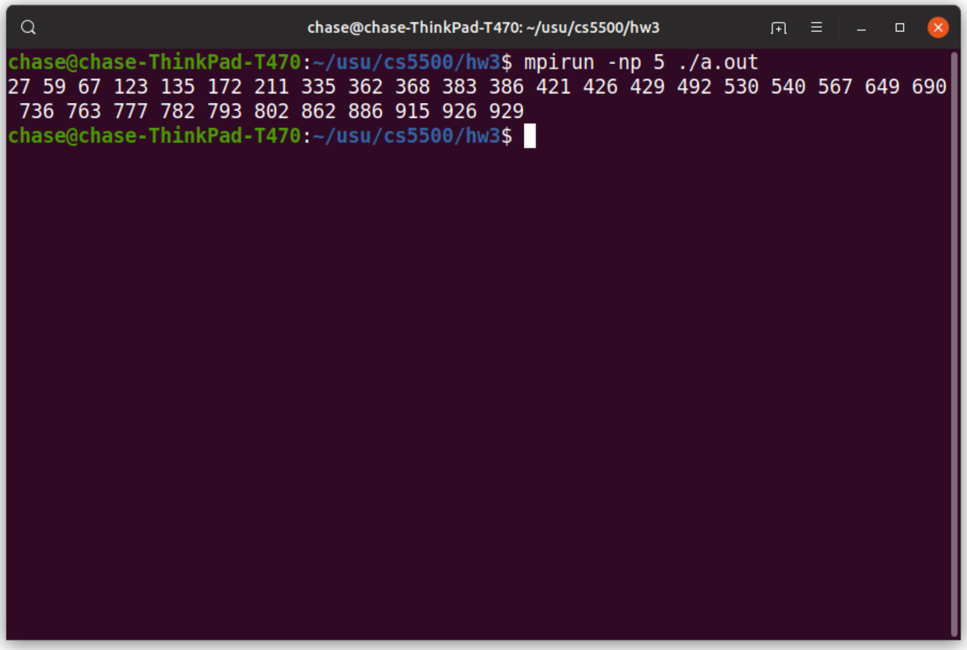
My implementation is based off of the animalfarm.cpp program written in class. It is simply a demonstration of the master/slave (or pig/horse) style. In this case, I implemented a single merger process which divides the vector to sort and sends the chunks to sorter processes. Running with 5 processes results in one merger process and four sorter processes. I am sorting 32 integers in this program, so each sorting process sorts a sub-list of 8 elements and returns it to the merger process, which finally prints out the sorted list.

2 Program

```
#include <iostream>
#include <mpi.h>
#include <unistd.h>
#include <stdlib.h>
#include <bits/stdc++.h>

#define MCW MPI_COMM_WORLD
using namespace std;

int main(int argc, char **argv){

    int rank, size;
    vector<int> vec(32);
    for (int i = 0; i < vec.size(); i++){
        vec[i] = rand()%1000;
    }
    MPI_Init(&argc, &argv);
    MPI_Comm_rank(MCW, &rank);
    MPI_Comm_size(MCW, &size);
    srand(rank);

    int numSorters = size - 1;
    int subVecLength = (int)vec.size() / numSorters;
    vector<int> subVec;
    vector<int> sortedVec;

    if(rank){   // I am a sorter
        subVec.resize(subVecLength);
        MPI_Recv(&subVec[0],subVecLength,MPI_INT,0,0,MCW,MPI_STATUS_IGNORE);

        sort(subVec.begin(), subVec.end());
        sleep(1);

        MPI_Send(&subVec[0],subVecLength,MPI_INT,0,0,MCW);
    }
    else{   // I am the merger
        for(int i = 0; i < numSorters; i++){
            subVec.clear();
            for (int j = 0; j < subVecLength; j++) {
                subVec.push_back(vec[j + (i * subVecLength)]);
            }
            MPI_Send(&subVec[0],subVecLength,MPI_INT,i+1,0,MCW);
        }
        while(numSorters > 0){
            MPI_Recv(&subVec[0],subVecLength,MPI_INT,MPI_ANY_SOURCE,0,MCW,
            MPI_STATUS_IGNORE);

            for(int i = 0; i < subVecLength; i++){
                sortedVec.push_back(subVec[i]);
            }
            sort(sortedVec.begin(), sortedVec.end());
            numSorters--;
        }

        for(int i = 0; i < sortedVec.size(); i++) {
            cout << sortedVec[i] << " ";
        }
        cout << endl;
    }
    MPI_Finalize();
    return 0;
}
```

3 Output

```
$ mpic++ integer_sort.cpp
$ mpirun -np 5 ./a.out

27 59 67 123 135 172 211 335 362 368 383 386 421 426 429
492 530 540 567 649 690 736 763 777 782 793 802 862 886
915 926 929
```



```
chase@chase-ThinkPad-T470: ~/usu/cs5500/hw3
chase@chase-ThinkPad-T470:~/usu/cs5500/hw3$ mpirun -np 5 ./a.out
27 59 67 123 135 172 211 335 362 368 383 386 421 426 429 492 530 540 567 649 690
 736 763 777 782 793 802 862 886 915 926 929
chase@chase-ThinkPad-T470:~/usu/cs5500/hw3$
```