F21_MPI_Proj01 Report

Jiale Shi

```
// processes with rank value from “1” to “total-1”
if (0 != my_rank)
{
        // step 1:  send toal_loc_value for processes with rank value
        // from “1” to “total-1”  to process “0”
        MPI_Send(&toal_loc_value,1, MPI_INT, 0, 12345,
MPI_COMM_WORLD);

        // step 2:  send sorted values in new_arr with size equal
        // toal_loc_value for processes with rank value from “1” to
        //“total-1”  to process “0”
        MPI_Send(new_arr, toal_loc_value, MPI_INT, 0, 123456,
MPI_COMM_WORLD);
}
```

```
// for process “0”
   if(0 == my_rank){

  // copy the new_arr of process0 to array temp
   for (j = 0; j<toal_loc_value; j++){
         temp[j] = new_arr[j];
   }

   for (j =1; j < total; j++)
   {

         //receive the toal_loc_value from process “j”
         MPI_Recv(&receive_toal_loc_value,1, MPI_INT, j, 12345,
MPI_COMM_WORLD,  &status);

         //received the sorted array in v_receive from process “j”
         MPI_Recv(v_receive, receive_toal_loc_value, MPI_INT, j,
123456, MPI_COMM_WORLD,  &status);

         // merge the temp with the v_receive
         merge(temp, toal_loc_value , v_receive,
receive_toal_loc_value, sorted);
```

```
        //update the length of the combined array
        toal_loc_value = toal_loc_value + receive_toal_loc_value;
        //update the combined array
        for (i = 0; i<toal_loc_value; i++){
        temp[i] = sorted[i];
        }

}
// save answer in file
ost <<"Whole sorted array: "<<endl;
for(i = 0; i < N; i++){
        ost <<sorted[i]<<endl;
}
}
```

For computing time complexity, the test results are

N=10, np =3

Proc 0 tooks 0.002530 seconds.

N=10, np =4

Proc 0 tooks 0.001563 seconds.

N=10, np =5

Proc 0 tooks 0.001870 seconds.

N=100, np =3

Proc 0 tooks 0.002183 seconds.

N=100, np =4

Proc 0 tooks 0.004041 seconds.

N=100, np =5

Proc 0 tooks 0.003024 seconds.

N=1000, np =3

Proc 0 tooks 0.014032 seconds.

N=1000, np =4

Proc 0 tooks 0.016774 seconds.

N=1000, np =5

Proc 0 tooks 0.017180 seconds.

N=10000, np =3

Proc 0 tooks 0.122993 seconds.

N=10000, np =4

Proc 0 tooks 0.169700 seconds.

N=10000, np =5

Proc 0 tooks 0.161537 seconds.