

CSCI 176 (Parallel Processing) Spring 2017 Prog. Assignment 5 (10 pts) Due: May 04 (Th)

Parallel merge sort with MPI.

Make a MPI version of the merge sort that you implemented in the previous assignment. A frame code is provided and you should complete the program, i.e., complete only 5 locations marked with “*****” – sending/receiving operation parts.

Logic used:

1. Process_0 accesses command line argument for n (total number of elements in the list), and broadcast it to all other processes;
2. Each process creates a dynamic array of integers with n/p elements, and fills it with random integer numbers; for the random numbers, please generate each number less than 100;
3. local lists (unsorted) are displayed in each process;
4. Each process performs qsort (GNU library quicksort) to sort the local list in the ascending order; please study any Internet sources for the usage of the qsort(..), if you haven't used it so far; Note: please do not use the ridiculous recursion based approach for sorting the local list;
5. Implement the tree-reduction for the merging operation, i.e., merging locally sorted lists into one, with the method of determining partner and sender/receiver; receiving processes should perform the merging operation and yield double-length merged list in each step;
6. Finally, process_0 produces the global sorted list and displays it.

Submission:

Submit the hard copies of the completed source code and output.

For the output to submit, please use n=400 and test with p=2, 4, 8.

Sample output is shown below.

Sample output (n=40, p=4 case):

```
process_0: 83 86 77 15 93 35 86 92 49 21
process_1: 90 19 88 75 61 98 64 77 45 27
process_2: 46 85 68 40 25 40 72 76 1 64
process_3: 1 83 74 26 63 37 25 63 28 85

sorted list:
1 1 15 19 21 25 25 26 27 28 35 37 40 40 45 46 49 61 63 63 64 64 68
72 74 75 76 77 77 83 83 85 85 86 86 88 90 92 93 98
```