Rohit Katariya

2020CSZ8844

Assignment 3

Parallel Sorting Using MPI

COL 730

2021

# Processor count vs Execution time graph

|  |  |  |
| --- | --- | --- |
| problem size | nproc | Time |
| 10000000 | 1 | 12.389 |
| 10000000 | 2 | 9.743 |
| 10000000 | 3 | 8.592 |
| 10000000 | 4 | 8.227 |
| 10000000 | 5 | 7.06 |
| 10000000 | 6 | 8.354 |

# Approach/ Algorithm used

* Designed parallelized version of quick sort
* **Why not merge sort**? Started with merge sort but thought it would require a lot of message passing and a lot of auxiliary memory creation required
* Parallelized quick sort:
  + Step 1: Message passing (all gather) between processors how many elements each has.
  + Step 2: Processor 1 sends pivot element to each processor(MPI\_Isend, MPI\_recv)
  + Step 3: Each processor divides itself into 2 parts left(containing items < pivot) and right(containing items >=pivot)
  + Step 4: Message passing to other processors that how many elements smaller than pivot are present with itself
  + Step 5: Processors compute whether they are left, right or pivot processors themselves; and start sending elements between each other swapping elements to get to a state where there are first all elements less than pivot then all with values more than pivot.
  + Step 6: pace pivot at the correct place by message passing to correct processor
  + Step 7: do recursive call on left and right arrays of pivot.