

# Assignment 5

ENGI 9869/7894 Adv. Concurrent Programming  
Electrical and Computer Engineering  
Memorial University

Note that the work that you turn in for this assignment must represent your individual effort. You are welcome to help your fellow students to understand the material of the course and the meaning of the assignment questions, however, the answer that you submit must be created by you alone. Assignments will be assessed on your efforts and approach you are using for particular solution.

## 1 Merge Sort

Consider a merge sort algorithm which breaks the given data set, sort and merge the broken sets to a sorted data set i.e. Integer Array. See the Figure 1 that with four array elements.

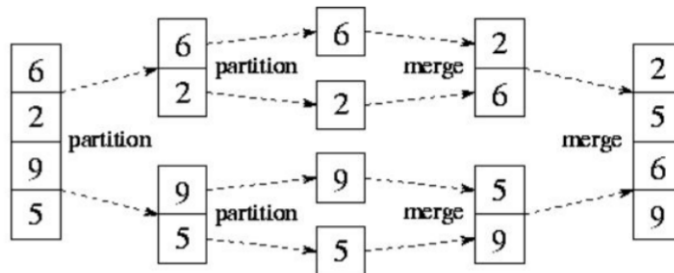
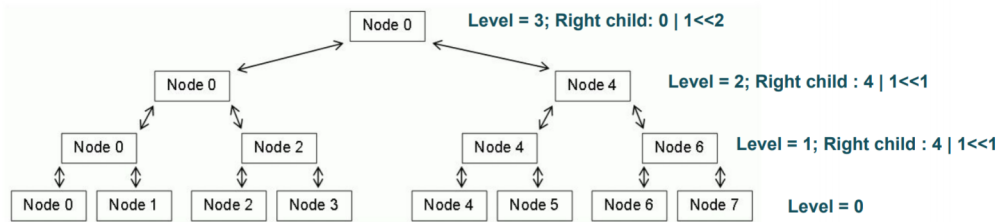


Figure 1: Deadlocked intersection with four vehicles

Suppose we have 8 communicating nodes to perform the parallel merge sort. Refer the Figure 2 for 8 nodes with Node 0 as host node. We have proposed a parallel algorithm in Section 2.

## 2 Proposed Parallel Algorithm:

1. Node with rank 0 is the host node. It computes the height of the node and get the entire dataset
2. Node 0 initiates the parallel merge operation



3. For internal nodes (height  $> 0$ ) including node 0,
  - (a) Divide the data in half and send the right half to the right child.
  - (b) Recursively call parallel merge operation for the left half on the same node
  - (c) Receive the sorted data from right child
  - (d) Merge the sorted left and right halves
4. If it is a leaf node, just do internal sorting
5. If parent's rank  $<$  node's rank, send the sorted data to parent node
6. Finally, node 0 will have the entire sorted result

For this assignment you have to use Message Passing (OpenMPI) and any programming language supporting MPI libraries. You have to test the algorithm with following data sets (Have the arrays with values of your choice).

1. An array of integers i.e. 4096 elements and 8 nodes to divide the work on.
2. An array of real numbers i.e 8192 elements and 8 nodes to divide the work on.
3. Record the time taken by the system to completely sort the both arrays, half the number of nodes for 1 and 2 and compare the time taken to sort the same arrays. Explain the impact of communication on by this change on the time.

### 3 Grading

Successful Implementation: 50

Results Comparisons: 20

Code Comments: 15

Readme file (Explains the process to build your solution): 15

*Note: A good source to start with MPI is <https://mpitutorial.com/tutorials/>*