The Metropolis Parallel Sort Algorithm

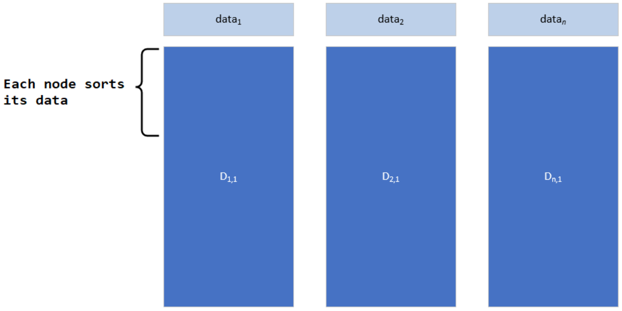
Jamshid Farzidayeri, Graham West, JJ Lay

Wednesday, January 16, 2019

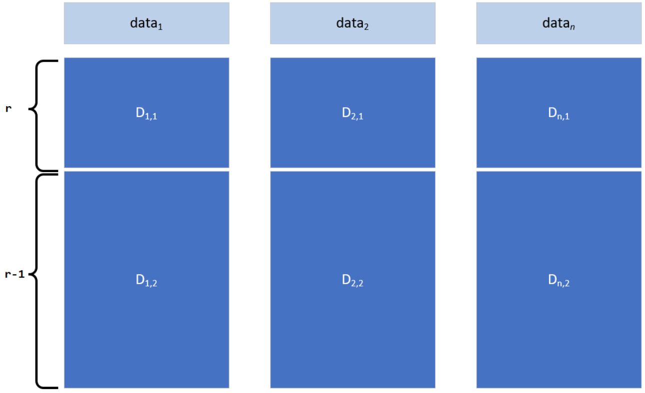
1 Algorithm

1. Distribute data to each node

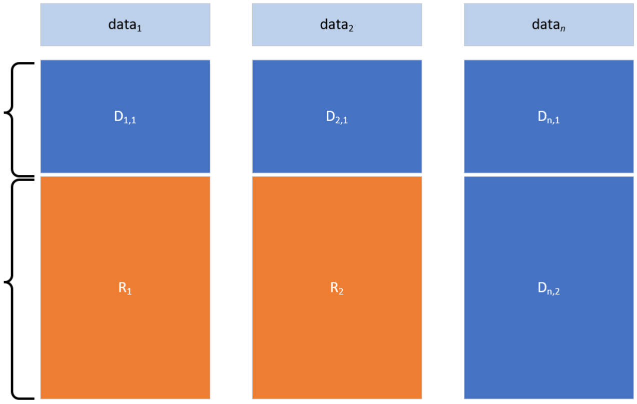
2. Each node sorts its data



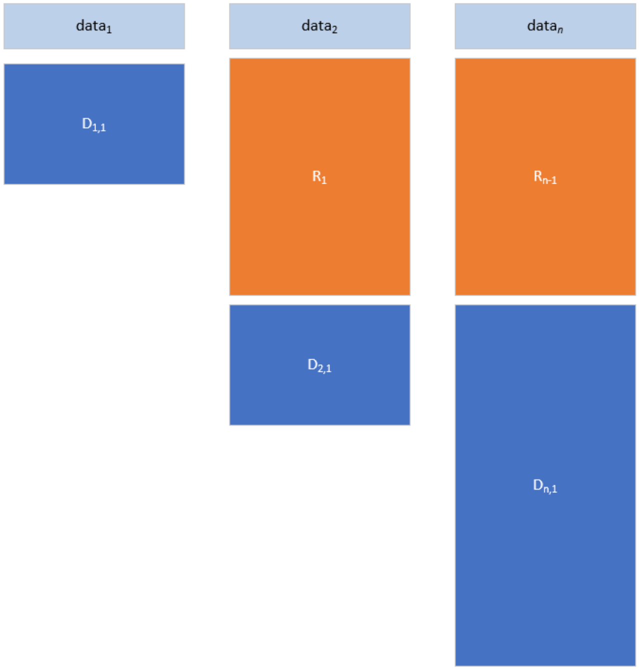
3. Data on each node is divided into two sections of sizes $N \times r$ and $N \times (1 - r)$ (where r is a ratio selected by the user and N is the number of values to be sorted)



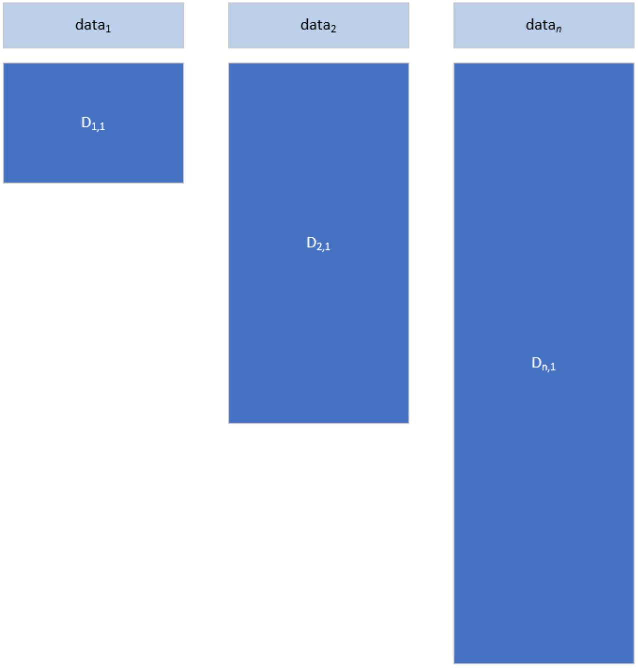
4. For the first $n-1$ nodes, the largest $(r-1)/N$ data is marked for transmission



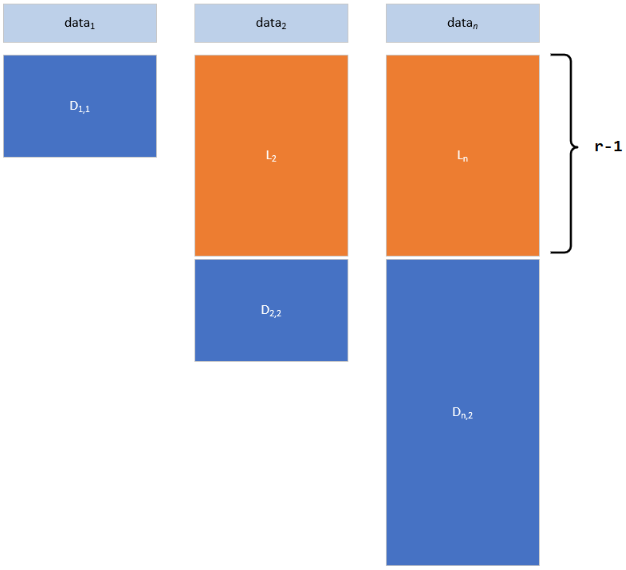
5. The largest values from node j is shifted to the front of the list on node $j+1$



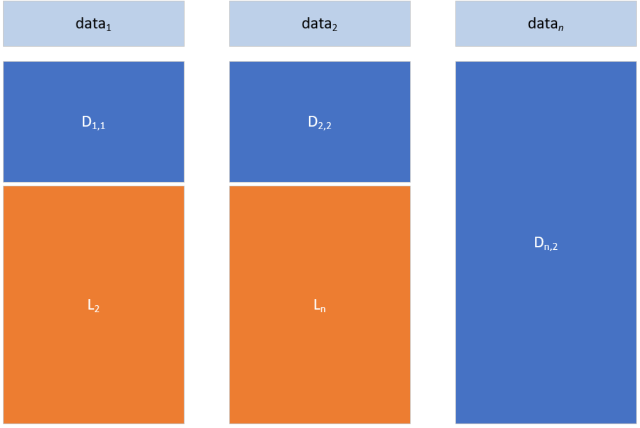
6. Each node sorts its new data set



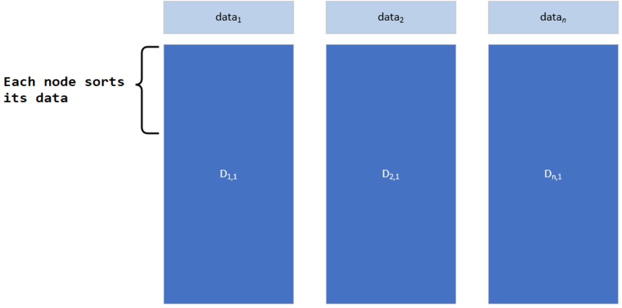
7. For the last $n-1$ nodes, the smallest $N \times (r-1)$ data is marked for transmission



8. The smallest values from node j is shifted to back of the list on node $j - 1$



9. Each node sorts its data



10. Nodes exchange max values

11. If any node j has a min value less than the max value of a node less than j, then loop to step 3

12. Nodes return results to node 0

13. Node 0 exports final results