CSE 512: DISTRIBUTED AND PARALLEL DATABASE SYSTEMS

GROUP 18 - PHASE 2 REPORT

MEMBERS

Vivek Singh(1209521349),Saurabh Singh(1209404713),Tarun Shimoga(1208478709)
Anand Ganesh(1209408847),Lakshmisagar Hiragappa Kusnoor(1211009498)

TASK B

1) For Task 2, difference between (a) and (b)

1000 iterations	Execution Time	Average Memory	CPU Utilization(%)
2(a)	100	5G	31.3
2(b)	200	5.1G	31

GANGLIA SCREENSHOTS:

Fig 1.2(a) Aggregated Cluster Report

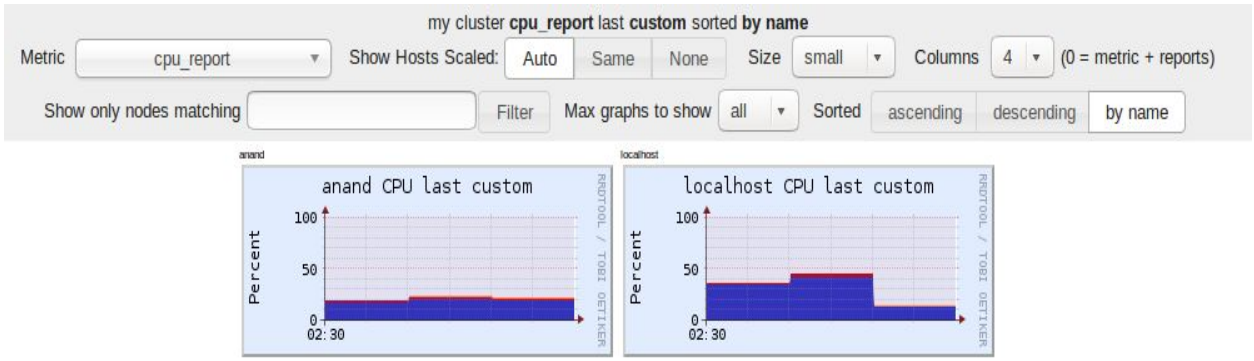


Fig 2.2(a) CPU Utilization

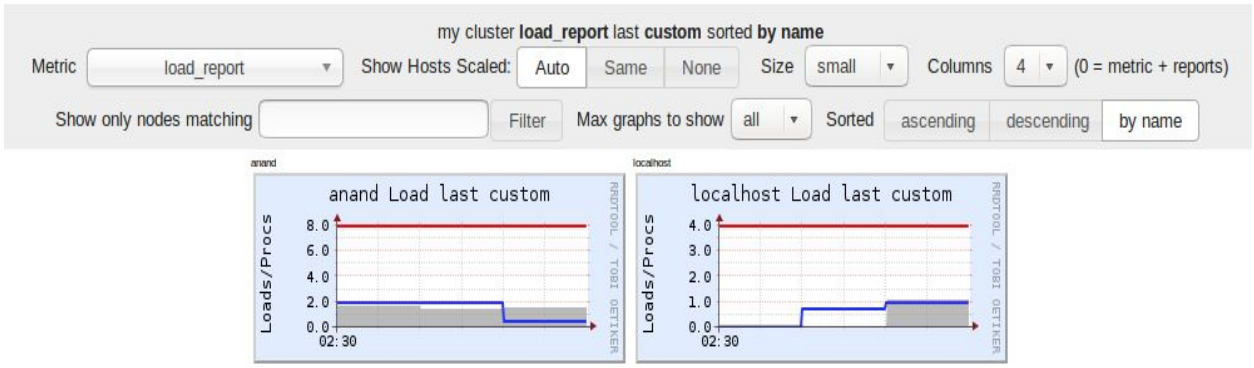


Fig 3.2(a)Cluster Load

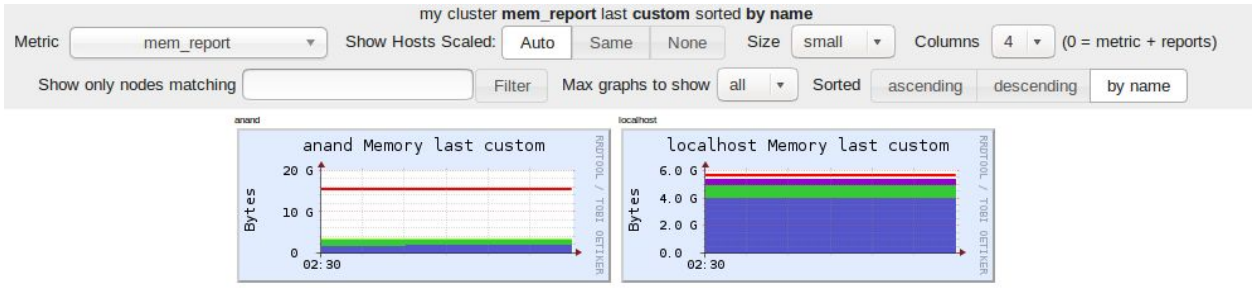


Fig 4.2(a) Memory Utilization

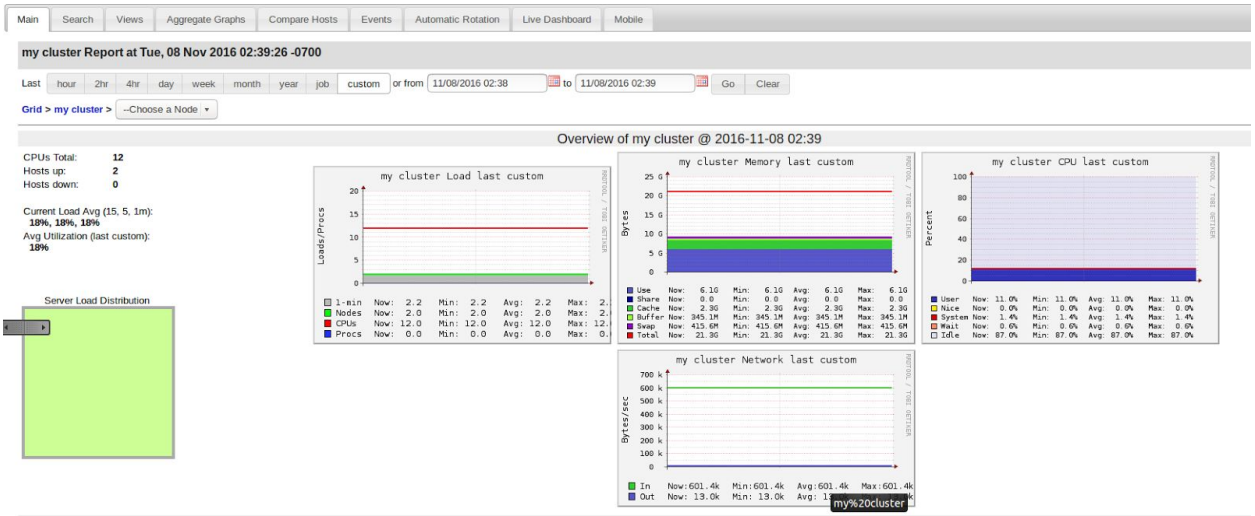


Fig 5.2(b) Aggregated Cluster Report

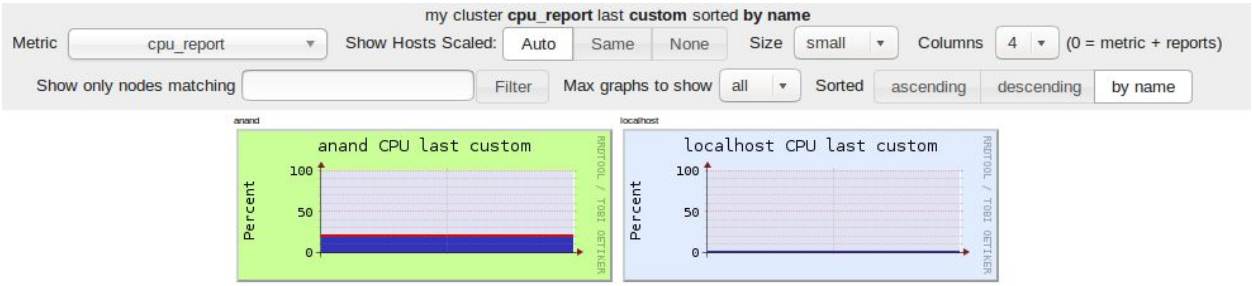


Fig 6.2(b) CPU Utilization

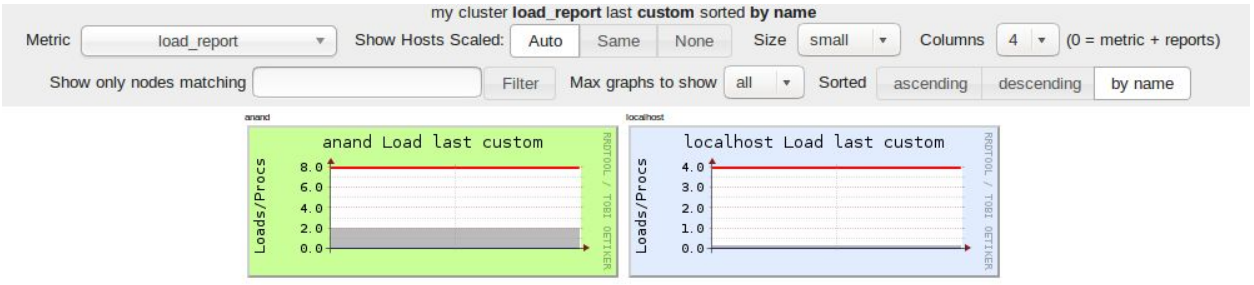


Fig 7.2(b)Cluster Load

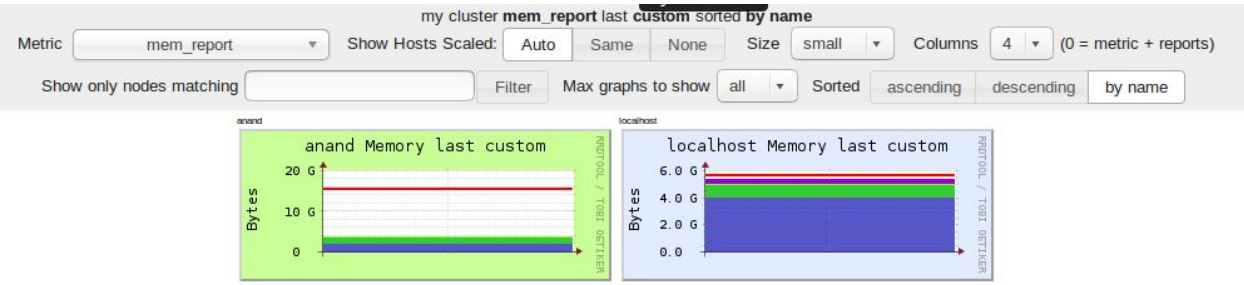


Fig 7.2(b)Memory Utilization

Conclusion:

The execution time, average memory and average CPU utilization is greater when Rtree indexing not used to build the PointRDD.

Explanation:

The Rtree indexing runs faster because the Point RDD's are stored in a Rtree data structure which is a tree data structure for spatial access methods i.e., for indexing multi-dimensional information.The Rtree data structure groups nearby objects and represents with their minimum bounding rectangle.So it becomes much faster to run Spatial range query on such grouped points rather than points distributed randomly.

2) For Task 3, difference between (a) and (b)

1000 Iterations	Execution Time(sec)	Average Memory	CPU Utilization(%)
3(a)	90	5.1G	31.9
3(b)	40	5.2G	31.3

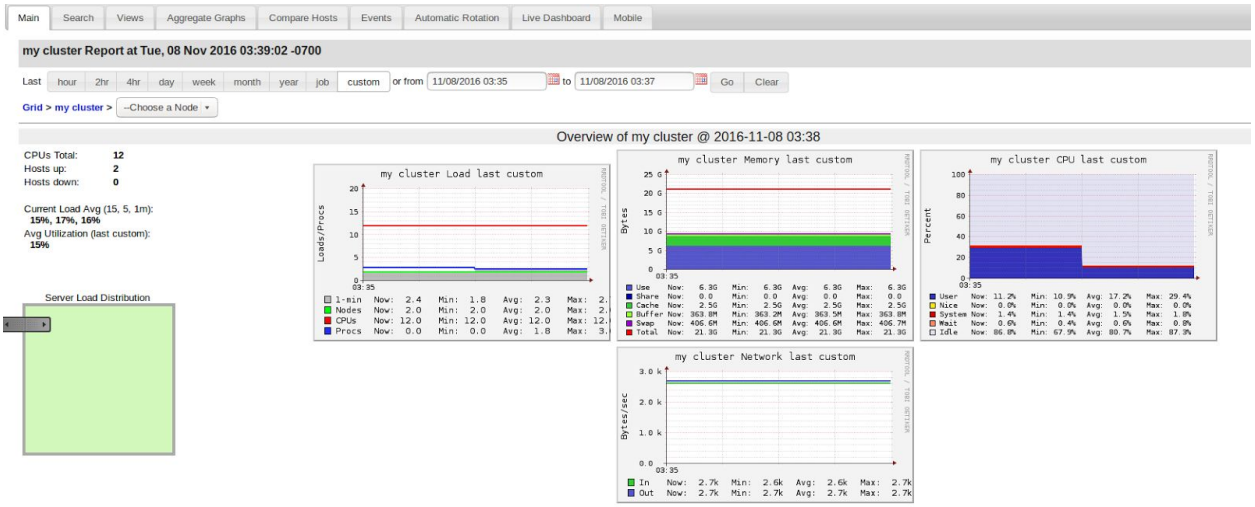


Fig 8.3(a) Aggregated Cluster Report

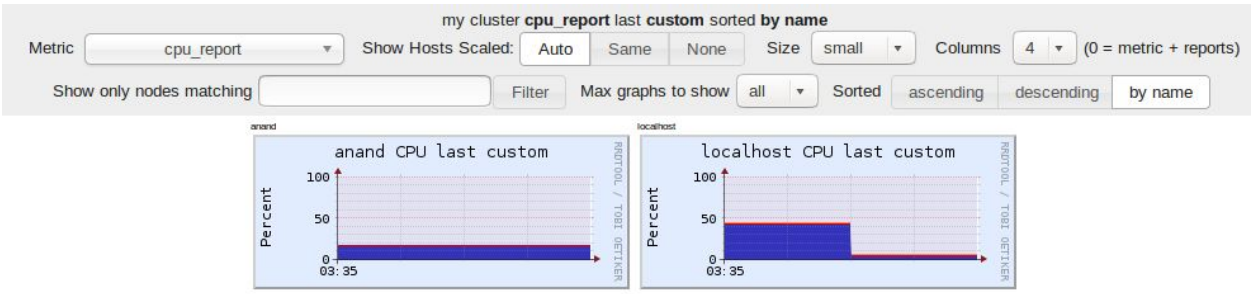


Fig 9.3(a) CPU Utilization

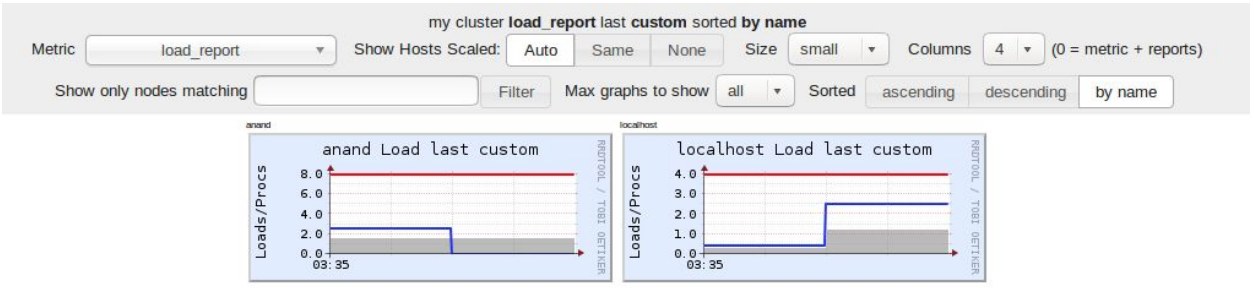


Fig 10.3(a)Cluster Load

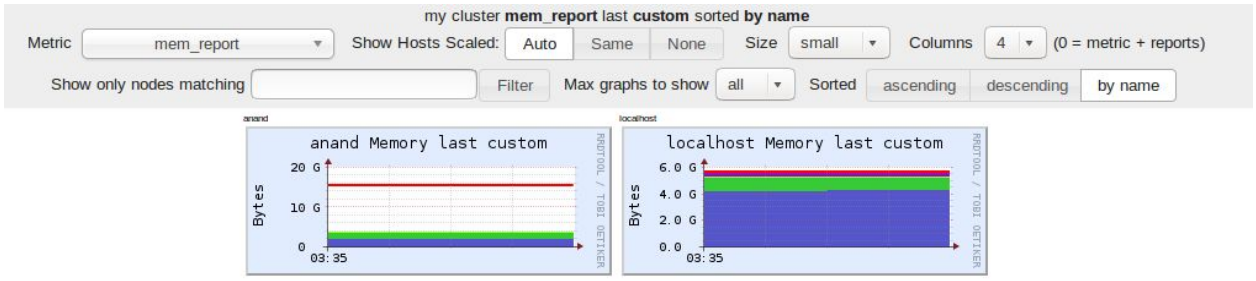


Fig 11.3(a)Memory Utilization

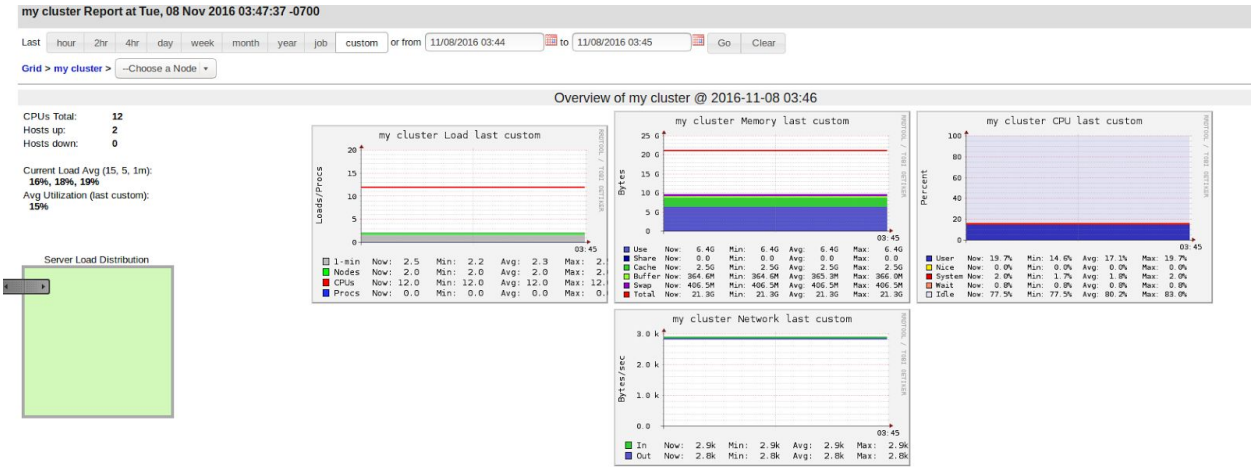


Fig 12.3(b) Aggregated Cluster Report

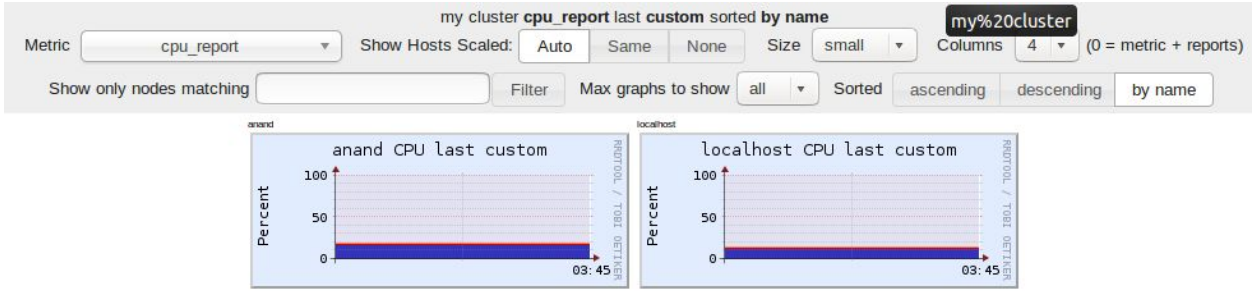


Fig 13.3(b) CPU Utilization

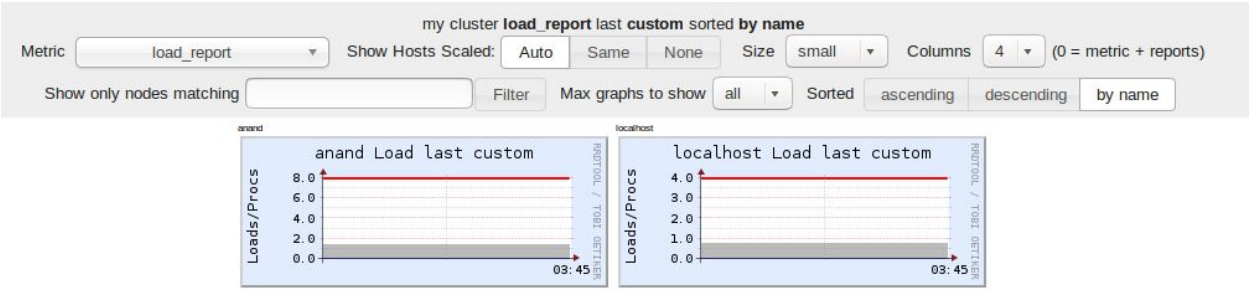


Fig 14.3(b)Cluster Load

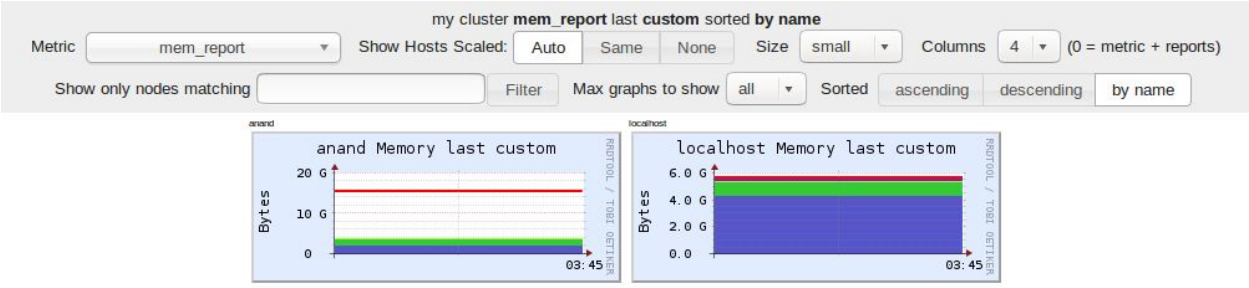


Fig 15.3(b)Memory Utilization

Conclusion:

The execution time, average memory and average CPU utilization is greater when Rtree indexing is not used to build PointRDD

Explanation:

The Rtree indexing runs faster because the Point RDD's are stored in a Rtree data structure which is a tree data structure for spatial access methods i.e., for indexing multi-dimensional information.The Rtree data structure groups nearby objects and represents with their minimum bounding rectangle.So it becomes much faster to run KNN query on nearby grouped points.

3) For Task 4, difference between (a) and (b); difference between (a) and (c).

10 iteration	Execution Time(MIN)	Memory	CPU Utilization(%)
4(a)	13	5G	34.2
4(b)	10	5.1G	37.3
4(c)	7	5.1G	37.3

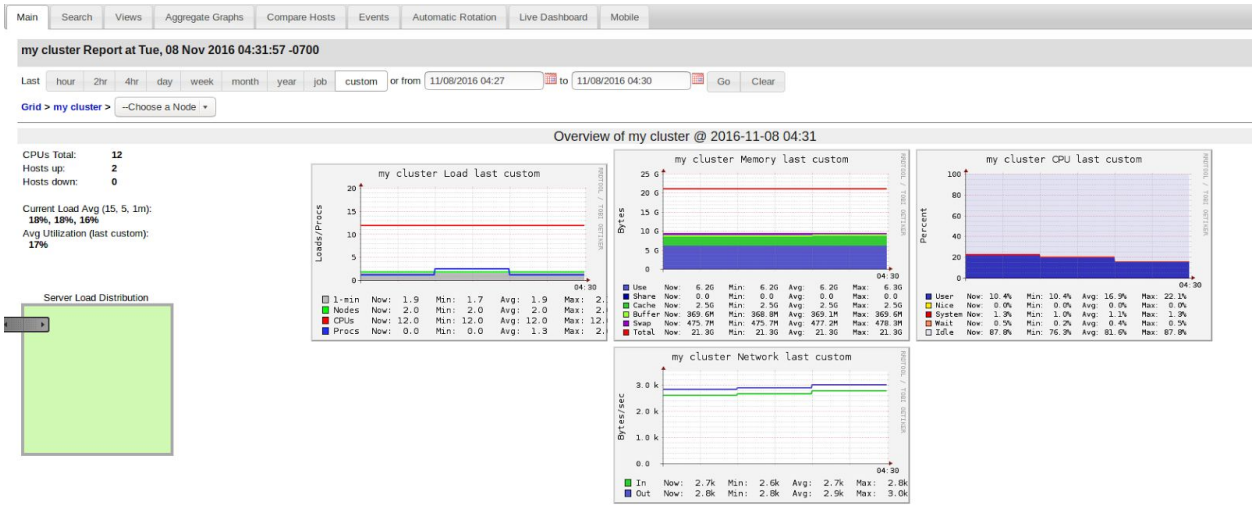


Fig 16.4(a) Aggregated Cluster Report

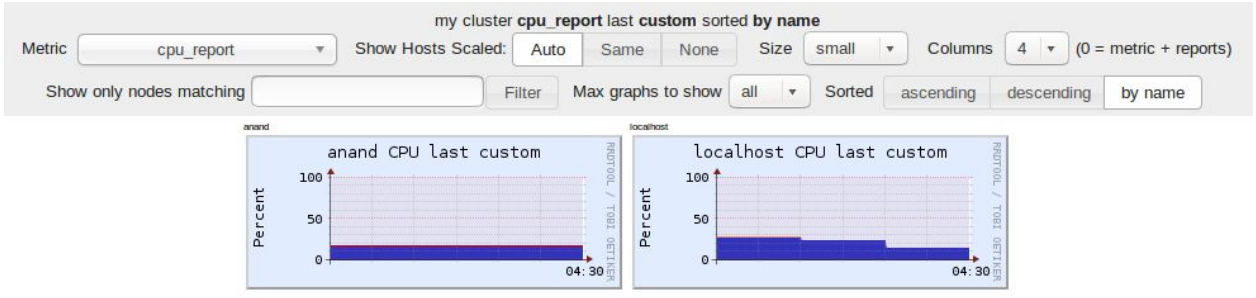


Fig 17.4(a) CPU Utilization

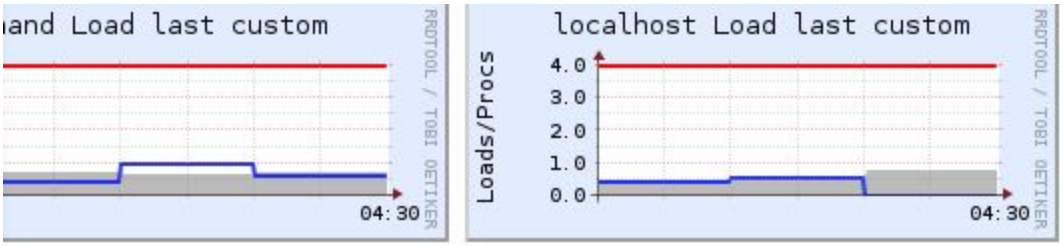


Fig 18.4(a)Cluster Load

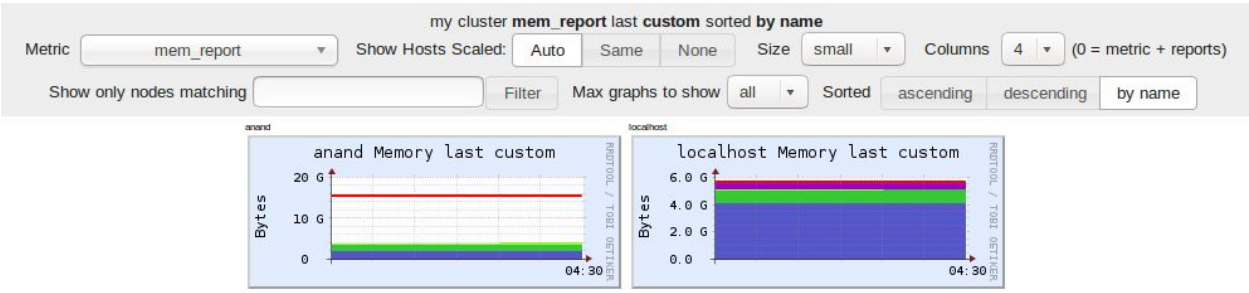


Fig 19.4(a)Memory Utilization

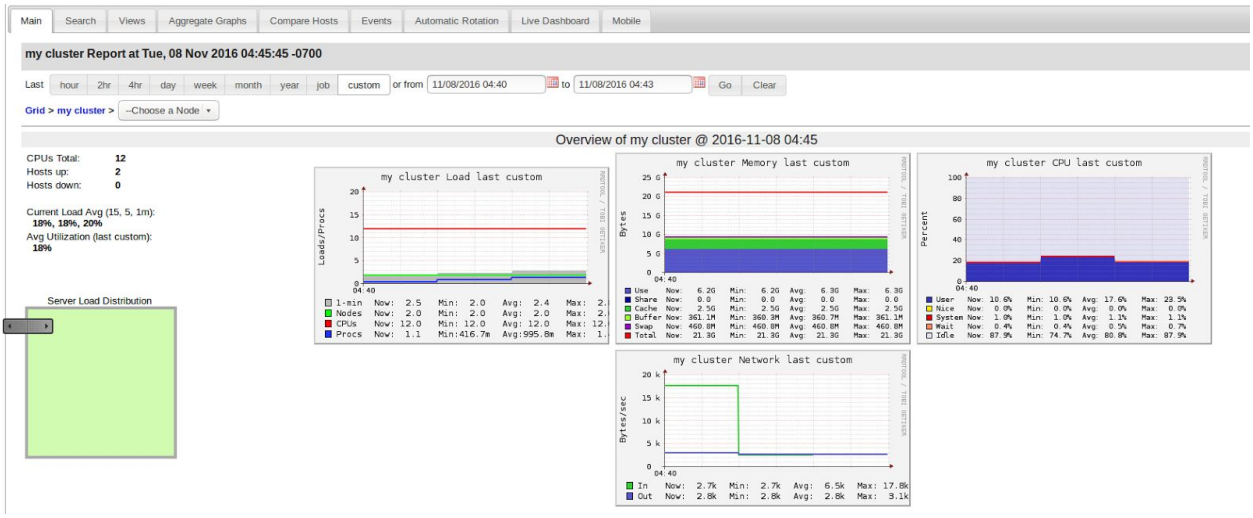


Fig 20..4(b) Aggregated Cluster Report

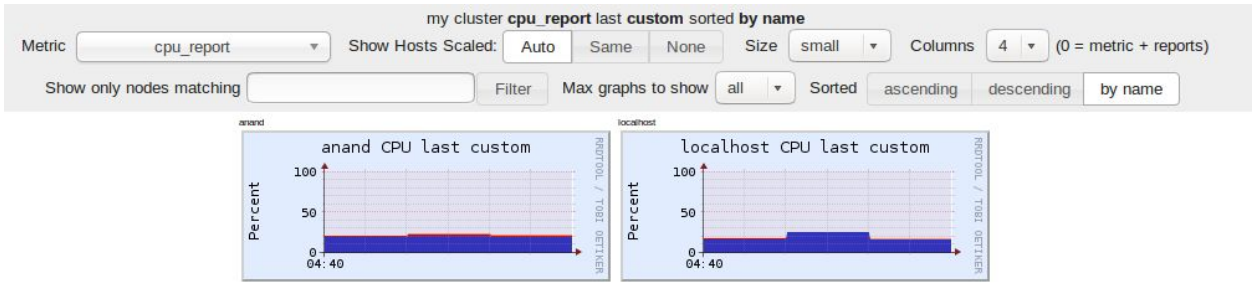


Fig 21.4(b) CPU Utilization

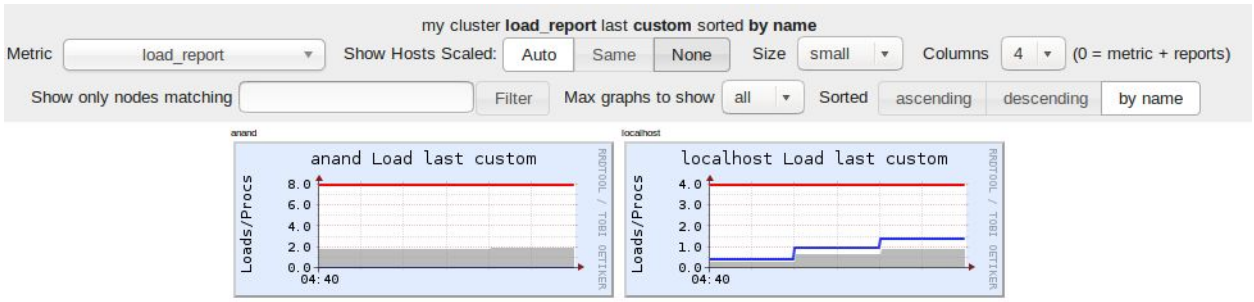


Fig 22.4(b)Cluster Load

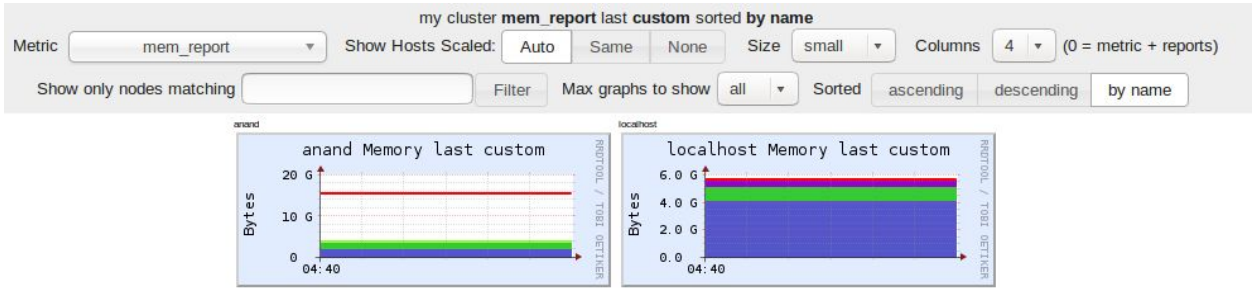


Fig 23.4(b)Memory Utilization

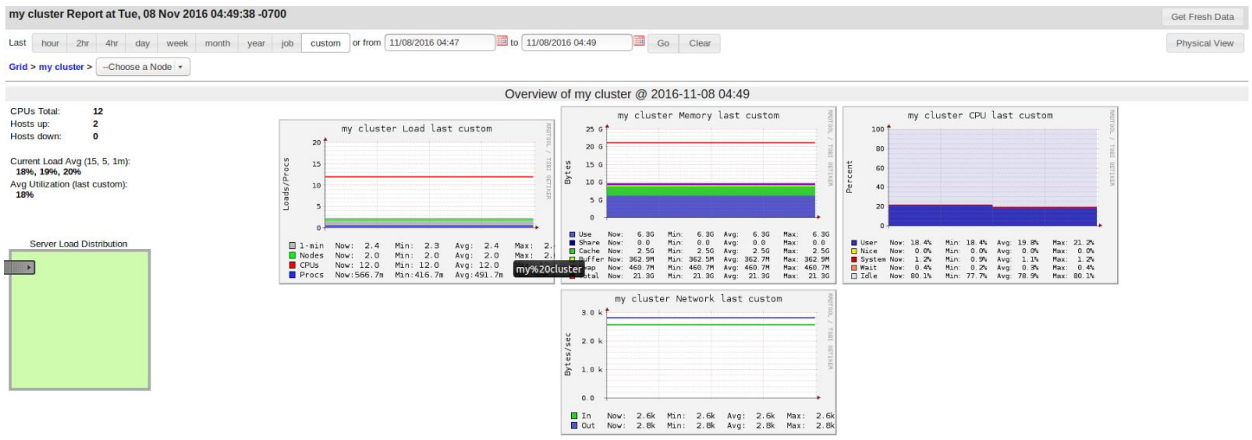


Fig 24..4(c) Aggregated Cluster Report

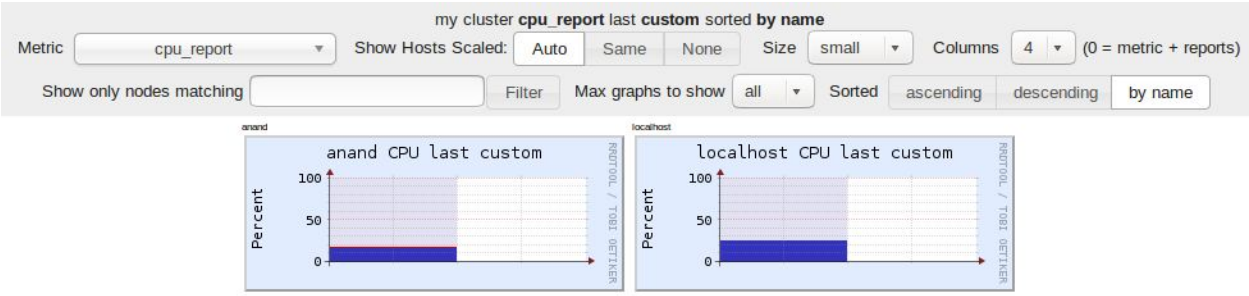


Fig 25.4(c) CPU Utilization

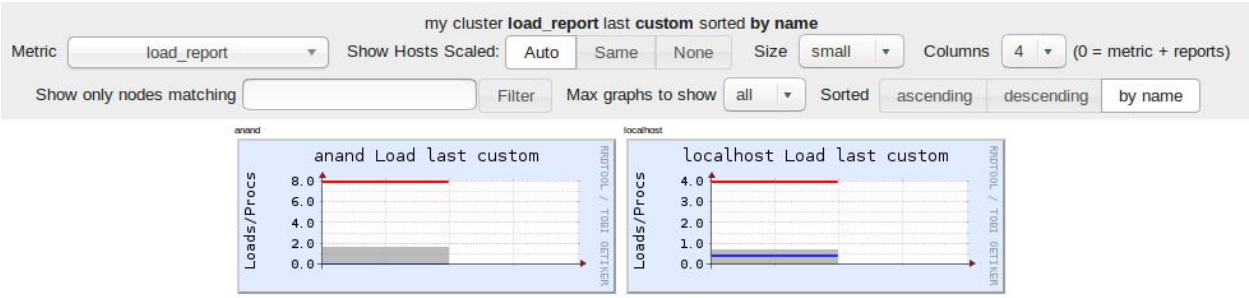


Fig 26.4(c)Cluster Load

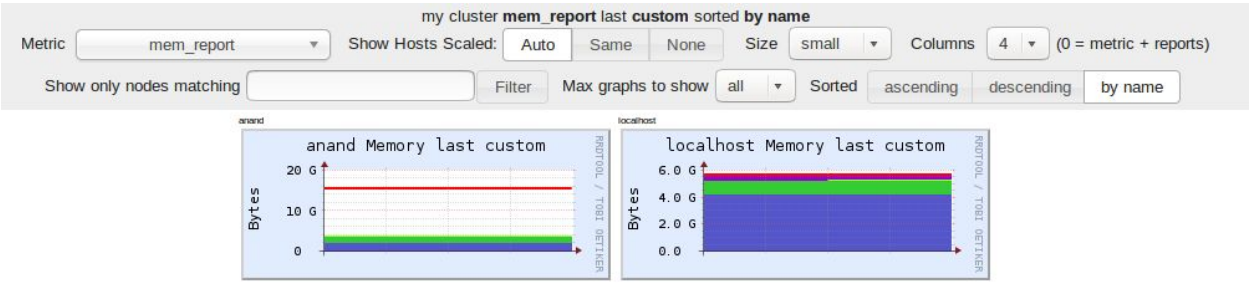


Fig 27.4(c)Memory Utilization

Conclusion:

The execution time, average memory and average CPU utilization is least for 4(c) i.e. Joining the PointRDD using R-Tree grid without R-Tree index and maximum for 4(a) i.e Joining the PointRDD using equal grid without R-Tree index

Explanation:

Comparing 4(a) and 4(b) we see that 4(b) takes lesser execution time, average memory and average CPU utilization.4(b) uses equal grid which effectively partitions the RDD

and then indexing is done using Rtree which groups nearby points together.Thus performing join on such RDD becomes much faster as compared to PoinRDD in 4(a).

Now comparing 4(a) and 4(c) we see that 4(c) takes lesser execution time, average memory and average CPU utilization.In case of 4(c) the spatial partitioning is done using Rtree which divides the RDD and gruops nearby points together while partitioning.Thus increase the performane of join perfromed on such RDDs.Also partitioning Rtree resuces the newtork utiliztaion to greater extent as it avoids the sort and shuffle phase.

2.Phase 1 Task 4 (a)(b)(c) and Phase 2 Task A
Phase 2 Task A Execution time(Full data)
2hrs 36 minutes

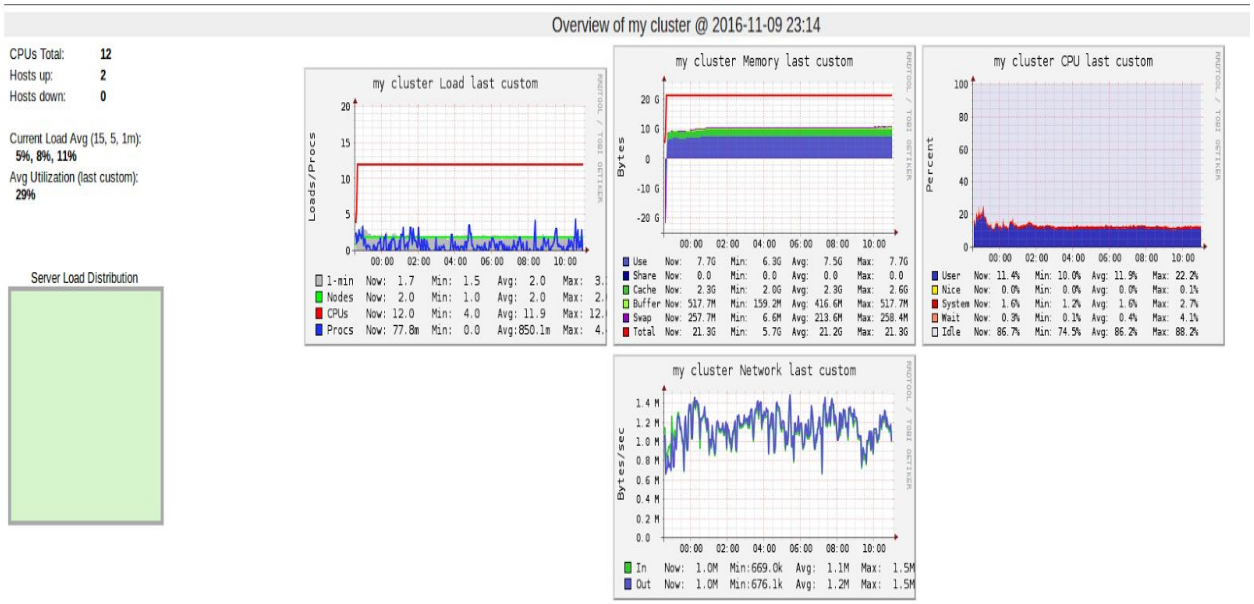


Fig 28..Task A Aggregated Cluster Report

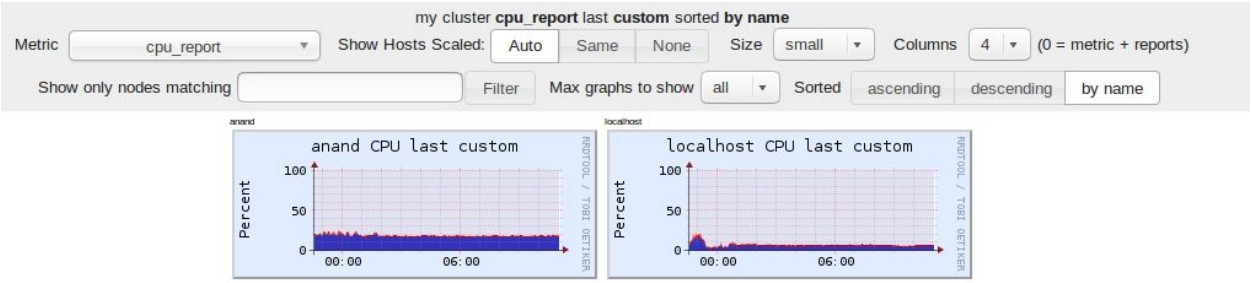


Fig 29. Task A CPU Utilization

Fig 30.. Task A Cluster Load

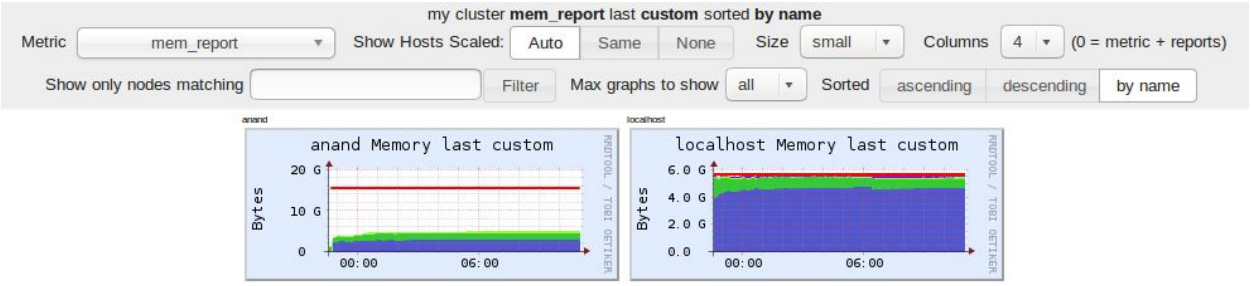
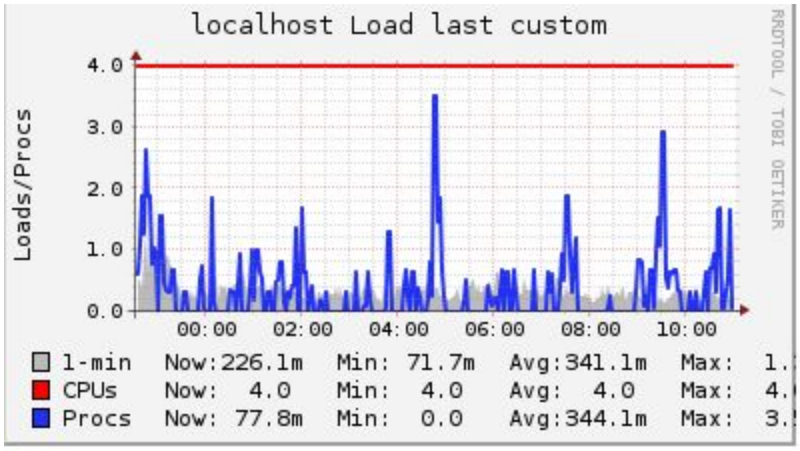
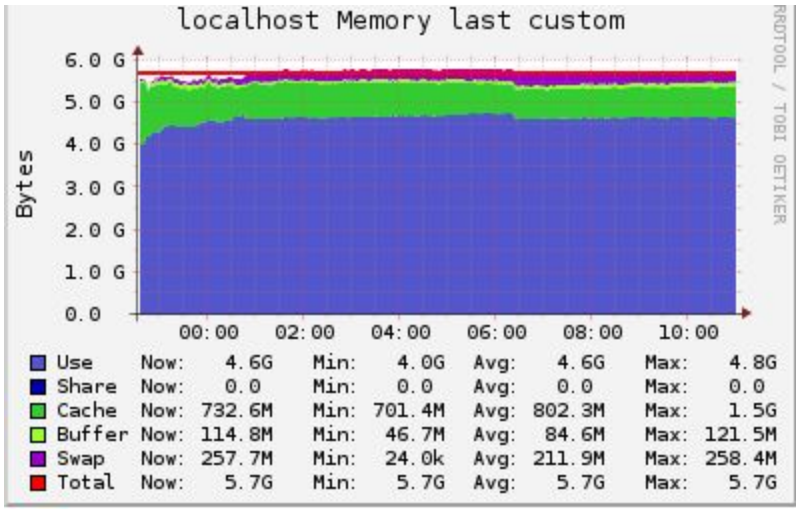
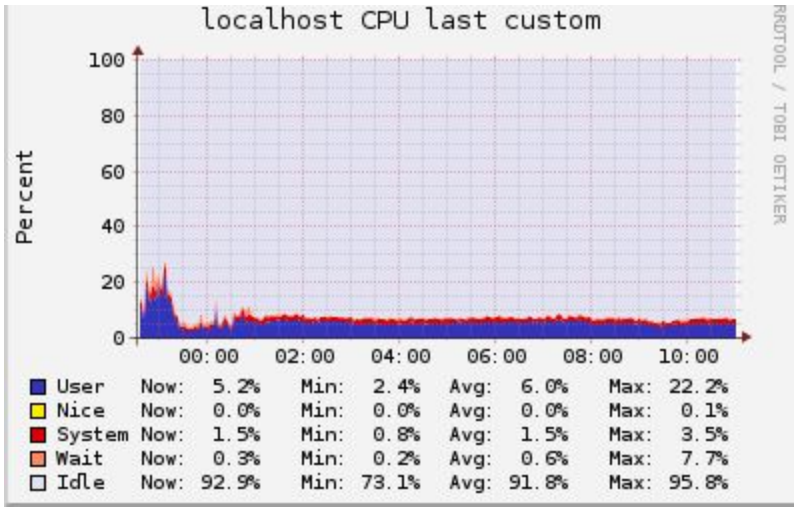


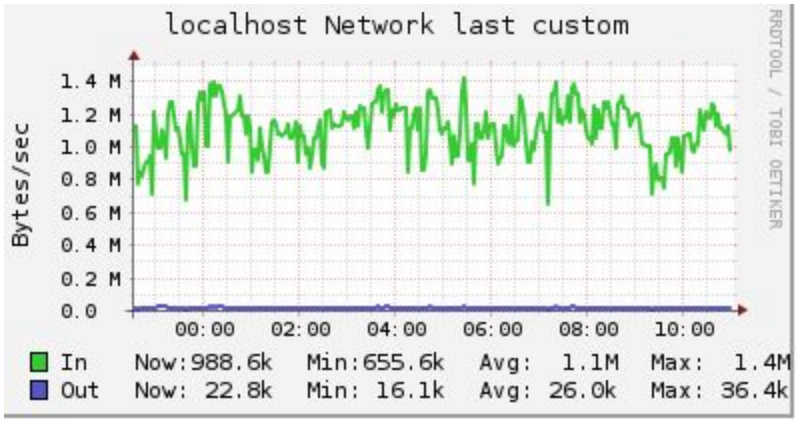
Fig 31. Task A Memory Utilization

Part A - Detailed Reports for Worker 1

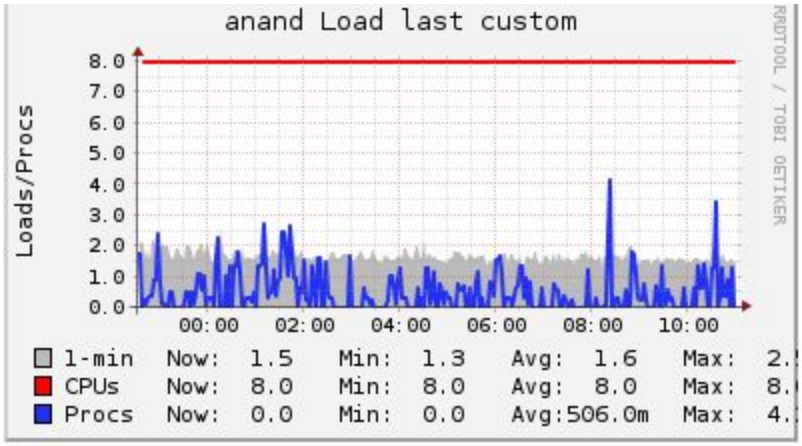


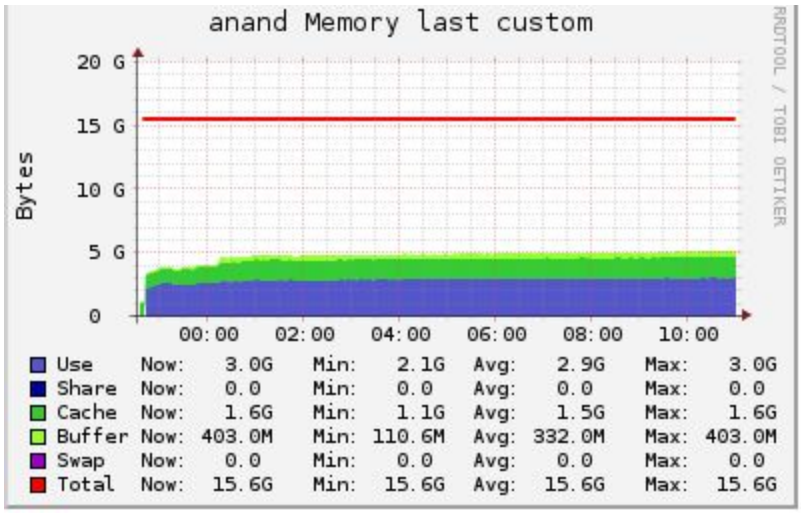


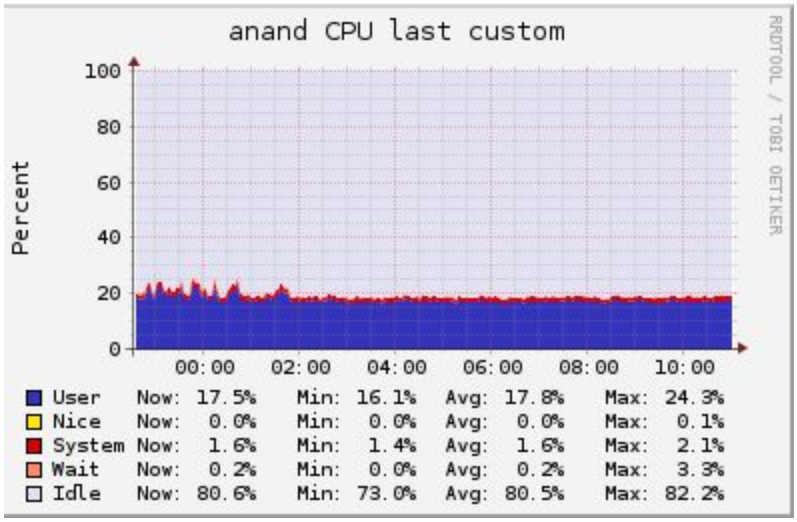


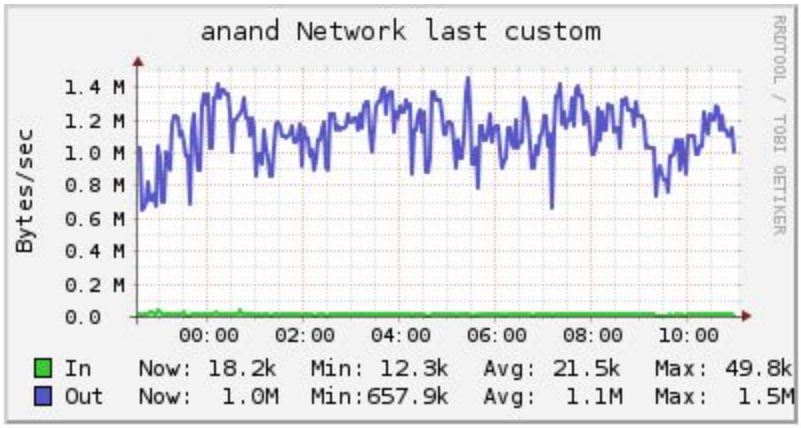


Part A -Detailed Reports for Worker 2









Conclusion: The spatial join using cartesian product takes greater amount of time compared to 4(a),4(b),4(c).

Explanation:
The spatial join in cartesian product does not use co-grouping of envelops before joining.The spatial join using cartesian product is naive algorithm which does not optimize or increase the performance.