Task A

Write a Java function to do a Spatial Join Query using the simple Cartesian Product algorithm. Specifically, for each rectangle from the query window dataset, check this rectangle against the point datasets using the regular GeoSpark Spatial Range Query: RangeQuery.SpatialRangeQuery(objectRDD, queryEnvelope, 0)

1. This function should be integrated into GeoSpark source code. You should fork or download GeoSpark Github master branch **LATEST** source code and practice.
2. The result format should be same as GeoSpark spatial join query: JavaPairRDD<Envelope, HashSet<Point>>.
3. A piece of pseudo code is available here: https://gist.github.com/jiayuasu/28cb8edb3287f705ed4d30c30de07639
4. Your Scala API usage should be like that in this link: https://gist.github.com/jiayuasu/84a0f4e5e9e246e50c32648bfd9de906

Task B

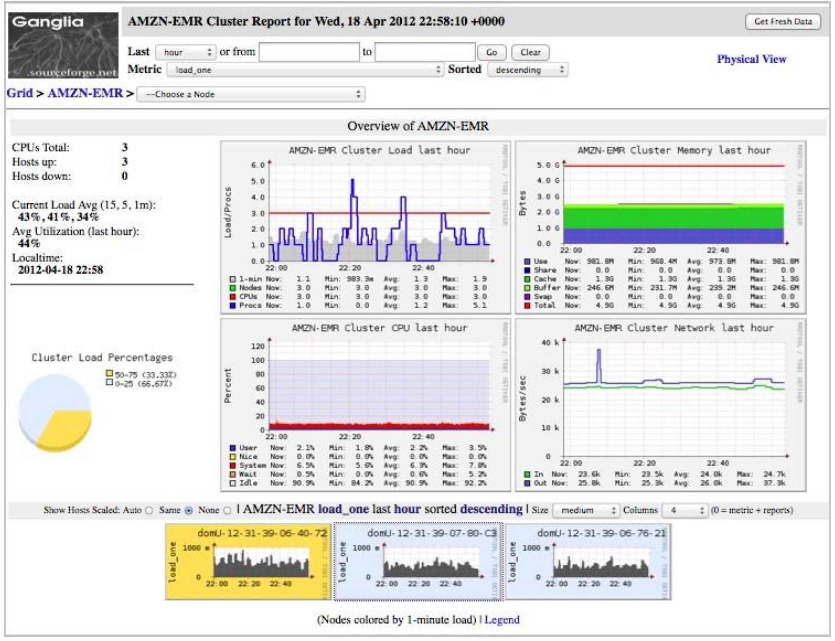
Write a report:

1. For the tasks in Phase 1, compare the execution time / average memory / average CPU utilization of the entire cluster and explain differences of the following operations if any:
    1) For Task 2, difference between (a) and (b)
    2) For Task 3, difference between (a) and (b)
    3) For Task 4, difference between (a) and (b); difference between (a) and (c).
2. For Phase 1 Task 4 (a)(b)(c) and Phase 2 Task A, compare the execution time / average memory / average CPU utilization of the entire cluster and explain the differences if any.
3. The comparison above should be done in Spark Scala shell. You might need to run each program many times to get an accurate result.
4. Clearly mention your group number, member name, 10 digits Student ID and ASU Email in the report.

Notes:

1. You have to run Apache Spark on a cluster. This means you should have at least three machines (One master and two workers). The master should be able to communicate with workers using bi-directional Password-less SSH.
2. You have to pre-load data to HDFS and use Spark to read data from HDFS.
3. You should use zcta510 as the query window datasets (use GeoSpark RectangleRDD) and use arealm as the point datasets (use GeoSpark PointRDD). The datasets are put into Blackboard. This time the PointRDD in all Tasks should have 10 RDD partitions.
4. You have to use Apache Spark 2.0.1.
5. You'd better use some cluster monitoring software such as Ganglia to monitor the resources of your cluster. Single machine monitoring software like dstat is OK but it may take you lots of time to collect and process data from the machines in your cluster. You

need to explain what software you are using and how you collect and process statistic data. You also need to provide screenshots of the software which is running on your cluster such as Figure 1.



AMZN-EMR Cluster Report for Wed, 18 Apr 2012 22:58:10 +0000

Last: hour, or from ___ to ___ Go Clear

Metric: load_one Sorted: descending

Grid > AMZN-EMR > --Choose a Node

Overview of AMZN-EMR

CPUs Total: 3
Hosts up: 3
Hosts down: 0

Current Load Avg (15, 5, 1m):
43%, 41%, 34%
Avg Utilization (last hour):
44%
Localtime:
2012-04-18 22:58

Show Hosts Scaled: Auto ○ Same ● None ○ | AMZN-EMR **load_one** last **hour** sorted **descending** | Size: medium Columns: 4 (0 = metric + reports)

(Nodes colored by 1-minute load) | Legend

Ganglia
.sourceforge.net
Get Fresh Data
Physical View
AMZN-EMR Cluster Load last hour
Load/Procs
6.0
5.0
4.0
3.0
2.0
1.0
0.0
22:00
22:20
22:40
1-min Now: 1.1 Min: 983.3m Avg: 1.3 Max: 1.9
Nodes Now: 3.0 Min: 3.0 Avg: 3.0 Max: 3.0
CPUs Now: 3.0 Min: 3.0 Avg: 3.0 Max: 3.0
Procs Now: 1.0 Min: 0.0 Avg: 1.2 Max: 5.1
AMZN-EMR Cluster Memory last hour
Bytes
5.0 G
4.0 G
3.0 G
2.0 G
1.0 G
0.0
Use Now: 981.8M Min: 968.4M Avg: 973.8M Max: 981.8M
Share Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Cache Now: 1.3G Min: 1.3G Avg: 1.3G Max: 1.3G
Buffer Now: 246.6M Min: 231.7M Avg: 239.2M Max: 246.6M
Swap Now: 0.0 Min: 0.0 Avg: 0.0 Max: 0.0
Total Now: 4.9G Min: 4.9G Avg: 4.9G Max: 4.9G
Cluster Load Percentages
50-75 (33.33%)
0-25 (66.67%)
AMZN-EMR Cluster CPU last hour
Percent
120
100
80
60
40
20
0
User Now: 2.1% Min: 1.8% Avg: 2.2% Max: 3.5%
Nice Now: 0.0% Min: 0.0% Avg: 0.0% Max: 0.0%
System Now: 6.5% Min: 5.6% Avg: 6.3% Max: 7.8%
Wait Now: 0.5% Min: 0.0% Avg: 0.6% Max: 5.2%
Idle Now: 90.9% Min: 84.2% Avg: 90.9% Max: 92.2%
AMZN-EMR Cluster Network last hour
Bytes/sec
40 k
30 k
20 k
10 k
0
In Now: 23.6k Min: 23.5k Avg: 24.0k Max: 24.7k
Out Now: 25.8k Min: 25.3k Avg: 26.0k Max: 37.3k
domU-12-31-39-06-40-72
domU-12-31-39-07-80-C3
domU-12-31-39-06-76-21
load_one
1000 m
0

Figure 1 Ganglia Web Interface