----------Dataset 1: NREL Solar Power Data----------

Dataset 1 was obtained from NREL Website which provided us with the Solar Power data for various locations. The link to download this dataset is: http://www.nrel.gov/grid/solar-power-data.html. This data is hosted and processed on Dumbo.

Code:

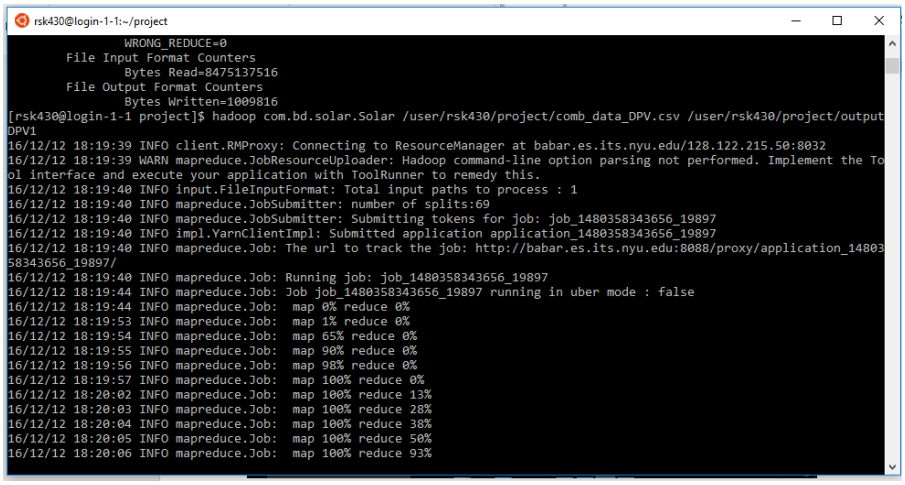
Combiner.py: Code to combine all the files into a single file

Combiner.py-UPV_DPV.py: Code to combine UPV and DPV data seperately.

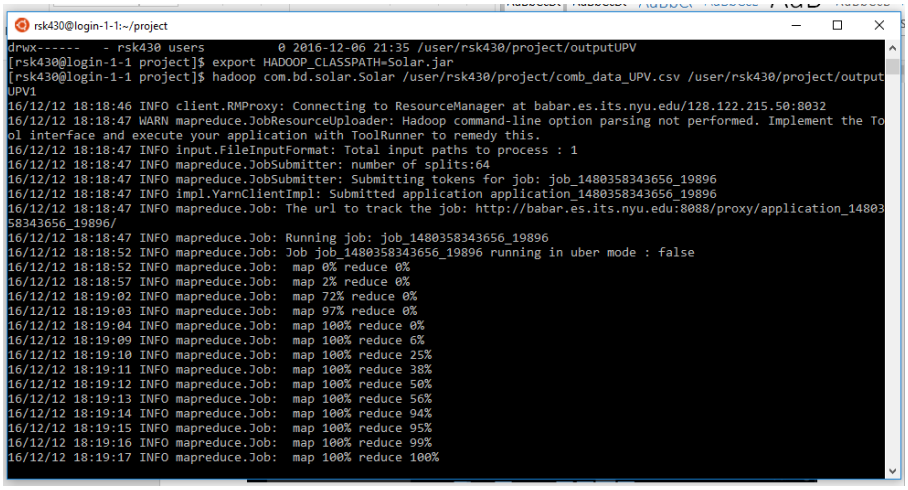
Solar.zip: MapReduce code to profile data and perform analytic on data

SolarCodeUPV_DPV.zip: MapReduce code to extract monthly power generated by UPV and DPV units.

Snapshot for MapReduce on UPV:



Snapshot for MapReduce on DPV:



----------Dataset 2: NLCD Data----------

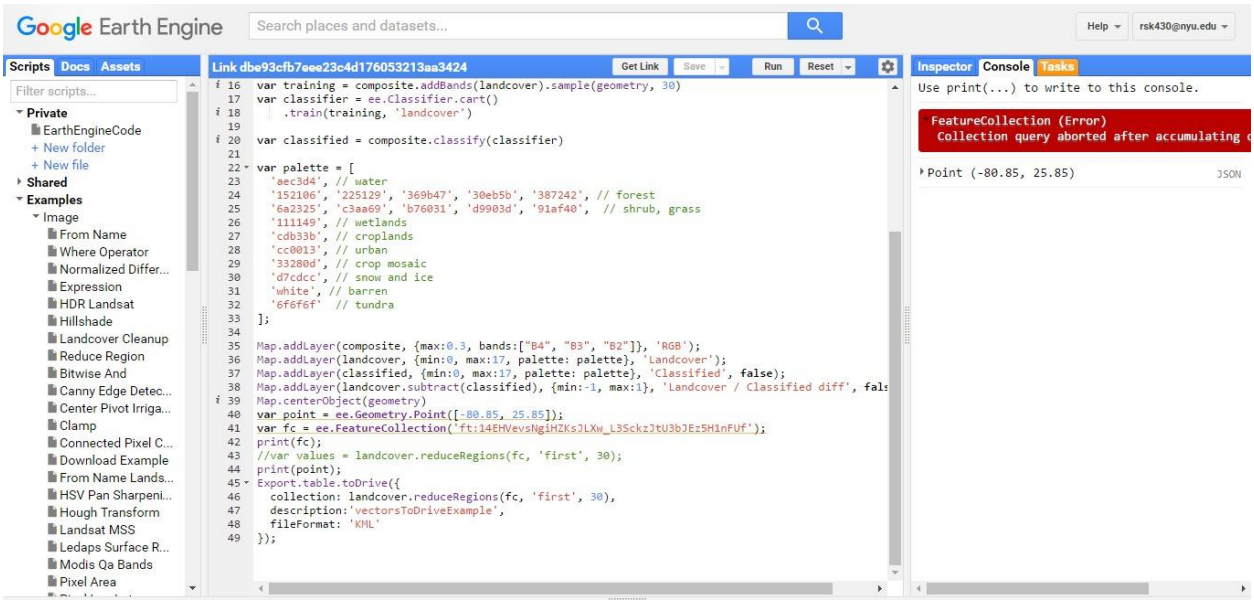
Dataset 2 was obtained from Multi-Resolution Land Characteristics (MRLC) Consortium website. As we were unable to process this data. We made use of Google earth Engine to extract the information from this data (Dataset was already hosted on Google Earth Engine). The link to download this dataset is: http://www.mrlc.gov/nlcd2011.php. But It's better to extract data directly from Google Earth Engine as special software (ArcGIS, ArcMap, FME workbench) are required to work with the data. This data was hosted and processed on Quickstart VM)

Google Earth Engine.txt: Code used to extract Landcover from NLCD data.

kmltogeoJSON.py: Code to convert LandCover data from KML format to GeoJSON.

LandCodeforJSON.zip: MapReduce code to convert the GeoJSON data to CSV and isolate it in terms of Landcover so that it can be used in Impala.

Snapshot for Google Earth Engine:



----------**ANALYTICS**----------

Impala Queries:

Queries to create tables in Impala for the Solar(UPV and DPV) and Landcover_solar dataset

1. create table upv(latitude string, longitude string, month int, sum double) row format delimited fields terminated by ',' location '/user/cloudera/project/upv/';

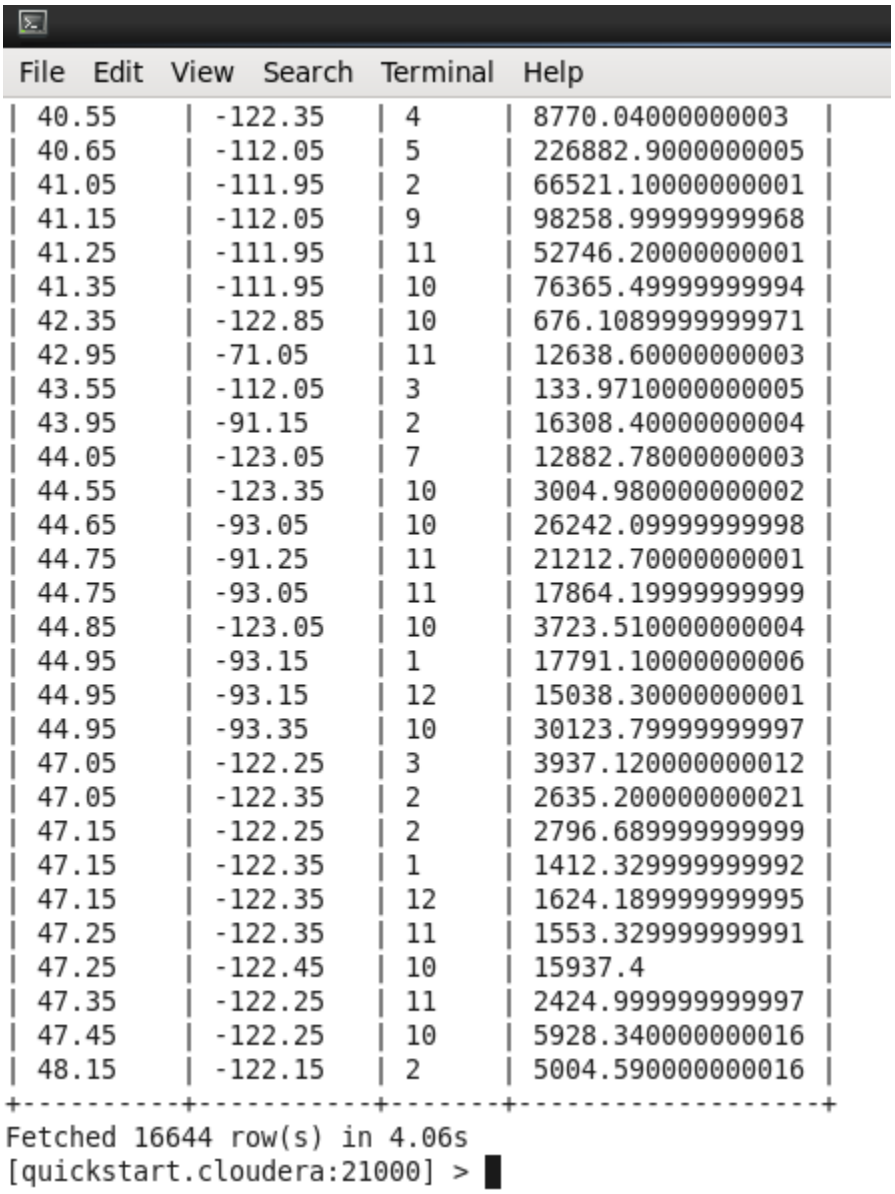
2. create table dpv(latitude string, longitude string, month int, sum double) row format delimited fields terminated by ',' location '/user/cloudera/project/dpv/';

3. create table landsolar(longitude string, latitude string, landcover int) row format delimited fields terminated by ',' location '/user/cloudera/project/landcover_solar/';

Query for Distributed PV (DPV) that can be used for solar power generation in houses and buildings and hence will include landcover like Urban, CropLands and Crop, Natural Veg. Mosaic.

4. select dpv.latitude, dpv.longitude, dpv.month, dpv.sum from dpv, landsolar where CAST(dpv.latitude as double) = CAST(landsolar.latitude as double) and CAST(dpv.longitude as double) = CAST(landsolar.longitude as double) and landsolar.landcover in (12,13,14);

Result of Analytic:



File Edit View Search Terminal Help

40.55	-122.35	4	8770.04000000003
40.65	-112.05	5	226882.9000000005
41.05	-111.95	2	66521.10000000001
41.15	-112.05	9	98258.99999999968
41.25	-111.95	11	52746.20000000001
41.35	-111.95	10	76365.49999999994
42.35	-122.85	10	676.1089999999971
42.95	-71.05	11	12638.60000000003
43.55	-112.05	3	133.9710000000005
43.95	-91.15	2	16308.40000000004
44.05	-123.05	7	12882.78000000003
44.55	-123.35	10	3004.980000000002
44.65	-93.05	10	26242.09999999998
44.75	-91.25	11	21212.70000000001
44.75	-93.05	11	17864.19999999999
44.85	-123.05	10	3723.510000000004
44.95	-93.15	1	17791.10000000006
44.95	-93.15	12	15038.30000000001
44.95	-93.35	10	30123.79999999997
47.05	-122.25	3	3937.120000000012
47.05	-122.35	2	2635.200000000021
47.15	-122.25	2	2796.689999999999
47.15	-122.35	1	1412.329999999992
47.15	-122.35	12	1624.189999999995
47.25	-122.35	11	1553.329999999991
47.25	-122.45	10	15937.4
47.35	-122.25	11	2424.999999999997
47.45	-122.25	10	5928.340000000016
48.15	-122.15	2	5004.590000000016

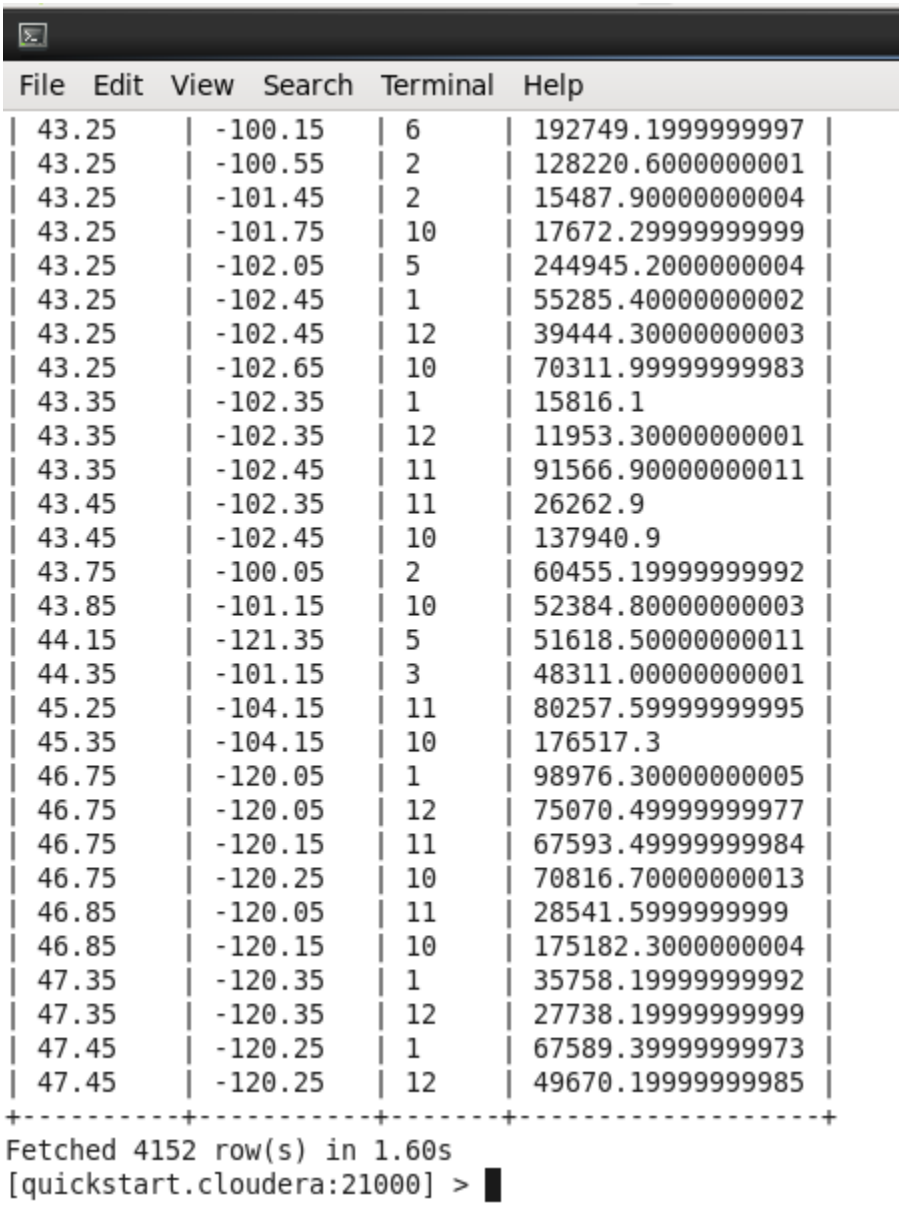
Fetched 16644 row(s) in 4.06s

[quickstart.cloudera:21000] >

Query for Utility-scale PV (UPV) that can be used for Solar power farms, large space is required and hence will iinclude Closed Shrubland, Savannas, Desert, Barren and Grasslands. The query for this is given below:

5. select upv.latitude, upv.longitude, upv.month, upv.sum from upv, landsolar where CAST(upv.latitude as double) = CAST(landsolar.latitude as double) and CAST(upv.longitude as double) = CAST(landsolar.longitude as double) and landsolar.landcover in (6,9,10,16);

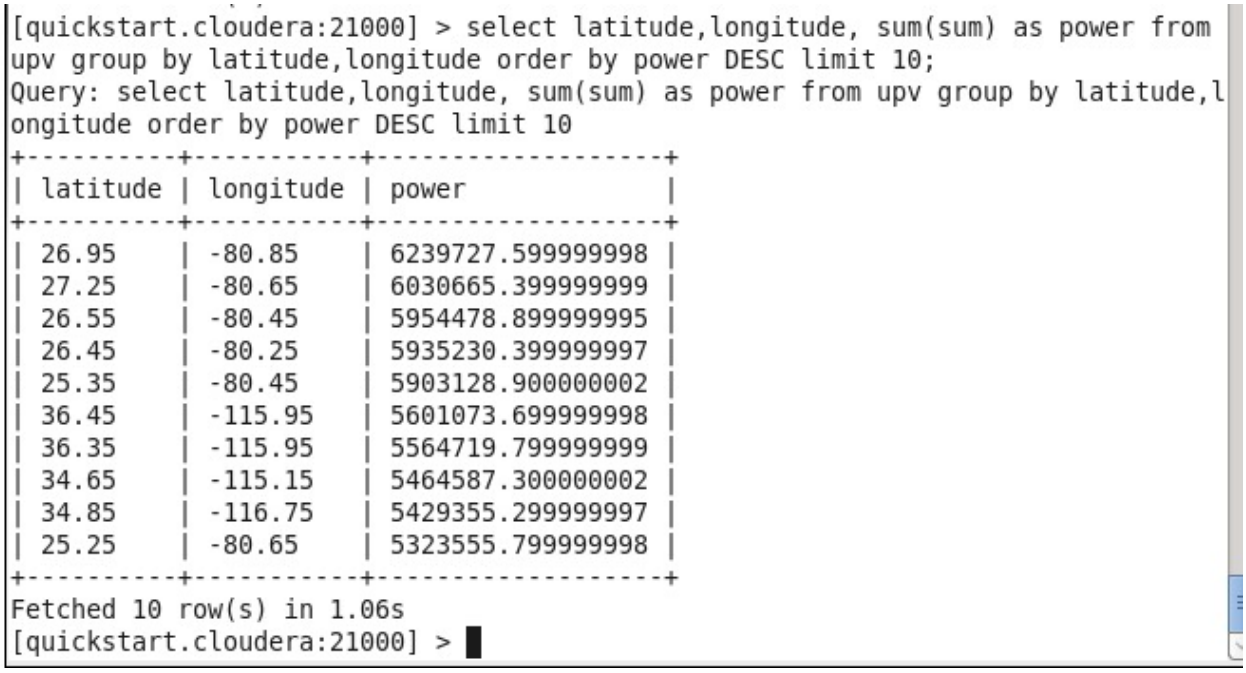
Result for the analytic:



Query to determine locations with highest UPV power generation potential (without combining with landcover data)

6. select latitude, longitude, sum(sum) as power from upv group by latitude, longitude order by power DESC limit 10;

Snapshot for the query:



```
[quickstart.cloudera:21000] > select latitude,longitude, sum(sum) as power from
upv group by latitude,longitude order by power DESC limit 10;
Query: select latitude,longitude, sum(sum) as power from upv group by latitude,l
ongitude order by power DESC limit 10
+----------+-----------+-------------------+
| latitude | longitude | power             |
+----------+-----------+-------------------+
| 26.95    | -80.85    | 6239727.599999998 |
| 27.25    | -80.65    | 6030665.399999999 |
| 26.55    | -80.45    | 5954478.899999995 |
| 26.45    | -80.25    | 5935230.399999997 |
| 25.35    | -80.45    | 5903128.900000002 |
| 36.45    | -115.95   | 5601073.699999998 |
| 36.35    | -115.95   | 5564719.799999999 |
| 34.65    | -115.15   | 5464587.300000002 |
| 34.85    | -116.75   | 5429355.299999997 |
| 25.25    | -80.65    | 5323555.799999998 |
+----------+-----------+-------------------+
Fetched 10 row(s) in 1.06s
[quickstart.cloudera:21000] >
```

Result of Analytic:

latitude	longitude	power	Location
26.95	-80.85	6239727.6	FL, USA
27.25	-80.65	6030665.4	FL, USA
26.55	-80.45	5954478.9	FL, USA
26.45	-80.25	5935230.4	FL, USA
25.35	-80.45	5903128.9	FL, USA
36.45	-115.95	5601073.7	NV, USA
36.35	-115.95	5564719.8	NV, USA
34.65	-115.15	5464587.3	CA, USA
34.85	-116.75	5429355.3	CA, USA
36.95	-115.15	5158296.7	NV, USA
36.35	-119.95	4895116.1	NV, USA

Query to determine locations with highest UPV power generation potential (without combining with landcover data)

7. select latitude, longitude, sum(sum) as power from upv group by latitude, longitude order by power DESC limit 10;

Snapshot for the query:

```
[quickstart.cloudera:21000] > select latitude,longitude, sum(sum) as power from
dpv group by latitude,longitude order by power DESC limit 10;
Query: select latitude,longitude, sum(sum) as power from dpv group by latitude,l
ongitude order by power DESC limit 10
+----------+-----------+-------------------+
| latitude | longitude | power             |
+----------+-----------+-------------------+
| 33.35    | -111.85   | 2340599.000000003 |
| 33.25    | -111.85   | 2331590           |
| 34.25    | -118.35   | 2263532.500000002 |
| 34.05    | -118.55   | 2263496.900000001 |
| 34.15    | -118.65   | 2254228.600000002 |
| 34.25    | -118.55   | 2250810.7         |
| 34.05    | -118.45   | 2249373.400000005 |
| 34.15    | -118.55   | 2245489.100000005 |
| 33.55    | -112.25   | 2243462.900000003 |
| 34.15    | -118.35   | 2233602.000000005 |
+----------+-----------+-------------------+
Fetched 10 row(s) in 0.93s
[quickstart.cloudera:21000] >
```

Result of analytic:

latitude	longitude	power	Location
33.35	-111.85	2340599	AZ, USA
33.25	-111.85	2331590	AZ, USA
34.25	-118.35	2263532.5	CA, USA
34.05	-118.55	2263496.9	CA, USA
34.15	-118.65	2254228.6	CA, USA
34.25	-118.55	2250810.7	CA, USA
34.05	-118.45	2249373.4	CA, USA
34.15	-118.55	2245489.1	CA, USA
33.55	-112.25	2243462.9	AZ, USA
34.15	-118.35	2233602	CA, USA