SPARK SHARED VARIABLE: BROADCAST VARIABLE HANDS-ON LAB

By www.HadoopExam.com

Note: These instructions should be used with the HadoopExam Apache Spark: Professional Trainings. Where it is executed and you can do hands on with trainer.

1. Broadcast variables are:

- Immutable
- Distributed to the cluster
- Fit in memory

2. Create Employee.txt file with following content (Huge table)

```
E01,John,36
E02,Rakesh,27
E03,Amit,45
E04,Venkat,34
E05,Chirag,29
E06,Jeevan,1
E07,Rupesh,2
E08,Lokesh,7
E09,Nitin,10
E011,A.John,36
E012,B.Rakesh,27
E013,C.Amit,45
E014,D.Venkat,34
E015,E.Chirag,29
E016,F.Jeevan,1
E017,G.Rupesh,2
E018,H.Lokesh,7
E019,K.Nitin,10
```

3. Create another file with City.txt with following content (Small lookup table)

```
1,Mumbai
2,Delhi
7,Newyork
10,Kolkata
27,Bangluru
29,Chennai
34,Jaipur
36,Ahmedabad
45,Indore
```

4. Load the data

```
//load employee data
val emp = sc.textFile("shared_variable/Employee.txt")
val empRDD = emp.map(x=> (x.split(",")(1),x.split(",")(2)))

//load cities data
val lookup  = sc.textFile("shared_variable/City.txt").map( v => v.split(","))
val cities = lookup.map(x => (x.split(",")(0),x.split(",")(1)))
cities.collect

//broadcast the cities now
```

```
val bcities = sc.broadcast(cities.collectAsMap())

//Join the data
val joined = empRDD.mapPartitions({row =>
 row.map(x => (x._1, x._2, bcities.value.getOrElse(x._2, -1)))
}, preservesPartitioning = true)
```