Kmeans

K-means Clustering on MRQL

The following instructions assume that you have already installed Hadoop in your cluster and you have tested it using some examples. The following tests use the k-means clustering query kmeans.mrql.

Run K-means Clustering on a Hadoop MapReduce Cluster

First, you need to generate random ponts and store them in a HDFS file using the MRQL program points.mrql:

```
mrql -dist points.mrql 1000000
```

This will create 1M points and will store them in HDFS as the sequence file `points.bin`. You can adjust this number to fit your cluster. Then, run the k-means clustering in MapReduce mode using:

```
mrql -dist -nodes 10 kmeans.mrql
```

where `-nodes` specifies the max number of MapReduce reducers.

Run K-means Clustering on a Hama Cluster

To run the same query using Hama, you need to know the number of simultaneous BSP tasks that can run in parallel on your Hama cluster without a problem. For example, if you have 16 nodes with 4 cores each, you need to set `-nodes` less than 64, eg 50. First, you need to generate random points and store them in a HDFS file (if you haven't done so for the MapReduce example):

```
mrql.bsp -dist -nodes 50 points.mrql 1000000
```

This will create 1M points and will store them in HDFS as the sequence file `points.bin`. You can adjust this number to fit your cluster. Then, run the k-means clustering in BSP mode using:

```
mrql.bsp -dist -nodes 50 kmeans.mrql
```

Run K-means Clustering on a Spark Standalone Cluster

To run the same query using Spark, change the SPARK_MASTER and FS_DEFAULT_NAME in `conf/mrql-conf.sh` to point to your Spark cluster URLs. Then, you need to generate random points and store them in a HDFS file (if you haven't done so for the other examples):

```
mrql.spark -dist -nodes 50 points.mrql 1000000
```

This will create 1M points and will store them in HDFS as the sequence file `points.bin`. You can adjust this number to fit your cluster. Then, run the k-means clustering in BSP mode using:

```
mrql.spark -dist -nodes 50 kmeans.mrql
```