Pagerank

PageRank 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 PageRank query pagerank.mrql.

Run PageRank on a Hadoop MapReduce Cluster

First, you need to generate a random graph and store it in a HDFS file using the MRQL program RMAT.mrql:

```
mrql -dist RMAT.mrql 100000 1000000
```

This will create a graph with 100K nodes and 1M edges using the RMAT algorithm, will remove duplicate edges, and store the graph in HDFS as the sequence file `graph.bin`. You can adjust these numbers to fit your cluster. Then, run PageRank in MapReduce mode using:

```
mrql -dist -nodes 10 pagerank.mrql
```

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

Run PageRank 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 a random graph and store it in a HDFS file (if you haven't done so for the MapReduce example):

```
mrql.bsp -dist -nodes 50 RMAT.mrql 100000 1000000
```

This will create a graph with 100K nodes and 1M edges using the RMAT algorithm, will remove duplicate edges, and store the graph in HDFS as the sequence file `graph.bin`. You can adjust these numbers to fit your cluster. Then, run PageRank in BSP mode using:

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

Run PageRank 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 URIs. Then, you need to generate a random graph and store it in a HDFS file (if you haven't done so for the others examples):

```
mrql.spark -dist -nodes 50 RMAT.mrql 100000 1000000
```

This will create a graph with 100K nodes and 1M edges using the RMAT algorithm, will remove duplicate edges, and store the graph in HDFS as the sequence file `graph.bin`. You can adjust these numbers to fit your cluster. Then, run PageRank in BSP mode using:

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