Getting Started With R-Store

Contents

1 System Configuration

1.1 Hadoop

(1) Enter the directory of hadoop: **hadoop-1.0.1**
(2) Configure **conf/masters**, **conf/slaves**
(3) Configure the parameter **HADOOP_LOG_DIR** in **conf/hadoop-env.sh**
(4) Configure the parameter "**hadoop.tmp.dir**" and "**fs.default.name**" in **conf/core-site.xml**
(5) Configure the parameter "**mapred.job.tracker**" in **conf/mapred-site.xml**

1.2 HBase

(1) Enter the directory of HBase: **hbase-stable/target/hbase-0.92.1/hbase-0.92.1**
(2) Configure **conf/regionservers**
(3) Configure "**hbase.rootdir**" in **conf/hbase-site.xml**. This value is set to the value of the "fs.default.name" (set in 1.1.(4)) contac the directory of name of HBase.
E.g. "hdfs://awan-0-00-0:44111/hbase"
(4) Configure "**hbase.tmp.dir**",
"**hbase.zookeeper.quorum**", "**hbase.zookeeper.property.datadir**" in **conf/hbase-site.xml**
(5) Configure "**HBASE_LOG_DIR**" in **conf/hbase-env.sh**

1.3 HStreaming

HStreaming is run on top of Hadoop. This Hadoop must be configured differently from the Hadoop in 1.1.

(1) Enter the directory of hadoop in Hstreaming: **hstreaming/hadoop-1.0.1**
(2) Configure **conf/masters**, **conf/slaves**
(3) Configure the parameter **HADOOP_LOG_DIR** in **conf/hadoop-env.sh**
(4) Configure the parameter "**hadoop.tmp.dir**" and "**fs.default.name**" in **conf/core-site.xml**
(5) Configure the parameter "**mapred.job.tracker**" in **conf/mapred-site.xml**

2 System Startup

We suggest that the three systems (Hadoop, HBase, Hadoop in HStreaming) are started one by one. And once one system is started, it is better to check whether it is started correctly before starting the next systems. An easy way is to open the websites listed below to check the cluster status. Replace "URL_OF_MASTER_NODE" with the correct master node in Hadoop, Hbase and Hadoop in Hstreaming, respectively.

(1) Start Hadoop: **bin/start-all.sh**
HDFS website: **http://URL_OF_MASTER_NODE:44070/dfshealth.jsp**
MapReduce website: **http:// URL_OF_MASTER_NODE:44030/jobtracker.jsp**

(2) Start HBase: **bin/start-hbase.sh**
HBase website: **http:// URL_OF_MASTER_NODE:55010/master-status**

(3) Start Hadoop in hstreaming: **bin/start-all.sh**
HDFS website: **http:// URL_OF_MASTER_NODE:34070/dfshealth.jsp**
MapReduce website: **http:// URL_OF_MASTER_NODE:34030/jobtracker.jsp**

3 Experimental Startup

First, you need to enter the directory of script "**scripts**" and configure the parameter "**ROLAP_HOME**" in "**exp.conf**" to be the home folder of the entire project.

3.1 Data Generator

(1) Enter the directory "**scripts/hstreaming**"

(2) Edit "**StreamMachines**" file in order to decide the number of machines that generate data. It is better to set this file to be the same as the "**slaves**" file in Hadoop of HStreaming.

(3) Execute "**startDataGen.sh**". This file will call the main function in java class "DataCubeRefresh.StreamGenerator". It has 8 parameters:
0: host name
1: port number
2: wait time and **3**: wait count. These two parameters let the client to sleep for a few ms after generating a number of tuples.
4: index of the current client
5: number of keys generated by this client
The key generated by this client is Parameter4 * (a random number between 0 and parameter 5)
6: number of updates after generating one value for each key
7: skew factor. E.g., "0.2" means 20% of the users are hot users.

3.2 Data Cube Refresher

(1) Enter the directory "**scripts/hstreaming**"

(2) Execute "**refreshcube.sh**". This file will call the main function in the java class "**DataCubeRefresh.CubeRefresher**". It also has 9+ parameters
0: path of berkeleydb for storing local data
1: interval for reduce function
2: number of reducers
3: interval for refreshing the datacube
4: the table of the refreshed datacube
5: the path of the schema file
6: the hdfs address for Hadoop (note that this is not the hadoop in hstreaming, but the hadoop in the project home folder)

7: the output path of this hstreaming job (note that the url must be that of hadoop in hstreaming)

8 to n: the urls for the dataGen clients.

3.3 Query

(1) Enter directory "**scripts**"

(2) Execute the script "**runQuery.sh**". It calls an example query "QueryProcessor.Query1", which has two parameters:

0: the path of the schema file

1: the folder for the hdfs output

3.3.1 Schema

An example schema for a table (TPCH part table in the data generator) is

```
part,200000
p_partkey,7,1,p_name,27.5,0,p_mfgr,15,0,p_brand,10,0,p_type,25,0,p_size,2,0,p_
container,10,0,p_retailprice,7,0
p_mfgr,p_brand,p_type,p_container,p_retailprice
```

(1) The first line is the table name and the number of keys for this table.

(2) The second line contains the column information. For each column is represented by three parameters: column name, size of the column, and whether it is the key ("1" stands for key). The column size is used in the cost-based query optimization.

(3) The third line contains the data cube information. The last column must be the numerical column. We assume that all the columns are stored in text format.