Introduction to Hive

Feng Li

feng.li@cufe.edu.cn

School of Statistics and Mathematics
Central University of Finance and Economics

Revised on December 14, 2017

Today we are going to learn...

1. Introduction
2. Basic Hive commands
3. Databases in Hive
4. HiveQL: Queries and Functions

Hive in the Hadoop Ecosystem

- Many of those low-level details in Hadoop are actually quite repetitive from one job to the next, from low-level chores like wiring together Mappers and Reducers to certain data manipulation constructs, like filtering for just the data you want and performing SQL- like joins on data sets.
- Hive not only provides a familiar programming model for people who know SQL, it also eliminates lots of boilerplate and sometimes-tricky coding you would have to do in Hadoop
- When MapReduce jobs are required, Hive doesn't generate Java MapReduce programs. Instead, it uses built-in, generic Mapper and Reducer modules that are driven by an XML file representing the "job plan." In other words, these generic modules function like mini language interpreters and the "language" to drive the computation is encoded in XML.

Installing and starting Hive

- Installing Hive is similar to installing Hadoop. We will download and extract a tarball for Hive, which does not include an embedded version of Hadoop.
- Hive uses the environment variable `HADOOP_HOME` to locate the Hadoop JARs and con- figuration files. So, make sure you set that variable as discussed above before proceeding.
- To start Hive, type hive from the terminal

```
[lifeng@emr-header-1 ~]$ hive

Logging initialized using configuration in jar...
Hive-on-MR is deprecated in Hive 2...
hive>
```

Basic Hive commands I

- Hadoop dfs Commands from Inside Hive

```
hive> dfs -ls / ;
```

- This method of accessing hadoop commands is actually more efficient than using the `hadoop dfs ...` equivalent at the bash shell, because the latter starts up a new JVM instance each time, whereas Hive just runs the same code in its current process.
- You can see a full listing of help on the options supported by dfs using this command:

```
hive> dfs -help;
```

- You don't need to leave the hive CLI to run simple bash shell commands. Simply type ! followed by the command and terminate the line with a semicolon (;). But shell "pipes" don't work.

```
hive> ! pwd;
/home/lifeng
```

- If you want to quit hive, type

```
hive> exit;
```

Basic Hive commands II

- Hive "One Shot" Commands: The user may wish to run one or more queries (semicolon separated) and then have the hive CLI exit immediately after completion. The CLI accepts a `-e` command argument that enables this feature.

```
[lifeng@emr-header-1 ~]$ hive -e "dfs -ls /;"
```

- Adding the `-S` for silent mode removes the `OK and Time taken ...` lines, as well as other inessential output
- Executing Hive Queries from Files. Hive can execute one or more queries that were saved to a file using the -f file argu- ment. By convention, saved Hive query files use the `.q` or `.hql` extension.

```
[lifeng@emr-header-1 ~]$hive -f /path/to/file/withqueries.hql
```

Databases in Hive I

- The Hive concept of a database is essentially just a catalog or namespace of tables.
- If you don't specify a database, the `default` database is used.
- The simplest syntax for creating a database is shown in the following example

```
hive> CREATE DATABASE IF NOT EXISTS financials;
OK
Time taken: 0.378 seconds
```

- At any time, you can see the databases that already exist as follows:

```
hive> SHOW DATABASES;
OK
airdata
bikedata
default
financials
mobile
mobileall
Time taken: 0.228 seconds, Fetched: 6 row(s)
```

Databases in Hive II

- If you have a lot of databases, you can restrict the ones listed using a regular expression

```
hive> SHOW DATABASES Like 'de*';
OK
default
Time taken: 0.037 seconds, Fetched: 1 row(s)
```

- Hive will create a directory in HDFS for each database. The database directory is created under a top-level directory specified by the property `hive.metastore.warehouse.dir`. Assuming you are using the default value for this property, `/user/hive/warehouse`, when the financials database is created, Hive will create the directory `/user/hive/warehouse/financials.db`. Note the `.db` extension.
- You can override this default location for the new directory as shown in this example:

Databases in Hive III

```
hive> CREATE DATABASE IF NOT EXISTS financials2
      LOCATION '/user/lifeng/myhive';
OK
Time taken: 0.268 seconds
hive> DESCRIBE DATABASE financials2;
OK
financials2 hdfs://hadoop:9000/user/lifeng/myhive lifeng USER
Time taken: 0.078 seconds, Fetched: 1 row(s)
```

- The USE command sets a database as your working database, analogous to changing working directories in a filesystem

```
hive> USE financials2;
OK
Time taken: 0.108 seconds
hive> SHOW tables;
OK
Time taken: 0.114 seconds
```

- Finally, you can drop a database:

Databases in Hive IV

```
hive> SHOW DATABASES Like 'fi*';
OK
financials
financials2
Time taken: 0.033 seconds, Fetched: 2 row(s)
hive> DROP DATABASE IF EXISTS financials;
OK
Time taken: 0.159 seconds
hive> SHOW DATABASES Like 'fi*';
OK
financials2
Time taken: 0.034 seconds, Fetched: 1 row(s)
```

- The `CREATE TABLE` statement follows SQL conventions, but HiveâĂŹs version offers sig- nificant extensions to support a wide range of flexibility where the data files for tables are stored, the formats used, etc.

Databases in Hive V

```
CREATE TABLE IF NOT EXISTS mydb.employees (
name
STRING COMMENT 'Employee name',
salary
FLOAT COMMENT 'Employee salary',
subordinates ARRAY<STRING> COMMENT 'Names of subordinates',
deductions MAP<STRING, FLOAT>
COMMENT 'Keys are deductions names, values are percentages',
address
STRUCT<street:STRING, city:STRING, state:STRING, zip:INT>
COMMENT 'Home address')
COMMENT 'Description of the table'
TBLPROPERTIES ('creator'='me', 'created_at'='2012-01-02 10:00:00
LOCATION '/user/hive/warehouse/mydb.db/employees';
```

- Suppose we are analyzing data from the stock markets. Let's assume the data files are in the distributed filesystem directory /user/lifeng/data/

Databases in Hive VI

```
[lifeng@emr-header-1 000lifeng]$ cat stock.hql
CREATE EXTERNAL TABLE IF NOT EXISTS stocks  (
exchanges STRING, symbol STRING, ymd STRING,
price_open FLOAT, price_high FLOAT, price_low FLOAT,
price_close FLOAT, volume INT, price_adj_close FLOAT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
LOCATION '/user/lifeng/data/';

[lifeng@emr-header-1 000lifeng]$ hive -f stock.hql

Logging initialized using configuration in...
OK
Time taken: 1.079 seconds
```

HiveQL: Queries

- Ch6 in Programming Hive

HiveQL: Functions

- The `SHOW FUNCTIONS` command lists the functions currently loaded in the Hive session, both built-in and any user-defined functions.
- To use a function, simply call it by name in a query, passing in any required arguments.

```
hive> SELECT concat(column1,column2) AS x FROM table;
```

- Or

```
hive> SELECT avg(price_close)
> FROM stocks
> WHERE exchange = 'NASDAQ' AND symbol = 'AAPL';
```

Suggested reading

- Capriolo, Edward, Dean Wampler, and Jason Rutherglen. **Programming Hive**: Data warehouse and query language for Hadoop. " O'Reilly Media, Inc.", 2012.