Starting with Hive

May 16, 2012

By Raul Overa
Software Engineer

So you have Big Data stored in Hadoop and want to make it accessible to Non-Java programmers? Hives lets you access your data without the need to create Map Reduce jobs . They let you access your data with SQL-like language and takes cares of translating it into Map Reduce jobs, so if you already know SQL, you can start using hive almost immediately.

Now if you have Hive installed and configured, there is a couple of small steps you need to take in order to be able to extract data with SQL-like queries.

You need to create a table structure. This will define each of your columns and data types. Here is an example of the creation of a games database:

```
CREATE TABLE games
(id INT, name STRING, published_year INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ‘,’
STORED AS TEXTFILE
```

The first two lines are really similar to the usual way we would declare a table in SQL. Let me explain why the last 3 lines are necessary:

- **ROW FORMAT DELIMITED:** This line is telling Hive to expect the file to contain one row per line. So basically, we are telling Hive that when it finds a new line character that means is a new records
- **FIELDS TERMINATED BY ‘,’:** This is really similar to the one above, but instead of meaning rows this one means columns, this way Hive knows what delimiter you are using in your files to separate each column. If none is set the default will be used which is ctrl-A
- **STORED AS TEXTFILE:** This is to tell Hive what type of file to expect. The other type of file that can be consumed is Sequence Files (Hadoop’s binary file format).

After doing this a directory will be created under “/user/hive/warehouse/tablename.” If you already have the data stored somewhere you can also create an external table like this:

```
CREATE EXTERNAL TABLE games
(id INT, name STRING, published_year INT)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ‘,’
STORED AS TEXTFILE
LOCATION ‘/user/Raul/gameFiles’.
```

Now you have your table created, all you need to do is move your files to your table directory. You can use one of this commands to do it:

hadoop fs –mv /path/to/local_file /user/hive/warehouse/table_name/

After you have created the structure you can alter the table in many ways. One of the most important to me, is the creation of partitions. I like to think of partitions as clues you would use when trying to find something. When you are searching for your cellphone, it is faster to find it if you at least know it is in a certain area, rather than have to look all over your house for it. Here is how a partition:

ALTER TABLE games ADD PARTITION(year_published=2010);

Now, this means that you need to create a new folder inside your table files directory named year_published=2010 and the files inside this should only contain data for 2010. This way Hive will quickly know where to look when you specify this partition in your where clause.

Now you are ready to start querying your tables, analyzing and extracting data out of Hadoop without the need to create Map Reduce jobs.

A couple of tips before you start using Hive:

1. Always know your data. Depending on how many clusters you have and how big is your data, your queries can take from a couple of minutes to an hour (maybe more). Knowing your data is important, so that you don't make small mistakes and have to redo a query with a small change. I always take a small chunk of data and understand it before trying to slice and dice it.
2. Take advantages of your partitions. They will speed your queries a lot. Nobody likes waiting for their queries to be done, so use this advantage whenever you can. If you are not the creator of a table, you can check what partitions are available in a table with the Hive command **describe formatted <tableName>.**
3. When you are expecting a lot of results, having them output on your Cygwin screen is not really a good idea. You can get your results into a text file by doing something like this: **./bin/hive –e 'use databasename; Select name, year from games where published_year = "2010" limit 10000'.**
4. Last but not least, take advantages of some cool functions Hive has,; like the ability to do regular expressions in your queries or be able to parse a URL by its hostname, query string etc. You can take a look at the functions available in the screenshot to the right. https://cwiki.apache.org/Hive/languagemanual-udf.html