IMPALA LAB

Overview: Impala is a product created by Cloudera to utilize the metadata schema provided through Hive (HCatalog) to run SQL for real-time (non-batch) queries. Cloudera avoids the use of MapReduce, bypassing the incurred time cost of spinning up new JVMs. Hortonworks is currently in the third and final phase of the “Stinger Initiative” to allow Hive to also have this ability to avoid MapReduce, possibly making Impala obsolete if this is done well.

1) In Cloudera Manager, start the Impala service if it is not already started.

2) Go to your File Browser (HDFS) in Hue. Upload the file Salaries.csv from the lahman baseball dataset used in previous labs.

3) Browse the Salaries.csv data in File Browser. You should see “columns”: “yearID”, “teamID”, “lgID”, “playerID”, “salary”

4) Now we need to set up Metadata (HCatalog) for this data so we can access it in Hive, Pig, and/or Impala. Go to the Metastore Manager in Hue.

5) In the Metastore Manager, select to “Create a new table from a file” and name the tables salaries.

6) On the final step, set up columns with the below names. Use your best judgment for the column types – think about them. (Remember, you're going to check for a salary greater than two million in a year less than 1988.)

   year, team, league, player, salary

7) Go to Impala in Hue and execute some SQL to select all rows where salary is greater than two million and the year is before 1988.

8) Hmmm... didn't work, did it? (╯ °□°) ╯ ︵ ┻━┻ Try running the same query in Hive UI. It should work there.

9) Why didn't it work in Impala? Impala needs to be refreshed to recognize the new HCatalog metatable for salaries. This can be done via the command line, by restarting the Impala service via Cloudera Manager, or

following the nice little tip you'll find on the Impala Query Editor page – look for “Sync table tips” on the left hand side.

Bonus: Do you have time left? Your SQL should have revealed a few players with a salary greater than two million before 1988, but you got the players' IDs instead of the players' actual names. The players' names can be found in Master.csv from the lahman dataset. Upload that Master data and create a Metastore (HCatalog) definition for it. Now, write and execute a SQL statement to JOIN master data so the query result will show the players' full names instead of their IDs.

Think about it: you can run virtually any SQL statements in Impala that work in Hive, but Impala runs faster because it does not use MapReduce. How much faster is Impala than Hive for you? Would either technology's query times be acceptable as a backend for your webapp's reports? Why/when would using these technologies be better than just a relational DB like PostgreSQL or MySQL?