Hadoop

Hive / BeeLine

Objectives

- What is Hive?
- What is BeeLine?
- Hive, Components, and Concepts
- SQL Overview
- Using Hive and BeeLine
- Labs

What is Hive?

- **Apache Hive** is a data warehouse software project built on top of Apache Hadoop for providing data summarization, query, and analysis. Hive gives an SQL-like interface to query data stored in various databases and file systems that integrate with Hadoop.
  - Wikipedia
  - Essentially, Hive translates SQL-like commands into MapReduce jobs. So, if you don't like coding MapReduce in Java, Python, Shell Scripts, Perl, you might consider Hive.

What is BeeLine?

- **Apache BeeLine** is a command shell that runs on top of HiveServer2.
- BeeLine executes queries identically to those directly submitted thru the Hive CLI, however, non-HQL commands are different than those used with Hive.
- In this class, we will be using BeeLine shell for executing our HQL.

Hive, Components, and Concepts

- Metastore – Stores the metadata for the RDMS (MySQL, Oracle, text files, etc.)
- Compiler – Compiles the hive request into a MapReduce request execution plan.
- Optimizer – Transforms the execution plan into an optimized executable.
- Executor – Actually executes the MapReduce job.
- CLI - Command Line Interface – Interactive Hive
- HiveQL – Essentially SQL with Hive extensions.

SQL Overview

– ***General HQL Rules***

- Terminate your statements with a semi-colon(;)
- Quote your strings with single quotes
- Statements are generally case-insensitive except in quoted strings. Database names and table names may be case sensitive. Column names should not be.
- Newline placement is insignificant
- Limiting rows is done in varying ways. Typically, a LIMIT clause may be used.

SQL Overview

- ***Getting Started***

- ***Show Databases;***
  - *Shows the databases you have access to.*

- ***USE Clause***
  - *Determines which database you want as your default*
    - Example: Use pilot;

- ***Show Tables;***
  - Shows you the tables in your default database.

- ***Describe pilot_basic;***
  - *Shows you the data columns in your table*

SQL Overview

- ***SELECT***
- *The SQL SELECT statement returns a result set of records from one or more tables.*
- WHERE specifies which rows to retrieve.
- GROUP BY groups rows sharing a property so that an aggregate function can be applied to each group.
- HAVING selects among the groups defined by the GROUP BY clause.
- ORDER BY specifies an order in which to return the rows.
- AS provides an alias which can be used to temporarily rename tables or columns.

SQL Overview

- ***SELECT***

  - Following are, in order, the clauses that may be used in a SELECT statement:

    1. SELECT (Get)
    2. FROM (Table or tables)
    3. WHERE (Condition criteria)
    4. GROUP BY (Categorize)
    5. HAVING (GROUP BY Criteria)
    6. ORDER BY (Sort)

SQL Overview

- ***Examples:***

- ***1. SELECT * FROM pilot_basic***
- ***WHERE last_name = 'GREEN'***
- ***ORDER BY first_middle_name;***
- ***2. SELECT first_middle_name, last_name FROM pilot_basic***
- ***WHERE last_name = 'WARWICK'***
- ***ORDER BY last_name, first_middle_name;***
- ***3. SELECT first_middle_name, last_name FROM pilot_basic***
- ***WHERE last_name = 'WARWICK'***
- ***ORDER BY last_name, first_middle_name DESC;***

SQL Overview

- ***Examples:***

- ***1. SELECT COUNT(*) FROM pilot_basic;***
- ***WHERE state = 'TX';***

- ***2. SELECT COUNT(*), state FROM pilot_basic***
- ***GROUP BY state;***

- ***3. SELECT COUNT(*) as pilots, state FROM pilot_basic***
- ***GROUP BY state***
ORDER BY pilots;

SQL Overview

- ***Examples:***

- ***4. SELECT COUNT(*) as pilots, state FROM pilot_basic***
  WHERE state != ‘ ’
- ***GROUP BY state***
  ORDER BY pilots;

Using Hive and BeeLine

- ***Interactive HIVE***
  - ***Command: hive***
  - ***Exiting: quit;***
  - ***For now, your commands are essentially SQL statements***
  - ***Type your command or query***
  - ***Command will be executed when hive encounters a semi-colon.***
  - ***Don't goof your command up! Type carefully.***

Using Hive and BeeLine

- *Interactive HIVE*
  - *You might want to put your hive query in a file and bring the file into interactive Hive. Traditionally, we store our queries in a file with a .hql extension.*
  - *From within Hive:*
    - *source /home/rico/hql/myscript.hql;*
  - *Preserving output:*
    - *Example:*
      - *INSERT OVERWRITE LOCAL DIRECTORY '/home/rico/hive_output' ROW DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE*
      - *SELECT LAST_NAME FROM pilot_basic*
      - *WHERE LAST_NAME LIKE 'W%'*
      - *ORDER BY LAST_NAME;*

Using Hive and BeeLine

- ***Non-Interactive HIVE***
  - ***Instead of interacting with the HIVE CLI (command line interface), you can pass your hive command or script via the shell.***
- ***Example 1:***
  - ***hive –e "Use pilot; SELECT count(*) from pilot_basic;"***
- ***Example 2:***
  - ***hive –e "Use pilot; SELECT count(*) from pilot_basic;" > myOutputFile.txt***

Using Hive and BeeLine

- ***Example 3:***
  - ***echo "use pilot; select count(*) from pilot_basic;" \| hive -S > myOutputFile.txt***
- ***Example 4:***
  - ***hive –S -f myscript.hql > myOutputFile.txt***
- ***Obviously (maybe) all of the above commands can be put into a shell script with conditional logic, loops, variables, pipes, etc. to control the behavior of the request.***

Using Hive and BeeLine

- *Interactive BeeLine*
  - *Command: beeline -u jdbc:hive2://localhost:10000*
  - *Exiting: !q*
  - *Statements beginning with a "!" are beeline commands. Otherwise, statements will be considered HQL statements.*
  - *Like your Hive HQL commands, terminate your HQL commands with a semi-colon.*

Using Hive and BeeLine

- *Interactive BeeLine*
  - *From within BeeLine:*
    - *!run /home/rico/hql/myscript.hql;*
  - *Preserving output:*
    - *Example:*
      - *INSERT OVERWRITE LOCAL DIRECTORY '/home/mark/out_stuff'*
      - *SELECT LAST_NAME FROM pilot_basic*
      - *WHERE LAST_NAME LIKE 'W%'*
      - *ORDER BY LAST_NAME;*
    - *Example:*
    - *INSERT OVERWRITE DIRECTORY '/user/mark/out_stuff'*
      - *select count(*) from pilot_basic_small;*

Using Hive and BeeLine

- *Non-Interactive BeeLine*
  - *Instead of interacting with the BeeLine CLI, you can pass your BeeLine command or script via the shell.*
- *Example 1:*
  - *echo "use pilot; select count(*) from pilot_basic_small;" | beeline --silent=true*
- *Example 2:*
  - *echo "use pilot; select count(*) from pilot_basic_small;" | beeline --silent=true > myOutputFile.txt*

Using Hive and BeeLine

- *Example 3:*
  - *echo "use pilot; select count(*) from pilot_basic_small;" | beeline --silent=true > myOutputFile.txt*
- *Example 4:*
  - *beeline -n hive --silent=true -f pilotsmall1.hql > myOutputFile.txt*
- *Again, putting the above in a shell script makes sense. Example 4 in a shell script named pilotsmall.sh would be invoked as:*
  - *pilotsmall.sh # Way better*

Labs

- ***Work with BeeLine interactively***
- ***Pass HQL to BeeLine from the command line***
- ***Build a shell script to launch BeeLine applications***