Hive

Definition

- The Apache Hive ™ data warehouse software facilitates querying and managing large datasets residing in distributed storage.
- Hive provides a mechanism to project structure onto this data and query the data using a SQL-like language called HiveQL.
- It was developed at Facebook and later open sourced via Apache
- HiveQL (HQL) also allows traditional map/reduce programmers to plug in their custom mappers and reducers when it is inconvenient or inefficient to express this logic in HiveQL.

Why

- Hive minimizes the effort of migrating to Hadoop, since data teams know SQL and all Data Warehouse apps are written in SQL,
- Ad-hoc queries of data.
- Analysis of large data sets.

Installation

- Download Hive from Apache site

  http://www.eu.apache.org/dist/hive/hive-2.0.0/apache-hive-2.0.0-bin.tar.gz
- Untar (tar xvfz apache-hive-2.0.0-bin.tar.gz)
- Move (mv apache-hive-2.0.0-bin hive2)
- Set Path etc…
  - export HIVE_HOME=/home/hdtester/hive2
  - export PATH=$PATH:$HIVE_HOME/bin
  - Also, its recommended to set the below variable to get rid of incompatabilities amongst dependencies

    HADOOP_USER_CLASSPATH_FIRST=true

First steps

- Hive requires a DataBase to store schema mapping.
- You can use MySQL/Oracle… to store Hive Schema, you could choose derby DB if you want to use Hive for a single node cluster.
- A Hive schema can be created using schematool, run the below command (if bin dir of hive is set on path)

  schematool -dbType derby –initSchema

- The above command creates the Hive Schema aka the metastore in your working directory (do a ls –ld metast*)
- Create the Directory /user/hive/warehouse in your HDFS

```
hdfs dfs -mkdir -p /user/hive/warehouse
```

Run Hive

- Have can be run in localmode and in cluster mode. Cluster mode is default.
- Consider using different metastores when switching modes
- LocalMode
  - HIVE_OPTS='-hiveconf mapred.job.tracker=local -hiveconf fs.default.name=file:///tmp‘ ; hive
- ClusterMode
  - Just type hive
- We would be using Cluster mode (that means we need HDFS+YARN+HS running)

Note: hive has released a new version (2.0.0) a week back in which MR is deprecated, alternate engines being spark and tez. You can set an alternate execution engine using the env variable HIVE_OPTS or use set on hive prompt –
export HIVE_OPTS='-hiveconf hive.execution.engine=spark‘
or at hive prompt do
set hive.execution.engine=spark (default is mr)

Lets start

- After you create the Metastore, start HDFS, YARN and HS
- Start hive (hive is a client program)
- We would be using ClickStream data from Wikipedia
  - Wikipedia published their clickstream data for 2015Jan and 2015, each dataset is ~1GB in size. Imagine the Clickstream data for the whole year?
  - The data is available at https://figshare.com/articles/Wikipedia_Clickstream/1305770
- You may face issues in analyzing the whole of the data from above, hence I prepared a small set from above consisting of 10K rows of data from Jan2015 CS data
- Use the trimmed dataset, the trimmed dataset is available at out Github site in 15-Hive-1 dir

DataSet format

PrevPage_ID	Curr_PageID	NumHits	Prev_PageTitle	Current_PageTitle
713020	2516600	56	Ju|'hoan_dialect	!Kung_language
657547	1118809	38	Stephen_Root	Crocodile_Dundee_II
	1118809	335	other-empty	Crocodile_Dundee_II
2321513	1118809	81	John_Meillon	Crocodile_Dundee_II
33437103	2321513	76	The_Picture_Show_Man	John_Meillon
1688639	2321513	27	They're_a_Weird_Mob_(film)	John_Meillon

The data is TAB delimited.
Explanation of the data – A Page having title Crocodile_Dundee_II (a movie) having an ID 1118809 was accessed 81 Times from the page titled John_Meillon (the Hero of the movie), the page John_Meillon has the ID 2321513, in total the Crocodile_Dundee_II page was accessed 38+335+81 times in the above sample.

Use Case

- Lets assume you have all the dataset of Wikipedia in a HDFS cluster
- I want to find how many hits in total have happened to the page titled Crocodile_Dundee_II
  - There are 4 options we know as of now (with the frameworks that we have seen until now), enumerate pros and cons of each…
    1. Conventional programming
    2. MapReduce
    3. Pig
    4. Hive
- Upload the data into /wikics HDFS dir

Hive in action

- Start hive cli
- Create a table

```
create table Wiki_data (PrevPage_ID BIGINT,Curr_PageID BIGINT,NumHits
BIGINT,Prev_PageTitle string,Current_PageTitle string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n' ;
```

- List tables

```
show tables
```

- Select data from Table

```
Select * from wiki_data
```


- Load data into Table, and do a select

```
LOAD DATA INPATH '/wikics' INTO TABLE Wiki_data;
```

- Select sum of Hits to our film

```
select sum(NumHits) from wiki_data where
Current_PageTitle='"Crocodile"_Dundee_II';
```

Hive in action

- Examine what you have in /wikics HDFS dir
- Examine the content in /user/hive/warehouse/wiki_data HDFS dir
- Now create another table and query the data, we wanted to check the different incomings to our movie

```
create external table Wiki_data_ext (PrevPage_ID BIGINT,Curr_PageID
BIGINT,NumHits BIGINT,Prev_PageTitle string,Current_PageTitle string)

ROW FORMAT DELIMITED

FIELDS TERMINATED BY '\t'

LINES TERMINATED BY '\n'

LOCATION '/wikics';


select count(*) from wiki_data_ext where
Current_PageTitle='"Crocodile"_Dundee_II';
```

- Examine what you have in /wikics and /user/hive/warehouse HDFS dirs.
- Drop both the tables from hive prompt, you could also use purge option while dropping. You could also do a truncate table <tableName>

```
Drop table wiki_data_ext;

Drop table wiki_data;
```

HiveQL

- The SQL Dialect that Hive uses
- Almost similar to SQL (with variations suited for file formats) with a lot of limitations (example, subqueries are not allowed anywhere except in from clause etc…)
- Language Reference - https://cwiki.apache.org/confluence/display/Hive/LanguageManual

A practical example

- Find the page which was accessed the most (exclude the main page, this has a title 0)

  select sum(numhits) as s_numhits from wiki_data_ext where current_PageTitle<>'0 group by curr_pageid

  - ~~Select max from above~~ (nesting aggregate functions is not yet supported), hence we need to do a SQ

  select max(inlinetable.s_numhits) from (select sum(numhits) as s_numhits from wiki_data_ext where Current_PageTitle<>'0' group by curr_pageid) inlinetable

  - We are tempted to do the below, but this is not yet supported.

```
select Current_PageTitle from wiki_data_ext group by Current_PageTitle having sum(numhits) =
(select max(inlinetable.s_numhits) from (select  sum(numhits) as s_numhits from wiki_data_ext where
Current_PageTitle<>'0'  group by curr_pageid) inlinetable)
```

A practical example

- We look for alternatives now. We take help of parameters…
- Create a file (sumhits.hql with the below query) you can store hive queries in a file with extension .hql and execute them with hive –f option. Unfortunately this prints a lot of mess which can be avoided by –S switch (caps S), to avoid all warnings we do a redirection also.
- `select max(a.s_numhits) from (select sum(numhits) as s_numhits from wiki_data_ext where Current_PageTitle<>'0' group by curr_pageid) a`

  You can execute the file with option

  ```
  hive -S -f sumhits.hql 2> /dev/null
  ```

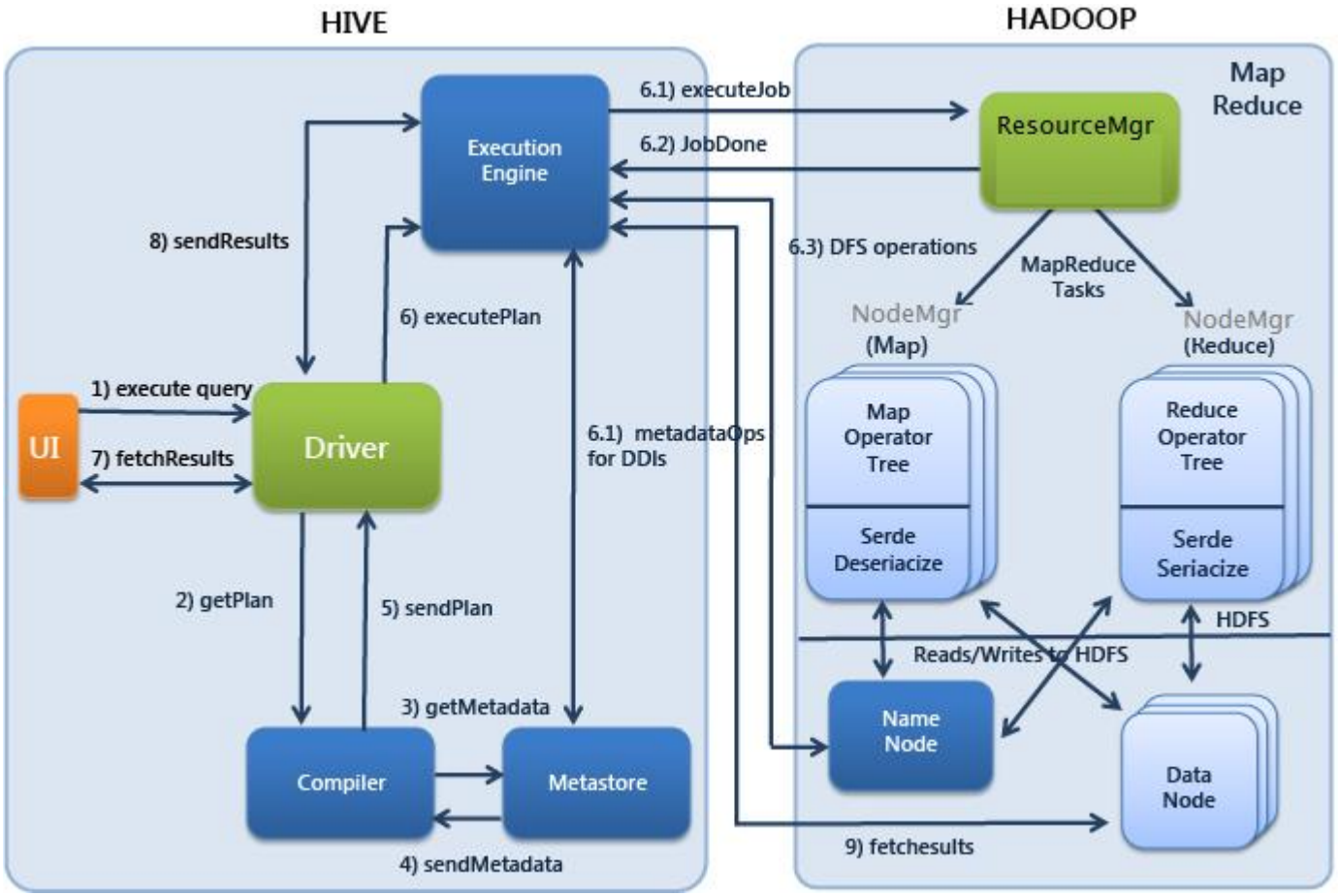
- Create another hsql file (maxhits.hql) with below content, note that the variable MAXHITS would be passed from command line

  ```
  select Current_PageTitle from wiki_data_ext where Current_PageTitle<>'0' group by Current_PageTitle having
  sum(numhits) = ${hiveconf:MAXHITS}
  ```

- Run the whole stuff, we are passing a variable named MAXHITS here, with the value that was fetched from sumhits.hql file, note the back quote to have the inner hive command executed by bash.

  ```
  hive -S -hiveconf MAXHITS=`hive -S -f sumhits.hql 2> /dev/null` -f maxhits.hql
  ```

Architecture of Hive



Architecture of Hive