THE UNIVERSITY OF SYDNEY

School of Information Technologies
A/Prof Uwe Roehm

COMP5349: Cloud Computing **1.Sem./2017**

Week 6: Hive Practice

13.04.2017

HIVE on Azure HDInsight

We will use Azure HDInsight to practice HIVE. Please start a HDInsight cluster as you did in tutorial 3. If your storage account and container for HDInsight still exist, please point to those. Allow for around 20 minutes for the cluster to start.

Lab exercise

Question 1: Simple Query With Sample Table

a) Access the HIVE View
Once your cluster is started, click the cluster URL and log in with your administrator user name (`admin`) and password. From the links at the top of the page, select `Hive View`. This is the Web UI you can use to explore HIVE tables and type in queries. Figure 1 shows you how to bring up the HIVE View and what it looks like initially.

`Query Editor` is where you can type and execute all your HIVE queries. You can have multiple work sheets there. Hive uses concepts that are quite similar to a RDBMS. These include database, tables and queries. The `Database Explorer` shows a default database with a `hivesampletable` in it. The sample table contains a number of fields. The actual data is stored as a text file in HDFS

`/hive/warehouse/hivesampletable/HiveSampleData.txt`

b) **Simple Query Using Sample Table**
We can run some simple queries to get an idea of the sample table. Type the following `SELECT` query in the worksheet of `Query Editor` and click the `execute` button. This query tries to find out the number of records in `hivesampletable`

```
select count(1) from hivesampletable;
```

After submitting the query, a query process results frame will appear beneath query editor. This frame has two tabs showing the logs and results. Your query may take a while to process. After it finishes the result table will display a count.

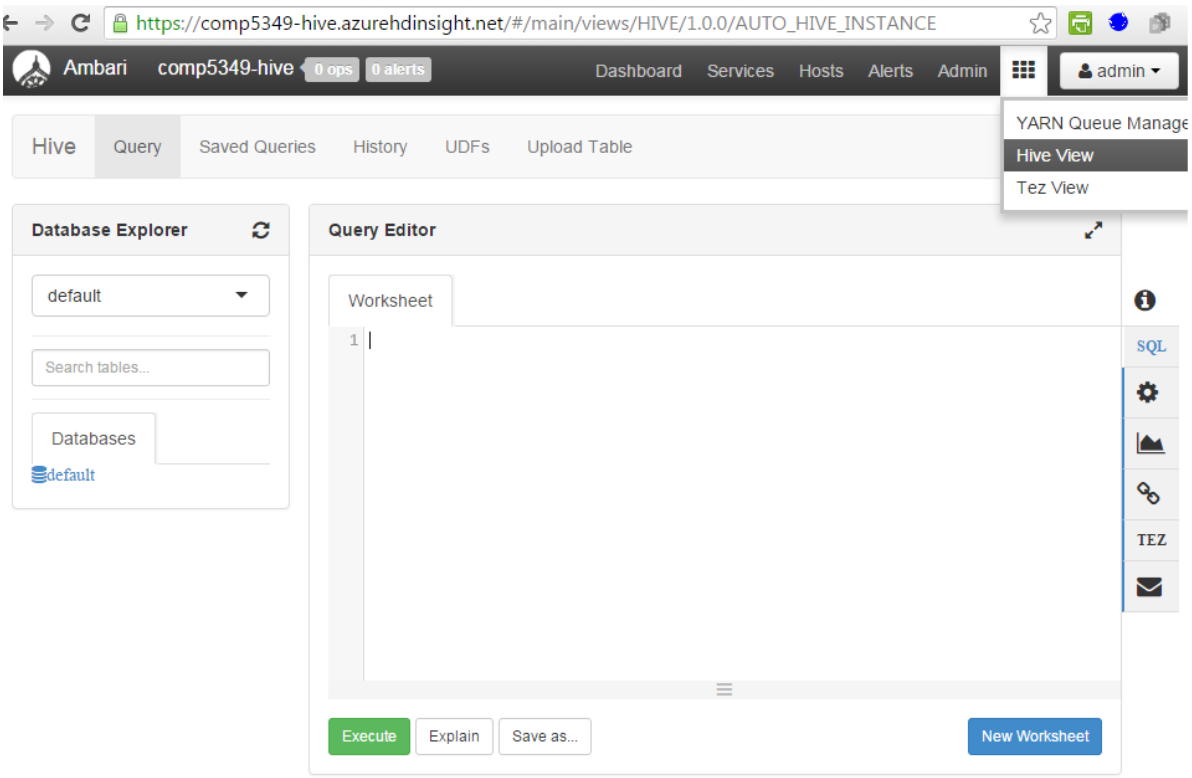

Figure 1: HDInsight Hive View

The query is executed as a Tez job consisting of many mappers and reducers. Complex HIVE queries may require a chain of `mapper-reducer` to implement. MapReduce framework would convert each `mapper-reducer` pair as a job and chain them together. Tez tries to optimize the overall flow by organizing a number of mappers and reducers as a DAG and execute them in a single job

To see how your query is executed, click `Tez View` from the links at the top of the page. All Tez DAGs will be listed (see Figure 2). There is only one so far.

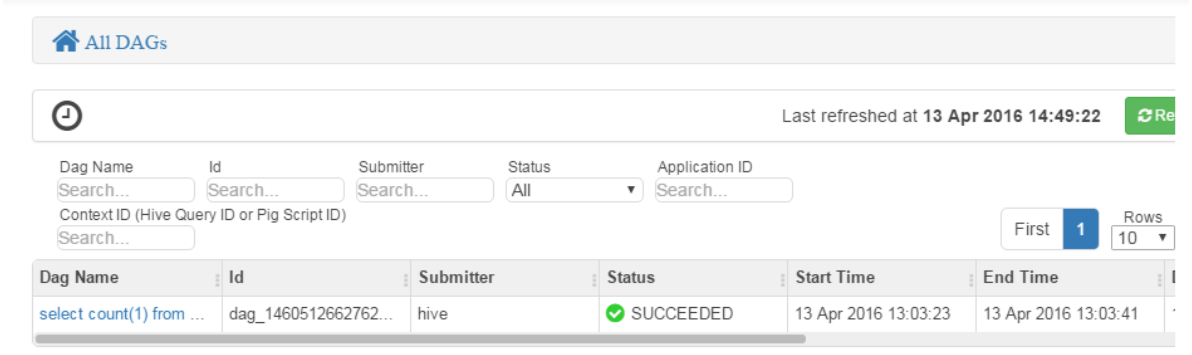

Figure 2: HDInsight Tez View

Click the DAG Name represented by your HIVE query will show you details of the DAG, such as how many mappers and reducers are included in the DAG. It also shows how many tasks are started during the execution. Our `select-count` query only requires a mapper and a reducer (see Figure 3), each runs as a single task.

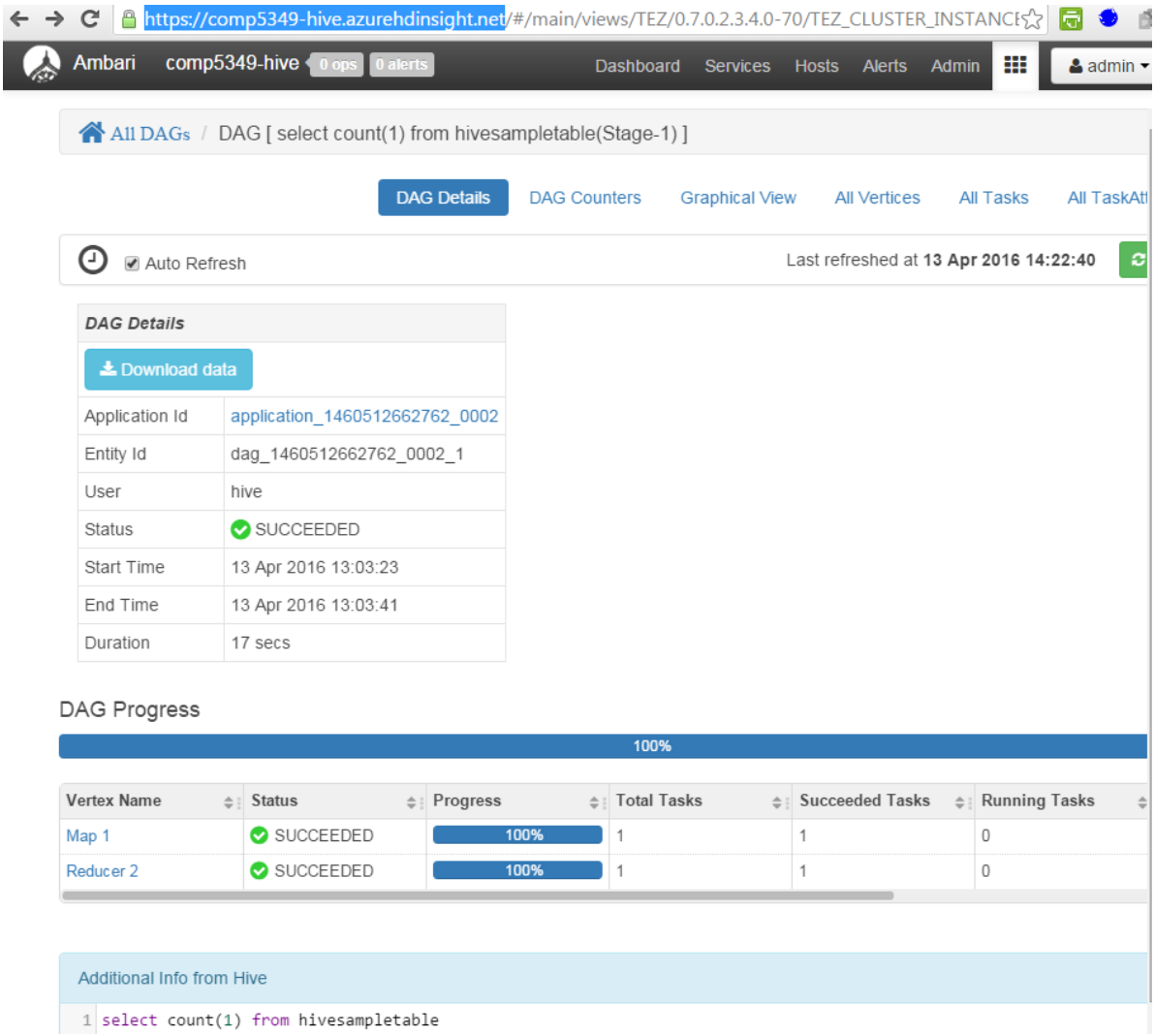

Figure 3: HDInsight Tez DAG degail of select-count query

Question 2: Work on Your Own Table

a) Prepare the table data

Follow the instruction on week 3 tutorial to SSH to your cluster node and, if you haven't already, create your user directory within HDFS.

Once connected and setup, create a `data` directory under your home directory and work on the directory. Run the following command to download a photo and a place data file.

```
wget https://2017sem1comp5349.blob.core.windows.net/photo-data/n07.txt
wget https://2017sem1comp5349.blob.core.windows.net/place-data/place.txt
```

Now run the following commands to create two HDFS folders and upload the data files to the respective folders.

```
hdfs dfs -mkdir -p hiveData/photo
hdfs dfs -put n07.txt hiveData/photo
hdfs dfs -mkdir -p hiveData/place
hdfs dfs -put place.txt hiveData/place
```

The `put n07.txt` command may take a while to run.

b) Create EXTERNAL tables based on existing data

Go back to the your cluster's HIVE view page. And type in the following query in `Query Editor`

```
create EXTERNAL table IF NOT EXISTS photo
( photo_id  STRING,
    owner   STRING,
    tags    STRING,
    date_taken     STRING,
    place_id       STRING,
    accuracy   INT)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n'
LOCATION '/user/sshuser/hiveData/photo';

create EXTERNAL table IF NOT EXISTS place
( place_id STRING,
    woeid   STRING,
    lat STRING,
    lon STRING,
    place_name  STRING,
    place_type_id   INT,
    place_url   STRING
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n'
LOCATION '/user/sshuser/hiveData/place';
```

The queries will create two tables based on data stored in HDFS. This is called external table and is indicated by the `EXTERNAL` keyword in the create table statement. The last line of each query specifies the location of the data. You should specify the directory, not the actual file. Also note that Hive needs to use an absolute HDFS path. The

query assumes that you have followed all instructions in week 3 tutorial and created a `sshuser` folder under `user`.

Click `execute` to execute the query.

You may run a select-count query on your own table to see how many records are there.

HIVE only manages meta data of an external table, which is the schema definition. When you drop an external table, the table is removed from the the database; but the actual table data is still there. HIVE can manage both the meta data and the actual data of a table. Such table is called managed table. Regular table creation statement without the `EXTERNAL` keyword will create a managed table. When you drop a managed table, both the meta data and the actual data will be deleted. HIVE managed tabled can be stored as regular text file or as its own storage format such as RC or ORC format. The storage format is specified in the table-creation statement.

c) Create empty table to store query results.

Add a new worksheet on your HIVE VIEW and execute the following table creation command:

```
create table placeFreq
( owner STRING, place_id STRING, count INT )
ROW FORMAT DELIMITED
FIELDS TERMINATED BY '\t' LINES TERMINATED BY '\n'
STORED AS TEXTFILE
LOCATION '/user/sshuser/hiveData/placeFreq';
```

This statement created a managed table stored as text file. After the table is created, you will see a directory `placeFreq` under `/user/sshuser/hiveData/`. The directory is currently empty because the table does not have any data. Note that hive table location should always point to a directory instead of a file.

This newly created table will be used to store data generated by a `GROUP BY` query which finds the number of photos a user took in a particular place. The following command runs the `GROUP BY` query on table `photo` and store the results in table `placefreq`.

```
INSERT OVERWRITE TABLE placefreq
SELECT owner, place_id, count(1)
FROM photo
GROUP BY owner,place_id;
```

Once the execution finishes, check the `/user/sshuser/hiveData/placeFreq` folder in HDFS, you will find one file in the folder, this file is generated by the single reducer of the job. It hold the data of `placefreq` table.

You can check the execution profile from Tez View as well. This query again only requires a pair of mapper and reducer tasks. There are 10 actual map tasks running.

Additional notes:

HIVE is designed for Data Warehousing type of applications. There is no command

to `insert/update/delete` a single or a few rows in a table. All data manipulations happen at the table level. You can create a table, overwrite the whole table, or delete the whole table. The command to delete a table is `DROP TABLE table-name`.

d) More Complex query

HIVE allows a query to include sub queries. Such complex query may require a few mappers and reducers to process. The following is an example of a complex query:

```
SELECT freq.owner, p.place_name, freq.count
FROM
    (SELECT owner, place_id, count(1) AS count FROM photo
    group by owner, place_id) freq
JOIN place p ON p.place_id=freq.place_id
WHERE freq.count > 5000
ORDER BY freq.count DESC;
```

This query produces a list of popular places with at least one user taking at least 5000 photos on that place. Figure 4 shows a graph view the execution DAG, which involves two mappers and two reducers. You may notice that the DAG does not follow the strict `mapper-reducer` sequence.

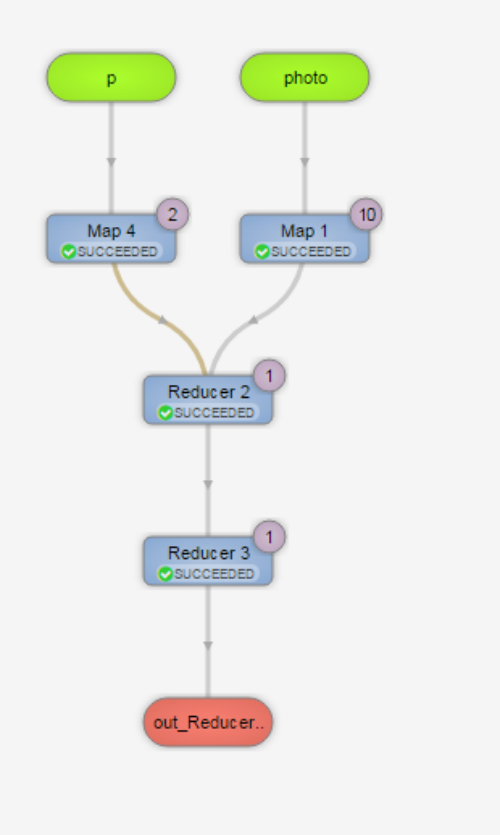

Figure 4: HDInsight Tez DAG Graph View

Question 3: Write Your Own Queries

Write a query to find all users who have taken photos in Australia. For each user, count the number of photos he/she takes in each place. The output should be printed to the console.

Submit the query to the cluster and inspect the execution DAG from Tez View