WQD7007 Big Data Management

Introduction to Pig

Introduction

- In this lab, we are going to practice how to analyze large amount of data as **data flows** using Apache Pig.
- Pig use Pig Latin scripting language, to achieve adhoc data analysis in an **iterative fashion**
- Pig sits on top of MapReduce, so all Pig scripts run as Map and Reduce task.

Installation

- Online reference: https://www.edureka.co/blog/apache-pig-installation

```
wget http://www-us.apache.org/dist/pig/pig-0.16.0/pig-0.16.0.tar.gz
tar -xzf pig-0.16.0.tar.gz
mv pig-0.16.0 /home/{yourname}/pig/
```

- In `.bashrc`:

```
export PATH=$PATH:/home/{yourname}/pig
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64/jre/
```

- Execute `source .bashrc`
- Execute `pig`

Load Data

- Write Pig script:

```
batting = load '/user/hdfs/batting.csv' using PigStorage(',');
raw_runs = FILTER batting BY $1>0;
```

- No result appeared even though the operation is completed.
  - This is because not DUMP command is called to display result or save to storage.
  - `DUMP raw_runs`
- Sample result (1st line):
  - `(aardsda01,2004,1,SFN,NL,11,11,0,0,0,0,0,0,0,0,0,0,0,0,0,0,11)`

Filter data

- Pig Characteristics: iterative. Means we an step into each intermediate step. Example:

```
Runs = FOREACH raw_runs GENERATE $0 as playerID,
$1 as year, $8 as runs;
```

Aggregate Data

- Data can be grouped based on elements e.g. according to the year by setting grp_data object to be indexed by year. Example:

```
grp_data = GROUP runs by (year);
max_runs = FOREACH grp_data GENERATE group as grp,MAX(run.runs) as max_runs;
DUMP max_runs
```

Join Data

- We have the maximum for each year but we need to join this with the runs data object.
- We want our output result in the form of (Year, PlayerID and Max Run). Example:

```
• join_max_run = JOIN max_runs by ($0, max_runs),
  runs by (year, runs);
• join_data = FOREACH join_max_run GENERATE $0 as
  year, $2 as playerID, $1 as run.
• DUMP join_data
```

Another example: Movie data

1. Download movies_data.csv and upload it to HDFS.
2. Run the following scripts in Pig:
   - movies = LOAD '/user/hdfs/movies_data.csv';
   - USING PigStorage(',') as (id, name, year, rating, duration);
   - DUMP movies;
3. Filter data iteratively (find movies that are worth watching → rating higher than 4.0):
   - `movies_greater_than_four = FILTER movies BY (float)rating>4.0;`
   - `DUMP movies_greater_than_four`

Another example: Movie data

4. Write outcome to persistent storage:
   - `STORE movies_greater_than_four into ‘/user/hdfs/movies_greater_than_four’;`
5. Look for classic movies that are between 50s and 60s:
   - `movies_between_50_60 = FILTER movies by year>1950 and year<1960;`
6. Retrieve movies that start with the character ‘A’
   - `movies_starting_with_A = FILTER movies by name matches ‘A.*’;`

* Use Hive query to retrieve the same information. What are the pros and cons between Hive and Pig?