Apache PIG

Apache Pig is a platform for analysing large data sets. It provides a high-level language, called Pig Latin, which abstracts the complexities of writing MapReduce programs in Java. Pig Latin scripts are translated into MapReduce jobs by the Pig runtime environment, allowing users to perform data processing tasks without having to write complex Java programs.

Pig Latin: This is the high-level scripting language used in Apache Pig. It resembles SQL and is designed to make it easy for users to express data processing tasks concisely.

Grunt Shell: Grunt is the interactive shell provided by Pig. It allows users to enter Pig Latin commands interactively and execute them directly.

Loading the data

Syntax:

- varName = LOAD 'path_of_data' USING PigStorage('delimiter') AS (metadata);
- dump varName;

Task 1: Load the u.data into Pig

```
ratings = LOAD  '/user/maria_dev/u.data'  USING PigStorage('	')  AS  (uid:int, fid:int, rating:int, ts:int);
dump ratings;
```

Task 2: Load the u.item into Pig. Just load the first three columns.

```
items = LOAD  '/user/maria_dev/u.item'  USING PigStorage('|')  AS  (fid:int, title:chararray, releaseDate:chararray);
dump items;
```

Note: The above variables are temporary.

Group by

Task 3: Group the ratings based upon filmid

```
ratings = LOAD  '/user/maria_dev/u.data'  USING PigStorage('	')  AS  (uid:int, fid:int, rating:int, ts:int);

ratingsByMovie = GROUP  ratings  BY  fid;

dump ratingsByMovie;
```

Modify the variables (FOREACH and GENERATE)

Task 4: Convert the releaseDate to Unixtime format. This will be useful in sorting the films based on time

```
items = LOAD  '/user/maria_dev/u.item'  USING PigStorage('|')  AS  (fid:int, title:chararray, releaseDate:chararray);

unixTime = FOREACH items GENERATE fid, title, ToUnixTime(ToDate(releaseDate, 'dd-MMM-YYYY')) AS unixTime;

dump unixTime;
```

Task 5: Find the average ratings based upon the film id.

```
ratings = LOAD  '/user/maria_dev/u.data'  USING PigStorage('	')  AS  (uid:int, fid:int, rating:int, ts:int);

ratingsByMovie = GROUP  ratings  BY  fid;

avgratings = FOREACH ratingsByMovie GENERATE group AS fid, AVG(ratings.rating) AS avgRating;

fourStarFilms = FILTER avgratings by avgRating > 4.0;

dump fourStarFilms;
```

Join

Task 6: Find the oldest four-star films

```
ratings = LOAD  '/user/maria_dev/u.data'  USING PigStorage('	')  AS  (uid:int, fid:int, rating:int, ts:int);

ratingsByMovie = GROUP  ratings  BY  fid;

avgratings = FOREACH ratingsByMovie GENERATE group AS fid, AVG(ratings.rating) AS avgRating;

fourStarFilms = FILTER avgratings by avgRating > 4.0;


items = LOAD  '/user/maria_dev/u.item'  USING PigStorage('|')  AS  (fid:int, title:chararray, releaseDate:chararray);

unixTime = FOREACH items GENERATE fid, title, ToUnixTime(ToDate(releaseDate, 'dd-MMM-YYYY')) AS unixTime;


bestFilmsData = JOIN fourStarFilms by fid, unixTime by fid;

bestOldFilms = ORDER bestFilmsData by unixTime::unixTime;

dump bestOldFilms;
```