Advanced Data Management (CMM524)

Laboratory #7: Introduction to Pig

1. Aims

- Use Pig to perform ETL and data analysis tasks.

2. Outcomes

In completing this exercise, you should be able to:
- Run the Grunt shell interactively.
- Run Pig scripts in local mode.
- Load datasets using Pig Latin.
- Write Pig Latin scripts to perform ETL and simple data analysis jobs.

3. ETL Processing of 1st Dataset

The company *Dualcore* uses two online advertisement networks to promote its business. The networks provide data on where an advertisement was placed, the date it was placed, the keywords used, whether a user clicked the advertisement, and the cost-per-click they charge *Dualcore*.

3.1. Preparing a Local Sample File

The first dataset `ad_data1.txt` is in the Unix folder `~/training_materials/analyst/data`.

- Show the first 25 lines of the dataset with the Unix command:

```
head -n 25 ~/training_materials/analyst/data/ad_data1.txt
```

To test out the Grunt shell, we will create a small sample file in the local Unix folder `~/training_materials/analyst/exercises/pig_etl`.

- Change into the desired Unix folder:

```
cd ~/training_materials/analyst/exercises/pig_etl
```

- Create a sample file `sample1.txt` with the following command (**in 1 single line!**):

```
head -n 25 ~/training_materials/analyst/data/ad_data1.txt > sample1.txt
```

- Check and make sure that `sample1.txt` is created.
  - What is the Unix command to use to show the file content?

© The Robert Gordon University

3.2. Running Grunt in Local Mode

It is faster to test out Pig with our sample data file `sample1.txt` in Unix rather than in HDFS. To use local data files, we need to run Grunt in local mode.

- Start the Grunt shell in local mode with the Unix command:

```
pig -x local
```

- You should see the "`grunt>`" prompt.

3.3. Loading 1st Sample Data using the Grunt Shell

Grunt accepts Pig Latin commands interactively.

- In the Grunt shell, load the dataset `sample1.txt` in our current folder into a relation:

```
grunt> data = LOAD 'sample1.txt';
```

- Once loaded, you can dump the relation to the screen:

```
grunt> DUMP data;
```

  - Can you see all tuples in the relation?

We can selectively load the first 2 columns of the data into a different relation, with a data schema.

- Type the following in *Grunt* (**in 1 single line!**):

```
grunt> first_2_columns = LOAD 'sample1.txt' AS
(keyword:chararray, campaign_id:chararray);
```

- Now print out the data loaded.
  - What is the Pig Latin statement to use?
  - Can you see the loaded data? How different are they from the `data` relation?

As our relation `first_2_columns` has a schema defined, we can ask *Grunt* to give us the information.

- Type the following Pig Latin in *Grunt*:

```
grunt> DESCRIBE first_2_columns;
```

- What can you tell about the columns and their data types?
- Printing the schema of the relation `data` you loaded a while ago.
  - What is the Pig Latin command to use?
  - What can you tell about the schema?

© The Robert Gordon University

- To quit *Grunt*, you can press *Ctrl-D* or use the command `QUIT`.
- Note: ***Knowing the schema of loaded data is a very important skill in debugging Pig scripts.***

3.4. Editing & Running Pig Scripts

To process the input dataset from the first advertisement network, we will create a Pig script in a file.

- Edit the `first_etl.pig` file to complete the `LOAD` statement to read the data from the `sample1.txt` file we created earlier. The structure of each line is described as follow:

Index	Field	Data Type	Description	Example
0	keyword	`chararray`	Keyword that triggered advertisement	`tablet`
1	campaign_id	`chararray`	Uniquely identifies the advertisement	`A3`
2	date	`chararray`	Date of advertisement displayed	`05/29/2014`
3	time	`chararray`	Time of display	`15:49:21`
4	display_site	`chararray`	Domain where ad was shown	`www.example.com`
5	was_clicked	`int`	Whether ad was clicked	`1`
6	cost_per_click	`int`	Cost-per-click, in cents	`106`
7	country	`chararray`	Name of country in which ad ran	`USA`
8	placement	`chararray`	Where on page was ad displayed	`TOP`

- Once you have finished the `LOAD` statement, add a `DUMP` to print the data to the screen.
- Test your Pig script in local mode with the Unix command:

```
pig -x local first_etl.pig
```

3.5. Filtering

Once data are loaded into a relation, you can filter out data that do not satisfy a condition.

- Update your script to filter out records where country field does not contain `USA`.
  - **That means you should keep records from USA only.**
- Test your script on the sample data.

© The Robert Gordon University

3.6. Generate a New Relation

We need to store the fields in a different order.

- Use a `FOREACH... GENERATE...` statement to create a new relation containing the fields in the following order:

Index	Field	Description
0	campaign_id	Uniquely identifies the advertisement
1	date	Date of advertisement displayed
2	time	Time of advertisement displayed
3	keyword	Keyword that triggered the advertisement
4	display_site	Domain where ad was shown
5	placement	Where on page was ad displayed
6	was_clicked	Whether the ad was clicked
7	cost_per_click	Cost-per-click, in cents

- Update your script to convert the `keyword` field to uppercase and to remove any leading and trailing white spaces.
  - Hint: Use the `UPPER` and `TRIM` functions.
- Test your script on the sample data.

3.7. Running Pig Script in MapReduce Mode

If your script runs fine on the sample data, you can now run it on the real data set in HDFS.

- Copy the first dataset `ad_data1.txt` in the Unix `~/training_materials/analyst/data` directory to HDFS.
- Edit the `first_etl.pig` script and change the path in the LOAD statement to match the path of the data file (above) in HDFS.
- Replace the `DUMP` statement in your script with a `STORE` to write the output of your processing as tab-limited records to the `/dualcore/ad_data1` HDFS directory.
- Run the script in MapReduce mode (**not local mode!**).
- Check the first 20 lines of your result in the `/dualcore/ad_data1` HDFS directory.
  - What is the Unix command to use?
    - (Hint: You can show the content of the result file in HDFS and then pipe it through the Unix filter "`head -n 20`".)
  - Are the fields in the correct order?
  - Are all keywords in uppercase?

© The Robert Gordon University

4. ETL Processing of 2nd Dataset

The 2nd dataset `ad_data2.txt` is in the Unix directory `~/training_materials/analyst/data/`.

4.1. Creating a Sample File

We will create a small sample of the 2nd dataset so that you can test your script locally.

- Examine the first 25 lines of `ad_data2.txt`.
  - What is the Unix command to use?
  - What is the delimiter in the data?
- Take the first 25 lines of `ad_data2.txt` to create a local file `sample2.txt`.
  - What is the Unix command to use?

4.2. Loading the 2nd Sample File

The following table explains the fields in the dataset:

Index	Field	Data Type	Description	Example
0	campaign_id	`chararray`	Uniquely identifies the ad	`A3`
1	date	`chararray`	Date of ad displayed	`05/29/2013`
2	time	`chararray`	Time of ad displayed	`15:49:21`
3	display_site	`chararray`	Domain where ad was shown	`www.example.com`
4	placement	`chararray`	Where on page was ad displayed	`TOP`
5	was_clicked	`int`	Whether ad was clicked	`Y`
6	cost_per_click	`int`	Cost-per-click, in cents	`106`
7	keyword	`chararray`	Keyword that triggered ad	`tablet`

- Edit `second_etl.pig` to complete the `LOAD` statement and read the data from the local file `sample2.txt`.
- Load the data in local mode and use the `DESCRIBE` command to check that the schema of the data matches the table above.
- If the schema is correct, replace `DESCRIBE` with `DUMP` to print the sample data to the screen.

4.3. Removing Duplicates

This 2nd dataset may contain duplicate entries.

- Use Pig Latin to remove duplicate entries in the data.
  - What is the Pig Latin command to use?
- Test your script to make sure that duplicates are removed.

© The Robert Gordon University

4.4. Re-arranging Fields

Again, we want the fields in a different order, as described in the following table:

Index	Field	Description
0	campaign_id	Uniquely identifies the ad
1	date	Date of ad displayed
2	time	Time of ad displayed
3	keyword	Keyword that triggered ad
4	display_site	Domain where ad was shown
5	placement	Where on page was ad displayed
6	was_clicked	Whether ad was clicked
7	cost_per_click	Cost-per-click, in cents

- Use Pig Latin to arrange the fields in the above order.
  - What is the Pig Latin command to use?
- Do a local run and check that the fields are in the desired order.

4.5. Fixing the Keyword Field

Again, the `keyword` field needs some fixing.

- Convert the `keyword` field to uppercase and remove all leading and trailing spaces.

4.6. Changing Date Format

The date field in the 2nd dataset is in the format MM-DD-YYYY.

- Use Pig Latin to change the date to the format MM/DD/YYYY.
  - Hint: Use the function `REPLACE(date,'-','/')` to correct this.

4.7. Processing 2nd Dataset in MapReduce Mode

Once you make sure that the script is correct, you can run it on the whole dataset.

- Upload the 2nd dataset (i.e. `ad_data2.txt`) to HDFS.
- Edit the script to load the 2nd dataset in HDFS (not the local sample!). Use a `STORE` statement to store your output to the HDFS `/dualcore/ad_data2` directory.
- Run your script in MapReduce mode.
- Check the first 15 lines of your output.
  - Do you see any duplicate record?
  - Are the fields in correct order?
  - Are all keywords in uppercase?
  - Is the date field in the correct MM/DD/YYYY format?

© The Robert Gordon University