Running Pig in Apache Zeppelin

Introduction

Apache Zeppelin is a web-based notebook that enables interactive data analytics while Apache Pig is a platform for analyzing large data sets that consists of a high-level language for expressing data analysis programs. Pig-latin is a very powerful languages for data flow processing.

One drawback of pig community complains about is that pig-latin is not a standard language like sql so very few BI tools integrate with it. So it is pretty to hard to visualize the result from pig. Now the good news is that Pig is integrated in zeppelin 0.7 where you can write pig latin and visualize the result.

Use pig interpreter

Pig interpreter is supported from zeppelin 0.7.0, so first you need to install zeppelin, you can refer this link for how to install and start zeppelin.

Zeppelin supports 2 kinds of pig interpreters for now.

- %pig (default interpreter)
- %pig.query

%pig is like the pig grunt shell. Anything you can run in pig grunt shell can be run in %pig.script interpreter, it is used for running pig script where you don’t need to visualize the data, it is suitable for data munging. %pig.query is a little different compared with %pig.script. It is used for exploratory data analysis by using pig latin where you can leverage zeppelin’s visualization ability. There're 2 minor differences in the last statement between %pig and %pig.query

- No pig alias in the last statement in %pig.query (read the examples below).
- The last statement must be in single line in %pig.query

Here I will give 4 simple examples to illustrate how to use these 2 interpreters. These 4 examples are another implementation of zeppelin tutorial where spark is used. We just do the same thing by using pig instead.

This script do the data preprocessing

```
%pig
bankText = load 'bank.csv' using PigStorage(';');
bank = foreach bankText generate $0 as age, $1 as job, $2 as marital, $3 as education, $5 as balance;
bank = filter bank by age != '"age"';
bank = foreach bank generate (int)age, REPLACE(job,'"','') as job, REPLACE(marital, '"', '') as
marital, (int)(REPLACE(balance, '"', '')) as balance;
store bank into 'clean_bank.csv' using PigStorage(';'); -- this statement is optional, it just show you
that most of time %pig.script is used for data munging before querying the data.
```

Get the number of each age where age is less than 30

```
%pig.query

bank_data = filter bank by age < 30;
b = group bank_data by age;
foreach b generate group, COUNT($1);
```

The same as above, but use dynamic text form so that use can specify the variable maxAge in textbox. (See screenshot below). Dynamic form is a very cool feature of zeppelin, you can refer this link for details.

```
%pig.query

bank_data = filter bank by age < ${maxAge=40};
b = group bank_data by age;
foreach b generate group, COUNT($1);
```

Get the number of each age for specific marital type, also use dynamic form here. User can choose the marital type in the dropdown list (see screenshot below).

```
%pig.query

bank_data = filter bank by marital=='${marital=single,single|divorced|married}';
b = group bank_data by age;
foreach b generate group, COUNT($1);
```

The following is a screenshot of these 4 examples. You can also check pig tutorial note which contains all the code of this blog in zeppelin.

blocked URL

Configuration

Pig interpreter in zeppelin supports all the execution engine that pig supports.

- Local Mode
  - Nothing needs to be done for local mode
- MapReduce Mode
  - HADOOP_CONF_DIR needs to be specified in ZEPPELIN_CONF_DIR/zeppelin-env.sh
- Tez Local Mode
  - Nothing needs to be done for tez local mode
- Tez Mode
  - HADOOP_CONF_DIR and TEZ_CONF_DIR needs to be specified in ZEPPELIN_CONF_DIR/zeppelin-env.sh

The default mode is mapreduce, but you can change that in interpreter setting. You can also set any pig configuration in the interpreter setting page. Here's one screenshot of that.

blocked URL

Future work

This is the first phase work to integrate pig into zeppelin. There’s lots of work needs to do in future. Here’s my current to-do list

- Integrate spark engine so that we can use spark sql together with pig-latin
- Integrate spark mllib so that we can use pig-latin to do machine learning
- Add new interpreter %pig.udf to allow user to write java udf in zeppelin
- Integrate more closely with datafu

If you have any other new ideas, please contact me at zjffdu@apache.org

or you can file ticket in apache zeppelin jira https://issues.apache.org/jira/browse/ZEPPELIN