CPrE/SE 419: SOFTWARE TOOLS FOR LARGE-SCALE DATA ANALYSIS, SPRING 2022

Lab 4: Pig

Purpose

The goal is to use the Pig platform and the scripting language Pig Latin, to analyze large Log Files and Trace Files. While you can write a MapReduce program for each such task, Pig can help you complete the development aspect of this task faster, since it helps you program at a higher level of abstraction than MapReduce. Scripts written in Pig Latin are automatically converted to MapReduce jobs.

During this lab, you will learn:

- The Pig platform and the scripting language Pig Latin
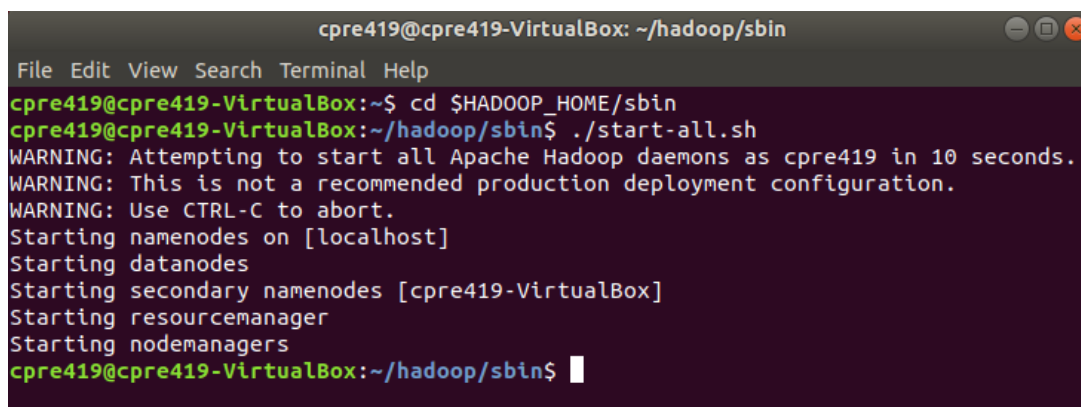- Some real world problems that can be solved using Pig

Submission

Create a single zip archive with the following and upload it on Canvas:

- Screenshots of your results for each individual experiment.
- Commented Code for your program. Include all source files needed for compilation and make sure it compiles successfully. Make sure you output the results to a specified folder of each experiment.
- DO NOT INCLUDE: jar files, IDE project files, assignment pdf, etc.

Pig

First, please check "VM_LAB_MACHINE.pdf" under "Lab Readings" module on Canvas and create VM properly.

Start HDFS and Yarn in VM,

In the local mode, Pig will not use Hadoop to distribute the work across the clusters. Instead it will perform all work locally. However, MapReduce mode is another option for Pig, which uses Hadoop and Yarn to execute the job in distributed manner.

You are provided on Canvas two Pig scripts for the Word Count problem (as well under "Downloads" folder in VM).

First, we test the Pig code in local mode. Copy Shakespeare data file under "Downloads" folder, and then execute the script with the following command

```
cpre419@cpre419-VirtualBox: ~/Downloads
File Edit View Search Terminal Help
cpre419@cpre419-VirtualBox:~/hadoop/sbin$ cd ~/Downloads/
cpre419@cpre419-VirtualBox:~/Downloads$ pig -x local wordcount_local.pig
```

The output should be available under "Downloads/lab_pig" folder.

Next, we will use MapReduce mode. Since Shakespeare data file is already on HDFS, execute the script with the following command

```
cpre419@cpre419-VirtualBox: ~/Downloads
File Edit View Search Terminal Help
cpre419@cpre419-VirtualBox:~/Downloads$ pig -x mapreduce wordcount_mr.pig
```

The output can be found at

```
File Edit View Search Terminal Help
cpre419@cpre419-VirtualBox:~/Downloads$ hadoop fs -ls /lab_pig/output
Found 2 items
-rw-r--r--   1 cpre419 supergroup          0 2020-03-03 17:02 /lab_pig/output/_SUCCESS
-rw-r--r--   1 cpre419 supergroup     260643 2020-03-03 17:02 /lab_pig/output/part-r-00000
cpre419@cpre419-VirtualBox:~/Downloads$
```

Want more examples?

Here is the link of pig scripts in book "Programming Pig" by Alan Gates

https://github.com/alanfgates/programmingpig

Look here for documentation on pig:

http://pig.apache.org/docs/r0.14.0/

The Pig Latin language can be learnt from the link:

http://pig.apache.org/docs/r0.14.0/basic

Pig utilities:

http://pig.apache.org/docs/r0.14.0/cmds

Experiment 1 (20 points)

For this experiment we will use Pig in local mode and analyze US demographic data.

It can be found at: http://www.census.gov/geo/maps-data/data/gazetteer2010.html

You may access the file from hdfs at `/cpre419/gaz_tracts_national.txt` The file has the following columns:

Column	Label	Description
Column 1	USPS	United States Postal Service State Abbreviation
Column 2	GEOID	Geographic Identifier - fully concatenated geographic code (State FIPS, County FIPS, census tract number)
Column 3	POP10	2010 Census population count.
Column 4	HU10	2010 Census housing unit count.
Column 5	ALAND	Land Area (square meters) - Created for statistical purposes only.
Column 6	AWATER	Water Area (square meters) - Created for statistical purposes only.
Column 7	ALAND_SQMI	Land Area (square miles) - Created for statistical purposes only.
Column 8	AWATER_SQMI	Water Area (square miles) - Created for statistical purposes only.
Column 9	INTPTLAT	Latitude (decimal degrees) First character is blank or "-" denoting North or South latitude respectively.

Column 10	INTPTLONG	Longitude (decimal degrees) First character is blank or "-" denoting East or West longitude respectively.

USPS	GEOID	POP10	HU10	ALAND	AWATER	ALAND_SQMI	AWATER_SQMI	INTPTLAT	INTPTLONG
AL	01001020100	1912	752	9809944	36312	3.788	0.014	32.4771112	-86.4903033
AL	01001020200	2170	822	3340505	5846	1.290	0.002	32.4757580	-86.4724678
AL	01001020300	3373	1326	5349274	9054	2.065	0.003	32.4740243	-86.4597033
AL	01001020400	4386	1823	6382705	16244	2.464	0.006	32.4710782	-86.4446805
AL	01001020500	10766	4308	11397725	48412	4.401	0.019	32.4589157	-86.4218165
AL	01001020600	3668	1452	8020366	60048	3.097	0.023	32.4473470	-86.4768023
AL	01001020700	2891	1301	22408265	781555	8.652	0.302	32.4305220	-86.4369107
AL	01001020801	3081	1169	124282110	8113962	47.986	3.133	32.4117228	-86.5316830
AL	01001020802	10435	4003	190810921	678023	73.673	0.262	32.5471327	-86.5315960
AL	01001020900	5675	2320	292756814	516158	113.034	0.199	32.6370123	-86.5149469
AL	01001021000	2894	1261	386855000	1587421	149.366	0.613	32.6049320	-86.7487062

We are interested in the field "ALAND – Land Area (square meters)"

Write a Pig script and run it in **local mode** on this data to find out the top 10 states according to the land area. Since you want to execute the script in local mode, do not save your answer in HDFS but in local file system (your cluster home directory). Include the source code and the output file in the submission. The land area of the top 10 states are in range of 15*10^11 and 25*10^10.

Experiment 2 (20 points)

Next we will use Pig in MapReduce mode.

For this experiment we have a network trace file from a network monitor.

It is in HDFS: `/cpre419/network_trace` along with a smaller file you can use for testing `/cpre419/test_data`

The file is a real life network trace that was dumped from a network monitor. These files can be queried for information about network traffic.

An example entry in the file is:

`10:20:00.000020 IP 244.131.189.196.22379 > 245.184.172.199.80: tcp 0`

The data format is as follows:

<Time> "IP" <Source IP> ">" <Destination IP> <protocol> <protocol dependent data>

The protocol dependent data will be different for TCP, UDP etc. and can be ignored for this lab.

Usually, IP addresses are of the format A.B.C.D However, this data presents IP addresses in the format: A.B.C.D.E. Thus, you will have to process the IP to get rid of the extra information including and after fourth "."

In network monitoring, it is useful to know the identity of those IP sources that connect to a large number of distinct IP destinations. Such sources are often malicious nodes, or compromised by malicious software, and maybe sending spam.

Write a pig script to find the top 10 source IP addresses ranked according to the number of unique destination IP addresses that they connect to using the TCP protocol. Please note that we are not interested in communication that does not use the TCP protocol. The number of unique destination IP addresses of the top 10 source IP addresses are in a range of 3000 and 13000. The script should be able to automatically save the output in the location:

`/lab6/exp2/output/`

Experiment 3 (30 points)

Suppose that there was a firewall blocked IP addresses that it believed potentially unsafe. The list of all IP connections that were blocked is stored in memory, and also in a log file, but the firewall log was lost, due to a failure. We want to regenerate this log file from the other data sources that we have. It is important to regenerate this information, as the IP addresses that are blocked regularly are added to a black list.

The lost firewall log contained details of all blocked connections and looks as follows:

<Time> <Connection ID> <Source IP> <Destination IP> "Blocked"

Your task is to regenerate the log file in the above format by combining information from others logs that are available. In particular, we have the following files:

`/cpre419/ip_trace` – An IP trace file having information about connections received from different source IP addresses, along with a connection ID and time.

`/cpre419/raw_block` - A file containing the connection IDs that were blocked

The IP trace file has the following format:

`0:0:0:9 8 215.160.81.159 > 174.83.200.101 UDP 943`

The format is similar (but not exactly the same) to the previous experiment

<Time> <Connection ID> <Source IP> ">" <Destination IP> <protocol> <protocol dependent data>

The firewall file has lines in the following format: <Connection ID> <Action Taken>

For instance, it could look as follows:

0 Allowed

1 Blocked

2 Allowed

3 Blocked

Your task is as follows.

A. Write a Pig script to regenerate the firewall log file. The output file should be at the location `/lab6/exp3/firewall`

B. Use the previous script to generate the list of all unique source IP addresses that were blocked and the number of times that they were blocked. This list should be sorted (by the script) by the number

of times each IP was blocked in descending order. The range of the frequencies for blocked IP addresses is 6000 and below. The result should be in the location `/lab6/exp3/output`

Finally, your solution should have one script that does both tasks A and B.

For this experiment, it is useful to use the "JOIN" operation. Joining data is a key feature of PIG, and works as follows. Consider the two data sets

Data A: u, x, y, z

Data B: a, b, c, u

These two data sets can be joined using the variable "u" to form: u, x, y, z, a, b, c if A::u == B::u