Project 2.2

**USA Consumer Forum Data Analysis**

Input:

* A .cvs file which contains the customer complaint resolution details.
* The data is comma delimited
* Attributes are:

Date received Product Sub-product Issue Sub-issue Consumer complaint narrative Company public response Company State ZIP code Submitted via Date sent to company Company response to consumer Timely response? Consumer disputed? Complaint ID

* The .cvs file is downloaded from
  + <https://drive.google.com/file/d/0B1QaXx7tpw3SQTlnQ0MzVW5HajA/view?usp=sharing>

Requirement

* Hadoop system installed (using the Oracle virtual Hadoop installation)
* The .cvs file is downloaded

Problem resolution

The first step involves getting the dataset into HDFS. For this flume is used.

**Moving Dataset from Local to HDFS using Flume**

**----------------------------------------------**

UsaCC.conf

-----------

#Specify source, channel and sink

usagent.sinks = usasink

usagent.sources = ussource

usagent.channels = fileChannel

#Flume Configuration Starts

# Define a file channel called fileChannel on usagent

usagent.channels.fileChannel.type = file

# on linux FS

usagent.channels.fileChannel.capacity = 200000

usagent.channels.fileChannel.transactionCapacity = 1000

# Define a source for usagent

usagent.sources.ussource.type = spooldir

# on linux FS

#Spooldir in my case is /home/acadgild/usaconcom

usagent.sources.ussource.spoolDir = /home/acadgild/usaconcom

usagent.sources.ussource.fileHeader = false

usagent.sources.ussource.fileSuffix = .COMPLETED

usagent.sinks.usasink.type = hdfs

#Sink is /usaimport under hdfs

usagent.sinks.usasink.hdfs.path = hdfs://localhost.localdomain:9000/usaimport

usagent.sinks.usasink.hdfs.batchSize = 1000

usagent.sinks.usasink.hdfs.rollSize = 268435456

usagent.sinks.usasink.hdfs.rollInterval = 0

usagent.sinks.usasink.hdfs.rollCount = 5000

usagent.sinks.usasink.hdfs.writeFormat=Text

usagent.sinks.usasink.hdfs.fileType = DataStream

usagent.sources.ussource.channels = fileChannel

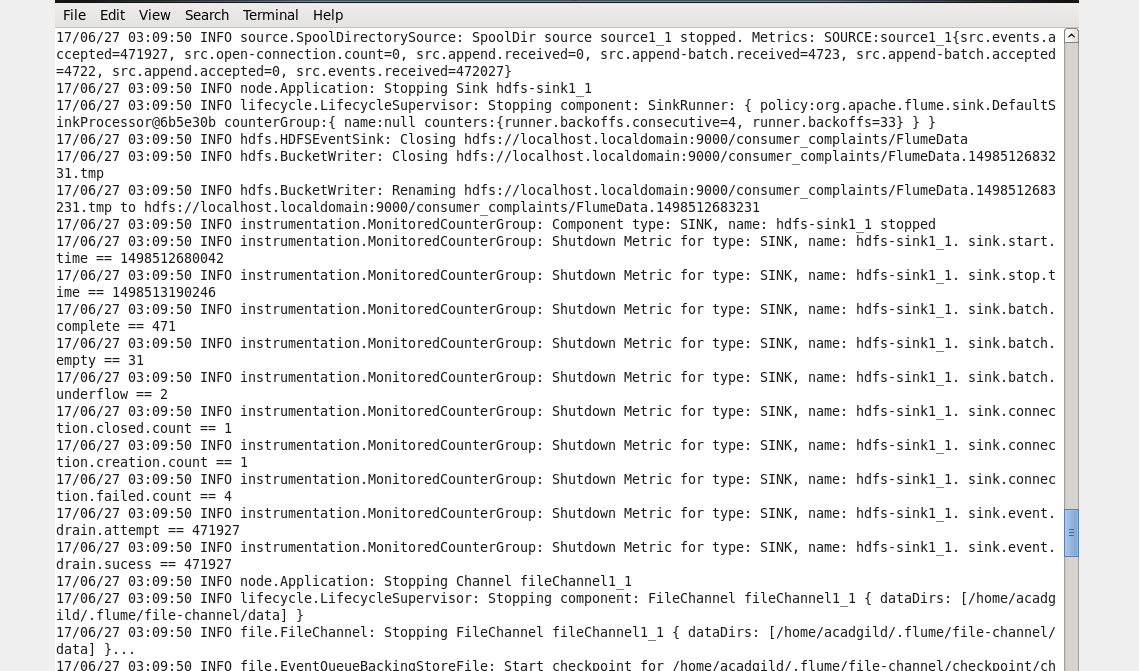
usagent.sinks.usasink.channel = fileChannel

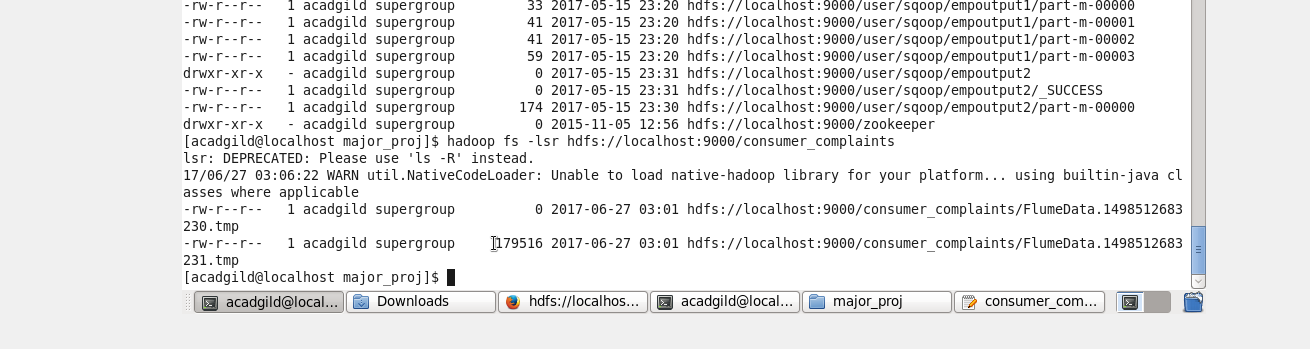
**Execute the Flume using the command:**

-----------------------------------

flume-ng agent -n usagent -f /home/acadgild/UsaCC.conf

Once the flume has copied the file from local to HDFS, the Dataset will also be found at /usaimport





**Problem statement 1:**

1. Write a pig script to find no of complaints which got timely response

File Name: USAPS1.pig

-----------------------

REGISTER '/usr/local/pig/lib/piggybank.jar';

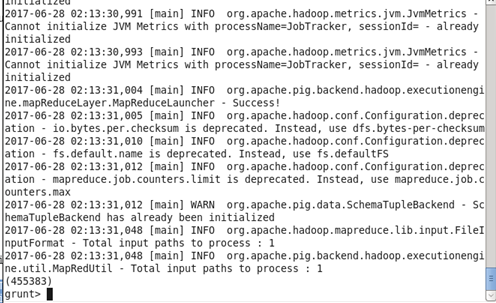
Data = LOAD '/home/acadgild/usaconcom/Consumer\_Complaints.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',' , 'NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER') as (date\_recd: chararray, product: chararray, sub\_prod: chararray, issue: chararray, sub\_issue: chararray, complaint: chararray, public\_response: chararray,company:chararray, state:chararray, zipcode:chararray, submitted\_channel:chararray, date\_submitted:chararray, company\_response:chararray, timely\_response:chararray, consumer\_disputed:chararray, complaint\_id:chararray);

USADet = FILTER Data BY timely\_response == 'Yes';

GroupData = GROUP USADet ALL;

ResponseCnt = FOREACH GroupData GENERATE COUNT(USADet.timely\_response);

STORE ResponseCnt INTO '/home/acadgild/UsaResponse' USING PigStorage(',');



Files Present after executing the script using:

-----------------------------------------------

[acadgild@localhost ~]$ cat /home/acadgild/UsaResponse/p\*

455383

Copy the output to HDFS

------------------------

[acadgild@localhost ~]$ hadoop fs -put UsaResponse /user/acadgild

Open another terminal:Terminal 2 and runl all the daemons using

1) start-all.sh

2) Make sure the daemons are up using

[acadgild@localhost ~]$ jps

16884 Jps

3382 SecondaryNameNode

3560 ResourceManager

3118 NameNode

3663 NodeManager

3215 DataNode

3) Run MySQL using the following commands

3.1) sudo service mysqld start

3.2) mysql -u root

4) Create a new Database:

mysql> create database Usa;

Query OK, 1 row affected (0.00 sec)

mysql>use Usa;

Database changed

Create a table similar to the structure of the output file contents:

mysql>create table Response (Timely\_Response\_Cnt int);

Query OK, 0 rows affected (0.01 sec)

mysql>show tables;

+-----------------+

| Tables\_in\_Usa |

+-----------------+

| Complaints |

| Debt2015 |

| Response |

| SameDateProcess |

+-----------------+

4 rows in set (0.00 sec)

In Terminal1, run the sqoop command to Export the Pig Script analysis from HDFS to MySQL:

----------------------------------------------------------------------------------------

sqoop export --connect jdbc:mysql://localhost/Usa --username 'root' -P --table Response --export-dir '/user/acadgild/UsaResponse/part-r-00000' --input-fields-terminated-by ',' -m 1

In Terminal2, check for if the Tables have been populated with data:

mysql> select \* from Response;

+---------------------+

| Timely\_Response\_Cnt |

+---------------------+

| 455383 |

+---------------------+

1 row in set (0.00 sec)

**Problem statement 2:**

Write a pig script to find no of complaints where consumer forum forwarded the complaint same day they received to respective company

File Name: USAPS2.pig

-----------------------

REGISTER '/usr/local/pig/lib/piggybank.jar';

Data = LOAD '/home/acadgild/usaconcom/Consumer\_Complaints.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',' , 'NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER') as (date\_recd: chararray, product: chararray, sub\_prod: chararray, issue: chararray, sub\_issue: chararray, complaint: chararray, public\_response: chararray,company:chararray, state:chararray, zipcode:chararray, submitted\_channel:chararray, date\_submitted:chararray, company\_response:chararray, timely\_response:chararray, consumer\_disputed:chararray, complaint\_id:chararray);

USADet = FILTER Data BY date\_recd == date\_submitted;

GroupedData = GROUP USADet ALL;

SameDayProcessCnt = FOREACH GroupedData GENERATE COUNT(USADet.date\_recd);

STORE SameDayProcessCnt INTO '/home/acadgild/UsaProcessCnt' USING PigStorage(',');

Execute the Pig Script using the command

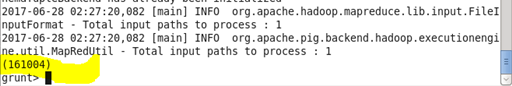
[acadgild@localhost ~]$ pig -x local USAPS2.pig

Files Present after executing the script using :

-----------------------------------------------

[acadgild@localhost ~]$ cat /home/acadgild/UsaProcessCnt/p\*

161004



Copy the output to HDFS

------------------------

[acadgild@localhost ~]$ hadoop fs -put UsaProcessCnt /user/acadgild

Open another terminal:Terminal 2 and runl all the daemons using

1) start-all.sh

2) Make sure the daemons are up using

[acadgild@localhost ~]$ jps

16884 Jps

3382 SecondaryNameNode

3560 ResourceManager

3118 NameNode

3663 NodeManager

3215 DataNode

3) Run MySQL using the following commands

3.1) sudo service mysqld start

3.2) mysql -u root

4) Create a new Database:

mysql> create database Usa;

Query OK, 1 row affected (0.00 sec)

mysql>use Usa;

Database changed

Create a table similar to the structure of the output file contents:

mysql>create table SameDateProcess ( SameDateProcess\_Cnt int);

Query OK, 0 rows affected (0.01 sec)

mysql>show tables;

+-----------------+

| Tables\_in\_Usa |

+-----------------+

| Complaints |

| Debt2015 |

| Response |

| SameDateProcess |

+-----------------+

4 rows in set (0.00 sec)

In Terminal1, run the sqoop command to Export the Pig Script analysis from HDFS to MySQL:

----------------------------------------------------------------------------------------

sqoop export --connect jdbc:mysql://localhost/Usa --username 'root' -P --table SameDateProcess --export-dir '/user/acadgild/UsaProcessCnt/part-r-00000' --input-fields-terminated-by ',' -m 1

In Terminal2, check for if the Tables have been populated with data:

mysql> select \* from SameDateProcess;

+---------------------+

| SameDateProcess\_Cnt |

+---------------------+

| 161004 |

+---------------------+

1 row in set (0.00 sec)

**Problem statement 3:**

Write a pig script to find list of companies toping in complaint chart (companies with maximum number of complaints)

File Name: USAPS3.pig

-----------------------

REGISTER '/usr/local/pig/lib/piggybank.jar';

Data = LOAD '/home/acadgild/usaconcom/Consumer\_Complaints.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',' , 'NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER') as (date\_recd: chararray, product: chararray, sub\_prod: chararray, issue: chararray, sub\_issue: chararray, complaint: chararray, public\_response: chararray,company:chararray, state:chararray, zipcode:chararray, submitted\_channel:chararray, date\_submitted:chararray, company\_response:chararray, timely\_response:chararray, consumer\_disputed:chararray, complaint\_id:chararray);

USADet = GROUP Data BY company;

GroupedData = FOREACH USADet GENERATE FLATTEN(group) AS company, COUNT(Data.company) AS cnt;

OrderData = ORDER GroupedData by cnt DESC;

Max10 = LIMIT OrderData 10;

STORE Max10 INTO '/home/acadgild/UsaMaxComplaint' USING PigStorage(',');

Execute the Pig Script using the command

-----------------------------------------------

[acadgild@localhost ~]$ pig -x local USAPS3.pig

Files Present after executing the script using :

-----------------------------------------------

[acadgild@localhost ~]$ cat /home/acadgild/UsaMaxComplaint/p\*

Bank of America,51127

Wells Fargo,37182

JPMorgan Chase,29583

Experian,24720

Equifax,24706

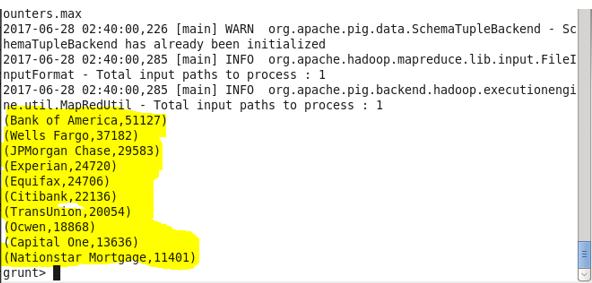
Citibank,22136

TransUnion,20054

Ocwen,18868

Capital One,13636

Nationstar Mortgage,11401



Copy the output to HDFS

------------------------

[acadgild@localhost ~]$ hadoop fs -put UsaMaxComplaint /user/acadgild

Open another terminal:Terminal 2 and runl all the daemons using

1) start-all.sh

2) Make sure the daemons are up using

[acadgild@localhost ~]$ jps

16884 Jps

3382 SecondaryNameNode

3560 ResourceManager

3118 NameNode

3663 NodeManager

3215 DataNode

3) Run MySQL using the following commands

3.1) sudo service mysqld start

3.2) mysql -u root

4) Create a new Database:

mysql> create database Usa;

Query OK, 1 row affected (0.00 sec)

mysql>use Usa;

Database changed

Create a table similar to the structure of the output file contents:

mysql>create table Complaints (Company\_Name varchar(20),Complaint\_Cnt int);

Query OK, 0 rows affected (0.01 sec)

mysql>show tables;

+-----------------+

| Tables\_in\_Usa |

+-----------------+

| Complaints |

| Debt2015 |

| Response |

| SameDateProcess |

+-----------------+

4 rows in set (0.00 sec)

In Terminal1, run the sqoop command to Export the Pig Script analysis from HDFS to MySQL:

----------------------------------------------------------------------------------------

sqoop export --connect jdbc:mysql://localhost/Usa --username 'root' -P --table Complaints --export-dir '/user/acadgild/UsaMaxComplaint/part-r-00000' --input-fields-terminated-by ',' -m 1

In Terminal2, check for if the Tables have been populated with data:

mysql> select \* from Complaints;

+---------------------+---------------+

| Company\_Name | Complaint\_Cnt |

+---------------------+---------------+

| Bank of America | 51127 |

| Wells Fargo | 37182 |

| JPMorgan Chase | 29583 |

| Experian | 24720 |

| Equifax | 24706 |

| Citibank | 22136 |

| TransUnion | 20054 |

| Ocwen | 18868 |

| Capital One | 13636 |

| Nationstar Mortgage | 11401 |

+---------------------+---------------+

10 rows in set (0.01 sec)

**Problem statement 4:**

Write a pig script to find no of complaints filed with product type has &quot;Debt collection&quot; for the year 2015

File Name: USAPS4.pig

-----------------------

REGISTER '/usr/local/pig/lib/piggybank.jar';

Data = LOAD '/home/acadgild/usaconcom/Consumer\_Complaints.csv' using org.apache.pig.piggybank.storage.CSVExcelStorage(',' , 'NO\_MULTILINE','UNIX','SKIP\_INPUT\_HEADER') as (date\_recd: chararray, product: chararray, sub\_prod: chararray, issue: chararray, sub\_issue: chararray, complaint: chararray, public\_response: chararray,company:chararray, state:chararray, zipcode:chararray, submitted\_channel:chararray, date\_submitted:chararray, company\_response:chararray, timely\_response:chararray, consumer\_disputed:chararray, complaint\_id:chararray);

DebtFilterData = FILTER Data BY product == 'Debt collection';

YrFilterData = FILTER DebtFilterData BY SUBSTRING(date\_recd,6,10) == '2015';

GroupData = GROUP YrFilterData ALL;

FinalCnt = FOREACH GroupData GENERATE COUNT(YrFilterData);

STORE FinalCnt INTO '/home/acadgild/UsaDebt2015' USING PigStorage(',');

Execute the script

------------------

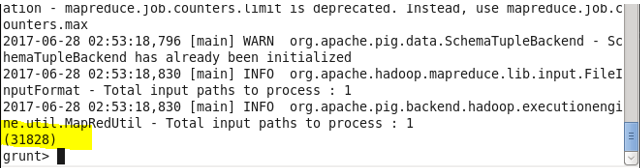
[acadgild@localhost ~]$pig -x local USAPS4.pig

Output after executing the script using :

-----------------------------------------

[acadgild@localhost ~]$ cat /home/acadgild/UsaDebt2015/p\*

31828



Copy the output to HDFS

------------------------

[acadgild@localhost ~]$ hadoop fs -put UsaDebt2015 /user/acadgild

Open another terminal:Terminal 2 and runl all the daemons using

1) start-all.sh

2) Make sure the daemons are up using

[acadgild@localhost ~]$ jps

16884 Jps

3382 SecondaryNameNode

3560 ResourceManager

3118 NameNode

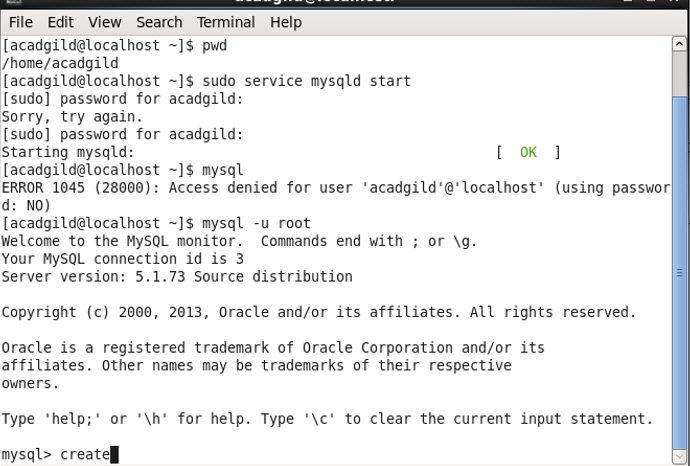
3663 NodeManager

3215 DataNode

3) Run MySQL using the following commands

3.1) sudo service mysqld start

3.2) mysql -u root



4) Create a new Database:

mysql> create database Usa;

Query OK, 1 row affected (0.00 sec)

mysql>use Usa;

Database changed

Create a table similar to the structure of the output file contents:

mysql>create table Debt2015 (Debt\_Cnt int);

Query OK, 0 rows affected (0.01 sec)

mysql>show tables;

+-----------------+

| Tables\_in\_Usa |

+-----------------+

| Complaints |

| Debt2015 |

| Response |

| SameDateProcess |

+-----------------+

4 rows in set (0.00 sec)

In Terminal1, run the sqoop command to Export the Pig Script analysis from HDFS to MySQL:

----------------------------------------------------------------------------------------

sqoop export --connect jdbc:mysql://localhost/Usa --username 'root' -P --table Debt2015 --export-dir '/user/acadgild/UsaDebt2015/part-r-00000' --input-fields-terminated-by ',' -m 1

In Terminal2, check for if the Tables have been populated with data:

mysql> select \* from Debt2015;

+----------+

| Debt\_Cnt |

+----------+

| 31828 |

+----------+

1 row in set (0.00 sec)

