INTERNSHIP REPORT

Submitted by:

UTKARSH KUMAR SINGH, MASTERS, DATA SCIENCE
Indiana University, Bloomington

Supervisor: Mr. Brian Fioravanti
Reviewer: Prof. Dr. David Wild

Start Date for Internship: 30th May, 2017
End Date for Internship: 18th August, 2017

Report Date: 21st August, 2017

Preface

This report documents the work done during the summer internship at Westfield Gas and Electric under the supervision of Mr. Brian Fioravanti. The report first shall give an overview of the tasks completed during the period of internship with technical details. Then the results obtained shall be discussed and analyzed.

Report shall also elaborate on the the future works which can be persuaded as an advancement of the current work.

I have tried my best to keep report compact and simple yet technically correct. I hope I succeed in my attempt.

UTKARSH KUMAR SINGH

Acknowledgments

Simply put, I could not have done this work without the lots of help I received cheerfully from whole Westfield Gas and Electric(WG&E). The work culture in WG&E really motivates.

Everybody is such a friendly and cheerful companion here that work stress is never comes in way.
I would specially like to thank Mr. Michael Mastroianni for proving the nice ideas to work upon. Not only did they advised about my project
but listening to their discussions in IPeT meeting have evoked a good interest in Smart meter data analysis. I am also highly indebted to my supervisor Mr. Brian Fioravanti, who seemed to have solutions to all my problems.

Abstract

The report presents the tasks completed during summer internship at WG&E which are listed below:

- Installed the latest stable version Apache Hadoop-2.8.0 on the Oracle VM Virtual box (OS-UBUNTU) on the remote machine and its other dependency Softwares.

- Ran test cases to check whether all important components of Hadoop (such as namenode, datanode, secondary namenode, resource manager, job tracker) are working fine or not so as to be used for future data analysis purposes.

- Installed the latest version of Apache Flume 1.7.0 and completed batch ingestion of smart meter data for the month of June into HDFS(Hadoop Distributed File System) using a flume agent.

- Developed a procedure to do real-time smart meter data ingestion from the application servers (source) to the HDFS (sink).

INTRODUCTION

Westfield Gas + Electric provides natural gas and electrical service to residents and businesses across the city of Westfield, Mass. They have recently installed new smart meters that records consumption of electric energy in intervals of 5 minutes and communicates that information at least daily back to the utility for monitoring and billing. Smart metering offers potential benefits to householders. These include, a) an end to estimated bills, which are a major source of complaints for many customers b) a tool to help consumers better manage their energy purchases - stating that smart meters with a display outside their homes could provide up-to-date information on gas and electricity consumption and in doing so help people to manage their energy use and reduce their energy bills. Electricity pricing usually peaks at certain predictable times of the day and the season. In particular, if generation is constrained, prices can rise if power from other jurisdictions or more costly generation is brought online. Proponents assert that billing customers at a higher rate for peak times will encourage consumers to adjust their consumption habits to be more responsive to market prices and assert further, that regulatory and market design agencies hope these "price signals" could delay the construction of additional generation or at least the purchase of energy from higher priced sources, thereby controlling the steady and rapid increase of electricity prices. In order to handle such huge volume of data and to get useful insights we used Apache Hadoop and Flume.

APACHE HADOOP INSTALLATION STEPS

STEP I

Download Oracle VM Virtual Box.

The link for downloading the virtual box is provided below :

https://drive.google.com/drive/u/0/folders/0B4er4Ai3anHjUnBXOE1FZHVCYUk

For windows user : VirtualBox-5.1.2-108956-Windows.exe

For MAC users : VirtualBox-5.1.2-108956-MAC OSX.dmg

STEP II

Install Virtual Box.

Right click on “VirtualBox-5.1.2-108956-Windows.exe” and click on Run as Administrator.

Click on Next until you find finish button and then click on Finish.

Now Oracle VM Virtual box is installed on your machine.

STEP III

Now Download Ubuntu version 16.04.2 ISO file using the following link:

https://www.ubuntu.com/download/desktop/contribute?version=16.04.2&architecture=amd6
4

Click on 'Not now, take me to Download' link.

STEP IV

Configure VM in Oracle VM Virtual box.

Open Virtual Machine.

Click on New button and provide the name of the VM, select Type of OS as Linux and version as Ubuntu (64-bit). (Make sure the virtual technology is enabled on the machine otherwise enable it via the BIOS settings).

Now provide RAM (Memory Size) as per the requirement (here 4 GB).

Next select 'Create a Virtual disk now' option.

Now Select 'Virtual Disk Image' as Hard Disk file type and select 'Dynamically allocated' option.

Next provide Hard disk space for the Ubuntu machine (Here 40GB).

Now click 'Create'. This will configure VM in Oracle VM Virtual Box.

STEP V

Install Ubuntu on the configured VM.

Click on the Configured VM in Oracle VM Virtual Box and then click on the start button in the menu to start the machine.

Now select the downloaded Ubuntu ISO file. This will start loading the ISO file.

Now select 'Install Ubuntu' option.

Select 'Erase disk and install ubuntu' option. (Ignore the Warning)

Click on 'Install Now' option. The installation will begin and finish in about 15-20 minutes.

Proceed further with the default options.

After a few minutes Ubuntu will be installed on the VM.

STEP VI

Install Java on Ubuntu.

Open the terminal and update the source list using the command:

```
sudo apt-get update
```

Install Java using the command:

```
sudo apt-get install openjdk-8-jdk
```

To check the Java version use the command:

```
java -version
```

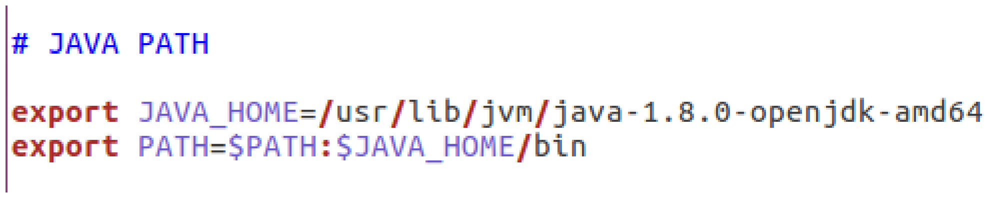
Find out the location of the java version installed using the command:

```
update-java-alternatives -l
```

Next open the .bashrc file from the home directory (important) using the command:

gedit .bashrc

Set the java path in the .bashrc file



```
# JAVA PATH

export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
export PATH=$PATH:$JAVA_HOME/bin
```

Save and close the .bashrc file and load the latest environment variables using the command:

source .bashrc

STEP VII

Install Apache Hadoop 2.8.0.

Get the access for /usr/local/ folder using the command:

sudo chown -R <username>:<groupname> /usr/local/

sudo chmod 777 /usr/local/

(To find out username and group name (mostly both are same) use 'll' command in terminal.)

Create a new directory named 'hadoop-env' using the command:

mkdir /usr/local/hadoop-env/

Download Apache Hadoop-2.8.0 using the following link:

https://archive.apache.org/dist/hadoop/core/hadoop-2.8.0/hadoop-2.8.0.tar.gz

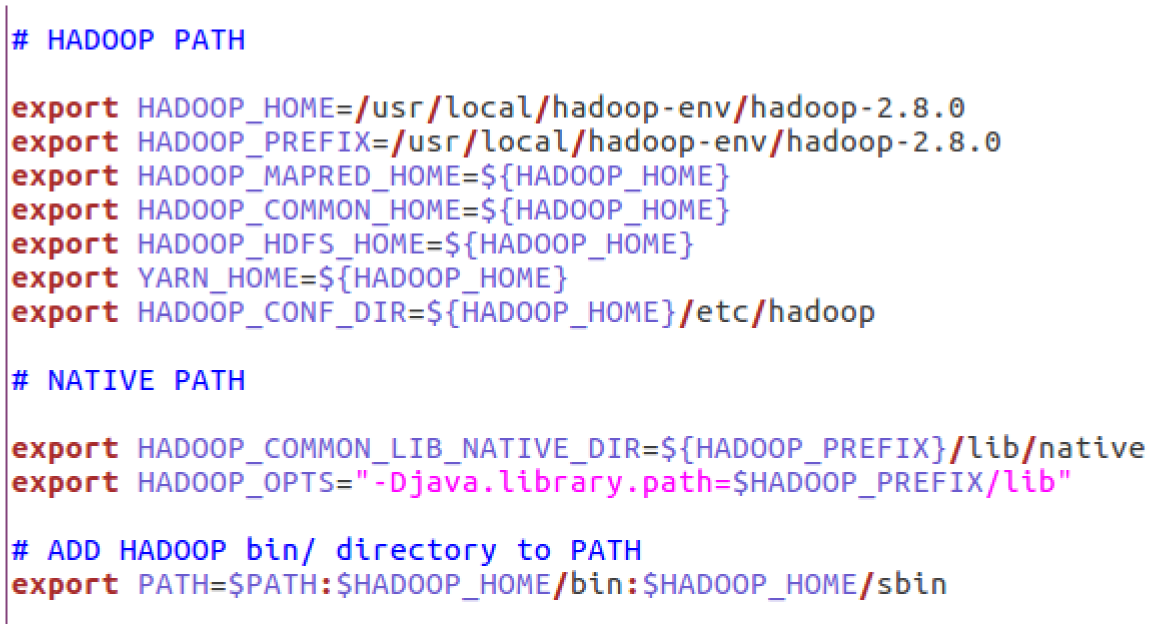
Now change the directory in which tar file has been downloaded and untar it using the command:

tar -xvf hadoop-2.8.0.tar.gz -C /usr/local/hadoop-env/

Next open the .bashrc file from the home directory (important) using the command:

gedit .bashrc

Set the environment variables in the .bashrc file :

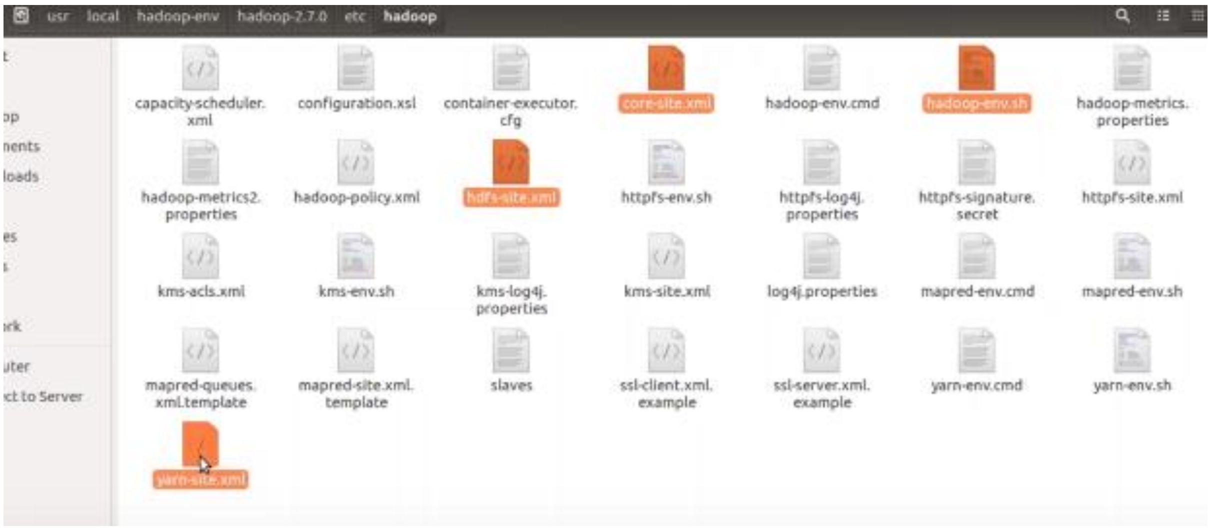


```
# HADOOP PATH

export HADOOP_HOME=/usr/local/hadoop-env/hadoop-2.8.0
export HADOOP_PREFIX=/usr/local/hadoop-env/hadoop-2.8.0
export HADOOP_MAPRED_HOME=${HADOOP_HOME}
export HADOOP_COMMON_HOME=${HADOOP_HOME}
export HADOOP_HDFS_HOME=${HADOOP_HOME}
export YARN_HOME=${HADOOP_HOME}
export HADOOP_CONF_DIR=${HADOOP_HOME}/etc/hadoop

# NATIVE PATH

export HADOOP_COMMON_LIB_NATIVE_DIR=${HADOOP_PREFIX}/lib/native
export HADOOP_OPTS="-Djava.library.path=$HADOOP_PREFIX/lib"

# ADD HADOOP bin/ directory to PATH
export PATH=$PATH:$HADOOP_HOME/bin:$HADOOP_HOME/sbin
```

Now go to /usr/local/hadoop-env/hadoop-2.8.0/etc/hadoop folder and create a copy of 'mapred-site.xml.template' file and name it as 'mapred-site.xml' and open it via gedit editor and make the following modifications:

```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>mapreduce.framework.name</name>
<value>yarn</value>
</property>
</configuration>
```

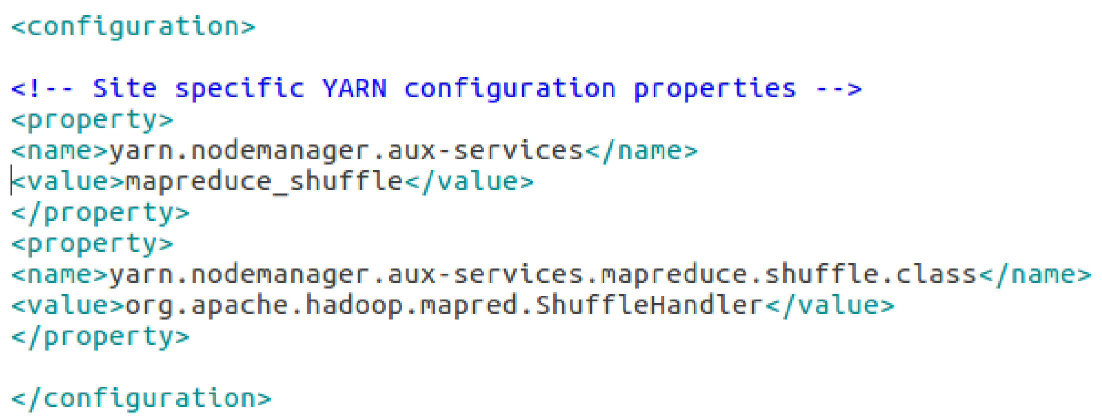
Now open the following files in gedit highlighted below:



Make the following modification in the hadoop-env.sh file:

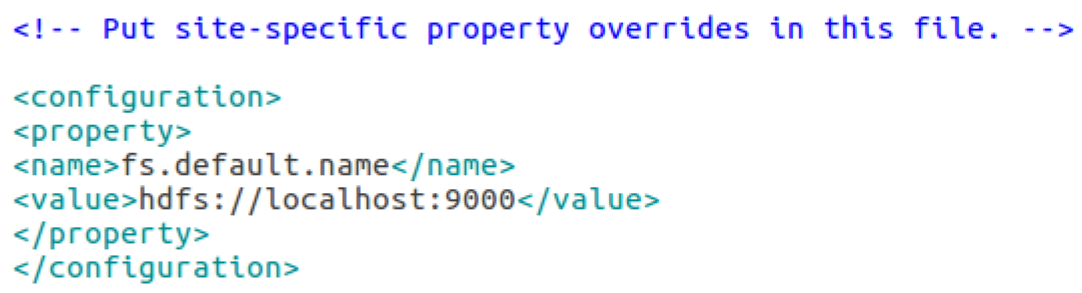
```
# The java implementation to use.
export JAVA_HOME=/usr/lib/jvm/java-1.8.0-openjdk-amd64
```

Make the following modification in the yarn-site.xml file:



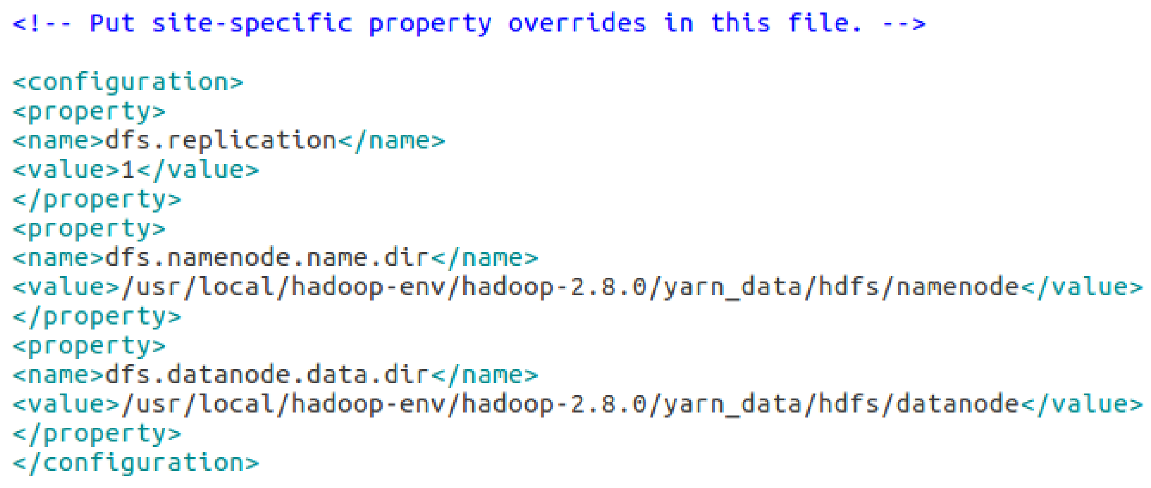
```
<configuration>

<!-- Site specific YARN configuration properties -->
<property>
<name>yarn.nodemanager.aux-services</name>
<value>mapreduce_shuffle</value>
</property>
<property>
<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>
<value>org.apache.hadoop.mapred.ShuffleHandler</value>
</property>

</configuration>
```

Make the following modification in the core-site.xml file:



```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>fs.default.name</name>
<value>hdfs://localhost:9000</value>
</property>
</configuration>
```

Make the following modification in the core-site.xml file:



```
<!-- Put site-specific property overrides in this file. -->

<configuration>
<property>
<name>dfs.replication</name>
<value>1</value>
</property>
<property>
<name>dfs.namenode.name.dir</name>
<value>/usr/local/hadoop-env/hadoop-2.8.0/yarn_data/hdfs/namenode</value>
</property>
<property>
<name>dfs.datanode.data.dir</name>
<value>/usr/local/hadoop-env/hadoop-2.8.0/yarn_data/hdfs/datanode</value>
</property>
</configuration>
```

Save and close the .bashrc file and load the latest environment variables using the command:

```
source .bashrc
```

Hadoop-2.8.0 is installed now and to verify the installation use the following command:

```
hadoop version
```

STEP VIII

Now install SSH-Server and Client using the command:

```
sudo apt-get update
sudo apt-get install openssh-server openssh-client
```

Next type in the following command to start all the services:

```
start-all.sh
```

To check the actives services use the command:

```
jps
```

or go to the browser and type 'localhost:50070' and the interface will display all the details regarding all the active services.

To stop the services use the command:

```
stop-all.sh
```

APACHE FLUME INSTALLATION STEPS

STEP I

Download and install Apache Flume-1.7.0

Download the Flume -1.7.0 using the link:

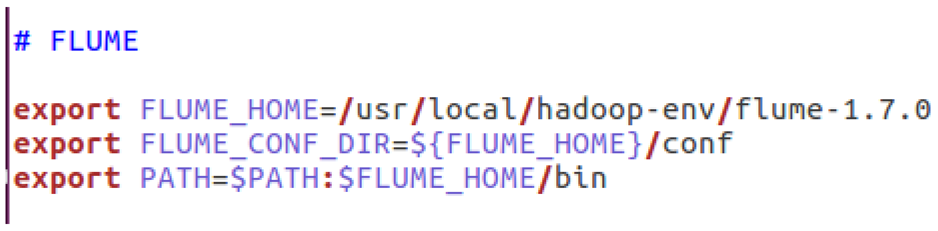
http://apache.spinellicreations.com/flume/1.7.0/apache-flume-1.7.0-bin.tar.gz

Now untar the file in the 'hadoop-env' folder and rename it as 'flume-1.7.0'.

Next open the .bashrc file from the home directory (important) using the command:

gedit .bashrc

Set the environment variables in the .bashrc file :



```
# FLUME

export FLUME_HOME=/usr/local/hadoop-env/flume-1.7.0
export FLUME_CONF_DIR=${FLUME_HOME}/conf
export PATH=$PATH:$FLUME_HOME/bin
```

Save and close the .bashrc file and load the latest environment variables using the command:

source .bashrc

The real time data ingestion is done by modifying the 'flume-conf.properties.template' file (located at /usr/local/hadoop-env/flume-1.7.0/conf/) which may vary depending upon the type of source, channel and sinks.

Future work & Conclusion

The whole experience of working at WG&E was great. This organization has a superb work culture, great minds and very high quality of work. I learned a lot of about smart meter data ingestion, preprocessing and analysis. The work I could complete here was very satisfactory. I hope my work on Smart meter data helps the utility meet its goals. For future We can do the preprocessing of the ingested data in order to treat the missing and duplicate data and then analyze it to get some useful insights about improving customer satisfaction and retention rates by providing direct access to energy usage insights and reduce direct energy costs through more accurate load forecasts that are driven by powerful insights based on energy consumption patterns.