ITECH2302 Big Data Management Laboratory Hadoop pt.2

Objectives:

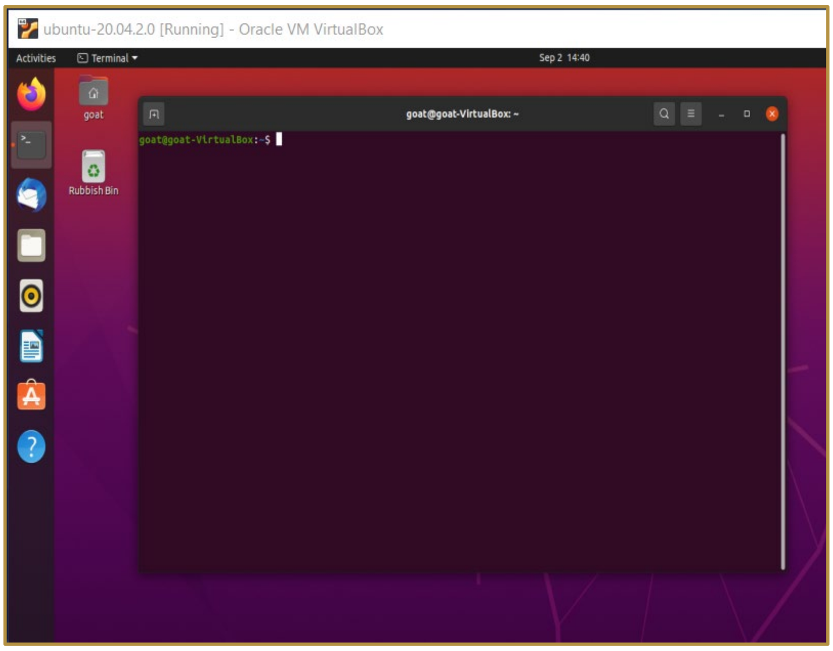
- Hadoop file system
- MapReduce

Activity 1

Apache Hadoop

1. Start Apache Hadoop

Open a terminal with the ubuntu operating system



```
goat@goat-VirtualBox:~$
```

Write the following commands:

```
$ ssh localhost
$ hdfs namenode -format
$ start-dfs.sh
$ start-yarn.sh
```

If the output from:

```
$ jps
```

..doesn't look like the following,

```
5042 DataNode
5299 SecondaryNameNode
4888 NameNode
```

```
5516 ResourceManager
5677 NodeManager
6046 Jps
```

Then maybe the *datanode* didn't start correctly because it was left in a corrupted state. This is easy to fix by using the following commands:

```
$ stop-all.sh
$ rm -rf /home/goat/hadoopdata/hdfs/datanode/*
$ start-all.sh
```

Try `jps` again, you should see the datanode listed now.

Activity 2

Managing the filesystem

You can create a folder in the Hadoop file system (HDFS) like this:

```
$ hadoop fs -mkdir /data
```

You can copy a file from the ubuntu file system into the Hadoop file system like this:

```
$ hadoop fs -put /home/goat/hadoop_spark/hadoop/lab_data/mapreduce_data/NYSE_DATA.txt /data
```

Now it is available to be operated on by Map/Reduce, or Pig, or Yarn etc. Check that the file was copied over correctly:

```
$ hadoop fs -ls /data
```

You can familiarise yourself with the file system commands if you like, here:

- https://hadoop.apache.org/docs/r2.4.1/hadoop-project-dist/hadoop-common/FileSystemShell.html

You can replace `hdfs dfs` with `hadoop fs`

Activity 3

Map/Reduce

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html#MapReduce+-+User+Interfaces

/home/goat/hadoop_spark/hadoop/jar-files/ hadoop-core-1.2.1.jar

/home/goat/hadoop_spark/hadoop/lab_data/mr/ ProcessUnits.java

```
javac -classpath hadoop-core-1.2.1.jar -d units ProcessUnits.java
```

javac -classpath /home/goat/hadoop_spark/hadoop/jar-files/hadoop-core-1.2.1.jar -d units

/home/goat/hadoop_spark/hadoop/lab_data/mr/ProcessUnits.java

```
$ jar -cvf units.jar -C units/ .
$HADOOP_HOME/bin/hadoop fs -mkdir /input_dir
$HADOOP_HOME/bin/
```

```
hadoop jar units.jar hadoop.ProcessUnits /input_dir output_dir
hadoop jar units.jar hadoop.ProcessUnits /input_dir output_dir
```

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

/home/goat/hadoop_spark/Hadoop/lab_data/xxx
/home/goat/hadoop_spark/Hadoop/lab_data/mapreduce_data/NYSE_mapper.py

https://www.tutorialspoint.com/map_reduce/index.htm

Run examples

Activity 4

Google BigQuery

You might want to check out Google BigQuery:

- https://cloud.google.com/bigquery

And its provisions for JSON in its query language:

- https://cloud.google.com/bigquery/docs/reference/standard-sql/json_functions
- https://docs.snowflake.com/en/sql-reference/functions/parse_json.html