Anubha Bhargava
Homework 1 – Big Data Analytics

I Installed Hadoop for Ubuntu. I downloaded the airline data for the year 2008 as well as the US Fish and Wildlife Service data for birds as shown below.

```
bigdata@ubuntu:~/Documents$ ls *.csv
2008.csv  birds.csv
```

DATA SET 1:

Downloaded Pig and ran it in local mode:

```
bigdata@ubuntu:/usr/local$ export PATH=/usr/local/pig-0.13.0/bin/:$PATH
bigdata@ubuntu:/usr/local$ pig -x local
15/10/01 09:42:19 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
15/10/01 09:42:19 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
15/10/01 09:42:20 WARN pig.Main: Cannot write to log file: /usr/local/pig_144371
7740025.log
2015-10-01 09:42:20,026 [main] INFO  org.apache.pig.Main - Apache Pig version 0.
13.0 (r1606446) compiled Jun 29 2014, 02:27:58
2015-10-01 09:42:20,079 [main] INFO  org.apache.pig.impl.util.Utils - Default bo
otup file /home/bigdata/.pigbootup not found
2015-10-01 09:42:20,385 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2015-10-01 09:42:20,392 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.addr
ess
2015-10-01 09:42:20,393 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to hadoop file system at: file:///
2015-10-01 09:42:21,054 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2015-10-01 09:42:21,056 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt>
```

Executed the commands to load and dump the data using PigStorage:

```
grunt> airplane = LOAD '/home/bigdata/Documents/2008.csv' USING PigStorage(',')
as (Year, Month, DayofMonth, DepTime, ArrTime);
2015-10-01 09:57:58,928 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2015-10-01 09:57:58,933 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
grunt> DUMP airplane;
```

Filtered the airplane data by arrival times between 700 to 1000.

```
grunt> airplane_arrtime = FILTER airplane BY (float)ArrTime<1000 and (float)ArrT
ime>700;
2015-10-01 10:06:08,729 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - E
ncountered Warning IMPLICIT_CAST_TO_FLOAT 2 time(s).
```

I typed store airplane_arrtime into '/home/bigdata/Documents/airplane_arrtime'; and received the following output:

```
Input(s):
Successfully read 7009729 records from: "/home/bigdata/Documents/2008.csv"

Output(s):
Successfully stored 1305605 records in: "/home/bigdata/Documents/airplane_arrtim
e"

Counters:
Total records written : 1305605
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1060255042_0001


2015-10-01 10:07:45,515 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 272494 time(s).
2015-10-01 10:07:45,515 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
```

I opened the file containing the output using vim and the following displayed. The columns correspond to year, month, day, departure time and arrival time.

```
2008    1       3       4       754
2008    1       3       4       926
2008    1       3       4       706
2008    1       3       4       715
2008    1       3       4       754
2008    1       3       4       706
2008    1       3       4       801
2008    1       3       4       734
2008    1       3       4       712
2008    1       3       4       831
2008    1       3       4       726
2008    1       3       4       831
2008    1       3       4       948
2008    1       3       4       850
2008    1       3       4       802
2008    1       3       4       821
2008    1       3       4       712
2008    1       3       4       958
2008    1       3       4       933
2008    1       3       4       905
2008    1       3       4       906
2008    1       3       4       816
2008    1       3       4       924
```

Moving forward, I generated the month and day of month for each line of the airplane data and stored it into the directory airplane_month_day.

```
airplane_month_day = foreach airplane generate Month, DayofMonth;
grunt> store airplane_month_day into '/home/bigdata/Documents/airplane_month_day
';
```

```
0       0       0       airplane,airplane_month_day     MAP_ONLY        /home/bi
gdata/Documents/airplane_month_day,

Input(s):
Successfully read 7009729 records from: "/home/bigdata/Documents/2008.csv"

Output(s):
Successfully stored 7009729 records in: "/home/bigdata/Documents/airplane_month_
day"

Counters:
Total records written : 7009729
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1625797435_0001


2015-10-01 10:26:48,413 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

When cd-ing into the output directory, the following displayed:

```
bigdata@ubuntu:~/Documents$ ls
2008.csv                airplane_month_day  pig_1443718497908.log
airplane_arrtime        NasaData            pig_1443719874489.log
airplane_arrtime_sort   nasadata.tar.gz     pig_1443720454248.log
bigdata@ubuntu:~/Documents$ cd airplane_month_day/
bigdata@ubuntu:~/Documents/airplane_month_day$ ls
part-m-00000  part-m-00005  part-m-00010  part-m-00015  part-m-00020
part-m-00001  part-m-00006  part-m-00011  part-m-00016  _SUCCESS
part-m-00002  part-m-00007  part-m-00012  part-m-00017
part-m-00003  part-m-00008  part-m-00013  part-m-00018
part-m-00004  part-m-00009  part-m-00014  part-m-00019
bigdata@ubuntu:~/Documents/airplane_month_day$
```

Below are the results I received in file part-m-00000. It shows the month and day for the airplanes.

Anubha Bhargava
Homework 1 – Big Data Analytics

```
10      31
10      1
10      2
10      3
10      4
10      5
10      6
10      7
10      8
10      9
10      10
10      11
10      12
10      13
10      14
10      15
10      16
10      17
10      18
10      19
10      20
10      21
10      22
"part-m-00017" 336886L, 1920718C
```

Afterwards, I sorted the airplane arrival times in descending order and stored them into the directory airplane_arrtime_sort.

```
grunt> airplane_arrtime_sort = order airplane by ArrTime desc;
grunt> store airplane_arrtime_sort into '/home/bigdata/Documents/airplane_arrtim
e_sort';
```

```
Input(s):
Successfully sampled 2100 records from: "/home/bigdata/Documents/2008.csv"
Successfully read 7009729 records from: "/home/bigdata/Documents/2008.csv"

Output(s):
Successfully stored 7009729 records in: "/home/bigdata/Documents/airplane_arrtim
e_sort"

Counters:
Total records written : 7009729
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local282985035_0001 ->      job_local341533399_0002,
job_local341533399_0002


2015-10-01 10:38:39,073 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

Anubha Bhargava
Homework 1 – Big Data Analytics

These are the results which displayed after sorting by arrival time. The columns correspond to year, month, day of month, departure time and arrival time. The Not Applicable ones displayed at the top, since these results were sorted in descending order.

2008	12	2	2	NA
2008	12	27	6	NA
2008	12	26	5	NA
2008	12	23	2	NA
2008	12	21	7	NA
2008	12	16	2	NA
2008	12	8	1	NA
2008	12	23	2	NA
2008	12	21	7	NA
2008	12	1	1	NA
2008	12	23	2	NA
2008	12	18	4	NA
2008	12	26	5	NA
2008	12	19	5	NA
2008	12	1	1	NA
2008	12	26	5	NA
2008	12	16	2	NA
2008	12	9	2	NA
2008	12	26	5	NA
2008	12	23	2	NA
2008	12	21	7	NA
2008	12	13	6	NA
2008	12	13	6	NA

After collecting and storing this data, I ssh-ed onto local host and ran pig.

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop-2.5.0/logs/hadoop-big
data-namenode-ubuntu.out
localhost: starting datanode, logging to /usr/local/hadoop-2.5.0/logs/hadoop-big
data-datanode-ubuntu.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.5.0/logs/had
oop-bigdata-secondarynamenode-ubuntu.out
```

Then I ran start-dfs using the following command: Cd /usr/local/Hadoop-2.5.0/sbin/, run start-dfs.sh

I uploaded the file onto HDFS as shown below:

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ ./bin/hdfs dfs -mkdir /PigSource
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ ./bin/hdfs dfs -put /home/bigdata/Docume
nts/2008.csv /PigSource
bigdata@ubuntu:/usr/local/hadoop-2.5.0$
```

Then, I ran pig in grunt using HDFS. I loaded and dumped the data as shown below:

```
grunt> airplane = LOAD '/PigSource/2008.csv' USING PigStorage(',') as (Year,Mont
h,DayasMonth,DepTime,ArrTime);
2015-10-01 11:18:45,375 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> DUMP airplane;
```

```
(2008,12,13,6,921)
(2008,12,13,6,1435)
(2008,12,13,6,1750)
(2008,12,13,6,706)
(2008,12,13,6,1552)
(2008,12,13,6,1250)
(2008,12,13,6,1033)
(2008,12,13,6,840)
(2008,12,13,6,810)
(2008,12,13,6,547)
(2008,12,13,6,848)
(2008,12,13,6,936)
(2008,12,13,6,657)
(2008,12,13,6,1007)
(2008,12,13,6,638)
(2008,12,13,6,756)
(2008,12,13,6,612)
(2008,12,13,6,749)
(2008,12,13,6,1002)
(2008,12,13,6,834)
(2008,12,13,6,655)
(2008,12,13,6,1251)
(2008,12,13,6,1110)
```

DATA SET 2:

I performed the same set of steps above with another set of data called birds.csv. I loaded the data as shown:

```
grunt> birds = LOAD '/home/bigdata/Documents/birds.csv' USING PigStorage(',') as
 (Species, Latitude, Longitude, Oiling, BirdCount);
2015-10-05 07:57:02,401 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2015-10-05 07:57:02,405 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
```

Afterwards, I dumped the data to receive the following results:

Anubha Bhargava
Homework 1 – Big Data Analytics

```
("Great Egret",29.4017,-90.79482,"Unknown","Dead")
("Clapper Rail",29.43601,-89.83241,"Unknown","Dead")
("Black Skimmer",29.3097,-89.89216,"Visibly Oiled","Dead")
("Laughing Gull",29.90611,-89.29635,"Unknown","Dead")
("Laughing Gull",29.90611,-89.29635,"Unknown","Dead")
("Clapper Rail",29.4671,-89.91194,"Unknown","Dead")
("Clapper Rail",29.47957,-89.90939,"Unknown","Dead")
("Laughing Gull",29.91961,-89.26281,"Unknown","Dead")
("Clapper Rail",29.44553,-89.87547,"Unknown","Dead")
("Clapper Rail",29.43795,-89.83844,"Not Visibly Oiled","Dead")
("Laughing Gull",29.1688,-89.53029,"Visibly Oiled","Dead")
("Laughing Gull",29.381,-89.72372,"Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Clapper Rail",29.47566,-89.71014,"Not Visibly Oiled","Dead")
("Clapper Rail",29.49309,-89.91873,"Unknown","Dead")
("Black Skimmer",29.26371,-89.96423,"Visibly Oiled","Dead")
("Clapper Rail",29.44239,-89.88561,"Visibly Oiled","Dead")
("Clapper Rail",29.50644,-89.86126,"Unknown","Dead")
grunt>
```

I filtered the data to only have values of latitude between 30.2 and 30 and stored that data into the directory species_lat_30_302.

```
grunt> species_lat_30_302 = FILTER birds BY (float)Latitude<30.2 and (float)Lati
 and (float)Latitude>30;
2015-10-05 08:29:12,896 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - E
ncountered Warning IMPLICIT_CAST_TO_FLOAT 1 time(s).
2015-10-05 08:29:12,896 [main] WARN  org.apache.pig.newplan.BaseOperatorPlan - E
ncountered Warning IMPLICIT_CAST_TO_DOUBLE 1 time(s).
grunt> store species_lat_30_302 into '/home/bigdata/Documents/species_lat_30_302
';
```

```
Input(s):
Successfully read 7230 records from: "/home/bigdata/Documents/birds.csv"

Output(s):
Successfully stored 501 records in: "/home/bigdata/Documents/species_lat_30_302"

Counters:
Total records written : 501
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local2077663004_0001


2015-10-05 08:30:05,957 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 2 time(s).
2015-10-05 08:30:05,957 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
```

This displays the results received after storing the data successfully (Column names are as follows: species, latitude, longitude, oiling, birdcount)

```
"Brown Pelican" 30.09232        -85.64811       "Visibly Oiled" "Live"
"Wilson's Storm-petrel" 30.17009        -87.33448       "Not Visibly Oiled"
"Live"
"Laughing Gull" 30.12509        -88.57988       "Not Visibly Oiled"     "Live"
"Rock Pigeon"   30.16888        -87.31127       "Not Visibly Oiled"     "Live"
"Northern Gannet"       30.15983        -87.35555       "Visibly Oiled" "Live"
"Brown Pelican" 30.13247        -88.31653       "Not Visibly Oiled"     "Live"
"Northern Gannet"       30.17642        -87.59083       "Visibly Oiled" "Live"
"Northern Gannet"       30.16975        -87.30252       "Visibly Oiled" "Live"
"Northern Gannet"       30.02362        -88.1468        "Visibly Oiled" "Live"
"Northern Gannet"       30.09   -86.43  "Visibly Oiled" "Live"
"Northern Gannet"       30.13927        -88.37061       "Visibly Oiled" "Live"
"Northern Gannet"       30.13051        -87.31092       "Not Visibly Oiled"
"Live"
"Northern Gannet"       30.15587        -87.26309       "Visibly Oiled" "Live"
"Mallard"       30.13956        -87.1438        "Visibly Oiled" "Live"
"Northern Gannet"       30.07678        -87.43327       "Visibly Oiled" "Live"
"Northern Gannet"       30.07678        -87.43327       "Visibly Oiled" "Live"
"Northern Gannet"       30.14933        -86.27367       "Visibly Oiled" "Live"
"Northern Gannet"       30.12797        -86.27788       "Visibly Oiled" "Live"
"Great Blue Heron"      30.1309 -88.07434       "Not Visibly Oiled"     "Live"
"Northern Gannet"       30.18052        -87.09736       "Visibly Oiled" "Live"
"Northern Gannet"       30.1562 -87.2483        "Visibly Oiled" "Live"
```

Then, I generated the species and oiling results for the birds by typing the following:

Anubha Bhargava
Homework 1 – Big Data Analytics

```
grunt> birds_species_oiling = foreach birds generate Species,(float)Oiling;
```

Then, I stored the data into birds_species_oiling:

```
Input(s):
Successfully read 7230 records from: "/home/bigdata/Documents/birds.csv"

Output(s):
Successfully stored 7230 records in: "/home/bigdata/Documents/birds_species_oili
ng"

Counters:
Total records written : 7230
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local668267525_0001


2015-10-05 08:43:06,549 [main] WARN  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_C
ONVERSION_FAILED 14460 time(s).
2015-10-05 08:43:06,549 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.mapReduceLayer.MapReduceLauncher - Success!
```

I received the following results that contains the species type and oiling:

```
"Species"       "Oiling"
"Northern Gannet"       "Not Visibly Oiled"
"Laughing Gull" "Not Visibly Oiled"
"Northern Gannet"       "Visibly Oiled"
"American White Pelican"        "Not Visibly Oil
"Brown Pelican" "Visibly Oiled"
"Brown Pelican" "Not Visibly Oiled"
"Northern Gannet"       "Unknown"
"Common Loon"   "Not Visibly Oiled"
"Brown Pelican" "Visibly Oiled"
"Northern Gannet"       "Not Visibly Oiled"
"Northern Gannet"       "Visibly Oiled"
"Brown Pelican" "Not Visibly Oiled"
"Laughing Gull" "Not Visibly Oiled"
"Ruddy Turnstone"       "Visibly Oiled"
"Brown Pelican" "Not Visibly Oiled"
"Herring Gull"  "Not Visibly Oiled"
"Northern Gannet"       "Not Visibly Oiled"
"Northern Gannet"       "Visibly Oiled"
"Northern Gannet"       "Not Visibly Oiled"
"Laughing Gull" "Not Visibly Oiled"
"Common Loon"   "Not Visibly Oiled"
"Northern Gannet"       "Not Visibly Oiled"
```

Lastly, I sorted the birds by latitude and stored them into variable birds_lat _sort:

```
grunt> birds_lat_sort = order birds by Latitude desc;
2015-10-05 08:45:08,096 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2015-10-05 08:45:08,114 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> store birds_lat_sort into '/home/bigdata/Documents/birds_lat_sort';
```

When opening the file containing the results, the following displayed. Note that they are sorted by descending latitude (the second column).

```
"Laughing Gull" 30.97813        -88.97813       "Not Visibly Oiled"     "Dead"
"Common Loon"   30.9259 -87.5559        "Not Visibly Oiled"     "Live"
"Herring Gull"  30.9117 -88.96537       "Not Visibly Oiled"     "Dead"
"Other" 30.87806        -89.00389       "Visibly Oiled" "Dead"
"Mallard"       30.787  -89.08532       "Not Visibly Oiled"     "Live"
"Laughing Gull" 30.787  -89.08532       "Visibly Oiled" "Live"
"Laughing Gull" 30.72833        -89.59889       "Visibly Oiled" "Live"
"Laughing Gull" 30.72833        -89.59889       "Visibly Oiled" "Dead"
"Brown Pelican" 30.72095        -88.04112       "Visibly Oiled" "Live"
"Northern Gannet"       30.68416        -86.48066       "Not Visibly Oiled"
"Live"
"Laughing Gull" 30.66285        -89.35222       "Visibly Oiled" "Live"
"Laughing Gull" 30.66285        -89.35222       "Visibly Oiled" "Dead"
"Herring Gull"  30.66204        -88.0365        "Not Visibly Oiled"     "Live"
"Herring Gull"  30.66204        -88.03654       "Not Visibly Oiled"     "Dead"
"Least Tern"    30.65569        -87.91173       "Visibly Oiled" "Live"
"Canada Goose"  30.64515        -87.76954       "Visibly Oiled" "Live"
"Laughing Gull" 30.62881        -88.1022        "Not Visibly Oiled"     "Live"
"Laughing Gull" 30.62881        -88.1022        "Not Visibly Oiled"     "Dead"
"Northern Gannet"       30.62642        -86.61698       "Not Visibly Oiled"
"Live"
"Black Crowned Night Heron"     30.62203        -88.12915       "Not Visibly Oil
ed"     "Dead"
```

Now, after collecting all this data, I ssh-ed into localhost, ran pig, and ran the commands to start HDFS.

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0/sbin$ ./start-dfs.sh
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop-2.5.0/logs/hadoop-big
data-namenode-ubuntu.out
localhost: starting datanode, logging to /usr/local/hadoop-2.5.0/logs/hadoop-big
data-datanode-ubuntu.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.5.0/logs/had
```

I created the directory for PigSource2 and moved the data from the birds.csv into this folder.

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0/bin$ ./hdfs dfs -mkdir /PigSource2
bigdata@ubuntu:/usr/local/hadoop-2.5.0/bin$ ./hdfs dfs -put /home/bigdata/Docume
nts/birds.csv /PigSource2
```

```
grunt> birds = LOAD '/PigSource2/birds.csv' USING PigStorage(',') as (Species, L
atitude, Longitude, Oiling, BirdCount);
2015-10-05 09:03:29,514 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt> DUMP birds;
```

I loaded and dumped the data, as shown above. I received the following results in Pig after performing the dump of the data:

```
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Black Skimmer",30.06896,-89.20712,"Not Visibly Oiled","Dead")
("Clapper Rail",29.47566,-89.71014,"Not Visibly Oiled","Dead")
("Clapper Rail",29.49309,-89.91873,"Unknown","Dead")
("Black Skimmer",29.26371,-89.96423,"Visibly Oiled","Dead")
("Clapper Rail",29.44239,-89.88561,"Visibly Oiled","Dead")
("Clapper Rail",29.50644,-89.86126,"Unknown","Dead")
grunt>
```

HBase:

I downloaded and configured HBase using the online instructions. The following shows a successful start using start-hbase.sh:

```
bigdata@ubuntu:~/hbase-0.98.14-hadoop1/bin$ ./start-hbase.sh
localhost: starting zookeeper, logging to /home/bigdata/hbase-0.98.14-hadoop1/bi
n/../logs/hbase-bigdata-zookeeper-ubuntu.out
starting master, logging to /home/bigdata/hbase-0.98.14-hadoop1/bin/../logs/hbas
e-bigdata-master-ubuntu.out
localhost: starting regionserver, logging to /home/bigdata/hbase-0.98.14-hadoop1
/bin/../logs/hbase-bigdata-regionserver-ubuntu.out
```

Then I entered the shell using ./hbase shell. I then created a table in HBase as demonstrated below:

```
hbase(main):001:0> create 'table','cf'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase-0.98.6.1-hadoop2/lib/slf4j-lo
g4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.5.0/share/hadoop/common/li
b/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
0 row(s) in 3.4410 seconds

=> Hbase::Table - table
hbase(main):002:0> put 'table','r1','cf:c1','value1'
0 row(s) in 1.0420 seconds

hbase(main):003:0> scan 'table'
ROW                    COLUMN+CELL
 r1                    column=cf:c1, timestamp=1444076196150, value=value1
1 row(s) in 0.1910 seconds

hbase(main):004:0>
```

Anubha Bhargava
Homework 1 – Big Data Analytics

Hive:

I successfully downloaded Hive, as shown in the directory below:

```
bigdata@ubuntu:/usr/local/apache-hive-0.13.1-bin$ ls
bin   examples  lib      NOTICE      RELEASE_NOTES.txt
conf  hcatalog  LICENSE  README.txt  scripts
```

I started Hadoop and created a test directory:

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0/bin$ ./hadoop fs -mkdir /Test
```

I created a text file to run in Hive for testing:

```
anubha_test ×
Name,Age,Occupation,Sex,Origin
Alice,14,Student,Female,U.S.A.
Patrick,60,Cashier,Male,Ireland
David,57,Professor,Male,England
Ben,20,Student,Male,China
Reena,30,Engineer,Female,India
Saba,40,Doctor,Female,Saudi Arabia
Kevin,37,Not Employed,Male,U.S.A.
Marco,24,Farmer,Male,Mexico
Simona,78,Retired,Female,Italy
```

I moved the file onto HDFS:

```
bigdata@ubuntu:/usr/local/apache-hive-0.13.1-bin/bin$ hadoop fs -ls /Test
Found 1 items
-rw-r--r--   1 bigdata supergroup        351 2015-10-07 14:16 /Test/anubha_test
bigdata@ubuntu:/usr/local/apache-hive-0.13.1-bin/bin$ ./hive

Logging initialized using configuration in jar:file:/usr/local/apache-hive-0.13.
```

Afterwards, I opened Hive and created a table called “anubha_table” with the same criteria as the text file. I loaded the text file into the table as shown below.

```
hive> create table anubha_table(
    > Name String,
    > Age INT,
    > Occupation String,
    > Sex String,
    > Origin String)
    > row format delimited
    > fields terminated by ','
    > lines terminated by '\n'
    > stored as textfile;
OK
Time taken: 0.611 seconds
hive> load data inpath '/Test/anubha_test' into table anubha_table;
Loading data to table default.anubha_table
Table default.anubha_table stats: [numFiles=1, numRows=0, totalSize=311, rawData
Size=0]
OK
Time taken: 0.833 seconds
```

Then, I ran the select command to return the column "Age." I got the following results:

```
hive> select Age from anubha_table;
Total jobs = 1
Launching Job 1 out of 1
Number of reduce tasks is set to 0 since there's no reduce operator
Starting Job = job_1444252543103_0006, Tracking URL = http://localhost:8088/prox
y/application_1444252543103_0006/
Kill Command = /usr/local/hadoop-2.5.0/bin/hadoop job  -kill job_1444252543103_0
006
Hadoop job information for Stage-1: number of mappers: 1; number of reducers: 0
2015-10-07 14:46:49,061 Stage-1 map = 0%,  reduce = 0%
2015-10-07 14:47:11,731 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU 2.8 sec
MapReduce Total cumulative CPU time: 2 seconds 800 msec
Ended Job = job_1444252543103_0006
MapReduce Jobs Launched:
Job 0: Map: 1   Cumulative CPU: 2.8 sec   HDFS Read: 932 HDFS Write: 60 SUCCESS
Total MapReduce CPU Time Spent: 2 seconds 800 msec
OK
NULL
14
60
57
20
30
40
37
24
78
```

I started the Hive Thrift Server using the commands below:

```
bigdata@ubuntu:/usr/local/apache-hive-0.13.1-bin/bin$ ./hiveserver2 --service hiveserver -p 1
0000
Starting Hive Thrift Server
```

Oozie:

I successfully ssh-ed into localhost, cd-ed into the Hadoop directory and ran the scripts start-dfs.sh and start-yarn.sh:

Anubha Bhargava
Homework 1 – Big Data Analytics

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ ssh localhost
Welcome to Ubuntu 14.04.1 LTS (GNU/Linux 3.13.0-32-generic x86_64)

 * Documentation:  https://help.ubuntu.com/

Last login: Sat Oct  3 16:10:06 2015 from localhost
bigdata@ubuntu:~$ cd /usr/local/hadoop-2.5.0/
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ ./sbin/start-dfs.sh
Starting namenodes on [localhost]
localhost: namenode running as process 23384. Stop it first.
localhost: datanode running as process 3896. Stop it first.
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.5.0/logs/hac
oop-bigdata-secondarynamenode-ubuntu.out
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ ./sbin/start-yarn.sh
starting yarn daemons
resourcemanager running as process 4207. Stop it first.
localhost: nodemanager running as process 4332. Stop it first.
bigdata@ubuntu:/usr/local/hadoop-2.5.0$
```

I ran Oozie with the following commands:

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ jps
4332 NodeManager
26142 RunJar
26779 SecondaryNameNode
3896 DataNode
4207 ResourceManager
27002 Jps
23384 NameNode
```

I started Oozie by running the oozied.sh script:

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0$ cd /usr/local/oozie-4.0.1/
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozied.sh start

Setting OOZIE_HOME:          /usr/local/oozie-4.0.1
Setting OOZIE_CONFIG:        /usr/local/oozie-4.0.1/conf
Sourcing:                    /usr/local/oozie-4.0.1/conf/oozie-env.sh
  setting CATALINA_OPTS="$CATALINA_OPTS -Xmx1024m"
Setting OOZIE_CONFIG_FILE:   oozie-site.xml
Setting OOZIE_DATA:          /usr/local/oozie-4.0.1/data
Setting OOZIE_LOG:           /usr/local/oozie-4.0.1/logs
Setting OOZIE_LOG4J_FILE:    oozie-log4j.properties
Setting OOZIE_LOG4J_RELOAD:  10
Setting OOZIE_HTTP_HOSTNAME: ubuntu
Setting OOZIE_HTTP_PORT:     11000
Setting OOZIE_ADMIN_PORT:     11001
Setting OOZIE_HTTPS_PORT:     11443
```

I checked the Oozie running status which displayed to be normal:

Anubha Bhargava
Homework 1 – Big Data Analytics

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ cd /usr/local/oozie-4.0.1
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozie admin -oozie http://localhos
:11000/oozie -status
System mode: NORMAL
```

I untarred the oozie example:

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ cd /usr/local/oozie-4.0.1
bigdata@ubuntu:/usr/local/oozie-4.0.1$ tar -zxvf oozie-examples.tar.gz
examples/src/
examples/src/org/
examples/src/org/apache/
examples/src/org/apache/oozie/
examples/src/org/apache/oozie/example/
examples/src/org/apache/oozie/example/DemoPigMain.java
examples/src/org/apache/oozie/example/DemoJavaMain.java
examples/src/org/apache/oozie/example/Repeatable.java
```

I changed the namenode port number from 8020 to 9000. I also changed the jobtracker port number from 8021 to 8088. The commands are shown below:

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ cd /usr/local/oozie-4.0.1/examples
bigdata@ubuntu:/usr/local/oozie-4.0.1/examples$ find ./ -type f -exec sed -i -e
's/8020/9000/g' {} \;
bigdata@ubuntu:/usr/local/oozie-4.0.1/examples$ cd /usr/local/oozie-4.0.1/exampl
es
bigdata@ubuntu:/usr/local/oozie-4.0.1/examples$ find ./ -type f -exec sed -i -e
's/8021/8088/g' {} \;
bigdata@ubuntu:/usr/local/oozie-4.0.1/examples$
```

I moved the files into the folder map-reduce on HDFS:

```
bigdata@ubuntu:/usr/local/hadoop-2.5.0/bin$ ./hadoop fs -put /usr/local/oozie-4.
0.1/examples/apps/map-reduce/* /user/bigdata/examples/apps/map-reduce/.
```

Then, I submitted a job to Oozie. The job ID is shown below:

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozie job -oozie http://localhost:1
1000/oozie -config examples/apps/map-reduce/job.properties -run
job: 0000000-151003161217108-oozie-bigd-W
```

When checking the job status, the following displayed:

Anubha Bhargava
Homework 1 – Big Data Analytics

```
bigdata@ubuntu:/usr/local/oozie-4.0.1/bin$ ./oozie job -oozie http://localhost:1
1000/oozie -info 0000000-151003161217108-oozie-bigd-W
Job ID : 0000000-151003161217108-oozie-bigd-W
------------------------------------------------------------------------------
-----------------------------------------------------
Workflow Name : map-reduce-wf
App Path      : hdfs://localhost:9000/user/bigdata/examples/apps/map-reduce
Status        : RUNNING
Run           : 0
User          : bigdata
Group         : -
Created       : 2015-10-04 00:58 GMT
Started       : 2015-10-04 00:58 GMT
Last Modified : 2015-10-04 00:58 GMT
Ended         : -
CoordAction ID: -
```

In order to test Oozie, I checked the Oozie log files and noted that Oozie did start properly.

```
oozie.log (/usr/local/oozie-4.0.1/logs) - gedit
Open  Save  Undo
oozie.log
Running coordinator status service from last instance time =  2015-10-04T01:00Z
2015-10-03 18:01:31,985  INFO StatusTransitService$StatusTransitRunnable:539 - USER[-] GROUP[-]
Running bundle status service from last instance time =  2015-10-04T01:00Z
2015-10-03 18:01:31,986  INFO StatusTransitService$StatusTransitRunnable:539 - USER[-] GROUP[-]
Released lock for [org.apache.oozie.service.StatusTransitService]
2015-10-03 18:01:31,989  INFO PauseTransitService:539 - USER[-] GROUP[-] Released lock for
[org.apache.oozie.service.PauseTransitService]
2015-10-03 18:02:31,987  INFO StatusTransitService$StatusTransitRunnable:539 - USER[-] GROUP[-]
Acquired lock for [org.apache.oozie.service.StatusTransitService]
2015-10-03 18:02:31,987  INFO StatusTransitService$StatusTransitRunnable:539 - USER[-] GROUP[-]
Running coordinator status service from last instance time =  2015-10-04T01:01Z
2015-10-03 18:02:31,999  INFO PauseTransitService:539 - USER[-] GROUP[-] Acquired lock for
[org.apache.oozie.service.PauseTransitService]
2015-10-03 18:02:32,007  INFO PauseTransitService:539 - USER[-] GROUP[-] Released lock for
[org.apache.oozie.service.PauseTransitService]
```

I checked the Oozie status on the command line and browser, and the results displayed as normal.

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ ./bin/oozie admin --oozie http://localhos
t:11000/oozie -status
System mode: NORMAL
```