

EECS6893 Homework 1

Zhuxi Cai zc2270

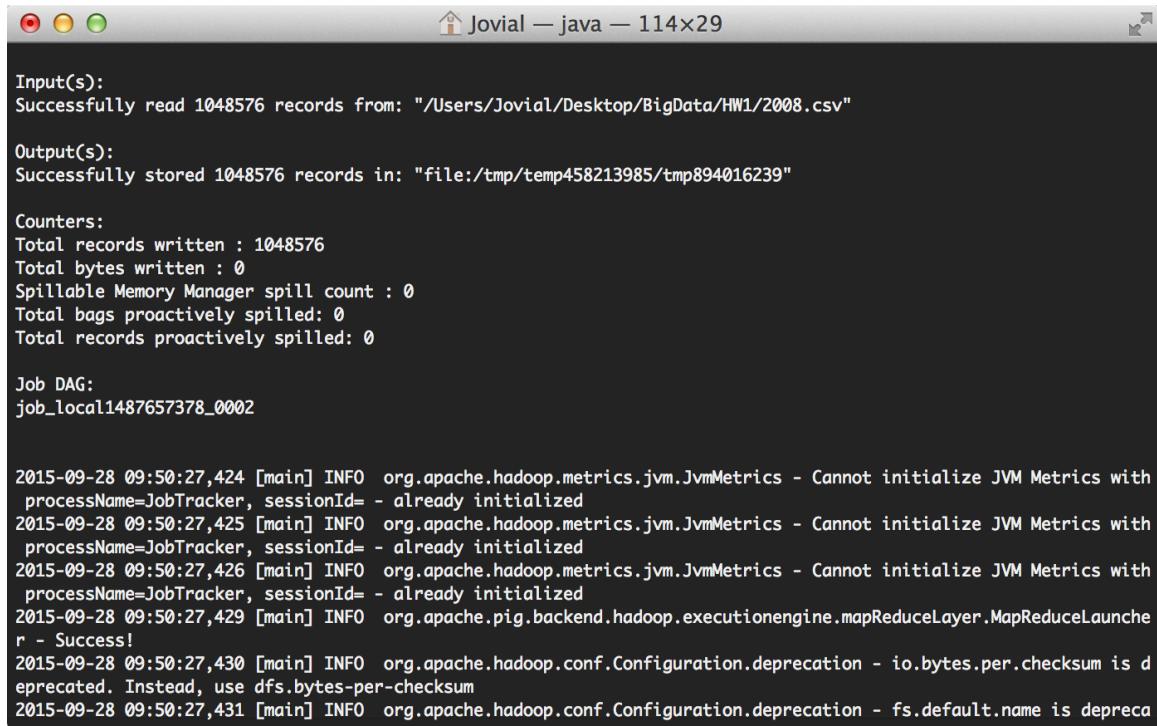
1. Learn to use PIG.

(1) Run pig in local mode:

(a) Install pig and read in 2008.csv as 'airline'

Code:

```
dyn-160-39-173-25:~ Jovial$ pig -x local
grunt> airline = load '/Users/Jovial/Desktop/BigData/HW1/2008.csv' using
PigStorage(',') as (Year,Month,DayofMonth,DayofWeek,DepTime,CRSDepTime,
ArrTime,CRSArrTime,UniqueCarr,FlightNum,Origin,Dest);
grunt> dump airline
```



```
Jovial — java — 114x29

Input(s):
Successfully read 1048576 records from: "/Users/Jovial/Desktop/BigData/HW1/2008.csv"

Output(s):
Successfully stored 1048576 records in: "file:/tmp/temp458213985/tmp894016239"

Counters:
Total records written : 1048576
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local1487657378_0002

2015-09-28 09:50:27,424 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 09:50:27,425 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 09:50:27,426 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 09:50:27,429 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLauncher.MapReduceLauncher -
Success!
2015-09-28 09:50:27,430 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is depre-
cated. Instead, use dfs.bytes-per-checksum
2015-09-28 09:50:27,431 [main] INFO org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is depre-
```

```

2015-09-28 09:50:27,470 [main] INFO org.apache.hadoop.mapreduce.lib.input.FileInputFormat - Total input paths to process : 2
2015-09-28 09:50:27,470 [main] INFO org.apache.pig.backend.hadoop.executionengine.util.MapRedUtil - Total input paths to process : 2
(Year,Month,DayofMonth,DayOfWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarrier,FlightNum,Origin,Dest)
(2008,1,3,4,2003,1955,2211,2225,WN,335,IAD,TPA)
(2008,1,3,4,754,735,1002,1000,WN,3231,IAD,TPA)
(2008,1,3,4,628,620,804,750,WN,448,IND,BWI)
(2008,1,3,4,926,930,1054,1100,WN,1746,IND,BWI)
(2008,1,3,4,1829,1755,1959,1925,WN,3920,IND,BWI)
(2008,1,3,4,1940,1915,2121,2110,WN,378,IND,JAX)
(2008,1,3,4,1937,1830,2037,1940,WN,509,IND,LAS)
(2008,1,3,4,1039,1040,1132,1150,WN,535,IND,LAS)
(2008,1,3,4,617,615,652,650,WN,11,IND,MCI)
(2008,1,3,4,1620,1620,1639,1655,WN,810,IND,MCI)
(2008,1,3,4,706,700,916,915,WN,100,IND,MCO)
(2008,1,3,4,1644,1510,1845,1725,WN,1333,IND,MCO)
(2008,1,3,4,1426,1430,1426,1425,WN,829,IND,MDW)
(2008,1,3,4,715,715,720,710,WN,1016,IND,MDW)
(2008,1,3,4,1702,1700,1651,1655,WN,1827,IND,MDW)
(2008,1,3,4,1029,1020,1021,1010,WN,2272,IND,MDW)
(2008,1,3,4,1452,1425,1640,1625,WN,675,IND,PHX)
(2008,1,3,4,754,745,940,955,WN,1144,IND,PHX)
(2008,1,3,4,1323,1255,1526,1510,WN,4,IND,TPA)
(2008,1,3,4,1416,1325,1512,1435,WN,54,ISP,BWI)
(2008,1,3,4,706,705,807,810,WN,68,ISP,BWI)
(2008,1,3,4,1657,1625,1754,1735,WN,623,ISP,BWI)
(2008,1,3,4,1900,1840,1956,1950,WN,717,ISP,BWI)
(2008,1,3,4,1039,1030,1133,1140,WN,1244,ISP,BWI)

```

(b) Filter: select the information of airline on 3rd January

Code:

```

grunt> airline0103 = filter airline by Month == 1 and DayofMonth == 3;
grunt> store airline0103 into '/Users/Jovial/Desktop/BigData/HW1/airline0103';
grunt> dump airline0103

```

```

Input(s): XLSX
Successfully read 1048576 records from: "/Users/Jovial/Desktop/BigData/HW1/2008.csv"

List-of-Cities-
Output(s): XLSX
Successfully stored 20937 records in: "/Users/Jovial/Desktop/BigData/HW1/airline0103"

Counters:
Total records written : 20937
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local180403461_0005

2015-09-28 11:13:01,681 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 11:13:01,682 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 11:13:01,682 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 11:13:01,686 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 2 time(s).
2015-09-28 11:13:01,686 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>

```

```

(2008,1,3,4,727,730,829,839,DL,1942,DCA,LGA)
(2008,1,3,4,728,730,828,839,DL,1943,LGA,DCA)
(2008,1,3,4,828,830,933,939,DL,1944,DCA,LGA)
(2008,1,3,4,829,830,944,943,DL,1945,LGA,DCA) filter airline by Month == 1 and DayofMonth == 3;
(2008,1,3,4,928,930,1038,1042,DL,1946,DCA,LGA)
(2008,1,3,4,929,930,1031,1051,DL,1947,LGA,DCA) 03 into '/Users/Jovial/Desktop/BigData/HW1/airline0103'
(2008,1,3,4,1028,1030,1137,1138,DL,1948,DCA,LGA)
(2008,1,3,4,1028,1030,1128,1148,DL,1949,LGA,DCA)
(2008,1,3,4,1129,1130,1239,1244,DL,1950,DCA,LGA)
(2008,1,3,4,1128,1130,1221,1250,DL,1951,LGA,DCA)
(2008,1,3,4,1228,1230,1329,1341,DL,1952,DCA,LGA)
(2008,1,3,4,1228,1230,1332,1341,DL,1953,LGA,DCA)
(2008,1,3,4,1328,1330,1427,1451,DL,1954,DCA,LGA) ds from: "/Users/Jovial/Desktop/BigData/HW1/2008.csv"
(2008,1,3,4,1328,1330,1436,1442,DL,1955,LGA,DCA)
(2008,1,3,4,1430,1430,1541,1544,DL,1956,DCA,LGA)
(2008,1,3,4,1429,1430,1529,1540,DL,1957,LGA,DCA) ds in: "/Users/Jovial/Desktop/BigData/HW1/airline0103"
(2008,1,3,4,1527,1530,1637,1645,DL,1958,DCA,LGA)
(2008,1,3,4,1528,1530,1626,1650,DL,1959,LGA,DCA)
(2008,1,3,4,1628,1630,1734,1742,DL,1960,DCA,LGA)
(2008,1,3,4,1628,1630,1725,1749,DL,1961,LGA,DCA)
(2008,1,3,4,1727,1730,1833,1853,DL,1962,DCA,LGA) count : 0
(2008,1,3,4,1728,1730,1846,1851,DL,1963,LGA,DCA) 0
(2008,1,3,4,1829,1830,1943,1944,DL,1964,DCA,LGA) add: 0
(2008,1,3,4,1829,1830,1923,1950,DL,1965,LGA,DCA)
(2008,1,3,4,1928,1930,2040,2047,DL,1966,DCA,LGA)
(2008,1,3,4,1928,1930,2033,2044,DL,1967,LGA,DCA)
(2008,1,3,4,2026,2030,2124,2148,DL,1968,DCA,LGA)
(2008,1,3,4,2028,2030,2125,2145,DL,1969,LGA,DCA) INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM processName=JobTracker, sessionId= - already initialized
grunt> processName=JobTracker, sessionId= - already initialized

```

(c) Foreach Generate: list airlines on 3rd January and their airtime (in minutes)

Code:

```

grunt> airline_airtime = foreach airline0103 generate UniqueCarr,FlightNum,Origin,
Dest,(ArrTime-DepTime);
grunt> store airline_airtime into '/Users/Jovial/Desktop/BigData/HW1/airline_airtime';
grunt> dump airline_airtime;

```

```

Input(s):
Successfully read 1048576 records from: "/Users/Jovial/Desktop/BigData/HW1/2008.csv"

Output(s):
Successfully stored 20937 records in: "/Users/Jovial/Desktop/BigData/HW1/airline_airtime"

Counters:
Total records written : 20937
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local627511387_0006
```

generate name, (float)duration

```

2015-09-28 11:23:01,455 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM Metrics with
processName=JobTracker, sessionId= - already initialized
2015-09-28 11:23:01,460 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 490 time(s).
2015-09-28 11:23:01,460 [main] INFO org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Success!
grunt>
```

```

Input(s):
Successfully stored 20937 records in: "/Users/Jovial/Desktop/BigData/HW1/airline_airtime"

Output(s):
records written : 20937
bytes written : 0
Spillable Memory Manager spill count : 0
bags proactively spilled: 0
records proactively spilled: 0

Counters:
Total records written : 20937
Total bytes written : 0
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_local627511387_0006
```

generate name, (float)duration

```

(DL,1942,DCA,LGA,102.0)
(DL,1943,LGA,DCA,100.0)
(DL,1944,DCA,LGA,105.0)
(DL,1945,LGA,DCA,115.0)
(DL,1946,DCA,LGA,110.0)
(DL,1947,LGA,DCA,102.0)
(DL,1948,DCA,LGA,109.0)
(DL,1949,LGA,DCA,100.0)
(DL,1950,DCA,LGA,110.0)
(DL,1951,LGA,DCA,93.0)
(DL,1952,DCA,LGA,101.0)
(DL,1953,LGA,DCA,104.0)
(DL,1954,DCA,LGA,99.0)
09-28 11:23:01,455 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM
processName=JobTracker, sessionId= - already initialized
(DL,1955,LGA,DCA,108.0)
09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM
processName=JobTracker, sessionId= - already initialized
(DL,1956,DCA,LGA,111.0)
09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM
processName=JobTracker, sessionId= - already initialized
(DL,1957,LGA,DCA,100.0)
09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM
processName=JobTracker, sessionId= - already initialized
(DL,1958,DCA,LGA,110.0)
09-28 11:23:01,456 [main] INFO org.apache.hadoop.metrics.jvm.JvmMetrics - Cannot initialize JVM
processName=JobTracker, sessionId= - already initialized
(DL,1959,LGA,DCA,98.0)
09-28 11:23:01,460 [main] WARN org.apache.pig.backend.hadoop.executionengine.mapReduceLayer.MapReduceLauncher - Encountered Warning FIELD_DISCARDED_TYPE_CONVERSION_FAILED 490 time(s).
(DL,1960,DCA,LGA,106.0)
(DL,1961,LGA,DCA,97.0)
(DL,1962,DCA,LGA,106.0)
(DL,1963,LGA,DCA,118.0)
(DL,1964,DCA,LGA,114.0)
(DL,1965,LGA,DCA,94.0)
(DL,1966,DCA,LGA,112.0)
(DL,1967,LGA,DCA,105.0)
(DL,1968,DCA,LGA,98.0)
(DL,1969,LGA,DCA,97.0)
grunt>
```

(d) Order: list all airlines on 3rd January in descending order of actual departure time

Code:

```
grunt> airline_deptime = order airline0103 by DepTime desc;  
grunt> store airline_deptime into '/Users/Jovial/Desktop/BigData/HW1/  
airline_deptime';
```

The terminal window displays the following information:

Job Stats (time in seconds):

JobId	Maps	Reduces	MaxMapTime	MinMapTime	AvgMapTime	MedianMapTime	MaxReduceTime	MinReduceT
job_local11334366011_0014	1	1	n/a	n/a	n/a	n/a	n/a	n/a
airline_deptime	SAMPLER							
job_local1925796400_0015	1	1	n/a	n/a	n/a	n/a	n/a	n/a
airline_deptime	ORDER_BY							
job_local543329715_0013	2	0	n/a	n/a	n/a	0	0	0
airline0103	MAP_ONLY							

Input(s):
Successfully read 1048576 records from: "/Users/Jovial/Desktop/BigData/HW1/2008.csv"

Output(s):
Successfully stored 20937 records in: "/Users/Jovial/Desktop/BigData/HW1/airline_deptime"

Counters:

- Total records written : 20937
- Total bytes written : 0
- Spillable Memory Manager spill count : 0
- Total bags proactively spilled: 0
- Total records proactively spilled: 0

er: list all airlines on 3rd january in descending order of actual departure

Job DAG:

```
job_local543329715_0013 -> job_local1334366011_0014,  
job_local11334366011_0014 -> job_local1925796400_0015,  
job_local1925796400_0015
```

airline_deptime = order airline0103 by DepTime desc;

descending order of actual departure

(2) Run pig with HDFS:

(a) Run HDFS first

Code:

```
dyn-160-39-173-137:~ Jovial$ cd /usr/local/Cellar/hadoop/2.7.0
```

```
dyn-160-39-173-137:2.7.0 Jovial$ ./sbin/start-dfs.sh
```

(b) Upload file to HDFS

Code:

```
dyn-160-39-173-137:2.7.0 Jovial$ ./bin/hdfs dfs -mkdir ./PigSource
```

```
dyn-160-39-173-137:2.7.0 Jovial$ ./bin/hdfs dfs -put /Users/Jovial/Desktop/BigData/HW1/2008.csv ./PigSource
```

(c) Run pig in grunt with HDFS

Code:

```
dyn-160-39-173-137:2.7.0 Jovial$ pig
```

```
grunt> airline = load './PigSource/2008.csv' using PigStorage(',') as (Year,Month,DayofMonth,DayofWeek,DepTime,CRSDepTime,ArrTime,CRSArrTime,UniqueCarr,FlightNum,Origin,Dest);
```

```
grunt> dump airline
```

```
(2008,1,3,4,2003,1955,2211,2225,WN,335,N712SW,128)
(2008,1,3,4,754,735,1002,1000,WN,3231,N772SW,128)
(2008,1,3,4,628,620,804,750,WN,448,N428WN,96)
(2008,1,3,4,926,930,1054,1100,WN,1746,N612SW,88)
(2008,1,3,4,1829,1755,1959,1925,WN,3920,N464WN,90)
(2008,1,3,4,1940,1915,2121,2110,WN,378,N726SW,101)
(2008,1,3,4,1937,1830,2037,1940,WN,509,N763SW,240)
(2008,1,3,4,1039,1040,1132,1150,WN,535,N428WN,233)
(2008,1,3,4,617,615,652,650,WN,11,N689SW,95) /local/Cellar/hadoop/2.7.0
(2008,1,3,4,1620,1620,1639,1655,WN,810,N648SW,79)
(2008,1,3,4,706,700,916,915,WN,100,N690SW,130) /Start-dfs.sh
(2008,1,3,4,1644,1510,1845,1725,WN,1333,N334SW,121)
(2008,1,3,4,1426,1430,1426,1425,WN,829,N476WN,60)
(2008,1,3,4,715,715,720,710,WN,1016,N765SW,65)
(2008,1,3,4,1702,1700,1651,1655,WN,1827,N420WN,49)
(2008,1,3,4,1029,1020,1021,1010,WN,2272,N263WN,52)
(2008,1,3,4,1452,1425,1640,1625,WN,675,N286WN,228)
(2008,1,3,4,754,745,940,955,WN,1144,N778SW,226)
(2008,1,3,4,1323,1255,1526,1510,WN,4,N674AA,123)
(2008,1,3,4,1416,1325,1512,1435,WN,54,N643SW,56)
(2008,1,3,4,706,705,807,810,WN,68,N497WN,61)
(2008,1,3,4,1657,1625,1754,1735,WN,623,N724SW,57)
(2008,1,3,4,1900,1840,1956,1950,WN,717,N786SW,56)
(2008,1,3,4,1039,1030,1133,1140,WN,1244,N714CB,54)
(2008,1,3,4,801,800,902,910,WN,2101,N222WN,61)
(2008,1,3,4,1520,1455,1619,1605,WN,2553,N394SW,59)
(2008,1,3,4,1422,1255,1657,1610,WN,188,N215WN,155)
(2008,1,3,4,1954,1925,2239,2235,WN,1754,N243WN,165)
(2008,1,3,4,636,635,921,945,WN,2275,N454WN,165)
```

(3) Summary of PIG

PIG has the huge advantage in reading and printing huge data set than Excel. In our example, we have a data set of 700 MB large. When I tried to use Excel to open this, it was really slow and my computer nearly crashed. However, I don't need to be afraid of this problem if I use PIG based on Hadoop.

2. Try HBase. Use my own example.

Code:

```
dyn-160-39-173-137:~ Jovial$ cd /usr/local/Cellar/hbase/1.0.1
dyn-160-39-173-137:1.0.1 Jovial$ ./bin/start-hbase.sh
dyn-160-39-173-137:1.0.1 Jovial$ ./bin/hbase shell
```

```
1.0.1 — java — 90x27
hbase(main):001:0> create 'table','university'
0 row(s) in 0.9570 seconds

=> Hbase::Table - table
hbase(main):002:0> put 'table','1','university:name','Columbia'
0 row(s) in 0.0840 seconds

hbase(main):003:0> put 'table','1','university:state','NY'
0 row(s) in 0.0030 seconds

hbase(main):004:0> put 'table','2','university:name','Harvard'
0 row(s) in 0.0110 seconds

hbase(main):005:0> put 'table','2','university:state','MA'
0 row(s) in 0.0040 seconds

hbase(main):006:0> put 'table','3','university:name','Stanford'
0 row(s) in 0.0040 seconds
    column=stats:daily, timestamp=1321296699190, value=test-daily-value
hbase(main):007:0> put 'table','3','university:state','CA'
0 row(s) in 0.0120 seconds
    column=stats:weekly, timestamp=1321296715892, value=test-weekly-value
hbase(main):008:0> put 'table','4','university:name','UIUC'
0 row(s) in 0.0090 seconds

hbase(main):009:0> put 'table','4','university:state','IL'
0 row(s) in 0.0030 seconds
```

```
1.0.1 — java — 90x27
hbase(main):009:0> put 'table','4','university:state','IL'
0 row(s) in 0.0030 seconds

hbase(main):010:0> scan 'table'
ROW          COLUMN+CELL
1            column=university:name, timestamp=1443492420040, value=Columbia
1            column=university:state, timestamp=1443492435654, value=NY
2            column=university:name, timestamp=1443492470485, value=Harvard
2            'stats:weekly', timestamp=1443492514933, value=test-weekly-value
2            column=university:state, timestamp=1443492482006, value=MA
3            column=university:name, timestamp=1443492504394, value=Stanford
4            column=university:state, timestamp=1443492595149, value=IL
4 row(s) in 0.0210 seconds

hbase(main):011:0> get 'table','1'
COLUMN+CELL
column=stats:weekly, timestamp=1321296715892, value=test-weekly-value
COLUMN+CELL
column=university:name, timestamp=1321296787444, value=test-weekly-value
    column=university:name, timestamp=1443492420040, value=Columbia
    column=university:state, timestamp=1443492435654, value=NY
2 row(s) in 0.0210 seconds

hbase(main):012:0> the contents of row 1. Sample output:
```

Summary of HBase:

By using HBase, I can create and apply some basic NoSQL technics in a faster and more high-technology way.

3. Try Hive. Use my own example.

(1) start Hadoop, create a file

Code:

```
Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -mkdir ./test
```

(2) put your dataset into hdfs

Code:

```
Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -put /Users/Jovial/Desktop/BigData/HW1/sotc-10.csv ./test
```

```
Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -ls test
```

```
Zhuxis-MacBook-Pro:~ Jovial$ hadoop fs -ls test
15/09/29 09:07:00 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 1 items
-rw-r--r-- 1 Jovial staff 27782556 2015-09-29 09:06 test/sotc-10.csv
Zhuxis-MacBook-Pro:~ Jovial$
```

(3) create the table (SQL) in Hive shell

Code:

```
dyn-160-39-173-104:~ Jovial$ hive
```

```
hive> create table university(id int, name string, city string, state string)
```

```
> row format delimited
> fields terminated by '\t'
> lines terminated by '\n'
> stored as textfile;
```

The screenshot shows a terminal window with the title "Jovial — java — 80x24". The terminal output is as follows:

```
>Password:
dyn-160-39-173-104:~ Jovial$ sudo chmod a+rwx /user/hive/warehouse
dyn-160-39-173-104:~ Jovial$ hive
readlink: illegal option -- f
usage: readlink [-n] [file ...]

Logging initialized using configuration in jar:file:/usr/local/Cellar/hive/1.1.0/libexec/lib/hive-common-1.1.0.jar!/hive-log4j.properties2014
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/Cellar/hadoop/2.7.0/libexec/share/hadoop/common/lib/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/Cellar/hive/1.1.0/libexec/lib/hive-jdbc-1.1.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
hive> create table university(id int, name string, city string, state string)
      > row format delimited
      > fields terminated by '\t'
      > lines terminated by '\n'
      > stored as textfile;
OK
Time taken: 0.925 seconds
hive>
```

(4) store the data into table

Code:

```
hive> load data inpath '/Users/Jovial/test/University.txt' overwrite into table university;
hive> select * from university;
```

```
hive> load data inpath '/Users/Jovial/test/University.txt' overwrite into table
university;
Loading data to table default.university
chgrp: changing ownership of 'file:///user/hive/warehouse/university/University.
txt': chown: you are not a member of group wheel
Table default.university stats: [numFiles=1, numRows=0, totalSize=123, rawDataSi
ze=0]
OK
Time taken: 0.149 seconds
hive> select * from university;
OK
1      Harvard      Boston
2      Columbia     NewYork NY
3      Stanford     SanFrancisco CA
4      Berkeley     Oakland CA
5      Chicago      Chicago
Time taken: 0.054 seconds, Fetched: 5 row(s)
```

Summary of Hive:

I think it is a very good RDBS based on Hadoop and much quicker than local SQL. However the installation and configure is much more complicated than PIG and HBase. I spend a lot of time configuring Hive, especially in making the directory /user/hive/warehouse and getting the permission to create a new table. At last, I used command ‘sudo’ to fix this problem. I don’t really have better idea.

4. Oozie installation.

(1) start Oozie and check Oozie running status

Code:

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

(2) untar Oozie example

Code:

```
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
```

```
bigdata@ubuntu: /usr/local/oozie-4.0.1
examples/apps/sqoop/db.hsqldb.script
examples/apps/sqoop/job.properties
examples/apps/sqoop/workflow.xml
examples/input-data/text/data.txt
examples/input-data/rawLogs/2010/01/00/20/log01.txt
examples/input-data/rawLogs/2010/01/00/20/_SUCCESS
examples/input-data/rawLogs/2010/01/00/40/log02.txt
examples/input-data/rawLogs/2010/01/00/40/_SUCCESS
examples/input-data/rawLogs/2010/01/01/00/log03.txt
examples/input-data/rawLogs/2010/01/01/00/_SUCCESS
examples/input-data/rawLogs/2010/01/01/20/log04.txt
examples/input-data/rawLogs/2010/01/01/20/_SUCCESS
examples/input-data/rawLogs/2010/01/01/40/log05.txt
examples/input-data/rawLogs/2010/01/01/40/_SUCCESS
examples/input-data/rawLogs/2010/01/02/00/log06.txt
examples/input-data/rawLogs/2010/01/02/00/_SUCCESS
examples/apps/aggregator/lib/oozie-examples-4.0.1.jar
examples/apps/custom-main/lib/oozie-examples-4.0.1.jar
examples/apps/demo/lib/oozie-examples-4.0.1.jar
examples/apps/hadoop-el/lib/oozie-examples-4.0.1.jar
examples/apps/java-main/lib/oozie-examples-4.0.1.jar
examples/apps/datelist-java-main/lib/oozie-examples-4.0.1.jar
examples/apps/map-reduce/lib/oozie-examples-4.0.1.jar
bigdata@ubuntu:/usr/local/oozie-4.0.1$ 
```

(3) change namenode port number from 8020 to 9000 and change jobtracker port number from 8021 to 8088

Code:

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

(4) submit a job to Oozie

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ bin/oozie job -oozie http://localhost:11000/oozie -config examples/apps/map-reduce/job.properties -run
job: 0000001-151004231153310-oozie-bigd-W
bigdata@ubuntu:/usr/local/oozie-4.0.1$ 
```

(5) check the job status

```
bigdata@ubuntu:/usr/local/oozie-4.0.1$ bin/oozie job -oozie http://localhost:11000/oozie -info 00-151004231153310-oozie-bigd-W
Job ID : 0000000-151004231153310-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       : 2015-10-05 07:08 GMT
Started       : 2015-10-05 07:08 GMT
Last Modified : 2015-10-05 07:08 GMT
Ended         : -
CoordAction ID: -

Actions
-----
ID          Status Ext ID
<           Ext Status Err Code
-----<
0000000-151004231153310-oozie-bigd-W:@:start:          OK      -
              -          -
-----<
0000000-151004231153310-oozie-bigd-W@mr-node          PREP    -
              -          -
-----<

bigdata@ubuntu:/usr/local/oozie-4.0.1$
```

Summary of Oozie:

Good to learn a lot on virtual machine and Linux environment. Now I am more familiar with Linux command lines and Oozie.