

1.

```
[root@sandbox ~]# hadoop fs -mkdir /final_project  
[root@sandbox ~]# hadoop fs -mkdir /final_project/data
```

The screenshot shows the Ambari Sandbox interface. At the top, there are several tabs including 'Ambari - Sandbox'. Below the tabs, the main area displays a file browser for the HDFS directory '/final_project/data'. The browser has a header with 'Total: 1 files or folders'. It lists one file, 'flightDelays.tar.gz', which is 63.0 MB in size, last modified on 2020-05-20 15:03, owned by 'admin', and belongs to the 'hdfs' group. There are buttons for 'Select All', 'New Folder', and 'Upload' at the top right. A search bar is also present.

2.

```
[root@sandbox ~]# mkdir FP  
[root@sandbox ~]# cd FP  
[root@sandbox FP]# hadoop fs -copyToLocal  
/final_project/data/flightDelays.tar.gz  
[root@sandbox FP]# gzip -d flightDelays.tar.gz  
[root@sandbox FP]# tar -xf flightDelays.tar  
[root@sandbox FP]# ls -lh
```

The screenshot shows a terminal window with the command 'ls -lh' run in the directory 'FP'. The output shows 764 files, all of which are CSV files named 'flightDelays_10.csv' through 'flightDelays_9.csv', indicating they are from April 2015. The files are all 64M in size and have the same modification date.

File Name	Size	Last Modified
flightDelays_10.csv	64M	Apr 2015
flightDelays_11.csv	64M	Apr 2015
flightDelays_12.csv	63M	Apr 2015
flightDelays_1.csv	61M	Apr 2015
flightDelays_2.csv	57M	Apr 2015
flightDelays_3.csv	67M	Apr 2015
flightDelays_4.csv	65M	Apr 2015
flightDelays_5.csv	66M	Apr 2015
flightDelays_6.csv	67M	Apr 2015
flightDelays_7.csv	78M	Apr 2015
flightDelays_8.csv	68M	Apr 2015
flightDelays_9.csv	61M	Apr 2015

3.

```
[root@sandbox FP]# hadoop fs -copyFromLocal flightDelays_1.csv  
flightDelays_2.csv flightDelays_3.csv flightDelays_4.csv  
flightDelays_5.csv flightDelays_6.csv flightDelays_7.csv  
flightDelays_8.csv flightDelays_9.csv flightDelays_10.csv  
flightDelays_11.csv flightDelays_12.csv /final_project/data
```

Name	Size	Last Modified	Owner	Group	Permission
flightDelays.tar.gz	63.0 MB	2020-05-20 15:03	admin	hdfs	-rW-f--f--
flightDelays_1.csv	60.8 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_10.csv	63.6 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_11.csv	59.8 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_12.csv	62.8 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_2.csv	56.0 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_3.csv	66.0 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_4.csv	64.4 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_5.csv	65.7 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_6.csv	66.9 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_7.csv	69.1 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_8.csv	67.5 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--
flightDelays_9.csv	60.7 MB	2020-05-20 15:43	root	hdfs	-rW-f--f--

The first time I used copyFromLocal from the FP folder only having to type out the file names to the HDFS directory.

```
[root@sandbox FP]# cd ~  
[root@sandbox ~]# hadoop fs -copyFromLocal  
~/FP/flightDelays_1.csv ~/FP/flightDelays_2.csv  
~/FP/flightDelays_3.csv ~/FP/flightDelays_4.csv  
~/FP/flightDelays_5.csv ~/FP/flightDelays_6.csv  
~/FP/flightDelays_7.csv ~/FP/flightDelays_8.csv  
~/FP/flightDelays_9.csv ~/FP/flightDelays_10.csv  
~/FP/flightDelays_11.csv ~/FP/flightDelays_12.csv  
/final_project/data
```

The second method I went back to the root and copied it from the root using the full address rather than from the FP folder.

4. [root@sandbox ~]# hive


```
hive> CREATE DATABASE FPdb
          > LOCATION '/final_project';
hive> DESCRIBE DATABASE FPdb;
```

```
hive> CREATE DATABASE FPdb
      > LOCATION '/final_project';
OK
Time taken: 1.344 seconds
hive> DESCRIBE DATABASE FPdb;
OK
fpdb          hdfs://sandbox.hortonworks.com:8020/final_project      root      USE
R
Time taken: 0.385 seconds, Fetched: 1 row(s)
hive>
```

5.

```
flightDelays = LOAD '/final_project/data/flightDelays_'
USING org.apache.pig.piggybank.storage.CSVExcelStorage()
as (YEAR:int,
FL_DATE:chararray,
UNIQUE_CARRIER:chararray,
CARRIER:chararray,
FL_NUM:chararray,
ORIGIN_AIRPORT_ID:chararray,
ORIGIN:chararray,
ORIGIN_CITY_NAME:chararray,
ORIGIN_STATE_ABR:chararray,
```

```
DEST_AIRPORT_ID:chararray,  
DEST:chararray,  
DEST_CITY_NAME:chararray,  
DEST_STATE_ABR:chararray,  
DEP_DELAY_NEW:float,  
ARR_DELAY:float,  
ARR_DELAY_NEW:float,  
CARRIER_DELAY:float,  
WEATHER_DELAY:float,  
NAS_DELAY:float,  
SECURITY_DELAY:float,  
LATE_AIRCRAFT_DELAY:float);
```

The screenshot shows the Ambari Sandbox interface. At the top, there's a navigation bar with links like 'Ambari - Sandbox', 'localhost:8080/#main/views/PIG/1.0.0/PIG_INSTANCE', and various system icons. Below the navigation bar is a header with tabs for 'Script' (selected), 'History', and 'flightDelays.pig - Completed'. A progress bar indicates the job is completed. On the left, a sidebar shows options for saving, copying, or deleting the script. The main area displays the 'Results' section, which is currently empty, and the 'Logs' section, which contains the following log output:

```
WARNING: Use "yarn jar" to launch YARN applications.  
20/05/22 23:41:13 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL  
20/05/22 23:41:13 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE  
20/05/22 23:41:13 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType  
2020-05-22 23:41:13,692 [main] INFO org.apache.pig.Main - Apache Pig version 0.16.0-2.5.0.0-1245 (reported) compiled:  
2020-05-22 23:41:13,693 [main] INFO org.apache.pig.Main - Logging error messages to: /hadoop/yarn/local/usercache/admin  
2020-05-22 23:41:14,305 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/yarn/.pigbootup not found  
2020-05-22 23:41:14,397 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop  
2020-05-22 23:41:14,836 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-script.pig-ea34cb71-  
2020-05-22 23:41:15,116 [main] INFO org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service address: http://  
http://127.0.0.1:12000/timeline
```

At the bottom of the screen, there's a Windows taskbar with a search bar, pinned icons for File Explorer, Task View, and others, and system status indicators.

The screenshot shows a terminal window with multiple tabs. The active tab displays Apache Pig logs from May 22, 2020, at 23:41:13. The logs include information about the Pig version (0.16.0), logging error messages to a local file, connecting to HDFS, and executing a script named PIG-script.pig-ea34cb71. The log ends with a message indicating the script completed in 2 seconds and 285 milliseconds.

Script Details

Script contents:

```

6   CARRIER:chararray,
7   FL_NUM:chararray,
8   ORIGIN_AIRPORT_ID:chararray,
9   ORIGIN:chararray,
10  ORIGIN_CITY_NAME:chararray,
11  ORIGIN_STATE_ABR:chararray,
12  DEST_AIRPORT_ID:chararray,
13  DEST:chararray,
14  DEST_CITY_NAME:chararray,
15  DEST_STATE_ABR:chararray,
    DEP_DELAY_NEW:float

```

Arguments:

This job was executed without arguments.

6.

```

flightDelays = LOAD '/final_project/data/flightDelays_'
USING org.apache.pig.piggybank.storage.CSVExcelStorage()
as (YEAR:int,
FL_DATE:chararray,
UNIQUE_CARRIER:chararray,
CARRIER:chararray,
FL_NUM:chararray,
ORIGIN_AIRPORT_ID:chararray,
ORIGIN:chararray,
ORIGIN_CITY_NAME:chararray,
ORIGIN_STATE_ABR:chararray,
DEST_AIRPORT_ID:chararray,
DEST:chararray,
DEST_CITY_NAME:chararray,
DEST_STATE_ABR:chararray,
DEP_DELAY_NEW:float,
ARR_DELAY:float,
ARR_DELAY_NEW:float,
CARRIER_DELAY:float,
WEATHER_DELAY:float,
NAS_DELAY:float,
SECURITY_DELAY:float,

```

```
LATE_AIRCRAFT_DELAY:float);
```

```
wanted_data = foreach flightDelays generate CARRIER_DELAY, WEATHER_DELAY,  
NAS_DELAY, SECURITY_DELAY, LATE_AIRCRAFT_DELAY;  
grpds = group wanted_data all;  
averageDelays = foreach grpds generate  
ROUND_TO( AVG(wanted_data.CARRIER_DELAY), 2),  
ROUND_TO( AVG(wanted_data.WEATHER_DELAY), 2),  
ROUND_TO( AVG(wanted_data.NAS_DELAY), 2),  
ROUND_TO( AVG(wanted_data.SECURITY_DELAY), 2),  
ROUND_TO( AVG(wanted_data.LATE_AIRCRAFT_DELAY), 2);  
dump averageDelays;
```

averageDelays.pig - COMPLETED

Save Copy Delete

Job ID: job_1590011432575_0064
Started: 2020-05-22 12:18

Results

(16.65,2.34,13.73,0.08,23.87)

Logs

Download

WARNING: Use "yarn jar" to launch YARN applications.
2020-05-22 16:19:05 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2020-05-22 16:19:05 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE
2020-05-22 16:19:05 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType
2020-05-22 16:19:05,335 [main] INFO org.apache.pig.Main - Apache Pig version 0.16.0-2.5.0.0-1245 (reported) compile
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:05,335 [main] INFO org.apache.pig.Main - Logging error messages to: /hadoop/yarn/local/usercache/ad
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:06,007 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/yarn/.pigbootup not f
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:06,134 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to h
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:06,646 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-script.pig-003b0d5
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:07,009 [main] INFO org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service add
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:07,156 [main] INFO org.apache.pig.backend.hadoop.PigATSClient - Created ATS Hook
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:08,053 [main] INFO org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GRO
(16.65,2.34,13.73,0.08,23.87)
2020-05-22 16:19:08,103 [main] INFO org.apache.pig.data.SchemaTupleBackend - Key [pig.schematuple] was not set... wi
(16.65,2.34,13.73,0.08,23.87)

The screenshot shows a Linux desktop environment with a terminal window and a browser window.

Terminal Window:

```

$ cat flightDelays*.txt
# File: flightDelays*.txt
# Line 1: LATE_AIRCRAFT_DELAY:float
# Line 2: wanted_data = foreach flightDelays generate CARRIER_DELAY, WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, LAT...
# Line 3: grpdt = group wanted_data all;
# Line 4: averageDelays = foreach grpdt generate
# Line 5: ROUND_TO(AVG(wanted_data.CARRIER_DELAY), 2),
# Line 6: ROUND_TO(AVG(wanted_data.WEATHER_DELAY), 2),
# Line 7: ROUND_TO(AVG(wanted_data.NAS_DELAY), 2),
# Line 8: ROUND_TO(AVG(wanted_data.SECURITY_DELAY), 2),
# Line 9: ROUND_TO(AVG(wanted_data.LATE_AIRCRAFT_DELAY), 2);
# Line 10: dump averageDelays;

```

Browser Window:

The browser window displays the output of a Pig Latin job. The output includes:

- Success!**
- Job Stats (time in seconds):**

JobId	Maps	Reduces	MaxMapTime	MinMapTime	AvgMapTime	MedianMapTime	MaxReduceTime	MinReduceTime
job_1590011432575_0065	7	1	18	7	14	15	12	12
- Input(s):** Successfully read 6369482 records (800288154 bytes) from: "/final_project/data/flightDelays_*
- Output(s):** Successfully stored 1 records (49 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp2048342054/tmp-289587415"
- Counters:**
 - Total records written : 1
 - Total bytes written : 49
 - Spillable Memory Manager spill count : 0
 - Total bags proactively spilled: 56
 - Total records proactively spilled: 33014424

7.

```

flightDelays = LOAD '/final_project/data/flightDelays_'
USING org.apache.pig.piggybank.storage.CSVExcelStorage()
as (YEAR:int,
FL_DATE:chararray,
UNIQUE_CARRIER:chararray,
CARRIER:chararray,
FL_NUM:chararray,
ORIGIN_AIRPORT_ID:chararray,
ORIGIN:chararray,
ORIGIN_CITY_NAME:chararray,
ORIGIN_STATE_ABR:chararray,
DEST_AIRPORT_ID:chararray,
DEST:chararray,
DEST_CITY_NAME:chararray,
DEST_STATE_ABR:chararray,
DEP_DELAY_NEW:float,
ARR_DELAY:float,
ARR_DELAY_NEW:float,
CARRIER_DELAY:float,
WEATHER_DELAY:float,

```

```
NAS_DELAY:float,  
SECURITY_DELAY:float,  
LATE_AIRCRAFT_DELAY:float);
```

```
wanted = foreach flightDelays generate CARRIER_DELAY, WEATHER_DELAY,  
NAS_DELAY,  
SECURITY_DELAY, LATE_AIRCRAFT_DELAY;  
grp = group wanted all;  
longDelays = foreach grp generate  
MAX(wanted.CARRIER_DELAY),  
MAX(wanted.WEATHER_DELAY),  
MAX(wanted.NAS_DELAY),  
MAX(wanted.SECURITY_DELAY),  
MAX(wanted.LATE_AIRCRAFT_DELAY);  
dump longDelays;
```

The screenshot shows the Ambari Sandbox interface on a Windows desktop. The browser window is titled 'localhost:8080/#/main/views/PIG/1.0.0/PIG_INSTANCE'. The main content area displays a completed Pig job named 'longestDelays.pig'. The job status is 'COMPLETED'. Job ID is 'job_1590011432575_0066' and it started at '2020-05-22 12:22'. Below the job details, there are two sections: 'Results' and 'Logs'. The 'Results' section contains a single tuple: '(1975.0,1591.0,1287.0,573.0,1182.0)'. The 'Logs' section shows the command-line output of the Pig script execution, including logs from the Apache Pig Main class and the Hadoop YARN Timeline service.

```
longestDelays.pig - COMPLETED  
Job ID      job_1590011432575_0066  
Started     2020-05-22 12:22  
  
Results  
(1975.0,1591.0,1287.0,573.0,1182.0)  
  
Logs  
WARNING: Use "yarn jar" to launch YARN applications.  
20/05/22 16:22:37 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL  
20/05/22 16:22:37 INFO pig.ExecTypeProvider: Trying ExecType : MAPREDUCE  
20/05/22 16:22:37 INFO pig.ExecTypeProvider: Picked MAPREDUCE as the ExecType  
2020-05-22 16:22:37,453 [main] INFO org.apache.pig.Main - Apache Pig version 0.16.0-2.5.0.0-1245 (reexported) compiled by  
2020-05-22 16:22:37,453 [main] INFO org.apache.pig.Main - Logging error messages to: /hadoop/yarn/local/usercache/ad  
2020-05-22 16:22:38,010 [main] INFO org.apache.pig.impl.util.Utils - Default bootup file /home/yarn/.pigbootup not f  
2020-05-22 16:22:38,170 [main] INFO org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to h  
2020-05-22 16:22:38,685 [main] INFO org.apache.pig.PigServer - Pig Script ID for the session: PIG-script.pig-d03daed  
2020-05-22 16:22:39,013 [main] INFO org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service add
```

Job Stats (time in seconds):

JobId	Maps	Reduces	MaxMapTime	MinMapTime	AvgMapTime	MedianMapTime	MaxReduceTime	MinReduceTime
job_1590011432575_0067	7	1	15	6	12	12	13	13

Input(s):
Successfully read 6369482 records (800288154 bytes) from: "/final_project/data/flightDelays_*

Output(s):
Successfully stored 1 records (29 bytes) in: "hdfs://sandbox.hortonworks.com:8020/tmp/temp-96195744/tmp1469156524"

Counters:

- Total records written : 1
- Total bytes written : 29
- Spillable Memory Manager spill count : 0
- Total bags proactively spilled: 42
- Total records proactively spilled: 34503794

Script Details

Script contents:

```

1 flightDelays = LOAD '/final_project/data/flightDelays_*'
2 USING org.apache.pig.piggybank.storage.CSVExcelStorage()
3 as (YEAR:int,
4 FL_DATE:chararray,
5 UNIQUE_CARRIER:chararray,
6 CARRIER:chararray,
7 FL_NUM:chararray,
8 ORIGIN_AIRPORT_ID:chararray,
9 ORIGIN:chararray,
10 ORIGIN_CITY_NAME:chararray,
11 ORIGIN_STATE_ABR:chararray,
12 DEST_AIRPORT_ID:chararray,
13 DEST:chararray,
14 DEST_CITY_NAME:chararray,
15 DEST_STATE_ABR:chararray,
16 DEP_DELAY_NEW:float,
17 ARR_DELAY:float,
18 ARR_DELAY_NEW:float,
```

Arguments:
This job was executed without arguments.

8.

flight_delays_udf.pig

```

flightDelays = LOAD '/final_project/data/flightDelays_*'
USING org.apache.pig.piggybank.storage.CSVExcelStorage()
as (YEAR:int,
FL_DATE:chararray,
UNIQUE_CARRIER:chararray,
CARRIER:chararray,
FL_NUM:chararray,
ORIGIN_AIRPORT_ID:chararray,
ORIGIN:chararray,
ORIGIN_CITY_NAME:chararray,
ORIGIN_STATE_ABR:chararray,
DEST_AIRPORT_ID:chararray,
DEST:chararray,
DEST_CITY_NAME:chararray,
DEST_STATE_ABR:chararray,
DEP_DELAY_NEW:float,
ARR_DELAY:float,
ARR_DELAY_NEW:float,
```

```
CARRIER_DELAY:float,  
WEATHER_DELAY:float,  
NAS_DELAY:float,  
SECURITY_DELAY:float,  
LATE_AIRCRAFT_DELAY:float);
```

```
REGISTER 'hdfs://sandbox.hortonworks.com:8020/final_project/flight_delay_udf.py'  
using jython as myudf;  
clean_c = filter flightDelays by CARRIER_DELAY == 1975.0;  
max_c_delay = foreach clean_c generate myudf.get_max(((0,'YEAR'),(1,'FL_DATE'),  
(2,'UNIQUE_CARRIER'),(3,'CARRIER'),(4,'FL_NUM'),(5,'ORIGIN_AIRPORT_ID'),  
(6, 'ORIGIN'),  
(7,'ORIGIN_CITY_NAME'),(8,'ORIGIN_STATE_ABR'),(9,'DEST_AIRPORT_ID'),  
(10,'DEST'),(11,'DEST_CITY_NAME'),(12,'DEST_STATE_ABR'),(13,'DEP_DELAY  
_NEW'),  
(14,'ARR_DELAY'),(15,'ARR_DELAY_NEW'),(16,'CARRIER_DELAY'),(17,'WEATH  
ER_DELAY'),  
(18,'NAS_DELAY'),(19,'SECURITY_DELAY'),(20,'LATE_AIRCRAFT_DELAY')));  
dump max_c_delay;
```

flight_delay_udf.py

```

%s:.1f \n \
%s:.1f \n \
%s:.1f \n \
%s:.1f \n \
%s:.1f \n' %(carrier,
maxdelay,data[0][1],data[0][0],data[1][1],data[1][0],data[2]
[1],data[2][0],data[3][1],data[3][0],data[4][1],data[4][0],data
[5][1],data[5][0],data[6][1],data[6][0],data[7][1],data[7][0],d
ata[8][1],data[8][0],data[9][1],data[9][0],data[10][1],data[10]
[0],data[11][1],data[11][0],data[12][1],data[12][0],data[13][1]
,data[13][0],data[14][1],data[14][0],data[15][1],data[15][0],da
ta[16][1],data[16][0],data[17][1],data[17][0],data[18][1],data[
18][0],data[19][1],data[19][0],data[20][1],data[20][0])
return dly

```

9.

```

flightDelays = LOAD '/final_project/data/flightDelays_'
USING org.apache.pig.piggybank.storage.CSVExcelStorage()
as (YEAR:int,
FL_DATE:chararray,
UNIQUE_CARRIER:chararray,
CARRIER:chararray,
FL_NUM:chararray,
ORIGIN_AIRPORT_ID:chararray,
ORIGIN:chararray,
ORIGIN_CITY_NAME:chararray,
ORIGIN_STATE_ABR:chararray,
DEST_AIRPORT_ID:chararray,
DEST:chararray,
DEST_CITY_NAME:chararray,
DEST_STATE_ABR:chararray,
DEP_DELAY_NEW:float,
ARR_DELAY:float,
ARR_DELAY_NEW:float,
CARRIER_DELAY:float,
WEATHER_DELAY:float,
NAS_DELAY:float,
SECURITY_DELAY:float,
LATE_AIRCRAFT_DELAY:float);
allTheDelays = foreach flightDelays generate FL_DATE, FL_NUM, CARRIER_DELAY,
WEATHER_DELAY, NAS_DELAY, SECURITY_DELAY, LATE_AIRCRAFT_DELAY;

```

```
theDelays = filter allTheDelays by NOT (CARRIER_DELAY is null or  
WEATHER_DELAY is null or NAS_DELAY is null or SECURITY_DELAY is null or  
LATE_AIRCRAFT_DELAY is null);  
STORE theDelays INTO '/final_project/theDelays';
```

The screenshot shows a web browser window with the URL `localhost:8080/#/main/views/PIG/1.0.0/PIG INSTANCE`. The page title is "Ambari - Sandbox". The main content area displays a completed Pig job titled "allTheDelays.pig - COMPLETED". The job details are as follows:

Job ID	StartedAt
job_1590011432575_0039	2020-05-21 17:36

Below the job details, there are two sections: "Results" and "Logs".

Results:

HadoopVersion PigVersion UserId StartedAt FinishedAt Features
2.7.3.2.5.0.0-1245 0.16.0.2.5.0.0-1245 yarn 2020-05-22 00:36:23 2020-05-22 00:37:02 FILTER

Success!

Job Stats (time in seconds):
Jobid Maps Reduces MaxMapTime MinMapTime AvgMapTime MedianMapTime MaxReduceTime MinReduceTime
job_1590011432575_0040 7 0 22 10 19 21 0 0 0 0 allTheDelays,f

Input(s):
Successfully read 6369482 records (800288154 bytes) from: "/final_project/data/flightDelays_**"

Output(s):
Successfully stored 1269277 records (46978677 bytes) in: "/final_project/theDelays"

Counters:

Ambari - Sandbox root@sandbox-~:/P - Shell In A | +

localhost:8080/#/main/views/PIG/1.0.0/PIG INSTANCE

Output(s):
Successfully stored 1269277 records (46978677 bytes) in: "/final_project/theDelays"

Counters:
Total records written : 1269277
Total bytes written : 46978677
Spillable Memory Manager spill count : 0
Total bags proactively spilled: 0
Total records proactively spilled: 0

Job DAG:
job_1590011432575_0040

2020-05-22 00:37:02,857 [main] INFO org.apache.hadoop.yarn.client.api.impl.TimelineClientImpl - Timeline service add

Script Details

Script contents:

```

1 flightDelays = LOAD '/final_project/data/flightDelays_*';
2 USING org.apache.pig.piggybank.storage.CSVExcelStorage()
3 as (YEAR:int,
4 FL_DATE:chararray,
5 UNIQUE_CARRIER:chararray,
6 CARRIER:chararray,
7 FL_NUM:chararray,
8 ORIGIN_AIRPORT_ID:chararray,
9 ORIGIN:chararray,
10 ORIGIN_CITY_NAME:chararray.

```

Arguments: ***

This job was executed without arguments.

Type here to search

Ambari - Sandbox root@sandbox-~:/P - Shell In A | +

localhost:8080/#/main/views/FILES/1.0.0/AUTO_FILES INSTANCE

Dashboard Services Hosts Alerts Admin admin

/ > final_project > theDelays Total: 7 files or folders

+ Select All New Folder Upload 0

Search in current directory... Q

Name	Size	Last Modified	Owner	Group	Permission
part-m-00000	6.8 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--
part-m-00001	8.2 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--
part-m-00002	6.7 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--
part-m-00003	6.3 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--
part-m-00004	8.0 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--
part-m-00005	3.7 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--
part-m-00006	5.1 MB	2020-05-21 17:36	admin	hdfs	-rw-r--r--

10.

a. In default database fd1_t:

The screenshot shows the Ambari Sandbox interface with the 'Upload Table' page open. The 'File type' is set to 'CSV', 'Database' to 'default', and 'Stored as' to 'ORC'. The 'HDFS Path' is set to '/final_project/theDelays/part-m-00000' and the 'Table name' is 'fd1_t'. The 'Contains endlines?' checkbox is unchecked. Below the form, there is a preview of the data with columns: FL_DATE, FL_NUM, CARRIER_DELAY, and WEATHER_DELAY. The data rows are as follows:

FL_DATE	FL_NUM	CARRIER_DELAY	WEATHER_DELAY
2013-04-10	3283	5.0	0.0
2013-04-20	3283	71.0	0.0
2013-04-26	3283	0.0	0.0
2013-04-07	3283	21.0	0.0
2013-04-07	3283	6.0	0.0
2013-04-06	3284	48.0	0.0
2013-04-08	3284	24.0	0.0

This screenshot is identical to the one above, showing the Ambari Sandbox interface with the 'Upload Table' page open. The configuration and data preview are the same, indicating no changes have been made between the two screenshots.

fd2_t:

Ambari - Sandbox

localhost:8080/#/main/views/HIVE/1.5.0/AUTO_HIVE_INSTANCE

Dashboard Services Hosts Alerts Admin

Hive Query Saved Queries History UDFs Upload Table

Upload from Local

File type: CSV

Database: default

Stored as: ORC

Upload from HDFS

HDFS Path: /final_project/theDelays/part-m-00001

Table name: fd2_t

Contains endlines?

FL_DATE FL_NUM CARRIER_DELAY WEATHER_DELAY

STRING STRING DOUBLE DOUBLE

FL_DATE	FL_NUM	CARRIER_DELAY	WEATHER_DELAY
2013-06-21	3189	0.0	0.0
2013-06-25	3189	0.0	0.0
2013-06-27	3189	0.0	0.0
2013-06-28	3189	0.0	0.0
2013-06-29	3189	51.0	0.0
2013-06-13	3191	0.0	0.0
2013-06-18	3191	0.0	0.0

Ambari - Sandbox

localhost:8080/#/main/views/HIVE/1.5.0/AUTO_HIVE_INSTANCE

Dashboard Services Hosts Alerts Admin

Hive Query Saved Queries History UDFs Upload Table

Upload from Local

File type: CSV

Database: default

Stored as: ORC

Upload from HDFS

HDFS Path: /final_project/theDelays/part-m-00001

Table name: fd2_t

Contains endlines?

_DELAY NAS_DELAY SECURITY_DELAY LATE_AIRCRAFT_DELAY

DOUBLE DOUBLE DOUBLE DOUBLE

_DELAY	NAS_DELAY	SECURITY_DELAY	LATE_AIRCRAFT_DELAY
106.0	0.0	5.0	
20.0	0.0	137.0	
21.0	0.0	0.0	
38.0	0.0	80.0	
0.0	0.0	22.0	
0.0	0.0	36.0	
0.0	0.0	58.0	

In FPdb database:

`SELECT * FROM fd1_t LIMIT 10;`

A screenshot of a web browser window titled "localhost:8080/#/main/views/HIVE/1.5.0/AUTO_HIVE_INSTANCE". The browser has multiple tabs open, including "root@sandbox:~/FP - Shell In A..." and "Ambari - Sandbox". The main content area displays the results of a query. At the top, there are buttons for "Execute", "Explain", and "Save as...". On the right, there is a "New Worksheet" button. Below these, a header bar shows "Query Process Results (Status: SUCCEEDED)" and a "Save results..." dropdown. There are two tabs: "Logs" (selected) and "Results". A "Filter columns..." input field and "previous" and "next" buttons are also present. The results table has columns: fd1_t.fl_date, fd1_t.fl_num, fd1_t.carrier_delay, fd1_t.weather_delay, fd1_t.nas_delay, fd1_t.security_delay, and fd1_t.late_aircraft_delay. The data rows are as follows:

fd1_t.fl_date	fd1_t.fl_num	fd1_t.carrier_delay	fd1_t.weather_delay	fd1_t.nas_delay	fd1_t.security_delay	fd1_t.late_aircraft_delay
2013-04-10	3283	5.0	0.0	45.0	0.0	16.0
2013-04-20	3283	71.0	0.0	0.0	0.0	5.0
2013-04-26	3283	0.0	0.0	17.0	0.0	0.0
2013-04-07	3283	21.0	0.0	0.0	0.0	0.0
2013-04-07	3283	6.0	0.0	2.0	0.0	7.0
2013-04-06	3284	48.0	0.0	23.0	0.0	0.0
2013-04-08	3284	24.0	0.0	10.0	0.0	0.0
2013-04-09	3284	100.0	0.0	0.0	0.0	0.0
2013-04-11	3284	23.0	0.0	9.0	0.0	0.0
2013-04-12	3284	0.0	0.0	35.0	0.0	0.0

A second screenshot of a web browser window titled "localhost:8080/#/main/views/HIVE/1.5.0/AUTO_HIVE_INSTANCE". The browser interface is identical to the first one, showing the same tabs and query results. The results table has columns: fl_num, fd1_t.carrier_delay, fd1_t.weather_delay, fd1_t.nas_delay, fd1_t.security_delay, and fd1_t.late_aircraft_delay. The data rows are as follows:

fl_num	fd1_t.carrier_delay	fd1_t.weather_delay	fd1_t.nas_delay	fd1_t.security_delay	fd1_t.late_aircraft_delay
5.0	0.0	45.0	0.0	16.0	
71.0	0.0	0.0	0.0	0.0	5.0
0.0	0.0	17.0	0.0	0.0	
21.0	0.0	0.0	0.0	0.0	
6.0	0.0	2.0	0.0	0.0	7.0
48.0	0.0	23.0	0.0	0.0	
24.0	0.0	10.0	0.0	0.0	
100.0	0.0	0.0	0.0	0.0	
23.0	0.0	9.0	0.0	0.0	
0.0	0.0	35.0	0.0	0.0	

SELECT * FROM fd2_t LIMIT 10;

The screenshot shows a web-based interface for querying a database. At the top, there is a navigation bar with various links and icons. Below the navigation bar is a toolbar with buttons for 'Execute', 'Explain', 'Save as...', and 'New Worksheet'. The main area displays the 'Query Process Results' with a status of 'SUCCEEDED'. There are two tabs: 'Logs' and 'Results'. The 'Results' tab is selected, showing a table with the following data:

fd2_t.fl_date	fd2_t.fl_num	fd2_t.carrier_delay	fd2_t.weather_delay	fd2_t.nas_delay	fd2_t.security_delay	fd2_t.late_aircraft_delay
2013-06-21	3189	0.0	0.0	106.0	0.0	5.0
2013-06-25	3189	0.0	0.0	20.0	0.0	137
2013-06-27	3189	0.0	0.0	21.0	0.0	0.0
2013-06-28	3189	0.0	0.0	38.0	0.0	80.0
2013-06-29	3189	51.0	0.0	0.0	0.0	22.0
2013-06-13	3191	0.0	0.0	0.0	0.0	36.0
2013-06-18	3191	0.0	0.0	0.0	0.0	58.0
2013-06-21	3191	0.0	0.0	302.0	0.0	0.0
2013-06-25	3191	0.0	0.0	32.0	0.0	22.0
2013-06-26	3191	0.0	0.0	54.0	0.0	88.0

This screenshot shows the same web-based interface as the first one, but with a different query result. The table now has an additional column at the end:

fl_num	fd2_t.carrier_delay	fd2_t.weather_delay	fd2_t.nas_delay	fd2_t.security_delay	fd2_t.late_aircraft_delay
0.0	0.0	106.0	0.0	5.0	
0.0	0.0	20.0	0.0	137.0	
0.0	0.0	21.0	0.0	0.0	
0.0	0.0	38.0	0.0	80.0	
51.0	0.0	0.0	0.0	22.0	
0.0	0.0	0.0	0.0	36.0	
0.0	0.0	302.0	0.0	0.0	
0.0	0.0	32.0	0.0	22.0	
0.0	0.0	54.0	0.0	88.0	



b.

```
CREATE TABLE IF NOT EXISTS FPdb.fd3_t (
FL_DATE string,
FL_NUM string,
CARRIER_DELAY double,
WEATHER_DELAY double,
NAS_DELAY double,
SECURITY_DELAY double,
LATE_AIRCRAFT_DELAY double)
row format delimited
fields terminated by '\t'
lines terminated by '\n';

load data inpath '/final_project/theDelays/part-m-00002'
overwrite into table FPdb.fd3_t;

SELECT * FROM FPdb.fd3_t LIMIT 10;
```

The screenshot shows the Ambari Sandbox interface. At the top, there are several tabs: 'Ambari - Sandbox', 'localhost:8080//main/views/HIVE/1.5.0/AUTO_HIVE_INSTANCE', 'Shell In A Box', and others. Below the tabs, the main menu includes 'Dashboard', 'Services', 'Hosts', 'Alerts', 'Admin', and a user dropdown for 'admin'. The central area has tabs for 'Hive', 'Query', 'Saved Queries', 'History', 'UDFs', and 'Upload Table'. On the left, the 'Database Explorer' shows databases like 'default', 'fd1', 'fd2', and 'fd3'. The right side features the 'Query Editor' with a SQL tab selected. A query is being typed into the editor:

```
Q10b.hive x fd1_t sample x fd2_t sample x
1 CREATE TABLE IF NOT EXISTS FPdb.fd3_t (
2   FL_DATE string,
3   TAIL_NUM string,
4   CARRIER_DELAY double,
5   WEATHER_DELAY double,
6   NAS_DELAY double,
7   SECURITY_DELAY double,
8   LATE_AIRCRAFT_DELAY double)
9 rowFormat delimited
10 fields terminated by '\t'
11 lines terminated by '\n';
12
13 load data inpath '/final_project/theDelays/part-m-00002'
14 overwrite into table FPdb.fd3_t;
15
16 SELECT * FROM FPdb.fd3_t LIMIT 10;
```

Buttons at the bottom of the editor include 'Execute', 'Explain', 'Save as...', and 'New Worksheet'.

The screenshot shows the Ambari Sandbox interface again. The top navigation and sidebar are identical to the previous screenshot. The central area displays the 'Query Process Results (Status: SUCCEEDED)'. The results are shown in a table with two tabs: 'Logs' (selected) and 'Results'. The table has columns: 'fd3_t.fl_date', 'fd3_t.fl_num', 'fd3_t.carrier_delay', 'fd3_t.weather_delay', 'fd3_t.nas_delay', 'fd3_t.security_delay', and 'fd3_t'. The data is as follows:

fd3_t.fl_date	fd3_t.fl_num	fd3_t.carrier_delay	fd3_t.weather_delay	fd3_t.nas_delay	fd3_t.security_delay	fd3_t
2013-08-07	3283	0.0	0.0	27.0	0.0	0.0
2013-08-19	3283	0.0	0.0	0.0	0.0	137
2013-08-20	3283	0.0	0.0	17.0	0.0	0.0
2013-08-24	3284	133.0	0.0	0.0	0.0	5.0
2013-08-01	3284	4.0	0.0	0.0	0.0	74.0
2013-08-06	3284	0.0	0.0	67.0	0.0	0.0
2013-08-07	3284	0.0	0.0	61.0	0.0	74.0
2013-08-09	3284	0.0	0.0	24.0	0.0	2.0
2013-08-24	3284	0.0	0.0	0.0	0.0	100
2013-08-03	3285	3.0	0.0	0.0	0.0	23.0

The screenshot shows the Ambari Sandbox interface. The top navigation and sidebar are identical to the previous screenshots. The central area displays the 'Apache License, Version 2.0' notice. The notice states: 'Licensed under the Apache License, Version 2.0. See the file LICENSE for details.' Below the notice, the Ambari interface is visible with its standard top bar and sidebar.

fl_num	fd3_t.carrier_delay	fd3_t.weather_delay	fd3_t.nas_delay	fd3_t.security_delay	fd3_t.late_aircraft_delay
0.0	0.0	27.0	0.0	0.0	
0.0	0.0	0.0	0.0	137.0	
0.0	0.0	17.0	0.0	0.0	
133.0	0.0	0.0	0.0	5.0	
4.0	0.0	0.0	0.0	74.0	
0.0	0.0	67.0	0.0	0.0	
0.0	0.0	61.0	0.0	74.0	
0.0	0.0	24.0	0.0	2.0	
0.0	0.0	0.0	0.0	100.0	
3.0	0.0	0.0	0.0	23.0	



C.

```
create table FPdb.fd4_t
LIKE FPdb.fd3_t;
```

```
load data inpath '/final_project/theDelays/part-m-00003'
overwrite into table FPdb.fd4_t;
```

```
create table FPdb.fd5_t
LIKE FPdb.fd3_t;
```

```
load data inpath '/final_project/theDelays/part-m-00004'
overwrite into table FPdb.fd5_t;
```

```
create table FPdb.fd6_t
LIKE FPdb.fd3_t;
```

```
load data inpath '/final_project/theDelays/part-m-00005'
overwrite into table FPdb.fd6_t;
```

```
create table FPdb.fd7_t
LIKE FPdb.fd3_t;
```

```
load data inpath '/final_project/theDelays/part-m-00006'
overwrite into table FPdb.fd7_t;
```

```
SELECT * FROM FPdb.fd7_t LIMIT 10;
```

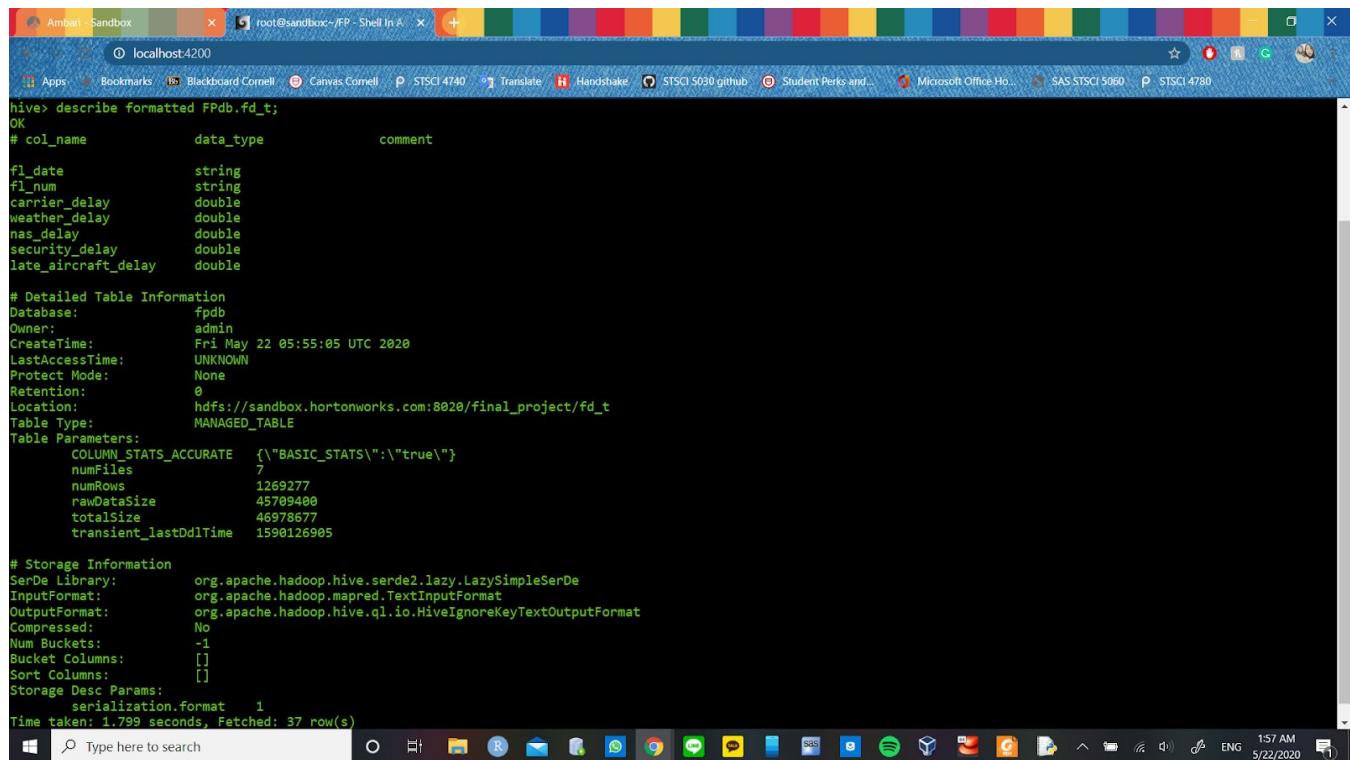
fd7_t.fl_date	fd7_t.fl_num	fd7_t.carrier_delay	fd7_t.weather_delay	fd7_t.nas_delay	fd7_t.security_delay	fd7_t.late_aircraft_delay
2013-12-02	2900	0.0	0.0	0.0	0.0	36.0
2013-12-06	2900	10.0	0.0	0.0	0.0	11.0
2013-12-10	2900	0.0	0.0	4.0	0.0	83.0
2013-12-11	2900	24.0	0.0	15.0	0.0	26.0
2013-12-14	2900	0.0	0.0	17.0	0.0	13.0
2013-12-15	2900	0.0	0.0	0.0	0.0	45.0
2013-12-16	2900	0.0	0.0	0.0	0.0	82.0
2013-12-18	2900	6.0	0.0	0.0	0.0	17.0
2013-12-22	2900	0.0	0.0	0.0	0.0	46.0
2013-12-23	2900	26.0	0.0	17.0	0.0	4.0

fl_num	fd7_t.carrier_delay	fd7_t.weather_delay	fd7_t.nas_delay	fd7_t.security_delay	fd7_t.late_aircraft_delay
0.0	0.0	0.0	0.0	36.0	
10.0	0.0	0.0	0.0	11.0	
0.0	0.0	4.0	0.0	83.0	
24.0	0.0	15.0	0.0	26.0	
0.0	0.0	17.0	0.0	13.0	
0.0	0.0	0.0	0.0	45.0	
0.0	0.0	0.0	0.0	82.0	
6.0	0.0	0.0	0.0	17.0	
0.0	0.0	0.0	0.0	46.0	
26.0	0.0	17.0	0.0	4.0	

fl_num	fd7_t.carrier_delay	fd7_t.weather_delay	fd7_t.nas_delay	fd7_t.security_delay	fd7_t.late_aircraft_delay
0.0	0.0	0.0	0.0	36.0	
10.0	0.0	0.0	0.0	11.0	
0.0	0.0	4.0	0.0	83.0	
24.0	0.0	15.0	0.0	26.0	
0.0	0.0	17.0	0.0	13.0	
0.0	0.0	0.0	0.0	45.0	
0.0	0.0	0.0	0.0	82.0	
6.0	0.0	0.0	0.0	17.0	
0.0	0.0	0.0	0.0	46.0	
26.0	0.0	17.0	0.0	4.0	

11.

```
CREATE TABLE IF NOT EXISTS FPdb.fd_t AS
SELECT * FROM FPdb.fd1_t
UNION ALL
SELECT * FROM FPdb.fd2_t
UNION ALL
SELECT * FROM FPdb.fd3_t
UNION ALL
SELECT * FROM FPdb.fd4_t
UNION ALL
SELECT * FROM FPdb.fd5_t
UNION ALL
SELECT * FROM FPdb.fd6_t
UNION ALL
SELECT * FROM FPdb.fd7_t;
```



A screenshot of a Windows desktop environment. The taskbar at the bottom shows various icons for apps like File Explorer, Edge, and other system tools. Above the taskbar is a window titled "localhost:4200" which contains a terminal session. The terminal output is a detailed description of a Hive table named "fd_t". The output includes columns, data types, and comments; detailed table information such as database (fpdb), owner (admin), creation time (Fri May 22 05:55:05 UTC 2020), and location (hdfs://sandbox.hortonworks.com:8020/final_project/fd_t); table parameters including column statistics and file counts; storage information like serde library (org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe) and input/output formats; and finally, execution metrics (Time taken: 1.799 seconds, Fetched: 37 row(s)).

```
hive> describe formatted FPdb.fd_t;
OK
# col_name          data_type          comment
f1_date            string
f1_num             string
carrier_delay      double
weather_delay      double
nas_delay          double
security_delay     double
late_aircraft_delay double

# Detailed Table Information
Database:          fpdb
Owner:              admin
CreateTime:        Fri May 22 05:55:05 UTC 2020
LastAccessTime:    UNKNOWN
Protect Mode:      None
Retention:         0
Location:          hdfs://sandbox.hortonworks.com:8020/final_project/fd_t
Table Type:        MANAGED_TABLE
Table Parameters:
  COLUMN_STATS_ACCURATE  {"BASIC_STATS":"true"}
  numFiles                7
  numRows                 1269277
  rawDataSize             45709400
  totalSize                46978677
  transient_lastDdlTime   1590126905

# Storage Information
Serde Library:      org.apache.hadoop.hive.serde2.lazy.LazySimpleSerDe
InputFormat:         org.apache.hadoop.mapred.TextInputFormat
OutputFormat:        org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat
Compressed:         No
Num Buckets:        -1
Bucket Columns:     []
Sort Columns:        []
Storage Desc Params:
  serialization.format  1
Time taken: 1.799 seconds, Fetched: 37 row(s)
```

12.

```
select max(CARRIER_DELAY) max_carrier_delay,
       max(WEATHER_DELAY) max_weather_delay,
       max(NAS_DELAY) max_nas_delay,
       max(SECURITY_DELAY) max_security_delay,
```

```
max(LATE_AIRCRAFT_DELAY) max_late_aircraft_delay  
from FPdb.fd_t;
```

The screenshot shows a web-based interface for managing a Hadoop cluster. At the top, there are tabs for 'Ambari - Sandbox' and 'root@sandbox~/FP - Shell In A...'. Below the tabs is a horizontal bar with various icons. The main area contains a search bar, a sidebar with 'Databases' (including default, flightsdb, foodmart, fpdb, stocksdb, xademo), and a code editor window. The code editor contains the following SQL query:

```
1 select max(CARRIER_DELAY) max_carrier_delay,  
2 max(WEATHER_DELAY) max_weather_delay,  
3 max(NAS_DELAY) max_nas_delay,  
4 max(SECURITY_DELAY) max_security_delay,  
5 max(LATE_AIRCRAFT_DELAY) max_late_aircraft_delay  
6 from FPdb.fd_t;
```

Below the code editor are buttons for 'Execute', 'Explain', and 'Save as...'. To the right, there are icons for 'SQL', 'TEZ', and 'New Worksheet'. The results section is titled 'Query Process Results (Status: SUCCEEDED)' and shows a table with the following data:

max_carrier_delay	max_weather_delay	max_nas_delay	max_security_delay	max_late_aircraft_delay
1975.0	1591.0	1287.0	573.0	1182.0

```
select round(avg(CARRIER_DELAY), 2) mean_carrier_delay,  
round(avg(WEATHER_DELAY), 2) mean_weather_delay,  
round(avg(NAS_DELAY), 2) mean_nas_delay,  
round(avg(SECURITY_DELAY), 2) mean_security_delay,  
round(avg(LATE_AIRCRAFT_DELAY), 2) mean_late_aircraft_delay  
from FPdb.fd_t;
```

The screenshot shows the Apache Ambari Sandbox interface. A query is being run in a worksheet:

```

1 select round(avg(CARRIER_DELAY), 2) mean_carrier_delay,
2 round(avg(WEATHER_DELAY), 2) mean_weather_delay,
3 round(avg(NAS_DELAY), 2) mean_nas_delay,
4 round(avg(SECURITY_DELAY), 2) mean_security_delay,
5 round(avg(LATE_AIRCRAFT_DELAY), 2) mean_late_aircraft_delay
6 from FPdb.fd_t;

```

The results of the query are displayed in a table:

mean_carrier_delay	mean_weather_delay	mean_nas_delay	mean_security_delay	mean_late_aircraft_delay
16.65	2.34	13.73	0.08	23.87

13.

```
hive> use FPdb;
```

```
hive> create view averageDelays_v as
    > select round(avg(CARRIER_DELAY), 2) mean_carrier_delay,
    > round(avg(WEATHER_DELAY), 2) mean_weather_delay,
    > round(avg(NAS_DELAY), 2) mean_nas_delay,
    > round(avg(SECURITY_DELAY), 2) mean_security_delay,
    > round(avg(LATE_AIRCRAFT_DELAY), 2)
mean_late_aircraft_delay
    > from fd_t;
```

FindMaxAverageDelayType.py

```
#!/usr/bin/python
```

```
import sys
```

```
def
findMaxAverageDelayType(carrier,weather,nas,security,late_aircr
aft):
```

```

delayTypes = {0:'carrier_delay', 1:'weather_delay',
2:'nas_delay', 3:'security_de
lay', 4:'late_aircraft_delay'}
delays = [carrier,weather,nas,security,late_aircraft]
maxAvgValue = max(delays)
return delayTypes[delays.index(maxAvgValue)], maxAvgValue

for line in sys.STDIN:
    print 'The delay category with the longest average delays is
%s; the average delay
time is %.2f'
%(findMaxAverageDelayType(float(carrier),float(weather),float(n
as),\
float(security),float(late_aircraft))[0],
findMaxAverageDelayType(float(carrier),float(weather),float(nas
),\float(security),float(late_aircraft))[1])

hive> ADD IN /FP/FindMaxAverageDelayType.py;
hive> SELECT
TRANSFORM(mean_carrier_delay,mean_weather_delay,mean_nas_delay,
mean_se
curity_delay, mean_late_aircraft_delay)
> USING 'python /FP/FindMaxAverageDelayType.py'
> AS maxAvgDelayType
> FROM FPdb.averageDelays_v;

```

14.

```

SELECT
ROUND(((count()*(sum(CARRIER_DELAY*WEATHER_DELAY)))-
(sum(CARRIER_DELAY)*(sum(WEATHER_DELAY)))/
sqrt(((count()*sum(pow(CARRIER_DELAY,2)))-(pow(sum(CARRIER_DELAY),2)))*
((count()*sum(pow(WEATHER_DELAY,2)))-(pow(sum(WEATHER_DELAY),2))))), 4)
w_c
FROM FPdb.fd_t;
```

A screenshot of a web-based SQL interface. The top navigation bar shows tabs for 'Arbami - Sandbox' and 'localhost:8080#/main/views/HIVE/1.5.0/AUTO_HIVE_INSTANCE'. Below the navigation is a toolbar with various icons. The main area has two panes: a 'Worksheet' pane on the left containing a code editor with the following SQL query:

```
1 SELECT
2 ROUND(((count(*)*(sum(CARRIER_DELAY*WEATHER_DELAY)) - (sum(CARRIER_DELAY)*(sum(WEATHER_DELAY))))/
3 sqrt(((count(*)*sum(pow(CARRIER_DELAY,2)))-(pow(sum(CARRIER_DELAY),2))))*
4 ((count(*)*sum(pow(WEATHER_DELAY,2)))-(pow(sum(WEATHER_DELAY),2)))), 4) w_c
5 FROM FPdb.fd_t;
```

The 'Worksheet' pane includes buttons for 'Execute', 'Explain', and 'Save as...'. To the right is a sidebar with icons for 'SQL', 'TEZ', and 'Logs'. The bottom section is a 'Query Process Results' panel with tabs for 'Logs' and 'Results'. The 'Results' tab shows a single row with the column 'w_c' and value '-0.0454'. A progress bar at the top of this panel indicates '100%'. The status is 'SUCCEEDED'.

```
select round(CORR(CARRIER_DELAY,WEATHER_DELAY),4) c_w,
round(CORR(CARRIER_DELAY, NAS_DELAY),4) c_n,
round(CORR(CARRIER_DELAY, SECURITY_DELAY),4) c_s,
round(CORR(CARRIER_DELAY, LATE_AIRCRAFT_DELAY),4) c_l,
round(CORR(WEATHER_DELAY, NAS_DELAY),4) w_n,
round(CORR(WEATHER_DELAY, SECURITY_DELAY),4) w_s,
round(CORR(WEATHER_DELAY, LATE_AIRCRAFT_DELAY),4) w_l,
round(CORR(NAS_DELAY,SECURITY_DELAY),4) n_s,
round(CORR(NAS_DELAY,LATE_AIRCRAFT_DELAY),4) n_l,
round(CORR(SECURITY_DELAY, LATE_AIRCRAFT_DELAY),4) s_l
from FPdb.fd_t;
```

Arbmi - Sandbox localhost:8080/#/main/views/HIVE/1.5.0/AUTO.HIVE.INSTANCE

Apps Bookmarks Blackboard Cornell Canvas Cornell STSCI 4740 Translate Handshake STSCI 5030 GitHub Student Perks and... Microsoft Office Ho... SAS STSCI 5060 STSCI 4780

Default Worksheet (1)

Search tables... Databases

Databases

- default
- flightsdbs
- goodmart
- fpdb
- stocksdb
- xademo

```
1 select round(CORR(CARRIER_DELAY,WEATHER_DELAY),4) c_w,
2 round(CORR(CARRIER_DELAY, NAS_DELAY),4) c_n,
3 round(CORR(CARRIER_DELAY, SECURITY_DELAY),4) c_s,
4 round(CORR(CARRIER_DELAY, LATE_AIRCRAFT_DELAY),4) c_l,
5 round(CORR(WEATHER_DELAY, NAS_DELAY),4) w_n,
6 round(CORR(WEATHER_DELAY, SECURITY_DELAY),4) w_s,
7 round(CORR(WEATHER_DELAY, LATE_AIRCRAFT_DELAY),4) w_l,
8 round(CORR(NAS_DELAY,SECURITY_DELAY),4) n_s,
9 round(CORR(NAS_DELAY,LATE_ATRCRAFT_DELAY),4) n_l,
10 round(CORR(SECURITY_DELAY, LATE_AIRCRAFT_DELAY),4) s_l
11 from FPdb.fdp_t;
```

Execute Explain Save as... New Worksheet

100%

Query Process Results (Status: SUCCEEDED) Save results... ▾

Logs Results

Filter columns previous next

c_w	c_n	c_s	c_l	w_n	w_s	w_l	n_s	n_l	s_l
-0.0454	-0.1142	-0.0103	-0.1217	-8.0E-4	-0.004	-0.0235	-0.0094	-0.1486	-0.0095

