Assignment 2

Submited by: Praneeth Krishna, Sakshi Arora, Prateek Gangwal

Kafka Implementations

Note: Since Kafka is not running in EMR 4.0. We have installed Spark 1.4.0 and kafka_2.10-0.8.2.1.tgz (asc, md5) on EMR 3.8

I. Scala

Step 1: Install Kafka and Spark.

Step 2: Lets start Kafka and see if its working.
All below commands needs to be run from kafka directory.

a. Start Zookeeper : bin/zookeeper-server-start.sh config/zookeeper.properties
Zookeeper starts at localhost:2181

b. Start Kafka Broker : bin/kakfa-server-start.sh config/server.properties
KafkaBroker starts on localhost:9092

c. Create Kafka Topic:
bin/kafka-topic.sh --create --zookeeper localhost:2181 –replication- factor 1 --partitions 1 --topic today

Creates a topic by name today

Step 3: Lets run spark streaming with kafka

a. Run the producer :
bin/run-example org.apache.spark.examples.streaming.KafkaWordCountProducer
localhost:9092 today 10 5

This runs the producer

b. Run the word Count :
bin/run-example org.apache.spark.examples.streaming.KafkaWordCount
localhost:2181 myconsumergroup today 1

```
15/08/08 03:22:37 WARN AppInfo$: Can't read Kafka version from MANIFEST.MF. Possible cause: java.lang.NullPointerException
15/08/08 03:22:37 WARN BlockManager: Block input-0-1439004157600 replicated to only 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 1439004158000 ms
-------------------------------------------
(4,11)
(8,6)
(6,5)
(0,2)
(2,7)
(7,3)
(5,4)
(9,5)
(3,3)
(1,4)

15/08/08 03:22:38 WARN BlockManager: Block input-0-1439004158600 replicated to only 0 peer(s) instead of 1 peers
15/08/08 03:22:39 WARN BlockManager: Block input-0-1439004159600 replicated to only 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 1439004160000 ms
-------------------------------------------
(4,24)
(8,13)
(6,16)
(0,15)
(2,15)
(7,9)
(5,14)
(9,15)
(3,11)
(1,18)
```

II. Python

Note: We need to download respective .jar file (PFA .jar file)

Step 1: Install Kafka and Spark.

Step 2. Lets start Kafka and see if its working.
All below commands needs to be run from kafka directory.

a. Start Zookeeper : bin/zookeeper-server-start.sh config/zookeeper.properties
Zookeeper starts at localhost:2181

b. Start Kafka Broker : bin/kakfa-server-start.sh config/server.properties
KafkaBroker starts on localhost:9092

c. Create Kafka Topic:
bin/kafka-topic.sh --create --zookeeper localhost:2181 –replication- factor 1 --partitions 1 --topic today

Creates a topic by name today

Step 3: Lets run spark streaming with kafka

a. Run the producer :
bin/run-example org.apache.spark.examples.streaming.KafkaWordCountProducer localhost:9092 today 10 5

This runs the producer
b. Run the word count example using the jar.
bin/spark-submit --jars /home/hadoop/spark-streaming-kafka-assembly_2.10-1.4.0.jar examples/src/main/python/streaming/kafka_wordcount.py localhost:2181 today

```
-------------------------------------------
Time: 2015-08-08 03:25:48
-------------------------------------------
(u'1', 5)
(u'9', 2)
(u'5', 5)
(u'0', 8)
(u'4', 8)
(u'8', 4)
(u'3', 11)
(u'7', 2)
(u'2', 1)
(u'6', 4)
()
15/08/08 03:25:48 INFO JobScheduler: Finished job streaming job 1439004348000 ms.0 from job set of time 1439004348000 ms
15/08/08 03:25:48 INFO JobScheduler: Total delay: 0.368 s for time 1439004348000 ms (execution: 0.338 s)
15/08/08 03:25:48 INFO PythonRDD: Removing RDD 38 from persistence list
15/08/08 03:25:48 INFO BlockRDD: Removing RDD 33 from persistence list
15/08/08 03:25:48 INFO BlockManager: Removing RDD 38
```

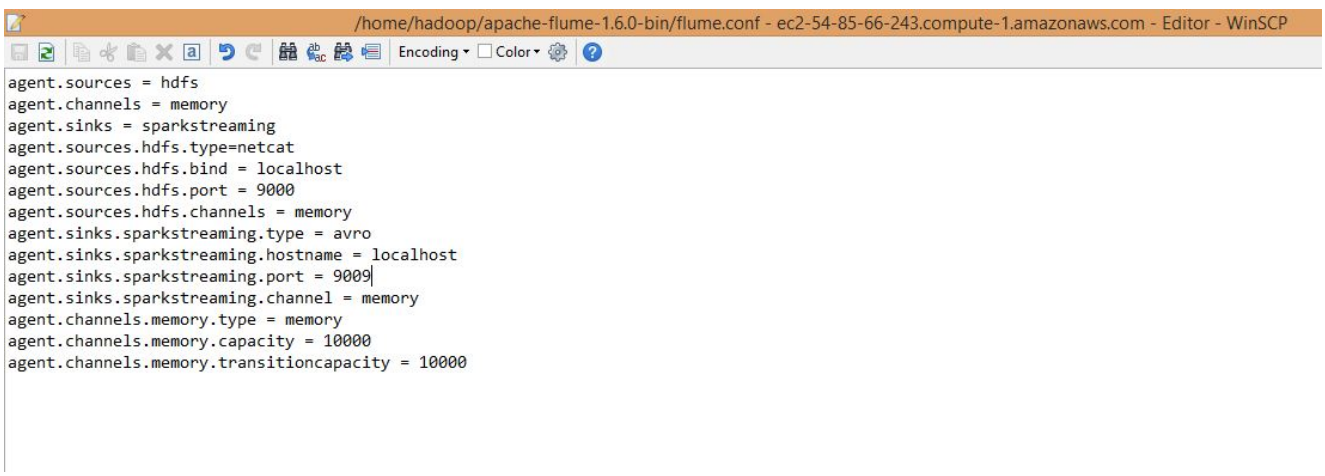
Flume Implementation

I. Scala

Step 1: Download Flume.

Step 2: Change the config file as follows:
agent.sources = hdfs
agent.channels = memory
agent.sinks = sparkstreaming
agent.sources.hdfs.type=netcat
agent.sources.hdfs.bind = localhost
agent.sources.hdfs.port = 9000
agent.sources.hdfs.channels = memory
agent.sinks.sparkstreaming.type = avro
agent.sinks.sparkstreaming.hostname = localhost
agent.sinks.sparkstreaming.port = 9009
agent.sinks.sparkstreaming.channel = memory
agent.channels.memory.type = memory
agent.channels.memory.capacity = 10000
agent.channels.memory.transitioncapacity = 10000

Note: We have open the port 9000 and sent all the information using this port.



Step 3: Start the Flume agent:
bin/flume-ng agent --conf conf --conf-file flume.conf --name agent -Dflume.root.avro=INFO,console

```
[hadoop@ip-172-31-12-167 flume]$ ls -ltr
total 152
-rw-r--r--  1 hadoop hadoop  1585 May  8 19:02 RELEASE-NOTES
-rw-r--r--  1 hadoop hadoop  1779 May  8 19:02 README
-rw-r--r--  1 hadoop hadoop   249 May  8 19:02 NOTICE
-rw-r--r--  1 hadoop hadoop 25903 May  8 19:02 LICENSE
-rw-r--r--  1 hadoop hadoop  6172 May  8 19:02 DEVNOTES
-rw-r--r--  1 hadoop hadoop 69856 May  8 19:02 CHANGELOG
drwxr-xr-x 10 hadoop hadoop  4096 May 11 18:14 docs
drwxrwxr-x  2 hadoop hadoop  4096 Aug  7 21:49 lib
drwxrwxr-x  2 hadoop hadoop  4096 Aug  7 21:49 tools
drwxr-xr-x  2 hadoop hadoop  4096 Aug  7 21:49 bin
drwxr-xr-x  2 hadoop hadoop  4096 Aug  7 21:51 conf
drwxrwxr-x  2 hadoop hadoop  4096 Aug  7 21:51 logs
-rw-r--r--  1 root   root     577 Aug  7 21:55 flume-conf.properties
-rw-r--r--  1 hadoop hadoop   576 Aug  7 22:00 flume.conf
[hadoop@ip-172-31-12-167 flume]$
[hadoop@ip-172-31-12-167 flume]$
[hadoop@ip-172-31-12-167 flume]$ bin/flume-ng agent --conf conf --conf-file flume.conf --name agent -Dflume.root.avro=INFO,console
```

Step 4: Run the example:
bin/run-example org.apache.spark.examples.streaming.FlumeEventCount localhost 9009

```
-------------------------------------------
Time: 1438986020000 ms
-------------------------------------------
Received 20 flume events.

-------------------------------------------
Time: 1438986022000 ms
-------------------------------------------
Received 0 flume events.

-------------------------------------------
Time: 1438986024000 ms
-------------------------------------------
Received 0 flume events.
```

II. Python

Step 1: Start the server
To run this on your local machine, you need to first run a Netcat server: $ nc -lk 9999
Step 2: Run the network word count example
$ bin/run-example org.apache.spark.examples.streaming.NetworkWordCount localhost 9999

```
15/08/07 22:56:23 INFO Executor: Finished task 198.0 in stage 87.0 (TID 2697). 1993 bytes result sent to driver
15/08/07 22:56:23 INFO TaskSetManager: Finished task 198.0 in stage 87.0 (TID 2697) in 4 ms on localhost (199/199)
15/08/07 22:56:23 INFO TaskSchedulerImpl: Removed TaskSet 87.0, whose tasks have all completed, from pool
15/08/07 22:56:23 INFO DAGScheduler: ResultStage 87 (showString at NativeMethodAccessorImpl.java:-2) finished in 0.388 s
15/08/07 22:56:23 INFO DAGScheduler: Job 61 finished: showString at NativeMethodAccessorImpl.java:-2, took 0.401236 s
+--------------------+-----+
|                word|total|
+--------------------+-----+
| errorRatePerZipCode|    1|
|        Serializable|    2|
|          Foundation|    1|
|              BASIS,|    1|
|             String)|    1|
|                The|    1|
|             String,|    1|
|         permissions|    1|
|                more|    1|
|               class|    1|
|                with|    4|
|                 not|    1|
|org.apache.spark....|    1|
|     parts(1).toInt,|    1|
|            PageView|    2|
|               under|    4|
|                10`|    1|
|       fromString(in|    1|
|                 //|    1|
|              NOTICE|    1|
+--------------------+-----+

15/08/07 22:56:23 INFO JobScheduler: Finished job streaming job 1438988182000 ms.0 from job set of time 1438988182000 ms
15/08/07 22:56:23 INFO JobScheduler: Total delay: 1.059 s for time 1438988182000 ms (execution: 1.053 s)
15/08/07 22:56:23 INFO JobScheduler: Starting job streaming job 1438988183000 ms.0 from job set of time 1438988183000 ms
```

Kinesis Implementation

Note: We have installed Spark 1.3.1 on EMR 3.8

I. Scala

Step 1: Run the generator
$ bin/run-example org.apache.spark.examples.streaming.clickstream.PageViewGenerator 44444 10

```
hadoop@ip-172-31-24-197:~/spark
[hadoop@ip-172-31-24-197 spark]$ bin/run-example org.apache.spark.examples.streaming.clickstream.PageViewGenerator 44444 10
Listening on port: 44444
Got client connected from: /127.0.0.1
Got client connected from: /127.0.0.1
```

Step 2: Process the generated stream (To run error rate per zip code)
$ bin/run-example \org.apache.spark.examples.streaming.clickstream.PageViewStream errorRatePerZipCode localhost 44444

```
[hadoop@ip-172-31-24-197 ~]$ cd spark/
[hadoop@ip-172-31-24-197 spark]$ bin/run-example org.apache.spark.examples.streaming.clickstream.PageViewStream errorRatePerZipCode localhost 44444
15/08/07 19:19:26 INFO spark.SparkContext: Running Spark version 1.3.1
15/08/07 19:19:26 WARN spark.SparkConf:
SPARK_CLASSPATH was detected (set to '/home/hadoop/spark/conf:/home/hadoop/conf:/home/hadoop/spark/classpath/emr/*:/home/hadoop/spark/classpath/emrfs/*:/home/hadoop/sha
re/hadoop/common/lib/*:/home/hadoop/share/hadoop/common/lib/hadoop-lzo.jar').
This is deprecated in Spark 1.0+.

Please instead use:
 - ./spark-submit with --driver-class-path to augment the driver classpath
 - spark.executor.extraClassPath to augment the executor classpath

15/08/07 19:19:26 WARN spark.SparkConf: Setting 'spark.executor.extraClassPath' to '/home/hadoop/spark/conf:/home/hadoop/conf:/home/hadoop/spark/classpath/emr/*:/home/h
adoop/spark/classpath/emrfs/*:/home/hadoop/share/hadoop/common/lib/*:/home/hadoop/share/hadoop/common/lib/hadoop-lzo.jar' as a work-around.
15/08/07 19:19:26 WARN spark.SparkConf: Setting 'spark.driver.extraClassPath' to '/home/hadoop/spark/conf:/home/hadoop/conf:/home/hadoop/spark/classpath/emr/*:/home/had
oop/spark/classpath/emrfs/*:/home/hadoop/share/hadoop/common/lib/*:/home/hadoop/share/hadoop/common/lib/hadoop-lzo.jar' as a work-around.
15/08/07 19:19:27 INFO spark.SecurityManager: Changing view acls to: hadoop
15/08/07 19:19:27 INFO spark.SecurityManager: Changing modify acls to: hadoop
```

Note: We can also process the generated stream to run the following functions:

I. pageCounts
II. slidingPageCounts
III. errorRatePerZipCode
IV. activeUserCount
V. popularUsersSeen

```
-------------------------------------------
Time: 1438975178000 ms
-------------------------------------------
94117: **0.0625**
94709: 0.03846154

15/08/07 19:19:38 INFO scheduler.JobScheduler: Finished job streaming job 1438975178000 ms.0 from job set of time 1438975178000 ms
15/08/07 19:19:38 INFO rdd.MapPartitionsRDD: Removing RDD 31 from persistence list
15/08/07 19:19:38 INFO scheduler.JobScheduler: Total delay: 0.411 s for time 1438975178000 ms (execution: 0.395 s)
15/08/07 19:19:38 INFO storage.BlockManager: Removing RDD 31
15/08/07 19:19:38 INFO rdd.ShuffledRDD: Removing RDD 30 from persistence list
15/08/07 19:19:38 INFO storage.BlockManager: Removing RDD 30
15/08/07 19:19:38 INFO rdd.MapPartitionsRDD: Removing RDD 29 from persistence list
15/08/07 19:19:38 INFO storage.BlockManager: Removing RDD 29
15/08/07 19:19:38 INFO rdd.UnionRDD: Removing RDD 28 from persistence list
15/08/07 19:19:38 INFO storage.BlockManager: Removing RDD 28
15/08/07 19:19:38 INFO scheduler.ReceivedBlockTracker: Deleting batches ArrayBuffer()
```

MQTT Implementation

I. Scala

Step 1: Download the repository

wget http://download.opensuse.org/repositories/home:/oojah:/mqtt/RedHat_RHEL-7/home:oojah:mqtt.repo

Step 2: Place the repository into yum.repos.d

```
hadoop@ip-172-31-45-134:/etc/yum.repos.d
[hadoop@ip-172-31-45-134 yum.repos.d]$ cd ~
[hadoop@ip-172-31-45-134 ~]$ cp -r home-oojah-mqtt.repo /etc/yum.repos.d/
cp: cannot create regular file â/etc/yum.repos.d/home-oojah-mqtt.repoâ: Permission denied
[hadoop@ip-172-31-45-134 ~]$ sudo cp -r home-oojah-mqtt.repo /etc/yum.repos.d/
[hadoop@ip-172-31-45-134 ~]$ cd /etc/yum.repos.d
[hadoop@ip-172-31-45-134 yum.repos.d]$ ls -ltr
total 32
-rw-r--r-- 1 root root 1056 Mar  1  2013 epel-testing.repo
-rw-r--r-- 1 root root  957 Mar  1  2013 epel.repo
-rw-r--r-- 1 root root  686 Feb 11 23:32 amzn-updates.repo
-rw-r--r-- 1 root root  686 Feb 11 23:32 amzn-preview.repo
-rw-r--r-- 1 root root  324 Feb 11 23:32 amzn-nosrc.repo
-rw-r--r-- 1 root root  668 Feb 11 23:32 amzn-main.repo
-rw-r--r-- 1 root root   96 Aug  7 18:33 Bigtop.repo
-rw-r--r-- 1 root root  265 Aug  7 18:38 home-oojah-mqtt.repo
```

Step 3: Install mosquitto
Sudo yum install mosquitto

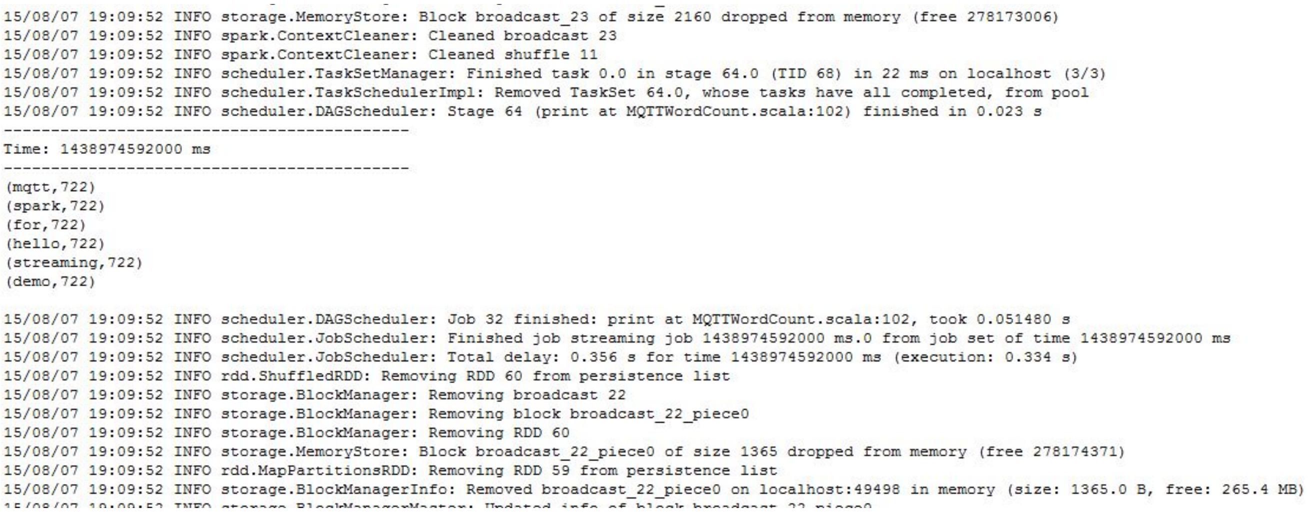
```
[hadoop@ip-172-31-45-134 yum.repos.d]$ sudo yum install mosquittp
Loaded plugins: priorities, update-motd, upgrade-helper
amzn-main/2015.03                                                          | 2.1 kB     00:00
amzn-updates/2015.03                                                       | 2.3 kB     00:00
home_oojah_mqtt                                                            | 1.2 kB     00:00
home_oojah_mqtt/primary                                                    | 2.2 kB     00:00
home_oojah_mqtt                                                                         8/8
No package mosquittp available.
Error: Nothing to do
[hadoop@ip-172-31-45-134 yum.repos.d]$ sudo yum install mosquitto
Loaded plugins: priorities, update-motd, upgrade-helper
Resolving Dependencies
--> Running transaction check
---> Package mosquitto.x86_64 0:1.4.2-3.1 will be installed
--> Processing Dependency: uuid for package: mosquitto-1.4.2-3.1.x86_64
--> Running transaction check
---> Package uuid.x86_64 0:1.6.2-27.22.amzn1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================================
 Package                    Arch                    Version                    Repository                                Size
================================================================================================================================
Installing:
 mosquitto                  x86_64                  1.4.2-3.1                  home_oojah_mqtt                          102 k
Installing for dependencies:
 uuid                       x86_64                  1.6.2-27.22.amzn1          amzn-main                                 58 k
```

Step 4: Start Mosquitto-default port number:1883

Step 5: Run this example, you may run publisher as
$ bin/run-example \org.apache.spark.examples.streaming.MQTTPublisher tcp://localhost:1883 topic

```
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
^CPublished data. topic: topic Message: hello mqtt demo for spark streaming
Published data. topic: topic Message: hello mqtt demo for spark streaming
[hadoop@ip-172-31-24-197 spark]$ bin/run-example org.apache.spark.examples.streaming.MQTTPublisher tcp://localhost:1883 topic
```

Step 6: Run the example as
$ bin/run-example \org.apache.spark.examples.streaming.MQTTWordCount tcp://localhost:1883 topic



```
15/08/07 19:09:52 INFO storage.MemoryStore: Block broadcast_23 of size 2160 dropped from memory (free 278173006)
15/08/07 19:09:52 INFO spark.ContextCleaner: Cleaned broadcast 23
15/08/07 19:09:52 INFO spark.ContextCleaner: Cleaned shuffle 11
15/08/07 19:09:52 INFO scheduler.TaskSetManager: Finished task 0.0 in stage 64.0 (TID 68) in 22 ms on localhost (3/3)
15/08/07 19:09:52 INFO scheduler.TaskSchedulerImpl: Removed TaskSet 64.0, whose tasks have all completed, from pool
15/08/07 19:09:52 INFO scheduler.DAGScheduler: Stage 64 (print at MQTTWordCount.scala:102) finished in 0.023 s
-------------------------------------------
Time: 1438974592000 ms
-------------------------------------------
(mqtt,722)
(spark,722)
(for,722)
(hello,722)
(streaming,722)
(demo,722)

15/08/07 19:09:52 INFO scheduler.DAGScheduler: Job 32 finished: print at MQTTWordCount.scala:102, took 0.051480 s
15/08/07 19:09:52 INFO scheduler.JobScheduler: Finished job streaming job 1438974592000 ms.0 from job set of time 1438974592000 ms
15/08/07 19:09:52 INFO scheduler.JobScheduler: Total delay: 0.356 s for time 1438974592000 ms (execution: 0.334 s)
15/08/07 19:09:52 INFO rdd.ShuffledRDD: Removing RDD 60 from persistence list
15/08/07 19:09:52 INFO storage.BlockManager: Removing broadcast 22
15/08/07 19:09:52 INFO storage.BlockManager: Removing block broadcast_22_piece0
15/08/07 19:09:52 INFO storage.BlockManager: Removing RDD 60
15/08/07 19:09:52 INFO storage.MemoryStore: Block broadcast_22_piece0 of size 1365 dropped from memory (free 278174371)
15/08/07 19:09:52 INFO rdd.MapPartitionsRDD: Removing RDD 59 from persistence list
15/08/07 19:09:52 INFO storage.BlockManagerInfo: Removed broadcast_22_piece0 on localhost:49498 in memory (size: 1365.0 B, free: 265.4 MB)
```

Twitter Implementation

I. Scala

Step 1: Visit the Twitter Developers' Site

Step 2: Sign in with your twitter account

Step 3: Go to apps.twitter.com

Step 4: Create a new application

Step 5: Fill in your application details

Step 6: Create your access token

Step 7: Run the code using these 4 parameters

bin/run-example org.apache.spark.examples.streaming.TwitterPopularTags
fMkkA28hy1f9WTqIGHhUQm42i
XNY9cCk2GKigPXjYPINeWMgVNWy7acNFePOQzDRBWQOhKFBNfE 63948391-
cwJ48AddECWlje63DdSge6tgDm0NUSwWW3sPiYDFy
DpoO7f3BuyzHltk41AhG6yagSHQ7AzEFvwuGybxWM7wcC

```
Popular topics in last 60 seconds (71 total):
#MTVHottest (3 tweets)
#RememberThem (1 tweets)
#soccer (1 tweets)
#powerrangerssamurai (1 tweets)
#Ø§ÙØ-Ø±Ø§Ù (1 tweets)
#MUFC (1 tweets)
#Ø³Ø-Ø§Ù (1 tweets)
#ÙÙÙ (1 tweets)
#Ø·ÙØ² (1 tweets)
#ããããã¼ã« (1 tweets)

Popular topics in last 10 seconds (71 total):
#MTVHottest (3 tweets)
#RememberThem (1 tweets)
#soccer (1 tweets)
#powerrangerssamurai (1 tweets)
#Ø§ÙØ-Ø±Ø§Ù (1 tweets)
#MUFC (1 tweets)
#Ø³Ø-Ø§Ù (1 tweets)
#ÙÙÙ (1 tweets)
#Ø·ÙØ² (1 tweets)
```

ZeroMQ Implementation

I. Scala

Step 1: Setup an Amazon EC2 instance and install Java, Spark 1.4.0, Scala 2.11.0

Step 2: Now, download zeromq 2.1.1 from http://download.zeromq.org/zeromq-2.1.1-rc.tar.gz and untar it.

Step 3: To build the environment Run ./configure and install the dependencies such as g++ ,uuid-dev by sudo apt-get install g++ and uuid-dev.

Step 4: Install make by sudo apt-get install make

Step 5: And run sudo install make

Step 6: And run sudo ldconfig

Step 7: Go to spark folder and start publisher

bin/run-example org.apache.spark.examples.streaming.SimpleZeroMQPublisher tcp://127.0.1.1:1234 foo.bar

Step 8: Once the publisher starts now run zeromq word count example

bin/run-example org.apache.spark.examples.streaming.ZeroMQWordCount tcp://127.0.1.1:1234 foo

```
Last login: Sat Aug  8 00:55:34 2015 from 129.10.18.33
ubuntu@ip-172-31-8-30:~$ cd spark-1.4.0-bin-hadoop2.6/
ubuntu@ip-172-31-8-30:~/spark-1.4.0-bin-hadoop2.6$ bin/run-example org.apache.spark.examples.streaming.ZeroMQWordCount tcp://127.0.1.1:
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
15/08/08 00:57:00 INFO StreamingExamples: Setting log level to [WARN] for streaming example. To override add a custom log4j.properties
15/08/08 00:57:00 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where app
OpenJDK 64-Bit Server VM warning: You have loaded library /tmp/jna5248674955257786015.tmp which might have disabled stack guard. The VM
now.
It's highly recommended that you fix the library with 'execstack -c <libfile>', or link it with '-z noexecstack'.
-------------------------------------------
Time: 1438995424000 ms
-------------------------------------------

15/08/08 00:57:04 WARN BlockManager: Block input-0-1438995423969 replicated to only 0 peer(s) instead of 1 peers
15/08/08 00:57:05 WARN BlockManager: Block input-0-1438995423970 replicated to only 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 1438995426000 ms
-------------------------------------------
(count,2)
(words,2)
(may,2)
```

HDFS Implementation

I. Scala

Step 1: Get the user input from text file
hadoop fs -put 13chil.txt /user/input

Step 2: Run the example for Scala word count by passing arguments
bin/run-example org.apache.spark.examples.streaming.HdfsWordCount /user/input/

```
2015-08-08 02:45:58,786 INFO  [task-result-getter-1] scheduler.TaskSchedulerImpl (Logging.scala:logInfo(59)) - Removed TaskSe
m pool
2015-08-08 02:45:58,786 INFO  [pool-16-thread-1] scheduler.DAGScheduler (Logging.scala:logInfo(59)) - Job 8 finished: print a
-------------------------------------------
Time: 1439001958000 ms
-------------------------------------------
(young,9)
(plump,1)
(mattered,1)
(paper,4)
(jump.,2)
(guide,1)
(opening,1)
(proof,2)
(afternoon,1)
(serves,1)
...

2015-08-08 02:45:58,788 INFO  [JobScheduler] scheduler.JobScheduler (Logging.scala:logInfo(59)) - Finished job streaming job
9001958000 ms
```

II. Python

Step 1: Get the user input from text file
hadoop fs -put 13chil.txt /user/input

Step 2: Run the example for Python word count by passing arguments
bin/spark-submit examples/src/main/python/streaming/hdfs_wordcount.py /user/input

```
n 0.058 s
2015-08-08 02:53:20,291 INFO  [Thread-51] scheduler.DAGScheduler (Logging.scala:logInfo(59)) - Job 313 finished: runJob at PythonRDD.sca
-------------------------------------------
Time: 2015-08-08 02:53:20
-------------------------------------------
(u'', 443)
(u'replied', 3)
(u'all', 6)
(u'shot', 1)
(u'fall,', 1)
(u'just', 2)
(u'hustled', 1)
(u'woodland', 1)
(u'over', 4)
(u'relief.', 1)
...
()
2015-08-08 02:53:20,294 INFO  [JobScheduler] scheduler.JobScheduler (Logging.scala:logInfo(59)) - Finished job streaming job 14390024000
```