Assignment 2 _Group 6

Baochen Hu & Ning Zhang

Kafka word count:

Python:

1. Start Zookeeper

bin/zookeeper-server-start.sh config/zookeeper.properties

2. configuration

/config/server.properties

host.name = localhost

advertised.host.name = localhost

3. Start Kafka Broker

bin/kafka-server-start.sh config/server.properties

4. create topic

bin/kafka-topics.sh —create —zookeeper

localhost:2181 —replication-factor 1 — partitions 1 —topic test

bin/kafka-topics.sh —list —zookeeper localhost:2181

5. Start a Producer

bin/kafka-console-producer.sh --broker-list localhost:9092 --topic test

6. Start a consumer

bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic test

--from-beginning

7. download a relieable jar, and put it in /external/kafka-assembly

wget http://search.maven.org/remotecontent?filepath=org/apache/spark/spark-streaming-kafka-assembly_2.10/1.4.0/spark-streaming-kafka-assembly_2.10-1.4.0.jar

6.

bin/spark-submit --jars

external/kafka-assembly/spark-streaming-kafka-assembly_2.10-1.4.0.jar

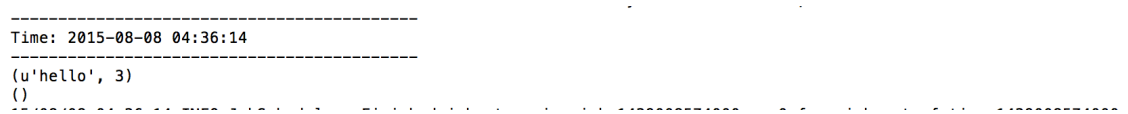
examples/src/main/python/streaming/kafka_wordcount.py localhost:2181

test

Scala:

bin/run-example org.apache.spark.examples.streaming.KafkaWordCount lo

calhost:2181 test-consumer-group test 1



Flume word count

1.

apt-get http://apache.arvixe.com/flume/1.6.0/apache-flume-1.6.0-bin.tar.g

z

2.tar -xvz apache-flume-1.6.0-bin.tar.gz

3.add config.properies file.

a1.sources = r1

a1.sinks = k1

a1.channels = c1

Describe/configure the source

a1.sources.r1.type = netcat

a1.sources.r1.bind = 127.0.0.1

```
a1.sources.r1.port = 44444



# Describe the sink

#a1.sinks = avroSink

a1.sinks.k1.type = avro

a1.sinks.k1.channel = memoryChannel

a1.sinks.k1.hostname = 127.0.0.1

a1.sinks.k1.port = 8989

#a1.sinks.k1.type = logger

# Use a channel which buffers events in memory

a1.channels.c1.type = memory

a1.channels.c1.capacity = 5

a1.channels.c1.transactionCapacity = 5

# Bind the source and sink to the channel

a1.sources.r1.channels = c1
```

a1.sinks.k1.channel = c1

4.bin/flume-ng agent --conf conf --conf-file

conf/flume-spark-integration.properties --name a1

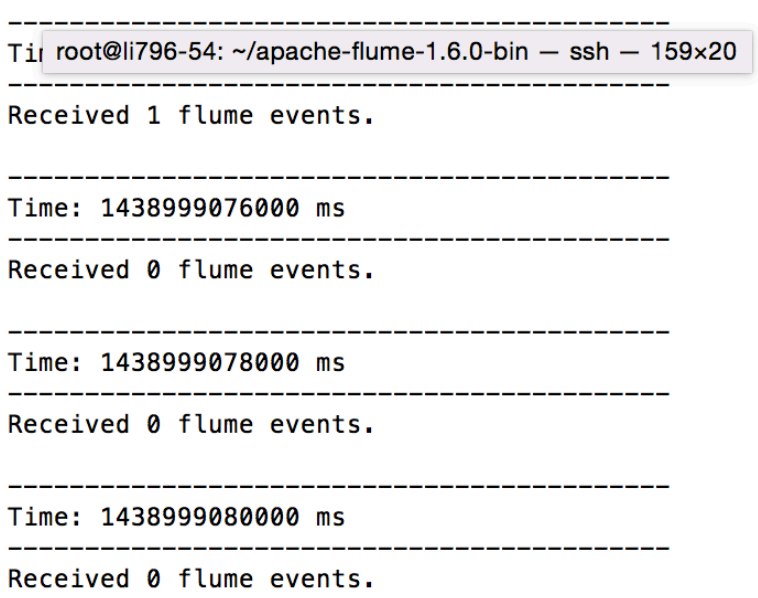
-Dflume.root.logger=INFO,console

5. open another terminal: telnet localhost 44444

6.open another terminal: bin/run-example

org.apache.spark.examples.streaming.FlumeEventCount 127.0.0.1 8989

NOTE: This should be ran first.



```
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
hello
OK
ni
OK
Connection closed by foreign host.
root@li796-54:~# telnet localhost 44444
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
hello
OK
hhllo
OK
^CConnection closed by foreign host.
```

HDFS word count:

hadoop fs -ls /user

hadoop fs -mkdir /user/ningzhang/sparkStreaming

scala:

Counts words in new text files created in the given directory

* Usage: HdfsWordCount <directory>

* <directory> is the directory that Spark Streaming will use to find and re

ad new text files.

*

* To run this on your local machine on directory `localdir`, run this example

* $ bin/run-example \

```
 *         org.apache.spark.examples.streaming.HdfsWordCount localdir
 *
 * Then create a text file in `localdir` and the words in the file will get count
ed.
 */
```

bin/run-example org.apache.spark.examples.streaming.HdfsWordCount

hdfs://ec2-52-2-200-145.compute-1.amazonaws.com/data/

hdfs dfs -copyFromLocal /home/hadoop/txtFile/hello.txt

hdfs://ec2-52-2-200-145.compute-1.amazonaws.com/data/hello1

```
-------------------------------------------
Time: 1438993764000 ms
-------------------------------------------
(hi,3)
(hello,2)
```

MQTT word count:

Install mosquitto:

1.http://mosquitto.org/download/

For Centos

Add the CentOS mosquitto repository to YUM's list of repositories

$ cd /etc/yum/yum.repos.d

$ sudo

wget http://download.opensuse.org/repositories/home:/oojah:/mqtt/Cent

OS_CentO...

$ sudo yum update

$ sudo yum install mosquitto

As of writing, no init.d script exists for the CentOS distribution of mosquitto. However, it is a simple enough matter to set it running as a daemon, you'll just need to restart it yourself whenever your machine gets restarted

1.$ sudo su

2.$ /usr/sbin/mosquitto -d -c /etc/mosquitto/mosquitto.conf >

/var/log/mosquitto.log 2>&1

$mosquitto

To run this example locally, you may run publisher as

bin/run-example

org.apache.spark.examples.streaming.MQTTPublisher tcp://localhost:1883

foo

bin/run-example

org.apache.spark.examples.streaming.MQTTWordCount tcp://localhost:18

83 foo

TwitterPopularTags:

scala:

1.

./bin/run-example

org.apache.spark.examples.streaming.TwitterPopularTags

fLFWZflOEZbUhPjhtfYvtmycy

iPGu1R7NGTt8SQkRpmFC1OfSs3VOC6vJwhzBP7yhABYTnzNvde

2612760793-cSRGyHgdIByaMeQZojsB6HYCJaPw9rgYsokbqjL

CAcXdrOf7XDnWOUrc94fdawucDs56Ylz7GZRtRpFRDnil

http://spark.apache.org/docs/latest/quick-start.html

1. install sbt

sudo yum update

curl https://bintray.com/sbt/rpm/rpm | sudo tee

/etc/yum.repos.d/bintray-sbt-rpm.repo

sudo yum install sbt

2. dependency: simple.sbt

```
name := "Simple Project"

version := "1.0"

scalaVersion := "2.10.4"

libraryDependencies ++= Seq(
 "org.apache.spark" %% "spark-core" % "1.4.1",
 "org.apache.spark" %% "spark-streaming" % "1.4.1",
 "org.apache.spark" %% "spark-streaming-twitter" % "1.4.1"
)
```

vi TwitterPopularTags.scala

```
/*     StreamingExamples.setStreamingLogLevels() */
```

3. sbt package

4. go to twitter developer to get token

https://apps.twitter.com/app/new

Consumer Key (API Key) fLFWZflOEZbUhPjhtfYvtmycy

Consumer Secret (API

Secret) iPGu1R7NGTt8SQkRpmFC1OfSs3VOC6vJwhzBP7yhABYTnzNvde

Access

Token 2612760793-cSRGyHgdIByaMeQZojsB6HYCJaPw9rgYsokbqjL

Access Token Secret

CAcXdrOf7XDnWOUrc94fdawucDs56Ylz7GZRtRpFRDnil

5.

./../../../usr/lib/spark/bin/spark-submit

twitterPopularTags/target/scala-2.10/simple-project_2.10-1.0.jar

fLFWZflOEZbUhPjhtfYvtmycy

iPGu1R7NGTt8SQkRpmFC1OfSs3VOC6vJwhzBP7yhABYTnzNvde

2612760793-cSRGyHgdIByaMeQZojsB6HYCJaPw9rgYsokbqjL

CAcXdrOf7XDnWOUrc94fdawucDs56Ylz7GZRtRpFRDnil

```
Popular topics in last 60 seconds (50 total):
#AlDubWeBelongTogether (2 tweets)
#flycosta (1 tweets)
#mizzoudg (1 tweets)
#騒音トラブル
http://t.co/WnXFWef8GC (1 tweets)
#sistas (1 tweets)
#online… (1 tweets)
#Internet (1 tweets)
#里親募集 (1 tweets)
#Follow (1 tweets)
#football (1 tweets)
15/08/08 05:01:24 WARN BlockManager: Block input-0-1439010084200 replicated to only 0 peer(s) instead of 1 peers

Popular topics in last 10 seconds (50 total):
#AlDubWeBelongTogether (2 tweets)
#flycosta (1 tweets)
#mizzoudg (1 tweets)
#騒音トラブル
http://t.co/WnXFWef8GC (1 tweets)
#sistas (1 tweets)
#online… (1 tweets)
#Internet (1 tweets)
#里親募集 (1 tweets)
#Follow (1 tweets)
#football (1 tweets)
```

ZeroMQ word count:

Install ZeroMQ:

https://www.digitalocean.com/community/tutorials/how-to-install-zeromq-from-source-on-a-centos-6-x64-vps

wget http://download.zeromq.org/zeromq-2.1.10.tar.gz

2.0.10

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

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

```
15/08/07 23:55:58 WARN BlockManager: Block input-0-1438991757105 replicated to only 0 peer(s) instead of 1 peers
15/08/07 23:55:59 WARN BlockManager: Block input-0-1438991757106 replicated to only 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 1438991760000 ms
-------------------------------------------
(count,2)
(words,2)
(may,2)
```

Kinesis Word Count:

https://github.com/apache/spark/blob/master/extras/kinesis-asl/src/main/scala/org/apache/spark/examples/streaming/KinesisWordCountASL.scala

Set up Kinesis stream (see earlier section) within AWS. Note the name of the Kinesis stream and the endpoint URL corresponding to the region where the stream was created.

https://console.aws.amazon.com/kinesis/home?region=us-east-1#stream-detail:myKinesisStream

Set up the environment variables AWS_ACCESS_KEY_ID and AWS_SECRET_KEY with your AWS credentials.

http://docs.aws.amazon.com/AWSSdkDocsJava/latest/DeveloperGuide/credentials.html

$ export AWS_ACCESS_KEY_ID=AKIAJVZXGXB72L4TZUUQ

$ export AWS_SECRET_KEY=vJCOq6o9XiRqKCas9zmiyp7fFGgqOMRIoTMSpXLE

./bin/run-example

org.apache.spark.examples.streaming.KinesisWordCountASL

myKinesisStream https://kinesis.us-east-1.amazonaws.com

./bin/run-example org.apache.spark.examples.streaming.KinesisWordCountProducerASL myKinesisStream https://kinesis.us-east-1.amazonaws.com

1000 10

```
^C[hadoop@ip-172-31-37-15 spark]$ ./bin/run-example  org.apache.spark.examples.streaming.KinesisWordCountPro
Putting records onto stream myKinesisStream and endpoint https://kinesis.us-east-1.amazonaws.com at a rate o
Sent 1000 records
Sent 1000 records
Sent 1000 records
Sent 1000 records
Sent 1000 records
Totals
(0,4976)
(1,4957)
(2,4978)
(3,5010)
(4,4945)
(5,5107)
(6,5175)
(7,4949)
(8,4948)
(9,4955)
```

```
-------------------------------------------
Time: 1439060860000 ms
-------------------------------------------
(4,152)
(8,151)
(0,153)
(5,150)
(9,141)
(1,162)
(6,161)
(2,155)
(7,136)
(3,169)

15/08/08 19:07:40 INFO scheduler.JobScheduler: Finished job stream
```