Big Data Analysis: Assignment 2

Baochen Hu
Ning Zhang

August 7, 2015

Abstract

The goal of this assignment is to try out different streaming algorithms in Apache Spark in AWS EC2/EMR.

Algorithms	Examples	Scala	Python
1	Kafka-wordcount	y	y
2	Flume-wordcount	y	y
3	HDFS-wordcount	y	y
4	Kinesis-Clickstream analysis	y	
5	MQTT-Hello world	y	
6	ZeroMQ	y	
7	Twitter-Twitter Popular Tags	y	

1 Development Environment

AWS emr-4.0.0
AWS emr-3.8.0
AWS ec2 instance
Amazon Hadoop 2.6.0
Spark 1.4.1
Spark 1.4.0
sbt

2 Tryout

2.1 Kafka-wordcount

2.1.1 Enviroment

Install kafka:

```
http://apache.mirrors.ionfish.org/kafka/0.8.2.1/kafka_2.9.1-0.8.2.1.tgz
```

Start Zookeeper:

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

Modify Configuration:

```
vi /config/server.properties

host.name = localhost
advertised.host.name = localhost
```

Start Kafka Broker:

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

Create topic test:

```
bin/kafka-topics.sh --create --zookeeper localhost:2181
       --replication-factor 1 --partitions 1 --topic test
bin/kafka-topics.sh --list --zookeeper localhost:2181
```

Start a Producer:

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

Start a Consumer:

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

Download a spark streaming kafka 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
```

2.1.2 Scala running command:

```
bin/run-example org.apache.spark.examples.streaming.KafkaWordCount
 localhost:2181 test-consumer-group test 1
```

2.1.3 Python running command:

```
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
```

2.1.4 Result:

```
-------------------------------------------------------
Time: 2015-08-08 04:36:14
-------------------------------------------------------
(u'hello', 3)
()
```

2.2 Flume-wordcount

2.2.1 Enviroment

Install flume:

```
apt-get http://apache.arvixe.com/flume/1.6.0/apache-flume-1.6.0-bin.tar.gz
tar -xvz apache-flume-1.6.0-bin.tar.gz
```

Add config.properties 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.k1.type = avro
a1.sinks.k1.channel = memoryChannel
a1.sinks.k1.hostname = 127.0.0.1
a1.sinks.k1.port = 8989

# 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
```

Start Flume:

```
bin/flume-ng agent --conf conf --conf-file conf/flume-spark-integration.properties
       --name a1 -Dflume.root.logger=INFO,console
```

Open another terminal:

```
telnet localhost 44444
```

2.2.2 Scala running command:

```
bin/run-example
org.apache.spark.examples.streaming.FlumeEventCount 127.0.0.1 8989
```

2.2.3 Python running command:

```
bin/spark-submit --jars
 examples/src/main/python/streaming/flume_wordcount.py
 localhost:2181 test
```

2.2.4 Result:

```
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
hello
OK
hhllo
OK
Connection closed by foreign host.



-------------------------------------------------------
Time: 2015-08-08 04:36:14
-------------------------------------------------------
Received 1 flume events.

-------------------------------------------------------
```

2.3 HDFS-wordcount

2.3.1 Enviroment

Create an hdfs directory and put a txt file in it

```
hdfs dfs -ls /user
hdfs dfs -mkdir /user/ningzhang/sparkStreaming
hdfs dfs -put hello.txt /user/ningzhang/sparkStreaming
```

2.3.2 Scala running command:

```
bin/run-example
 org.apache.spark.examples.streaming.HdfsWordCount
 hdfs://ec2-52-2-200-145.compute-1.amazonaws.com/data/
```

2.3.3 Python running command:

```
bin/spark-submit
 examples/src/main/python/streaming/hdfs_wordcount.py
 hdfs://ec2-52-0-97-103.compute-1.amazonaws.com/streaming
```

2.3.4 Result:

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

2.4 Kinesis-Clickstream analysis

2.4.1 Enviroment

```
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
```

2.4.2 Run command:

```
bin/spark-submit
./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
```

2.4.3 Result:

As shown in class.

2.5 MQTT-Hello world

2.5.1 Enviroment

Install mosquitto: 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/RedHat_RHEL-7/home:oojah:mqtt.repo
$ 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.

```
$ sudo su
$ /usr/sbin/mosquitto -d -c
 /etc/mosquitto/mosquitto.conf > /var/log/mosquitto.log 2>&1
```

Start mosquitto server:

```
$mosquitto
```

2.5.2 Run command:

```
bin/run-example
 org.apache.spark.examples.streaming.MQTTPublisher
 tcp://localhost:1883 foo

bin/run-example
 org.apache.spark.examples.streaming.MQTTWordCount
 tcp://localhost:1883 foo
```

2.5.3 Result:

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

2.6 ZeroMQ

2.6.1 Enviroment

Install ZeroMQ:

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

Run the following to get the tools:

```
$sudo yum install -y uuid-devel
$sudo yum install -y pkgconfig
$sudo yum install -y libtool
$sudo yum install -y gcc-c++
```

Configure the application build procedure:

```
./configure
```

Build the program using the Makefile:

```
$sudo make
$sudo make install
```

2.6.2 Run command:

```
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
```

2.6.3 Result:

```
-------------------------------------------------------
Time: 1438991760000 ms
-------------------------------------------------------
(count, 3)
(words, 2)
(may, 2)
```

2.7 Twitter-Twitter Popular Tags

2.7.1 Enviroment

Set the system properties so that Twitter4j library used by twitter stream can use them to generat OAuth credentials.

2.7.2 Run command:

```
./bin/run-example
 org.apache.spark.examples.streaming.TwitterPopularTags
  fLFWZflOEZbUhPjhtfYvtmycy
  iPGu1R7NGTt8SQkRpmFC1OfSs3VOC6vJwhzBP7yhABYTnzNvde
  2612760793-cSRGyHgdIByaMeQZojsB6HYCJaPw9rgYsokbqjL
  CAcXdrOf7XDnWOUrc94fdawucDs56Ylz7GZRtRpFRDnil
```

2.7.3 Result:

```
Popular topics in last 60 seconds (10 total):
#MTVHottest (2 tweets)
#FAceToFace (1 tweets)
#Followme (1 tweets)
#ElSocialismoEsPobreza (1 tweets)
#GAR (1 tweets)
#GH2015 (1 tweets)
#ThanksObama (1 tweets)
#YouThink (1 tweets)
#followrapido (1 tweets)
#diver (1 tweets)

Popular topics in last 10 seconds (10 total):
#MTVHottest (2 tweets)
#FAceToFace (1 tweets)
#Followme (1 tweets)
#ElSocialismoEsPobreza (1 tweets)
#GAR (1 tweets)
#GH2015 (1 tweets)
#ThanksObama (1 tweets)
#YouThink (1 tweets)
#followrapido (1 tweets)
#diver (1 tweets)
```