Hanjiao Qiu
CSCIE63 Big Data Analytics
Assignment 08

Assignment 08 Solution

Please, describe every step of your work and present all intermediate and final results in a Word document. Please, copy past text version of all essential command and snippets of results into the Word document. We cannot retype text that is in JPG images. Please, always submit a separate copy of the original, working scripts and/or class files you used as separate files. Sometimes we need to run your code and retyping is too costly. Please include in your MS Word document only relevant portions of the console output or output files. Sometime either console output or the result file is too long and including it into the MS Word document makes that document too hard to read. PLEASE DO NOT EMBED files into your MS Word document. Please, submit to the class drop box. For issues and comments visit the class Discussion Board. You are not obliged to use Java or Eclipse. You are welcome to use any language and any IDE of your choice.

Use most excellent and very detailed notes created by Marina Popova for Section 08 on Kafka and Streaming as you main guide for this assignment.

Problem 1) On your Cloudera VM or any other VM you might be using install Kafka. Just in case, install one of the recent Kafka 0.8 versions. Demonstrate that you can create a topic, publish messages to that topic and consume messages sent to that topic. Use Kafka command line interface.

Solution

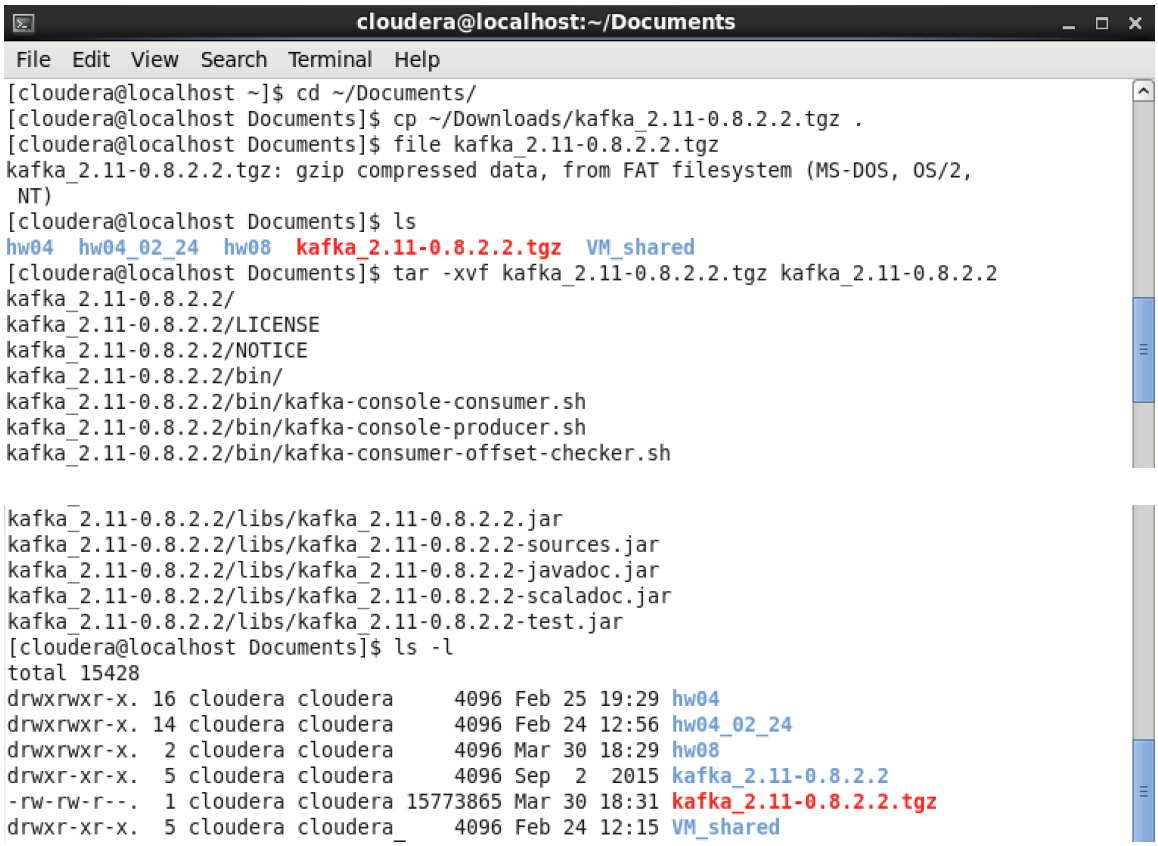
1. Install Kafka on my own VM, which is CentOS 6.7 with JDK 1.8 and Spark 1.6.2.

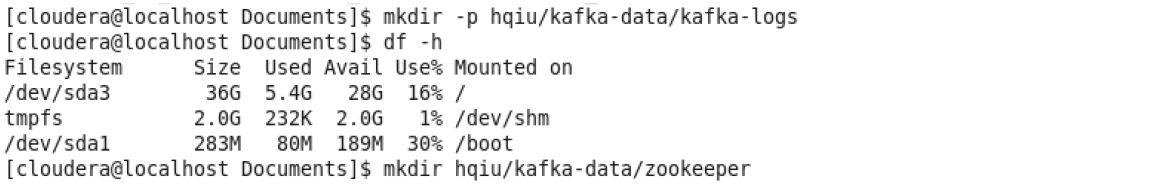
Download Kafka from:
http://kafka.apache.org/downloads.html

Choose Kafka 0.8.2.2
https://www.apache.org/dyn/closer.cgi?path=/kafka/0.8.2.2/kafka_2.11-0.8.2.2.tgz

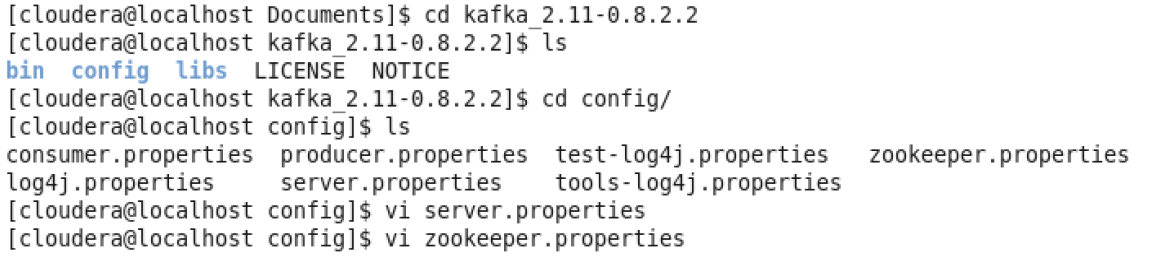
```
[cloudera@localhost Documents]$ file kafka_2.11-0.8.2.2.tgz

[cloudera@localhost Documents]$ tar -xvf kafka_2.11-0.8.2.2.tgz kafka_2.11-
0.8.2.2
```

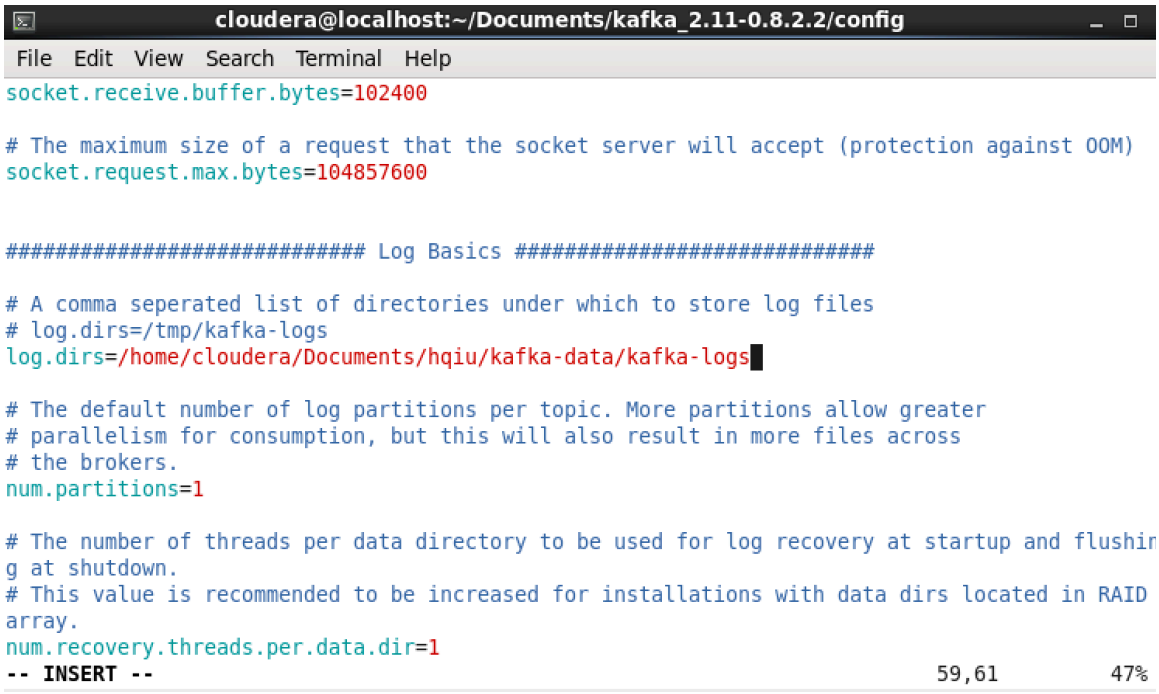
Check available disk space to store the logs for ZooKeeper and Kafka.

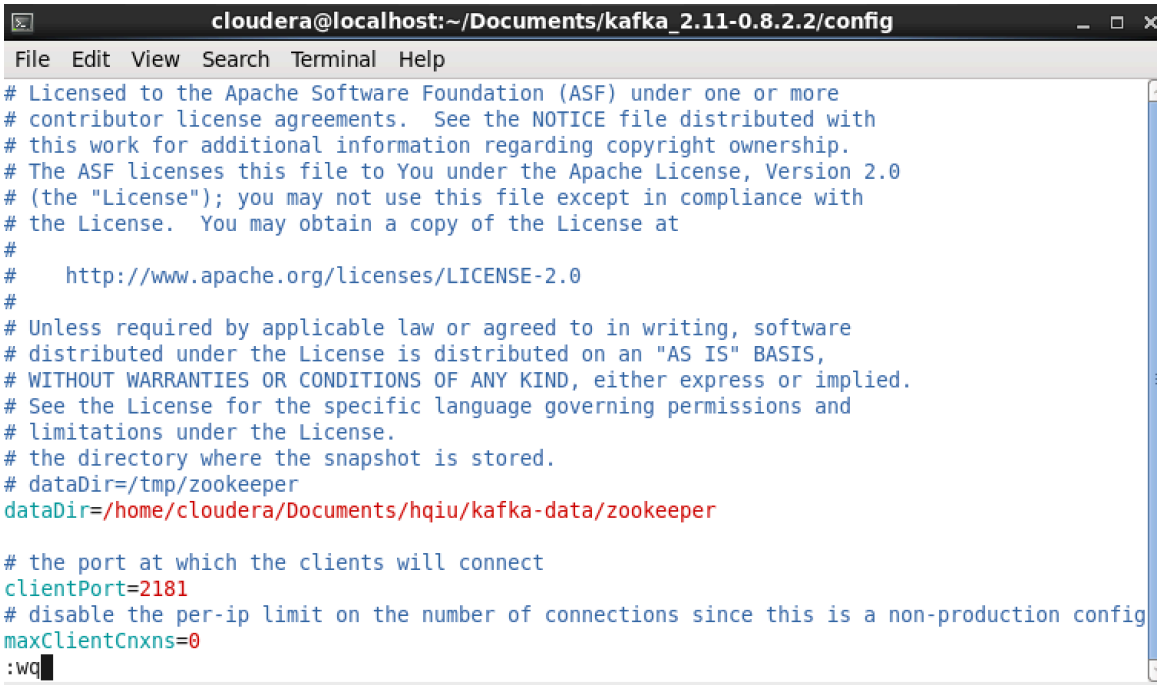




Adjust Kafka's server.properties and zookeeper properties. Modify the path to store logs.







2. Start ZooKeeper and Kafka server. Start ZooKeeper first, Kafka server second.

```
[cloudera@localhost Documents]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/zookeeper-server-start.sh /home/cloudera/Documents/kafka_2.11-
0.8.2.2/config/zookeeper.properties
```

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-server-start.sh /home/cloudera/Documents/kafka_2.11-
0.8.2.2/config/server.properties
```

```
[cloudera@localhost Documents]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/zookeeper-ser
ver-start.sh /home/cloudera/Documents/kafka_2.11-0.8.2.2/config/zookeeper.properties
[2016-03-30 18:47:42,060] INFO Reading configuration from: /home/cloudera/Documents/kafka_2.1
1-0.8.2.2/config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2016-03-30 18:47:42,066] INFO autopurge.snapRetainCount set to 3 (org.apache.zookeeper.serve
r.DatadirCleanupManager)
[2016-03-30 18:47:42,066] INFO autopurge.purgeInterval set to 0 (org.apache.zookeeper.server.
DatadirCleanupManager)
[2016-03-30 18:47:42,066] INFO Purge task is not scheduled. (org.apache.zookeeper.server.Data
dirCleanupManager)
[2016-03-30 18:47:42,066] WARN Either no config or no quorum defined in config, running  in s
tandalone mode (org.apache.zookeeper.server.quorum.QuorumPeerMain)
[2016-03-30 18:47:42,091] INFO Reading configuration from: /home/cloudera/Documents/kafka_2.1
1-0.8.2.2/config/zookeeper.properties (org.apache.zookeeper.server.quorum.QuorumPeerConfig)
[2016-03-30 18:47:42,093] INFO Starting server (org.apache.zookeeper.server.ZooKeeperServerMa
in)
[2016-03-30 18:47:42,113] INFO Server environment:zookeeper.version=3.4.6-1569965, built on 0
2/20/2014 09:09 GMT (org.apache.zookeeper.server.ZooKeeperServer)
[2016-03-30 18:47:42,113] INFO Server environment:host.name=localhost (org.apache.zookeeper.s
erver.ZooKeeperServer)
[2016-03-30 18:47:42,113] INFO Server environment:java.version=1.8.0_60 (org.apache.zookeeper
.server.ZooKeeperServer)

[2016-03-30 18:47:42,114] INFO Server environment:user.name=cloudera (org.apache.zookeeper.se
rver.ZooKeeperServer)
[2016-03-30 18:47:42,114] INFO Server environment:user.home=/home/cloudera (org.apache.zookee
per.server.ZooKeeperServer)
[2016-03-30 18:47:42,114] INFO Server environment:user.dir=/home/cloudera/Documents (org.apac
he.zookeeper.server.ZooKeeperServer)
[2016-03-30 18:47:42,128] INFO tickTime set to 3000 (org.apache.zookeeper.server.ZooKeeperSer
ver)
[2016-03-30 18:47:42,128] INFO minSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKe
eperServer)
[2016-03-30 18:47:42,128] INFO maxSessionTimeout set to -1 (org.apache.zookeeper.server.ZooKe
eperServer)
[2016-03-30 18:47:42,145] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.ser
ver.NIOServerCnxnFactory)
```

We will see “binding to port 0.0.0.0/0.0.0.0:2181” from ZooKeeper window.

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-server-start.sh /h
ome/cloudera/Documents/kafka_2.11-0.8.2.2/config/server.properties
[2016-03-30 18:53:06,275] INFO Verifying properties (kafka.utils.VerifiableProperties)
[2016-03-30 18:53:06,306] INFO Property broker.id is overridden to 0 (kafka.utils.VerifiableProp
erties)
[2016-03-30 18:53:06,306] INFO Property log.cleaner.enable is overridden to false (kafka.utils.V
erifiableProperties)
[2016-03-30 18:53:06,307] INFO Property log.dirs is overridden to /home/cloudera/Documents/hqiu/
kafka-data/kafka-logs (kafka.utils.VerifiableProperties)
[2016-03-30 18:53:06,307] INFO Property log.retention.check.interval.ms is overridden to 300000
(kafka.utils.VerifiableProperties)
[2016-03-30 18:53:06,309] INFO Property log.retention.hours is overridden to 168 (kafka.utils.Ve
rifiableProperties)
[2016-03-30 18:53:06,309] INFO Property log.segment.bytes is overridden to 1073741824 (kafka.uti
ls.VerifiableProperties)
[2016-03-30 18:53:06,310] INFO Property num.io.threads is overridden to 8 (kafka.utils.Verifiabl
eProperties)
[2016-03-30 18:53:06,310] INFO Property num.network.threads is overridden to 3 (kafka.utils.Veri
fiableProperties)
```

```
[2016-03-30 18:53:06,782] INFO Starting log flusher with a default period of 9223372036854775807
 ms. (kafka.log.LogManager)
[2016-03-30 18:53:06,815] INFO Awaiting socket connections on 0.0.0.0:9092. (kafka.network.Accep
tor)
[2016-03-30 18:53:06,816] INFO [Socket Server on Broker 0], Started (kafka.network.SocketServer)
[2016-03-30 18:53:06,931] INFO Will not load MX4J, mx4j-tools.jar is not in the classpath (kafka
.utils.Mx4jLoader$)
[2016-03-30 18:53:06,970] INFO 0 successfully elected as leader (kafka.server.ZookeeperLeaderEle
ctor)
[2016-03-30 18:53:07,068] INFO Registered broker 0 at path /brokers/ids/0 with address localhost
:9092. (kafka.utils.ZkUtils$)
[2016-03-30 18:53:07,084] INFO [Kafka Server 0], started (kafka.server.KafkaServer)
[2016-03-30 18:53:07,183] INFO New leader is 0 (kafka.server.ZookeeperLeaderElector$LeaderChange
Listener)
```

We will see “Kafka Server 0 started” from Kafka server window.

3. Create a new topic.

Describe Kafka cluster:

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-topics.sh --describe --zookeeper localhost:2181
```

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --descri
be --zookeeper localhost:2181
[cloudera@localhost ~]$
```

Since we haven’t created any topics yet, no topics show up.

Create new topics:

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-topics.sh --create --zookeeper localhost:2181 --
replication-factor 1 --partitions 4 --topic spark-topic
```

Check which topic are already created:

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-topics.sh --list --zookeeper localhost:2181
```

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --create
 --zookeeper localhost:2181 --replication-factor 1 --partitions 4 --topic spark-topic
Created topic "spark-topic".
```

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --descri
be --zookeeper localhost:2181
Topic:spark-topic       PartitionCount:4        ReplicationFactor:1     Configs:
        Topic: spark-topic      Partition: 0    Leader: 0       Replicas: 0     Isr: 0
        Topic: spark-topic      Partition: 1    Leader: 0       Replicas: 0     Isr: 0
        Topic: spark-topic      Partition: 2    Leader: 0       Replicas: 0     Isr: 0
        Topic: spark-topic      Partition: 3    Leader: 0       Replicas: 0     Isr: 0
```

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-topics.sh --list -
-zookeeper localhost:2181
spark-topic
```

Topic “spark-topic” has been created.

4. Test new topic. Start Kafka producer to publish messages to topic and consumer to consume messages sent to that topic.

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-console-producer.sh --broker-list localhost:9092 --topic
spark-topic

[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-console-consumer.sh --zookeeper localhost:2181 --topic
spark-topic --from-beginning
```

Type a message in the producer, one message a line. We can observe the messages being received in the consumer.

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-console-producer.s
h --broker-list localhost:9092 --topic spark-topic
[2016-03-30 19:05:52,367] WARN Property topic is not valid (kafka.utils.VerifiableProperties)
test message one from Hanjiao
test message two from Hanjiao
test message three from Hanjiao
```

```
[cloudera@localhost ~]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-console-
consumer.sh --zookeeper localhost:2181 --topic spark-topic --from-beginning
test message one from Hanjiao
test message two from Hanjiao
test message three from Hanjiao
```

5. Check the logs stored in our specified path (set in Kafka's server.properties).

```
[cloudera@localhost ~]$ cd Documents/hqiu/kafka-data/kafka-logs/
[cloudera@localhost kafka-logs]$ ls
recovery-point-offset-checkpoint  spark-topic-0  spark-topic-2
replication-offset-checkpoint     spark-topic-1  spark-topic-3
```

```
[cloudera@localhost kafka-logs]$ ls -l spark-topic-2
total 0
-rw-rw-r--. 1 cloudera cloudera 0 Mar 30 19:11 00000000000000000000.index
-rw-rw-r--. 1 cloudera cloudera 0 Mar 30 18:58 00000000000000000000.log
[cloudera@localhost kafka-logs]$ ls -l spark-topic-1
total 0
-rw-rw-r--. 1 cloudera cloudera 0 Mar 30 19:11 00000000000000000000.index
-rw-rw-r--. 1 cloudera cloudera 0 Mar 30 18:58 00000000000000000000.log
[cloudera@localhost kafka-logs]$ ls -l spark-topic-0
total 4
-rw-rw-r--. 1 cloudera cloudera   0 Mar 30 19:11 00000000000000000000.index
-rw-rw-r--. 1 cloudera cloudera 167 Mar 30 19:06 00000000000000000000.log
```

The messages are written into Partition 0. Check the log from Partition 0:

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-run-class.sh kafka.tools.DumpLogSegments --files
/home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-topic-
0/00000000000000000000.log --print-data-log
```

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-run-class
.sh kafka.tools.DumpLogSegments --files /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spar
k-topic-0/00000000000000000000.log --print-data-log
Dumping /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-topic-0/00000000000000000000.l
og
Starting offset: 0
offset: 0 position: 0 isvalid: true payloadsize: 29 magic: 0 compresscodec: NoCompressionCodec c
rc: 1411305222 payload: test message one from Hanjiao
offset: 1 position: 55 isvalid: true payloadsize: 29 magic: 0 compresscodec: NoCompressionCodec
crc: 147163294 payload: test message two from Hanjiao
offset: 2 position: 110 isvalid: true payloadsize: 31 magic: 0 compresscodec: NoCompressionCodec
 crc: 1130537238 payload: test message three from Hanjiao
```

The above three messages are what we just sent to topic “spark-topic”.

6. Shutdown Kafka server first, then shutdown Zookeeper (always do in this order!).

To stop the services, just type “Ctrl + C” in the terminal.

Kafka server:

```
[2016-03-30 19:11:46,362] INFO [Socket Server on Broker 0], Shutdown completed (kafka.network.So
cketServer)
[2016-03-30 19:11:46,362] INFO [Kafka Request Handler on Broker 0], shutting down (kafka.server.
KafkaRequestHandlerPool)
[2016-03-30 19:11:46,365] INFO [Kafka Request Handler on Broker 0], shut down completely (kafka.
server.KafkaRequestHandlerPool)
[2016-03-30 19:11:46,655] INFO [Replica Manager on Broker 0]: Shut down (kafka.server.ReplicaMan
ager)
[2016-03-30 19:11:46,656] INFO [ReplicaFetcherManager on broker 0] shutting down (kafka.server.R
eplicaFetcherManager)
[2016-03-30 19:11:46,657] INFO [ReplicaFetcherManager on broker 0] shutdown completed (kafka.ser
ver.ReplicaFetcherManager)
[2016-03-30 19:11:46,659] INFO [Replica Manager on Broker 0]: Shut down completely (kafka.server
.ReplicaManager)
[2016-03-30 19:11:46,660] INFO Shutting down. (kafka.log.LogManager)
[2016-03-30 19:11:46,680] INFO Shutdown complete. (kafka.log.LogManager)
[2016-03-30 19:11:46,688] INFO Terminate ZkClient event thread. (org.I0Itec.zkclient.ZkEventThre
ad)
[2016-03-30 19:11:46,691] INFO Session: 0x153ca5ceb980000 closed (org.apache.zookeeper.ZooKeeper
)
[2016-03-30 19:11:46,691] INFO EventThread shut down (org.apache.zookeeper.ClientCnxn)
[2016-03-30 19:11:46,691] INFO [Kafka Server 0], shut down completed (kafka.server.KafkaServer)
```

ZooKeeper:

```
[2016-03-30 19:08:12,358] INFO Closed socket connection for client /127.0.0.1:50339 which had
 sessionid 0x153ca5ceb980008 (org.apache.zookeeper.server.NIOServerCnxn)
[2016-03-30 19:08:33,000] INFO Expiring session 0x153ca5ceb980008, timeout of 30000ms exceede
d (org.apache.zookeeper.server.ZooKeeperServer)
[2016-03-30 19:08:33,001] INFO Processed session termination for sessionid: 0x153ca5ceb980008
 (org.apache.zookeeper.server.PrepRequestProcessor)
[2016-03-30 19:11:46,689] INFO Processed session termination for sessionid: 0x153ca5ceb980000
 (org.apache.zookeeper.server.PrepRequestProcessor)
[2016-03-30 19:11:46,690] INFO Closed socket connection for client /127.0.0.1:50263 which had
 sessionid 0x153ca5ceb980000 (org.apache.zookeeper.server.NIOServerCnxn)
```

Problem 2) Using Java or Python or any other (even scripting) language of your choice construct a producer and a consumer object. Let producer generate one random number

between 0 or 1 and 10 every second. Let both producer and consumer run indefinitely or until you kill them. Demonstrate that your consumer is receiving messages by printing both the stream if numbers generated on the producer and the stream of numbers fetched by the consumer. You might find it easier to print to files and examine files afterwards. Once you terminate the exchange, examine Kafka's log.
Instructions on how to write Java producer and consumer you can find on this URLs:
https://cwiki.apache.org/confluence/display/KAFKA/0.8.0+Producer+Example
https://cwiki.apache.org/confluence/display/KAFKA/0.8.0+SimpleConsumer+Example

Instructions on how to write Python clients for Kafka you could find on this URL:
https://cwiki.apache.org/confluence/display/KAFKA/Clients#Clients-Python

Instructions for Scala could be found here:
https://cwiki.apache.org/confluence/display/KAFKA/Clients#Clients-ScalaDSL

You are welcome to follow any other instructions and use any other programming or scripting language.

Solution

Source Code Manifest

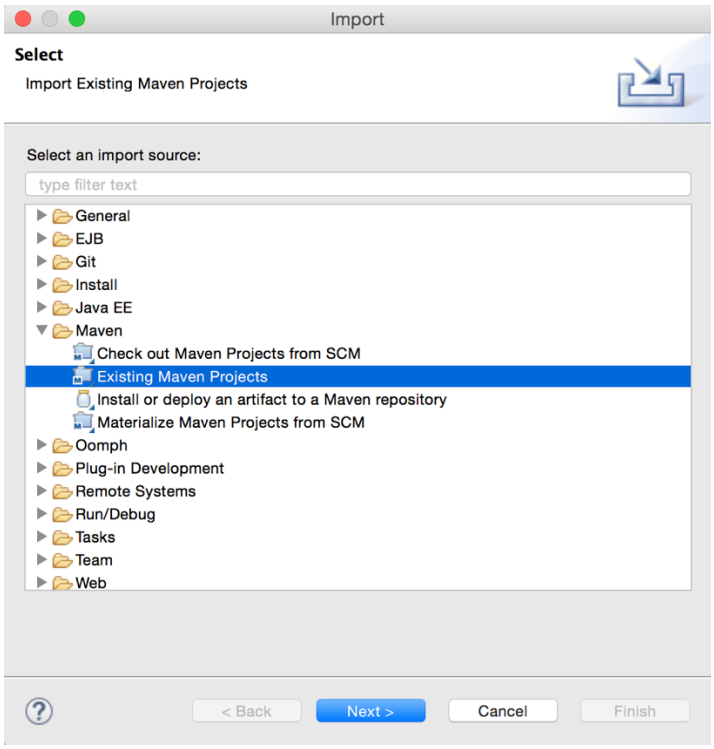
Java Solution	KafkaRandIntProducer.java KafkaRandIntProducerPT.java (Implement partition function) SimplePartitioner.java KafkaDirectRandIntConsumer.java KafkaDirectRandIntConsumerMT.java (Implement multi-threading) ConsumerThread.java
Python Solution	kafka_python_producer.py kafka_python_consumer.py

"KafkaRandIntProducer.java" doesn't include the step to send messages to a specific partition. Kafka will assign the message to a random partition.
"KafkaRandIntProducerPT.java" will assign messages to a specific partition based on the partition key.

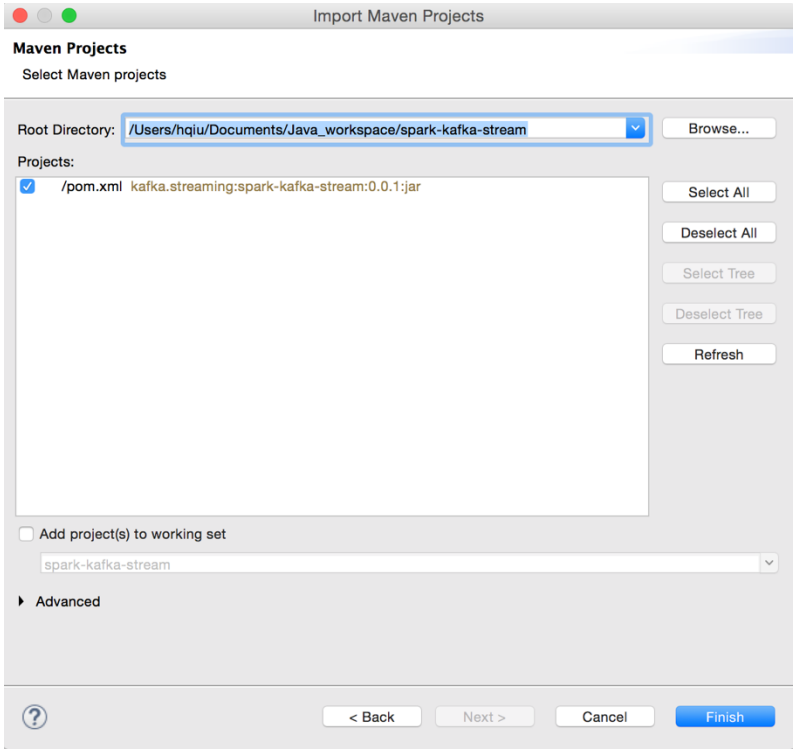
"KafkaDirectRandIntConsumer.java" is a single thread application.
"KafkaDirectRandIntConsumerPT.java uses multi-threading to consume messages.

1. Import the project to Eclipse on local machine (Mac OS). It's easier to debug locally and then run the code on VM CentOS 6.7.

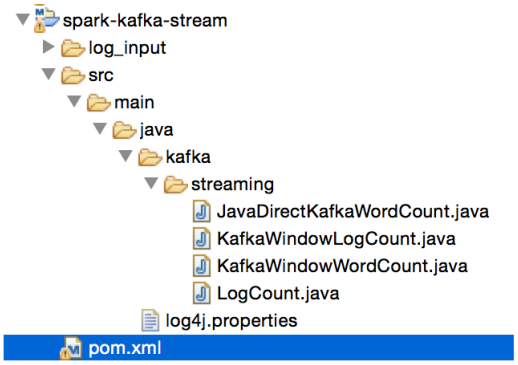
First create a new Java Maven project by choose "File->Import->…->Maven->Existing Maven Projects".



Navigate to the project's directory and choose "pom.xml".



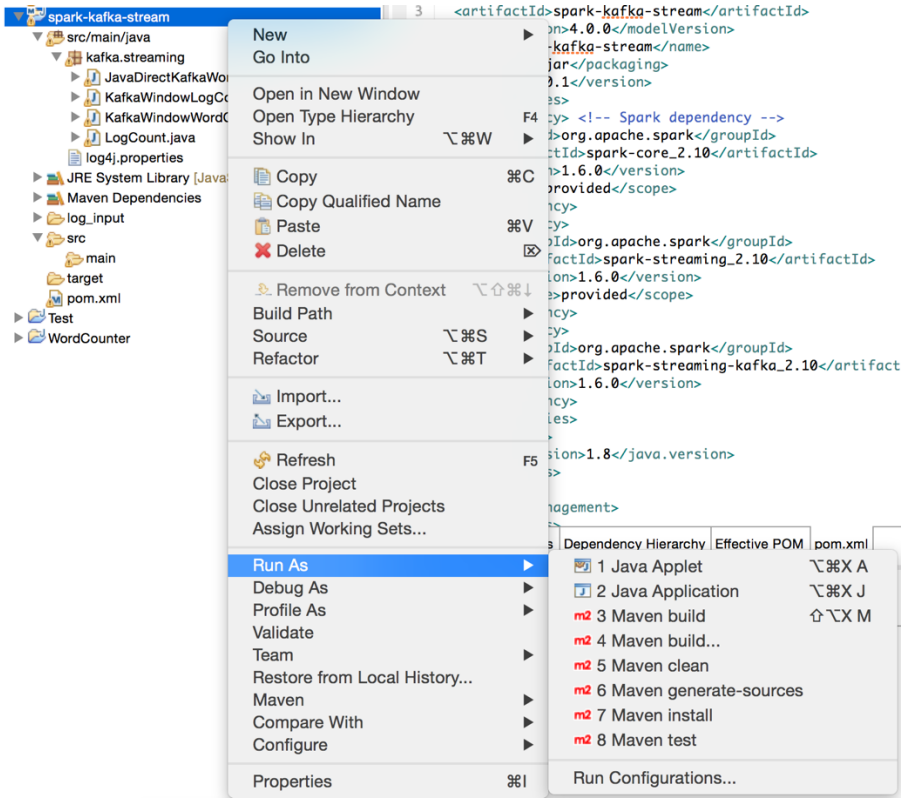
The imported file structure is like this:



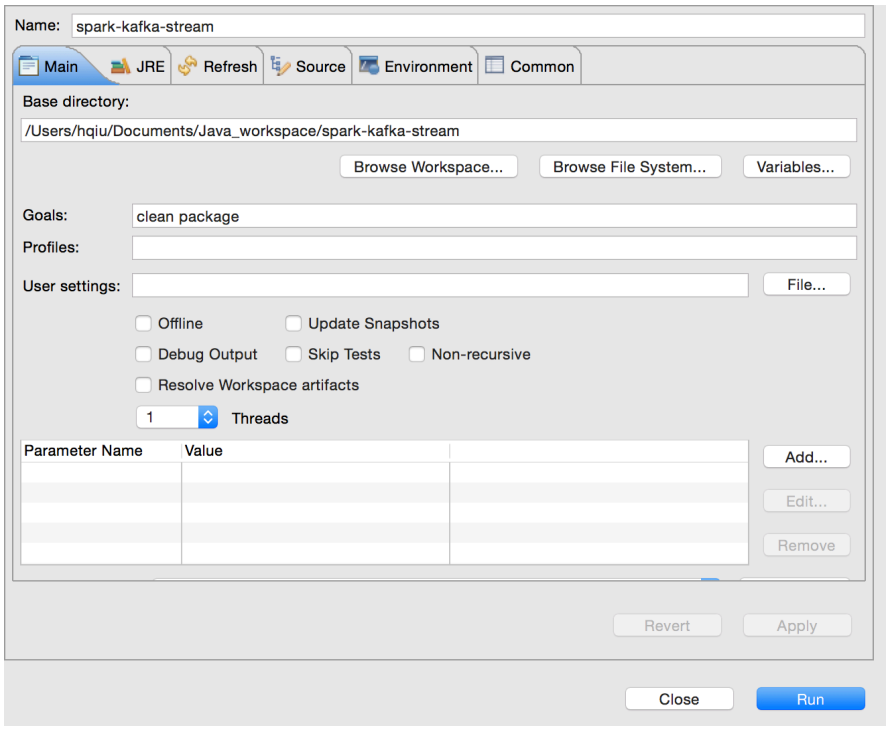
"pom.xml" will import all the necessary jars/libraries. To run the application, we can either build the jar using "Maven build", or directly run from Eclipse console.

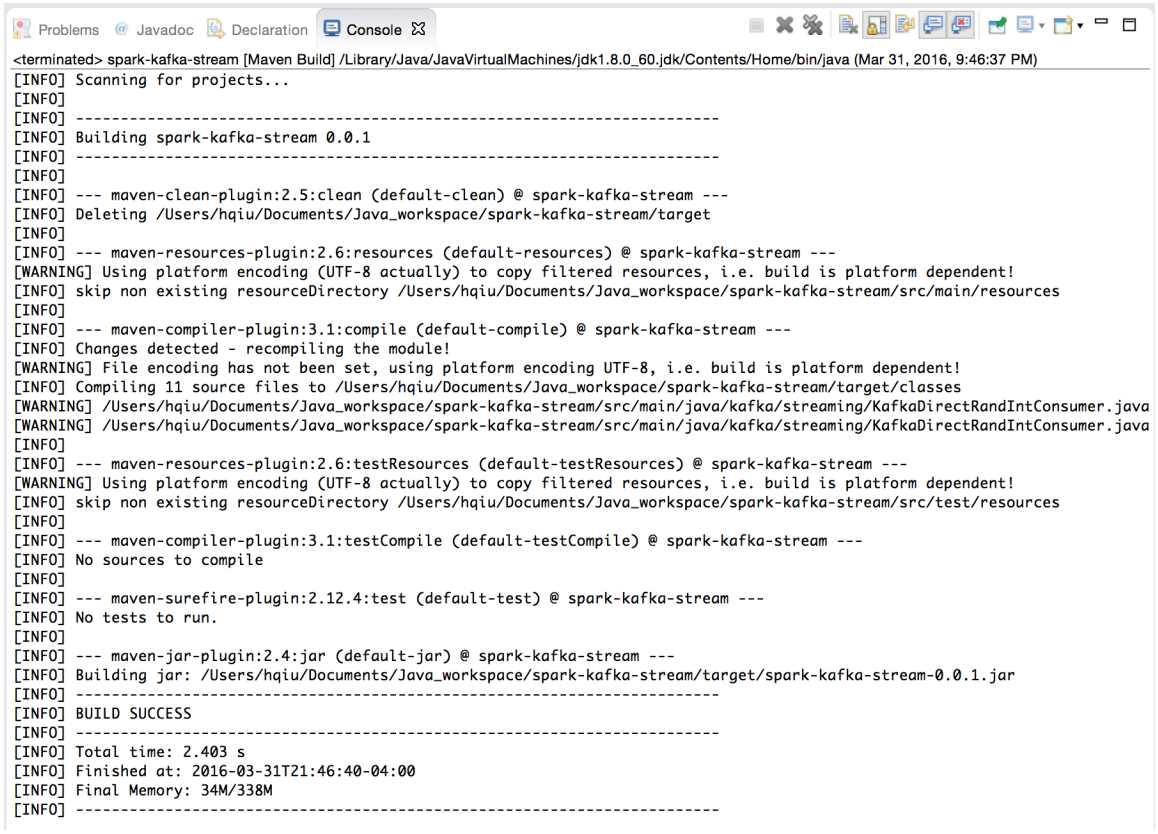
The configuration steps on VM CentOS are exactly the same.

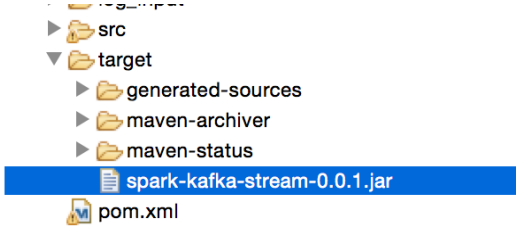
To build the project, right click the project, choose "Run As ->RunConfigurations".



Create a new run configuration, specify Goals “clean package”. Click “Run”, the file “spark-kafka-stream-0.0.1.jar” will be generated under folder “target”.







2. Create my own Java Kafka producer and consumer.

The instructions I followed are:
https://cwiki.apache.org/confluence/display/KAFKA/0.8.0+Producer+Example
https://cwiki.apache.org/confluence/display/KAFKA/Consumer+Group+Example

KafkaRandIntProducer.java:

```
package kafka.streaming;

import java.util.*;

import kafka.javaapi.producer.Producer;
import kafka.producer.KeyedMessage;
import kafka.producer.ProducerConfig;

public class KafkaRandIntProducer {
   public static void main(String[] args) {
          if (args.length < 2) {
                System.err.println("Usage: KafkaRandIntProducer <brokers> <topic> [<events>]\n" +
                             "  <brokers> is a list of one or more Kafka brokers\n" +
                             "  <topic> is the kafka topic to produce to \n" +
                             "  <events> is the total number of events to send \n\n");
                System.exit(1);
          }

          String brokers = args[0];
          String topic = args[1];

          /*
          long events = 50000;
          if (args.length == 3) {
                events = Long.parseLong(args[2]);
          }
          */

          Random rnd = new Random();

          Properties props = new Properties();
          props.put("metadata.broker.list", brokers);
          props.put("serializer.class", "kafka.serializer.StringEncoder");

          ProducerConfig config = new ProducerConfig(props);
```

```
        Producer<String, String> producer = new Producer<String, String>(config);

        // for (long nEvents = 0; nEvents < events; nEvents++) {
        while (true) {
                // Generate a random number between 1 to 10
                int number = rnd.nextInt(10) + 1;

                // Get current time stamp
                Date time = new Date(System.currentTimeMillis());

                // Compose the message
                String msg = "Message sent at " + time.toString() + ":\t" + number;

                // Send the message to topic
                // KeyedMessage<String, String> data = new KeyedMessage<String, String>(topic,
Integer.toString(number));
                KeyedMessage<String, String> data = new KeyedMessage<String, String>(topic, msg);
                producer.send(data);

                System.out.println(msg);

                try {
                        // Sleep for 1 second
                    Thread.sleep(1000);
                } catch(InterruptedException ex) {
                    Thread.currentThread().interrupt();
                }
        }

        /*
        if (producer != null) {
                producer.close();
        }
        */
   }
}
```

In this Producer:

- Class **ProducerConfig** is used wrap different properties those are required to establish connection with Kafka broker.
- Class **KeyedMessage** is used by Kafka producer to send message/data to Kafka broker. With this class we can define the topic name, message partition key and message.
- Class **Producer** is used to send data to the broker in form of KeyedMessage object. Message can be sent in both way synchronously or asynchronously.
- We haven't defined a partition class here. Kafka will use the default partitioner. From the results we can see the messages were sent to a random partition.

KafkaDirectRandIntConsumer.java:

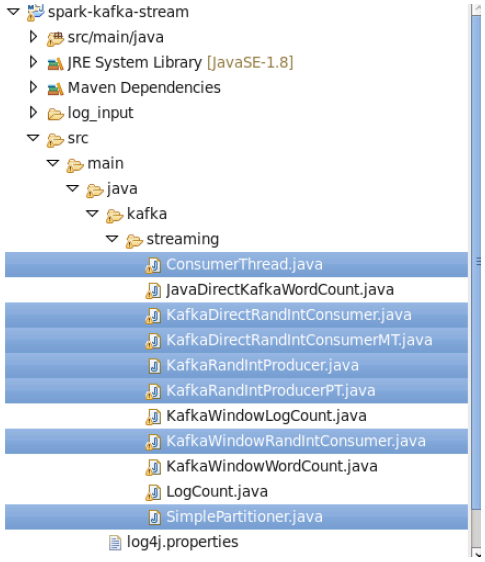
```
package kafka.streaming;

import java.util.*;

import kafka.consumer.Consumer;
import kafka.consumer.ConsumerConfig;
import kafka.consumer.ConsumerIterator;
import kafka.consumer.KafkaStream;
import kafka.javaapi.consumer.ConsumerConnector;

public final class KafkaDirectRandIntConsumer {

   public static void main(String[] args) {
          if (args.length < 1) {
                 System.err.println("Usage: KafkaDirectRandIntConsumer <topic>\n" +
                               "  <topic> is the kafka topic to consume from\n\n");
                 System.exit(1);
          }

          String topic = args[0];

          Properties props = new Properties();
          props.put("zookeeper.connect", "localhost:2181");
          props.put("group.id", "spark-app");

          ConsumerConnector consumer = Consumer.createJavaConsumerConnector(new
ConsumerConfig(props));

          Map<String, Integer> topicCountMap = new HashMap<String, Integer>();
          topicCountMap.put(topic, 1);

          Map<String, List<KafkaStream<byte[], byte[]>>> consumerStreams =
consumer.createMessageStreams(topicCountMap);

          List<KafkaStream<byte[], byte[]>> streams = consumerStreams.get(topic);
          for (final KafkaStream stream : streams) {
                 ConsumerIterator<byte[], byte[]> it = stream.iterator();
                 while (it.hasNext()) {
                        System.out.println("Message from Topic '" + topic + "': " + new
String(it.next().message()));
                 }
          }

          if (consumer != null) {
                 consumer.shutdown();
          }

   }
}
```

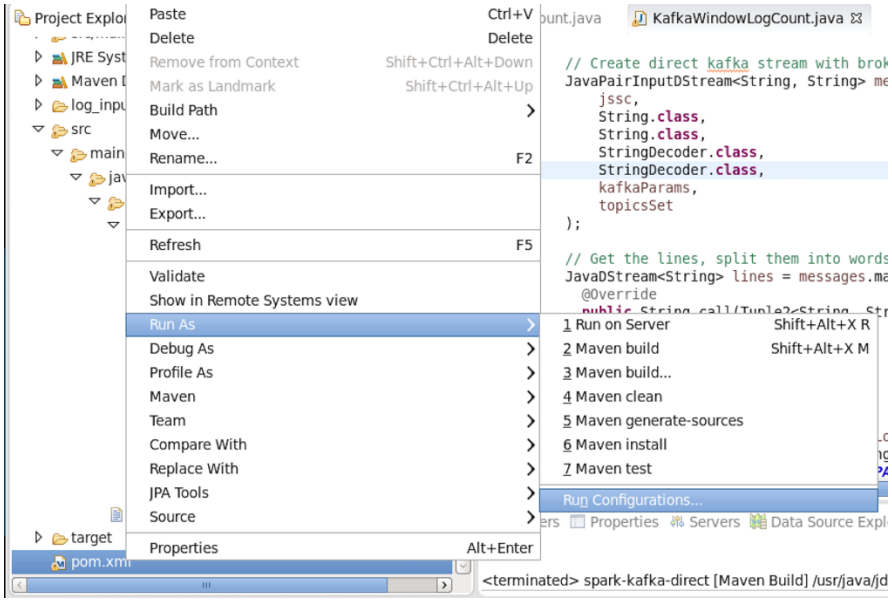
In this Consumer:

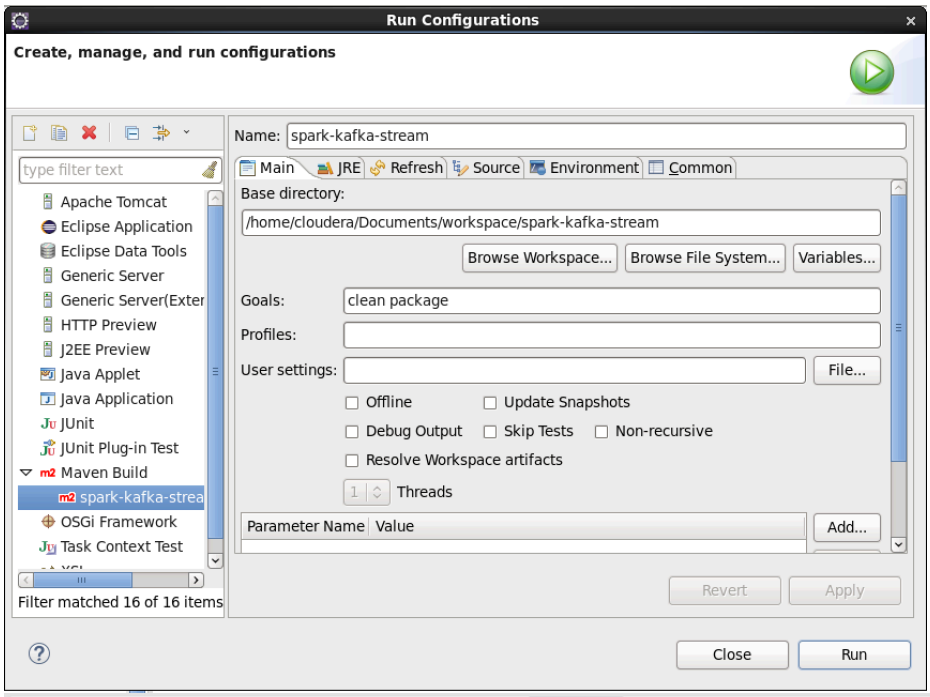
- Class **ConsumerConfig** is used to wrap different properties those are required to establish connection between consumer and Zookeeper.

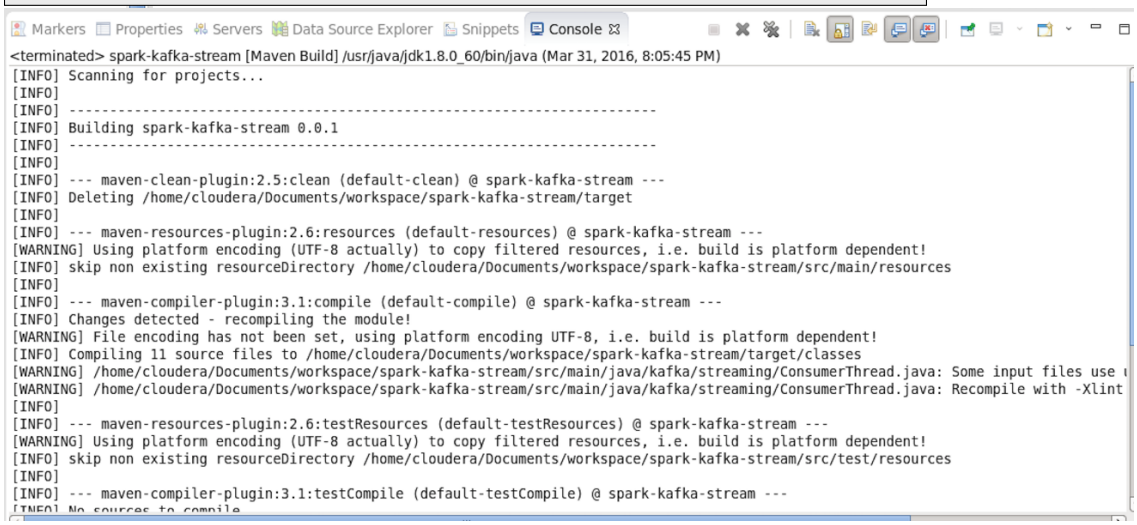
- **ConsumerConnector** is the Kafka interface.
- **ZookeeperConsumerConnector** is the implementer class for **ConsumerConnector**. This implementer class is used to establish connection with ZooKeeper. All the interactions with ZooKeeper are taken care by this implementer class class.
- **ConsumerConnector** returns the list of **KafkaStream** object for each topic wrapped in a map as mentioned below. Map key is the topic name and value is the list of KafkaStream objects. KafkaStream K is partition key type and V is the actual message Map<String, List<KafkaStream<K, V>>.
- **ConsumerIterator** is used to iterate **KafkaStream.**

3. Compile and run the code using Eclipse on VM CentOS 6.7.






Run Configurations
Create, manage, and run configurations
type filter text
Apache Tomcat
Eclipse Application
Eclipse Data Tools
Generic Server
Generic Server(Exter
HTTP Preview
J2EE Preview
Java Applet
Java Application
JUnit
JUnit Plug-in Test
Maven Build
spark-kafka-strea
OSGi Framework
Task Context Test
Filter matched 16 of 16 items
Name: spark-kafka-stream
Main
JRE
Refresh
Source
Environment
Common
Base directory:
/home/cloudera/Documents/workspace/spark-kafka-stream
Browse Workspace...
Browse File System...
Variables...
Goals: clean package
Profiles:
User settings:
File...
Offline
Update Snapshots
Debug Output
Skip Tests
Non-recursive
Resolve Workspace artifacts
Threads
Parameter Name
Value
Add...
Revert
Apply
Close
Run


Markers
Properties
Servers
Data Source Explorer
Snippets
Console
<terminated> spark-kafka-stream [Maven Build] /usr/java/jdk1.8.0_60/bin/java (Mar 31, 2016, 8:05:45 PM)
[INFO] Scanning for projects...
[INFO]
[INFO] --
[INFO] Building spark-kafka-stream 0.0.1
[INFO] --
[INFO]
[INFO] --- maven-clean-plugin:2.5:clean (default-clean) @ spark-kafka-stream ---
[INFO] Deleting /home/cloudera/Documents/workspace/spark-kafka-stream/target
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ spark-kafka-stream ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/cloudera/Documents/workspace/spark-kafka-stream/src/main/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ spark-kafka-stream ---
[INFO] Changes detected - recompiling the module!
[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 11 source files to /home/cloudera/Documents/workspace/spark-kafka-stream/target/classes
[WARNING] /home/cloudera/Documents/workspace/spark-kafka-stream/src/main/java/kafka/streaming/ConsumerThread.java: Some input files use
[WARNING] /home/cloudera/Documents/workspace/spark-kafka-stream/src/main/java/kafka/streaming/ConsumerThread.java: Recompile with -Xlint
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ spark-kafka-stream ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/cloudera/Documents/workspace/spark-kafka-stream/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ spark-kafka-stream ---


[WARNING] File encoding has not been set, using platform encoding UTF-8, i.e. build is platform dependent!
[INFO] Compiling 11 source files to /home/cloudera/Documents/workspace/spark-kafka-stream/target/classes
[WARNING] /home/cloudera/Documents/workspace/spark-kafka-stream/src/main/java/kafka/streaming/ConsumerThread.java: Some input files use
[WARNING] /home/cloudera/Documents/workspace/spark-kafka-stream/src/main/java/kafka/streaming/ConsumerThread.java: Recompile with -Xlint
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ spark-kafka-stream ---
[WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform dependent!
[INFO] skip non existing resourceDirectory /home/cloudera/Documents/workspace/spark-kafka-stream/src/test/resources
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ spark-kafka-stream ---
[INFO] No sources to compile
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ spark-kafka-stream ---
[INFO] No tests to run.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ spark-kafka-stream ---
[INFO] Building jar: /home/cloudera/Documents/workspace/spark-kafka-stream/target/spark-kafka-stream-0.0.1.jar
[INFO] --
[INFO] BUILD SUCCESS
[INFO] --
[INFO] Total time: 6.487 s
[INFO] Finished at: 2016-03-31T20:05:53-07:00
[INFO] Final Memory: 34M/81M
[INFO] --

Run from the command line:

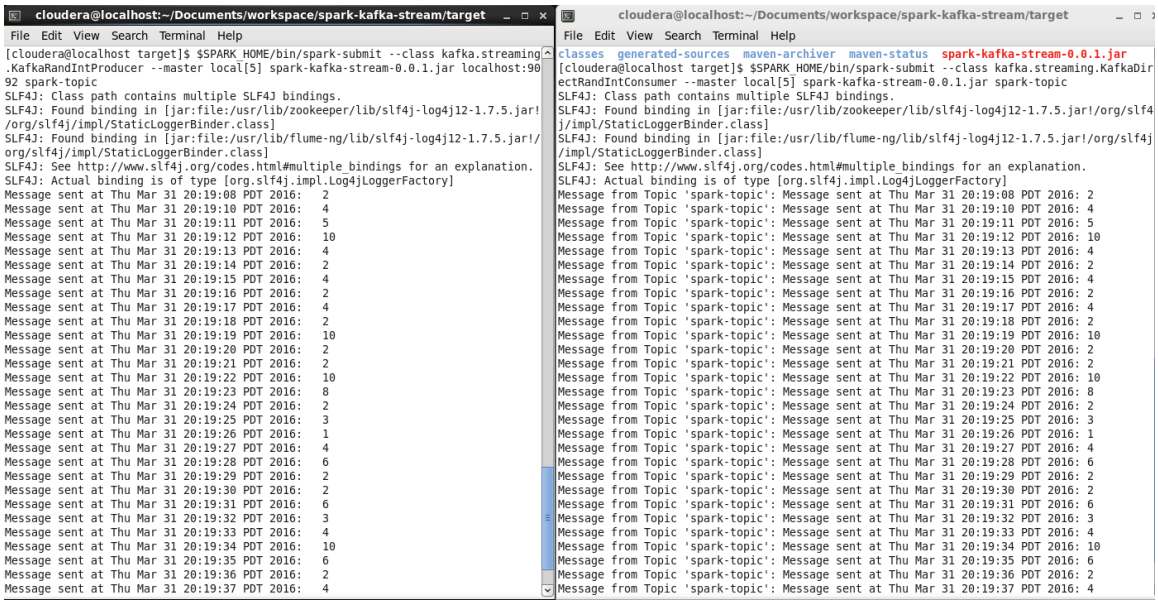
Launch Producer:

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaRandIntProducer --master local[5] spark-kafka-stream-
0.0.1.jar localhost:9092 spark-topic
```

Launch Consumer:

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaDirectRandIntConsumer --master local[5] spark-kafka-
stream-0.0.1.jar spark-topic
```

Set master nodes to at least 5 since we have 4 topics partitions and 1 current running application.



Left window shows the messages sent by Producer. I attached each message with a time stamp. Right window shows the messages consumed from Consumer. They are exactly the same. Below are the large snapshots of each window. What we got is just what we sent!

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class kafka.streaming
.KafkaRandIntProducer --master local[5] spark-kafka-stream-0.0.1.jar localhost:90
92 spark-topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!
/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.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]
Message sent at Thu Mar 31 20:19:08 PDT 2016:   2
Message sent at Thu Mar 31 20:19:10 PDT 2016:   4
Message sent at Thu Mar 31 20:19:11 PDT 2016:   5
Message sent at Thu Mar 31 20:19:12 PDT 2016:   10
Message sent at Thu Mar 31 20:19:13 PDT 2016:   4
Message sent at Thu Mar 31 20:19:14 PDT 2016:   2
Message sent at Thu Mar 31 20:19:15 PDT 2016:   4
Message sent at Thu Mar 31 20:19:16 PDT 2016:   2
Message sent at Thu Mar 31 20:19:17 PDT 2016:   4
Message sent at Thu Mar 31 20:19:18 PDT 2016:   2
Message sent at Thu Mar 31 20:19:19 PDT 2016:   10
Message sent at Thu Mar 31 20:19:20 PDT 2016:   2
Message sent at Thu Mar 31 20:19:21 PDT 2016:   2
Message sent at Thu Mar 31 20:19:22 PDT 2016:   10
Message sent at Thu Mar 31 20:19:23 PDT 2016:   8
Message sent at Thu Mar 31 20:19:24 PDT 2016:   2
Message sent at Thu Mar 31 20:19:25 PDT 2016:   3
Message sent at Thu Mar 31 20:19:26 PDT 2016:   1
Message sent at Thu Mar 31 20:19:27 PDT 2016:   4
Message sent at Thu Mar 31 20:19:28 PDT 2016:   6
Message sent at Thu Mar 31 20:19:29 PDT 2016:   2
Message sent at Thu Mar 31 20:19:30 PDT 2016:   2
Message sent at Thu Mar 31 20:19:31 PDT 2016:   6
Message sent at Thu Mar 31 20:19:32 PDT 2016:   3
Message sent at Thu Mar 31 20:19:33 PDT 2016:   4
Message sent at Thu Mar 31 20:19:34 PDT 2016:   10
Message sent at Thu Mar 31 20:19:35 PDT 2016:   6
Message sent at Thu Mar 31 20:19:36 PDT 2016:   2
Message sent at Thu Mar 31 20:19:37 PDT 2016:   4
```

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class kafka.streaming.KafkaDir
ectRandIntConsumer --master local[5] spark-kafka-stream-0.0.1.jar spark-topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/slf4
j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.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]
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:08 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:10 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:11 PDT 2016: 5
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:12 PDT 2016: 10
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:13 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:14 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:15 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:16 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:17 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:18 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:19 PDT 2016: 10
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:20 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:21 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:22 PDT 2016: 10
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:23 PDT 2016: 8
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:24 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:25 PDT 2016: 3
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:26 PDT 2016: 1
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:27 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:28 PDT 2016: 6
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:29 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:30 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:31 PDT 2016: 6
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:32 PDT 2016: 3
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:33 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:34 PDT 2016: 10
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:35 PDT 2016: 6
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:36 PDT 2016: 2
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:37 PDT 2016: 4
Message from Topic 'spark-topic': Message sent at Thu Mar 31 20:19:38 PDT 2016: 5
```

Check the log files:

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-
0.8.2.2/bin/kafka-run-class.sh kafka.tools.DumpLogSegments --files
/home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-topic-
0/00000000000000000000.log --print-data-log
```

We can see that all messages go to "spark-topic-2".

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-run-class.
sh kafka.tools.DumpLogSegments --files /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-
topic-2/00000000000000000000.log --print-data-log
Dumping /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-topic-2/00000000000000000000.lo
g
Starting offset: 0
offset: 0 position: 0 isvalid: true payloadsize: 205 magic: 0 compresscodec: NoCompressionCodec c
rc: 3215497133 payload: 66.249.67.3 - - [20/Jul/2009:20:12:25 -0700] "GET /gallery/main.php?g2_it
emId=15741&g2_fromNavId=x8fa12efc HTTP/1.1" 200 8068 "-" "Mozilla/5.0 (compatible; Googlebot/2.1;
 +http://www.google.com/bot.html)"
```

```
offset: 5 position: 1136 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 3145935320 payload: Message sent at Thu Mar 31 20:19:08 PDT 2016: 2
offset: 6 position: 1209 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 2515320566 payload: Message sent at Thu Mar 31 20:19:10 PDT 2016: 4
offset: 7 position: 1282 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 1065179109 payload: Message sent at Thu Mar 31 20:19:11 PDT 2016: 5
offset: 8 position: 1355 isvalid: true payloadsize: 48 magic: 0 compresscodec: NoCompressionCodec
 crc: 532358342 payload: Message sent at Thu Mar 31 20:19:12 PDT 2016:  10
offset: 9 position: 1429 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 673632824 payload: Message sent at Thu Mar 31 20:19:13 PDT 2016:  4
offset: 10 position: 1502 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3157684565 payload: Message sent at Thu Mar 31 20:19:14 PDT 2016:        2
offset: 11 position: 1575 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 2294484453 payload: Message sent at Thu Mar 31 20:19:15 PDT 2016:        4
offset: 12 position: 1648 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3697984542 payload: Message sent at Thu Mar 31 20:19:16 PDT 2016:        2
offset: 13 position: 1721 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3902781614 payload: Message sent at Thu Mar 31 20:19:17 PDT 2016:        4
offset: 14 position: 1794 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 646760622 payload: Message sent at Thu Mar 31 20:19:18 PDT 2016: 2
offset: 15 position: 1867 isvalid: true payloadsize: 48 magic: 0 compresscodec: NoCompressionCode
c crc: 1350770475 payload: Message sent at Thu Mar 31 20:19:19 PDT 2016:        10
offset: 16 position: 1941 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 15613464 payload: Message sent at Thu Mar 31 20:19:20 PDT 2016:  2
offset: 17 position: 2014 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3715688349 payload: Message sent at Thu Mar 31 20:19:21 PDT 2016:        2
offset: 18 position: 2087 isvalid: true payloadsize: 48 magic: 0 compresscodec: NoCompressionCode
c crc: 247877311 payload: Message sent at Thu Mar 31 20:19:22 PDT 2016: 10
offset: 19 position: 2161 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1576123336 payload: Message sent at Thu Mar 31 20:19:23 PDT 2016:        8
```

4. Build the Producer using our customized Partitioner, and the Consumer with multi-threads consuming from the same topic.

KafkaRandIntProducerPT.java:

```
package kafka.streaming;

import java.util.*;

import kafka.javaapi.producer.Producer;
import kafka.producer.KeyedMessage;
import kafka.producer.ProducerConfig;

public class KafkaRandIntProducerPT {
```

```
  public static void main(String[] args) {
        if (args.length < 2) {
              System.err.println("Usage: KafkaRandIntProducer <brokers> <topic> [<events>]\n" +
                          "  <brokers> is a list of one or more Kafka brokers\n" +
                          "  <topic> is the kafka topic to produce to \n" +
                          "  <events> is the total number of events to send \n");
              System.exit(1);
        }

        String brokers = args[0];
        String topic = args[1];

        /*
        long events = 50000;
        if (args.length == 3) {
              events = Long.parseLong(args[2]);
        }
        */

        Random rnd = new Random();

        Properties props = new Properties();
        props.put("metadata.broker.list", brokers);
        props.put("partitioner.class", "kafka.streaming.SimplePartitioner");
        props.put("serializer.class", "kafka.serializer.StringEncoder");

        ProducerConfig config = new ProducerConfig(props);

        Producer<String, String> producer = new Producer<String, String>(config);

        // for (long nEvents = 0; nEvents < events; nEvents++) {
        while (true) {
              // Generate a random number between 1 to 10
              int number = rnd.nextInt(10) + 1;

              // Get current time stamp
              Date time = new Date(System.currentTimeMillis());

              // Compose the message
              String msg = "Message sent at " + time.toString() + ":\t" + number;

              // Send the message to topic
              KeyedMessage<String, String> data = new KeyedMessage<String, String>(topic,
Integer.toString(number), msg);
              producer.send(data);

              System.out.println(msg);

              try {
                  // Sleep for 1 second
                  Thread.sleep(1000);
              } catch(InterruptedException ex) {
                  Thread.currentThread().interrupt();
```

```
                }
        }

        /*
        if (producer != null) {
                producer.close();
        }
        */
   }
}
```

Define our own Partitioner class, use the number as the partition key. The partition function is basically does a modulo operation on the number of partitions defined within Kafka for the topic. The benefit of this partitioning logic is the same number end up in the same Partition.

SimplePartitioner.java:

```
package kafka.streaming;

import kafka.producer.Partitioner;
import kafka.utils.VerifiableProperties;

public class SimplePartitioner implements Partitioner {
   public SimplePartitioner (VerifiableProperties props) {

   }

   public int partition(Object number, int numPartitions) {
          int partition = 0;

          String str = (String) number;
          int key = Integer.parseInt(str);

          if (key > 0) {
                partition = key % numPartitions;
          }

          return partition;
   }

}
```

KafkaDirectRandIntConsumerMT.java:

```
package kafka.streaming;

import kafka.consumer.Consumer;
import kafka.consumer.ConsumerConfig;
import kafka.consumer.KafkaStream;
import kafka.javaapi.consumer.ConsumerConnector;
```

```
import java.util.HashMap;
import java.util.List;
import java.util.Map;
import java.util.Properties;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.Executors;

public class KafkaDirectRandIntConsumerMT {

   private final ConsumerConnector consumer;
   private final String topic;
   private ExecutorService executor;

   public KafkaDirectRandIntConsumerMT (String topic) {

          Properties props = new Properties();
          props.put("zookeeper.connect", "localhost:2181");
          props.put("group.id", "spark-app");
          props.put("zookeeper.session.timeout.ms", "400");
          props.put("zookeeper.sync.time.ms", "200");
          props.put("auto.commit.interval.ms", "1000");

          consumer = Consumer.createJavaConsumerConnector(new ConsumerConfig(props));

          this.topic = topic;
   }

   public void run(int numThreads) {
          Map<String, Integer> topicCountMap = new HashMap<String, Integer>();
          topicCountMap.put(topic, new Integer(numThreads));

          Map<String, List<KafkaStream<byte[], byte[]>>> consumerMap =
consumer.createMessageStreams(topicCountMap);
          List<KafkaStream<byte[], byte[]>> streams = consumerMap.get(topic);

          // now launch all the threads
          executor = Executors.newFixedThreadPool(numThreads);

          // now create an object to consume the messages
          int threadNumber = 0;
          for (final KafkaStream stream : streams) {
                 System.out.println("Create thread " + threadNumber);
                 executor.submit(new ConsumerThread(stream, threadNumber));
                 threadNumber++;
          }
   }

   public static void main(String[] args) {
          String topic = args[0];
          int threads = Integer.parseInt(args[1]);

          KafkaDirectRandIntConsumerMT example = new KafkaDirectRandIntConsumerMT(topic);
```

```
        example.run(threads);
    }
}
```

The High Level Consumer is a multi-threaded application. The threading model revolves around the number of partitions in the topic and there are some very specific rules:

- If threads > partitions on the topic, some threads will never see a message.
- If threads < partitions, some threads will receive data from multiple partitions.
- Adding more processes/threads will cause Kafka to re-balance, possibly changing the assignment of a Partition to a Thread.

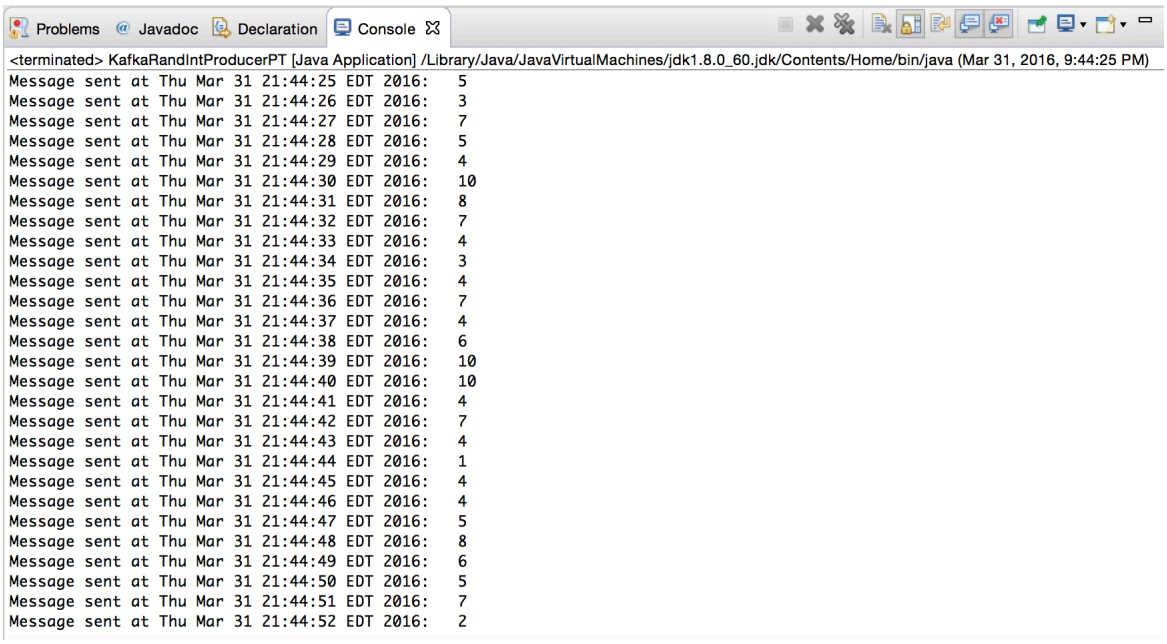
Function **run()** creates a thread pool and passes a new ConsumerTest object to each thread. The HashMap that tells Kafka how many threads we are providing for which topics.

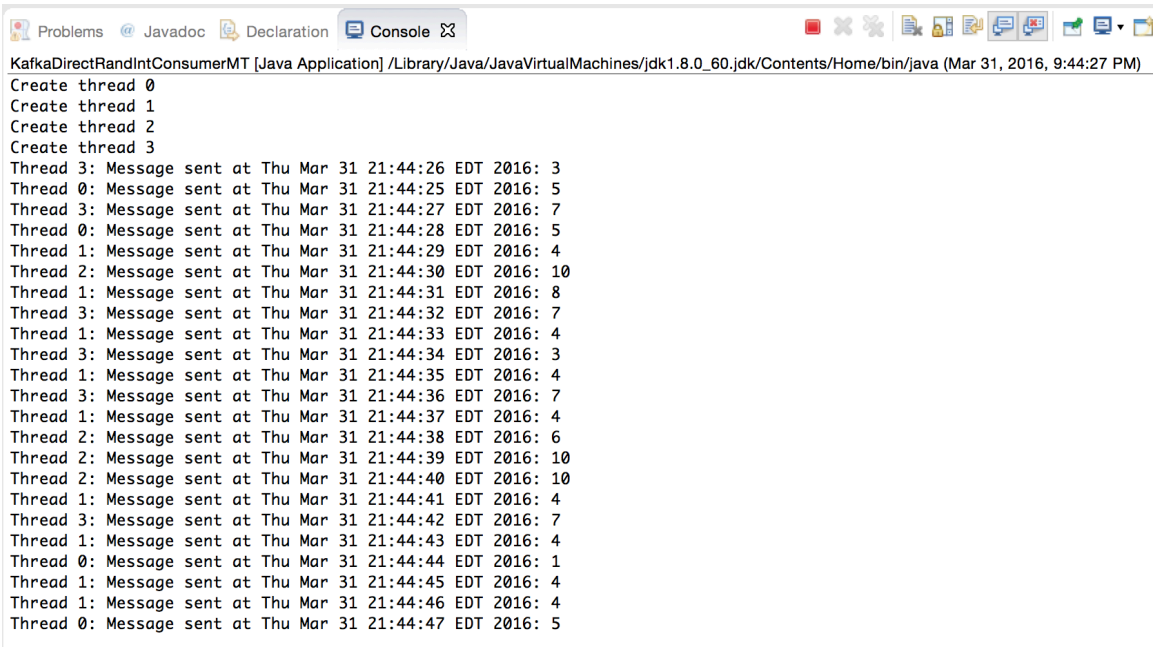
Here the number of partitions and the number of threads are both 4. So the Kafka system is almost in balance.

ConsumerThread.java:

```
package kafka.streaming;

import kafka.consumer.ConsumerIterator;
import kafka.consumer.KafkaStream;

public class ConsumerThread implements Runnable {
    private KafkaStream stream;
    private int threadNumber;

    public ConsumerThread(KafkaStream stream, int threadNumber) {
        this.threadNumber = threadNumber;
        this.stream = stream;
    }

    public void run() {
        ConsumerIterator<byte[], byte[]> it = stream.iterator();
        while (it.hasNext()) {
            System.out.println("Thread " + threadNumber + ": " + new String(it.next().message()));
        }
    }
}
```

Below are the results running from the Eclipse on MacOS. From the second window, we can see messages are consumed by **different threads**.

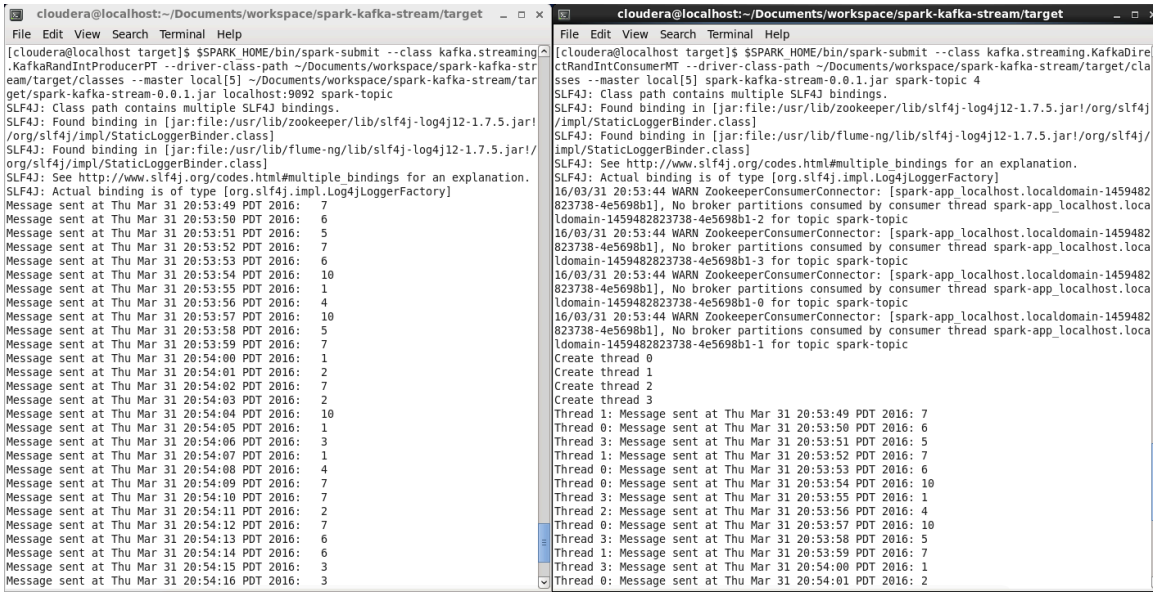




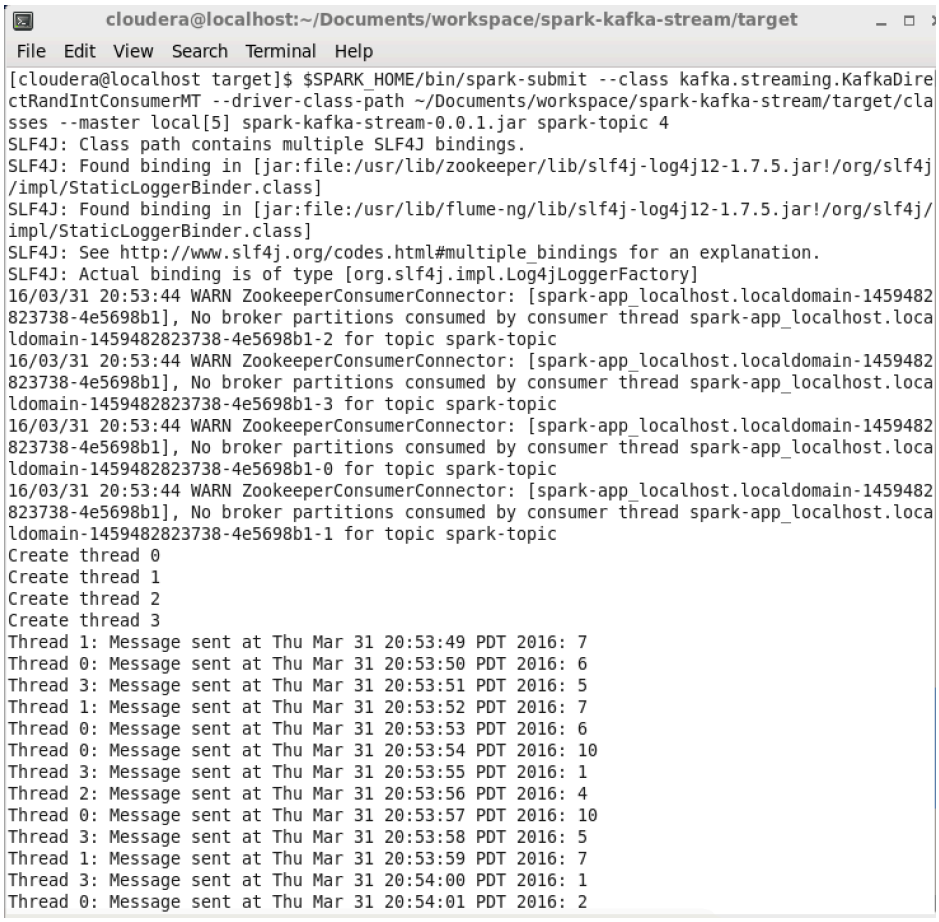
Run from the VM CentOS 6.7:

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaRandIntProducerPT --driver-class-path
~/Documents/workspace/spark-kafka-stream/target/classes --master local[5]
~/Documents/workspace/spark-kafka-stream/target/spark-kafka-stream-0.0.1.jar
localhost:9092 spark-topic

[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaDirectRandIntConsumerMT --driver-class-path
```

~/Documents/workspace/spark-kafka-stream/target/classes --master local[5] spark-kafka-stream-0.0.1.jar spark-topic 4



[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class kafka.streaming.KafkaRandIntProducerPT --driver-class-path ~/Documents/workspace/spark-kafka-stream/target/classes --master local[5] ~/Documents/workspace/spark-kafka-stream/target/spark-kafka-stream-0.0.1.jar localhost:9092 spark-topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.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]
Message sent at Thu Mar 31 20:53:49 PDT 2016: 7
Message sent at Thu Mar 31 20:53:50 PDT 2016: 6
Message sent at Thu Mar 31 20:53:51 PDT 2016: 5
Message sent at Thu Mar 31 20:53:52 PDT 2016: 7
Message sent at Thu Mar 31 20:53:53 PDT 2016: 6
Message sent at Thu Mar 31 20:53:54 PDT 2016: 10
Message sent at Thu Mar 31 20:53:55 PDT 2016: 1
Message sent at Thu Mar 31 20:53:56 PDT 2016: 4
Message sent at Thu Mar 31 20:53:57 PDT 2016: 10
Message sent at Thu Mar 31 20:53:58 PDT 2016: 5
Message sent at Thu Mar 31 20:53:59 PDT 2016: 7
Message sent at Thu Mar 31 20:54:00 PDT 2016: 1
Message sent at Thu Mar 31 20:54:01 PDT 2016: 2
Message sent at Thu Mar 31 20:54:02 PDT 2016: 7
Message sent at Thu Mar 31 20:54:03 PDT 2016: 2
Message sent at Thu Mar 31 20:54:04 PDT 2016: 10
Message sent at Thu Mar 31 20:54:05 PDT 2016: 1
Message sent at Thu Mar 31 20:54:06 PDT 2016: 3
Message sent at Thu Mar 31 20:54:07 PDT 2016: 1
Message sent at Thu Mar 31 20:54:08 PDT 2016: 4
Message sent at Thu Mar 31 20:54:09 PDT 2016: 7
Message sent at Thu Mar 31 20:54:10 PDT 2016: 7
Message sent at Thu Mar 31 20:54:11 PDT 2016: 2
Message sent at Thu Mar 31 20:54:12 PDT 2016: 7
Message sent at Thu Mar 31 20:54:13 PDT 2016: 6
Message sent at Thu Mar 31 20:54:14 PDT 2016: 6
Message sent at Thu Mar 31 20:54:15 PDT 2016: 3
Message sent at Thu Mar 31 20:54:16 PDT 2016: 3




```
Thread 1: Message sent at Thu Mar 31 20:54:02 PDT 2016: 7
Thread 0: Message sent at Thu Mar 31 20:54:03 PDT 2016: 2
Thread 0: Message sent at Thu Mar 31 20:54:04 PDT 2016: 10
Thread 3: Message sent at Thu Mar 31 20:54:05 PDT 2016: 1
Thread 1: Message sent at Thu Mar 31 20:54:06 PDT 2016: 3
Thread 3: Message sent at Thu Mar 31 20:54:07 PDT 2016: 1
Thread 2: Message sent at Thu Mar 31 20:54:08 PDT 2016: 4
Thread 1: Message sent at Thu Mar 31 20:54:09 PDT 2016: 7
Thread 1: Message sent at Thu Mar 31 20:54:10 PDT 2016: 7
Thread 0: Message sent at Thu Mar 31 20:54:11 PDT 2016: 2
Thread 1: Message sent at Thu Mar 31 20:54:12 PDT 2016: 7
Thread 0: Message sent at Thu Mar 31 20:54:13 PDT 2016: 6
Thread 0: Message sent at Thu Mar 31 20:54:14 PDT 2016: 6
Thread 1: Message sent at Thu Mar 31 20:54:15 PDT 2016: 3
Thread 1: Message sent at Thu Mar 31 20:54:16 PDT 2016: 3
Thread 3: Message sent at Thu Mar 31 20:54:17 PDT 2016: 5
Thread 1: Message sent at Thu Mar 31 20:54:18 PDT 2016: 3
Thread 0: Message sent at Thu Mar 31 20:54:19 PDT 2016: 6
Thread 3: Message sent at Thu Mar 31 20:54:20 PDT 2016: 5
```

Check log files:

Number 4, 8 always go to spark-topic-0:

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-run-class.sh kafka.tools.DumpLogSegments --files /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-topic-0/00000000000000000000.log --print-data-log
```

```
offset: 8 position: 1321 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 1509853096 keysize: 1 key: 8 payload: Message sent at Thu Mar 31 20:52:30 PDT 2016:       8
offset: 9 position: 1395 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 583062758 keysize: 1 key: 4 payload: Message sent at Thu Mar 31 20:53:56 PDT 2016:        4
offset: 10 position: 1469 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 2709278286 keysize: 1 key: 4 payload: Message sent at Thu Mar 31 20:54:08 PDT 2016:      4
offset: 11 position: 1543 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 715809780 keysize: 1 key: 8 payload: Message sent at Thu Mar 31 20:54:23 PDT 2016:       8
offset: 12 position: 1617 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 931516647 keysize: 1 key: 8 payload: Message sent at Thu Mar 31 20:54:26 PDT 2016:       8
```

Number 1, 5, 9 always go to spark-topic-1:

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-run-class.
sh kafka.tools.DumpLogSegments --files /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-
topic-1/00000000000000000000.log --print-data-log


offset: 13 position: 2641 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3381212728 keysize: 1 key: 1 payload: Message sent at Thu Mar 31 20:52:45 PDT 2016:      1
offset: 14 position: 2715 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1098391077 keysize: 1 key: 5 payload: Message sent at Thu Mar 31 20:53:51 PDT 2016:      5
offset: 15 position: 2789 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 4209919711 keysize: 1 key: 1 payload: Message sent at Thu Mar 31 20:53:55 PDT 2016:      1
offset: 16 position: 2863 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3337426125 keysize: 1 key: 5 payload: Message sent at Thu Mar 31 20:53:58 PDT 2016:      5
offset: 17 position: 2937 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 2660983764 keysize: 1 key: 1 payload: Message sent at Thu Mar 31 20:54:00 PDT 2016:      1
offset: 18 position: 3011 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 2209665223 keysize: 1 key: 1 payload: Message sent at Thu Mar 31 20:54:05 PDT 2016:      1
offset: 19 position: 3085 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3823658380 keysize: 1 key: 1 payload: Message sent at Thu Mar 31 20:54:07 PDT 2016:      1
offset: 20 position: 3159 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 97040022 keysize: 1 key: 5 payload: Message sent at Thu Mar 31 20:54:17 PDT 2016:        5
offset: 21 position: 3233 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 81433877 keysize: 1 key: 5 payload: Message sent at Thu Mar 31 20:54:20 PDT 2016:        5
offset: 22 position: 3307 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1413290532 keysize: 1 key: 9 payload: Message sent at Thu Mar 31 20:54:21 PDT 2016:      9
offset: 23 position: 3381 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3551006924 keysize: 1 key: 9 payload: Message sent at Thu Mar 31 20:54:28 PDT 2016:      9
```

Number 2, 6, 10 always go to spark-topic-2:

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-run-class.
sh kafka.tools.DumpLogSegments --files /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-
topic-2/00000000000000000000.log --print-data-log


offset: 47 position: 4221 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 4250799754 keysize: 1 key: 2 payload: Message sent at Thu Mar 31 20:54:11 PDT 2016:      2
offset: 48 position: 4295 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3861744045 keysize: 1 key: 6 payload: Message sent at Thu Mar 31 20:54:13 PDT 2016:      6
offset: 49 position: 4369 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 2606665717 keysize: 1 key: 6 payload: Message sent at Thu Mar 31 20:54:14 PDT 2016:      6
offset: 50 position: 4443 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 3698519947 keysize: 1 key: 6 payload: Message sent at Thu Mar 31 20:54:19 PDT 2016:      6
offset: 51 position: 4517 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1205499379 keysize: 1 key: 6 payload: Message sent at Thu Mar 31 20:54:22 PDT 2016:      6
offset: 52 position: 4591 isvalid: true payloadsize: 48 magic: 0 compresscodec: NoCompressionCode
c crc: 822379508 keysize: 2 key: 10 payload: Message sent at Thu Mar 31 20:54:24 PDT 2016:      1
0
offset: 53 position: 4667 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 567997580 keysize: 1 key: 2 payload: Message sent at Thu Mar 31 20:54:27 PDT 2016:       2
```

Number 3, 7 always go to spark-topic-:

```
[cloudera@localhost kafka-logs]$ /home/cloudera/Documents/kafka_2.11-0.8.2.2/bin/kafka-run-class.
sh kafka.tools.DumpLogSegments --files /home/cloudera/Documents/hqiu/kafka-data/kafka-logs/spark-
topic-3/00000000000000000000.log --print-data-log

offset: 43 position: 4847 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 65611071 keysize: 1 key: 3 payload: Message sent at Thu Mar 31 20:54:06 PDT 2016:        3
offset: 44 position: 4921 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1605812838 keysize: 1 key: 7 payload: Message sent at Thu Mar 31 20:54:09 PDT 2016:      7
offset: 45 position: 4995 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1160522232 keysize: 1 key: 7 payload: Message sent at Thu Mar 31 20:54:10 PDT 2016:      7
offset: 46 position: 5069 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 628137139 keysize: 1 key: 7 payload: Message sent at Thu Mar 31 20:54:12 PDT 2016:       7
offset: 47 position: 5143 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 590129287 keysize: 1 key: 3 payload: Message sent at Thu Mar 31 20:54:15 PDT 2016:       3
offset: 48 position: 5217 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 2665923657 keysize: 1 key: 3 payload: Message sent at Thu Mar 31 20:54:16 PDT 2016:      3
offset: 49 position: 5291 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 1677773049 keysize: 1 key: 3 payload: Message sent at Thu Mar 31 20:54:18 PDT 2016:      3
offset: 50 position: 5365 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCode
c crc: 610451248 keysize: 1 key: 7 payload: Message sent at Thu Mar 31 20:54:25 PDT 2016:       7
```

5. Create Kafka producer and consumer in Python.

Steps to install kafka-python:

```
cloudera@localhost ~]$ sudo rpm -ivh
http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm

[cloudera@localhost ~]$ sudo yum install -y python-pip

[cloudera@localhost ~]$ pip -V

[cloudera@localhost ~]$ pip list

[cloudera@localhost ~]$ sudo pip install kafka-python
```

kafka_python_producer.py:

```
from kafka import KafkaProducer
from datetime import datetime
from random import randint
import time

producer = KafkaProducer(bootstrap_servers='localhost:9092')

while True:
        num = randint(1, 10)
        message = "message sent at " + str(datetime.now()) + ":\t" + str(num)
        print message
        producer.send('spark-topic', message)
        time.sleep(1)

print 'Done sending messages'
```

kafka_python_consumer.py:

```
from kafka import KafkaConsumer

consumer = KafkaConsumer('spark-topic')
for msg in consumer:
        print msg
```

```
[cloudera@localhost hw08]$ python kafka_python_producer.py
message sent at 2016-03-31 21:55:20.229974:     2
message sent at 2016-03-31 21:55:21.232484:     3
message sent at 2016-03-31 21:55:22.234441:     10
message sent at 2016-03-31 21:55:23.235812:     4
message sent at 2016-03-31 21:55:24.237611:     6
message sent at 2016-03-31 21:55:25.239497:     9
message sent at 2016-03-31 21:55:26.241618:     5
message sent at 2016-03-31 21:55:27.244348:     3
message sent at 2016-03-31 21:55:28.246675:     1
message sent at 2016-03-31 21:55:29.248647:     7
message sent at 2016-03-31 21:55:30.251033:     5
message sent at 2016-03-31 21:55:31.253693:     8
message sent at 2016-03-31 21:55:32.256527:     2
message sent at 2016-03-31 21:55:33.258499:     10
message sent at 2016-03-31 21:55:34.260629:     6
message sent at 2016-03-31 21:55:35.262974:     8
message sent at 2016-03-31 21:55:36.265431:     1
```

```
[cloudera@localhost hw08]$ python kafka_python_consumer.py
ConsumerRecord(topic=u'spark-topic', partition=0, offset=13, key=None, value='message s
ent at 2016-03-31 21:55:20.229974:\t2')
ConsumerRecord(topic=u'spark-topic', partition=1, offset=84, key=None, value='message s
ent at 2016-03-31 21:55:21.232484:\t3')
ConsumerRecord(topic=u'spark-topic', partition=1, offset=85, key=None, value='message s
ent at 2016-03-31 21:55:22.234441:\t10')
ConsumerRecord(topic=u'spark-topic', partition=2, offset=140, key=None, value='message
sent at 2016-03-31 21:55:23.235812:\t4')
ConsumerRecord(topic=u'spark-topic', partition=1, offset=86, key=None, value='message s
ent at 2016-03-31 21:55:24.237611:\t6')
ConsumerRecord(topic=u'spark-topic', partition=0, offset=14, key=None, value='message s
ent at 2016-03-31 21:55:25.239497:\t9')
ConsumerRecord(topic=u'spark-topic', partition=3, offset=51, key=None, value='message s
ent at 2016-03-31 21:55:26.241618:\t5')
ConsumerRecord(topic=u'spark-topic', partition=2, offset=141, key=None, value='message
sent at 2016-03-31 21:55:27.244348:\t3')
ConsumerRecord(topic=u'spark-topic', partition=0, offset=15, key=None, value='message s
ent at 2016-03-31 21:55:28.246675:\t1')
ConsumerRecord(topic=u'spark-topic', partition=1, offset=87, key=None, value='message s
ent at 2016-03-31 21:55:29.248647:\t7')
ConsumerRecord(topic=u'spark-topic', partition=2, offset=142, key=None, value='message
sent at 2016-03-31 21:55:30.251033:\t5')
ConsumerRecord(topic=u'spark-topic', partition=3, offset=52, key=None, value='message s
ent at 2016-03-31 21:55:31.253693:\t8')
ConsumerRecord(topic=u'spark-topic', partition=1, offset=88, key=None, value='message s
ent at 2016-03-31 21:55:32.256527:\t2')
ConsumerRecord(topic=u'spark-topic', partition=3, offset=53, key=None, value='message s
ent at 2016-03-31 21:55:33.258499:\t10')
ConsumerRecord(topic=u'spark-topic', partition=3, offset=54, key=None, value='message s
ent at 2016-03-31 21:55:34.260629:\t6')
ConsumerRecord(topic=u'spark-topic', partition=0, offset=16, key=None, value='message s
ent at 2016-03-31 21:55:35.262974:\t8')
ConsumerRecord(topic=u'spark-topic', partition=0, offset=17, key=None, value='message s
ent at 2016-03-31 21:55:36.265431:\t1')
```

Check log files for 4 partitions:

In Python, even we haven't specified the partitioner, the messages are automatically distributed to different partitions randomly.

```
offset: 13 position: 1691 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 2290468257 payload: message sent at 2016-03-31 21:55:20.229974:   2
offset: 14 position: 1762 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 3014767268 payload: message sent at 2016-03-31 21:55:25.239497:   9
offset: 15 position: 1833 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 3531790720 payload: message sent at 2016-03-31 21:55:28.246675:   1
offset: 16 position: 1904 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 82810514 payload: message sent at 2016-03-31 21:55:35.262974:     8
offset: 17 position: 1975 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 1605776277 payload: message sent at 2016-03-31 21:55:36.265431:   1
```

```
offset: 81 position: 7624 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 795966280 payload: Message sent at Thu Mar 31 21:42:18 PDT 2016:  7
offset: 82 position: 7697 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 358584266 payload: Message sent at Thu Mar 31 21:42:19 PDT 2016:  9
offset: 83 position: 7770 isvalid: true payloadsize: 47 magic: 0 compresscodec: NoCompressionCodec
 crc: 2038004081 payload: Message sent at Thu Mar 31 21:42:20 PDT 2016: 2
offset: 84 position: 7843 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 3463445148 payload: message sent at 2016-03-31 21:55:21.232484:   3
offset: 85 position: 7914 isvalid: true payloadsize: 46 magic: 0 compresscodec: NoCompressionCodec
 crc: 2328003813 payload: message sent at 2016-03-31 21:55:22.234441:   10
offset: 86 position: 7986 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 3932995262 payload: message sent at 2016-03-31 21:55:24.237611:   6
offset: 87 position: 8057 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 1207528051 payload: message sent at 2016-03-31 21:55:29.248647:   7
offset: 88 position: 8128 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 441887002 payload: message sent at 2016-03-31 21:55:32.256527:    2
```

```
offset: 140 position: 11028 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2457408934 payload: message sent at 2016-03-31 21:55:23.235812: 4
offset: 141 position: 11099 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3723259645 payload: message sent at 2016-03-31 21:55:27.244348: 3
offset: 142 position: 11170 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 4214658210 payload: message sent at 2016-03-31 21:55:30.251033: 5
```

```
offset: 51 position: 5439 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 1058284908 payload: message sent at 2016-03-31 21:55:26.241618:   5
offset: 52 position: 5510 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 1333779786 payload: message sent at 2016-03-31 21:55:31.253693:   8
offset: 53 position: 5581 isvalid: true payloadsize: 46 magic: 0 compresscodec: NoCompressionCodec
 crc: 1688872405 payload: message sent at 2016-03-31 21:55:33.258499:   10
offset: 54 position: 5653 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCodec
 crc: 2124058628 payload: message sent at 2016-03-31 21:55:34.260629:   6
```

Problem 3) Starting from one of the attached Spark Streaming clients `DirectKafkaWordCount` in Java, Scala or Python write a consumer client that will replace the consumer from the previous problem. However, rather than simply printing every message it receives from the producer, let it print for us every 5 seconds the rolling count of numbers between 1 and 10 it received in the last 30 seconds. You might find it simpler to print to files and then examine those files afterwards. For Java build simple Maven Project with a single Java class and `pom.xml` file similar to the one provided. Build your projects following the process we used in Assignment 4.

You are welcome to follow any other instructions and use any other programming or scripting language to accomplish the above goals.

Solution

Source Code Manifest

Java Solution	KafkaWindowRandIntConsumer.java
Python Solution	kafka_window_consumer.py

1. Java code.

KafkaWindowRandIntConsumer.java:

```
package kafka.streaming;

import java.util.HashMap;
import java.util.HashSet;
import java.util.Arrays;
import java.util.regex.Pattern;

import scala.Tuple2;

import com.google.common.collect.Lists;
import kafka.serializer.StringDecoder;

import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaSparkContext;
import org.apache.spark.api.java.function.*;
import org.apache.spark.streaming.api.java.*;
import org.apache.spark.streaming.kafka.KafkaUtils;
import org.apache.spark.streaming.Durations;

public final class KafkaWindowRandIntConsumer {
   private static final Pattern TAB = Pattern.compile("\t");

   public static void main(String[] args) {
          if (args.length < 2) {
                System.err.println("Usage: KafkaWindowRandIntConsumer <brokers> <topics>
[<batch_interval> <window_duration> <sliding_window_duration>]\n" +
                                  "  <brokers> is a list of one or more Kafka brokers\n" +
                                  "  <topics> is a list of one or more kafka topics to consume
from\n\n");
                System.exit(1);
          }

          String brokers = args[0];
          String topics = args[1];

          int batchIntervalInSeconds = 1;
          int windowDurationInSeconds = 30;
```

```
        int slidingWindowDurationInSeconds = 5;

        if (args.length == 5) {
               batchIntervalInSeconds = Integer.parseInt(args[2]);
               windowDurationInSeconds = Integer.parseInt(args[3]);
               slidingWindowDurationInSeconds = Integer.parseInt(args[4]);
        }

        // Create context with a 1 second batch interval
        JavaSparkContext sparkConf = new JavaSparkContext("local[5]",
"KafkaWindowRandIntConsumer");
        JavaStreamingContext jssc = new JavaStreamingContext(sparkConf,
Durations.seconds(batchIntervalInSeconds));

        HashSet<String> topicsSet = new HashSet<String>(Arrays.asList(topics.split(",")));
        HashMap<String, String> kafkaParams = new HashMap<String, String>();
        kafkaParams.put("metadata.broker.list", brokers);
        kafkaParams.put("zookeeper.connect", "localhost:2181");
        kafkaParams.put("group.id", "spark-app");
        System.out.println("Kafka parameters: " + kafkaParams);
        System.out.println("KafkaWindowLogCount parameters: topics=" + topics +
                       "; batchIntervalInSeconds=" + batchIntervalInSeconds +
                       "; windowDurationInSeconds=" + windowDurationInSeconds +
                       "; slidingWindowDurationInSeconds=" + slidingWindowDurationInSeconds);

        // Create direct kafka stream with brokers and topics
        JavaPairInputDStream<String, String> messages = KafkaUtils.createDirectStream(
                       jssc,
                       String.class,
                       String.class,
                       StringDecoder.class,
                       StringDecoder.class,
                       kafkaParams,
                       topicsSet
                       );

        // Get the lines, split them into words
        JavaDStream<String> lines = messages.map(new Function<Tuple2<String, String>, String>() {
               @Override
               public String call(Tuple2<String, String> tuple2) {
                       System.out.println("processing lines: " + tuple2._2());
                       return tuple2._2();
               }
        });

        JavaDStream<String> words = lines.flatMap(new FlatMapFunction<String, String>() {
               @Override
               public Iterable<String> call(String x) {
                       return Lists.newArrayList(TAB.split(x)[1]);
               }
        });

        // Count each IP in each batch
```

```
        JavaPairDStream<String, Integer> pairs = words.mapToPair(
                        new PairFunction<String, String, Integer>() {
                                @Override public Tuple2<String, Integer> call(String s) {
                                        return new Tuple2<String, Integer>(s, 1);
                                }
                        });

        // Reduce function adding two integers, defined separately for clarity
        Function2<Integer, Integer, Integer> reduceFunc = new Function2<Integer, Integer,
Integer>() {
                @Override public Integer call(Integer i1, Integer i2) {
                        return i1 + i2;
                }
        };

        // Reduce last windowDurationInSeconds seconds of data, every
slidingWindowDurationInSeconds seconds
        JavaPairDStream<String, Integer> windowedWordCounts = pairs.reduceByKeyAndWindow(
                        reduceFunc, Durations.seconds(windowDurationInSeconds),
Durations.seconds(slidingWindowDurationInSeconds));

        windowedWordCounts.print();

        // Start the computation
        jssc.start();
        jssc.awaitTermination();
   }
}
```

Launch it from command line:

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaWindowRandIntConsumer --master local[5] spark-kafka-
stream-0.0.1.jar localhost:9092 spark-topic
```

Results:

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class kafka.streaming.K
afkaRandIntProducer --master local[5] spark-kafka-stream-0.0.1.jar localhost:9092 s
park-topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/o
rg/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!/or
g/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]
Message sent at Thu Mar 31 21:46:37 PDT 2016:   1
Message sent at Thu Mar 31 21:46:38 PDT 2016:   9
Message sent at Thu Mar 31 21:46:39 PDT 2016:   2
Message sent at Thu Mar 31 21:46:40 PDT 2016:   5
Message sent at Thu Mar 31 21:46:41 PDT 2016:   6
Message sent at Thu Mar 31 21:46:42 PDT 2016:   1
Message sent at Thu Mar 31 21:46:43 PDT 2016:   8
Message sent at Thu Mar 31 21:46:44 PDT 2016:   1
Message sent at Thu Mar 31 21:46:45 PDT 2016:   7
Message sent at Thu Mar 31 21:46:46 PDT 2016:   6
Message sent at Thu Mar 31 21:46:47 PDT 2016:   5
Message sent at Thu Mar 31 21:46:48 PDT 2016:   6
Message sent at Thu Mar 31 21:46:49 PDT 2016:   1
Message sent at Thu Mar 31 21:46:50 PDT 2016:   9
Message sent at Thu Mar 31 21:46:51 PDT 2016:   8
Message sent at Thu Mar 31 21:46:52 PDT 2016:   8
Message sent at Thu Mar 31 21:46:53 PDT 2016:   2
Message sent at Thu Mar 31 21:46:54 PDT 2016:   7
Message sent at Thu Mar 31 21:46:55 PDT 2016:   10
Message sent at Thu Mar 31 21:46:56 PDT 2016:   8
Message sent at Thu Mar 31 21:46:57 PDT 2016:   5
Message sent at Thu Mar 31 21:46:58 PDT 2016:   1
Message sent at Thu Mar 31 21:46:59 PDT 2016:   9
Message sent at Thu Mar 31 21:47:00 PDT 2016:   10
Message sent at Thu Mar 31 21:47:01 PDT 2016:   9
Message sent at Thu Mar 31 21:47:02 PDT 2016:   10
Message sent at Thu Mar 31 21:47:03 PDT 2016:   5
Message sent at Thu Mar 31 21:47:04 PDT 2016:   10
Message sent at Thu Mar 31 21:47:05 PDT 2016:   9
```

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class kafka.streaming.KafkaWi
ndowRandIntConsumer --master local[5] spark-kafka-stream-0.0.1.jar localhost:9092 spark-t
opic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/slf
4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!/org/slf4
j/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]
16/03/31 21:46:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your pl
atform... using builtin-java classes where applicable
16/03/31 21:46:40 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback
 address: 127.0.0.1; using 192.168.80.191 instead (on interface eth0)
16/03/31 21:46:40 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
16/03/31 21:46:42 WARN MetricsSystem: Using default name DAGScheduler for source because
spark.app.id is not set.
Kafka parameters: {zookeeper.connect=localhost:2181, group.id=spark-app, metadata.broker.
list=localhost:9092}
KafkaWindowLogCount parameters: topics=spark-topic; batchIntervalInSeconds=1; windowDurat
ionInSeconds=30; slidingWindowDurationInSeconds=5
[Stage 1:>                                                          (0 + 0) / 4]processin
g lines: Message sent at Thu Mar 31 21:46:44 PDT 2016:  1
processing lines: Message sent at Thu Mar 31 21:46:45 PDT 2016: 7
processing lines: Message sent at Thu Mar 31 21:46:46 PDT 2016: 6
processing lines: Message sent at Thu Mar 31 21:46:47 PDT 2016: 5
processing lines: Message sent at Thu Mar 31 21:46:48 PDT 2016: 6
-------------------------------------------
Time: 1459486009000 ms
-------------------------------------------
(7,1)
(5,1)
(6,2)
(1,1)

[Stage 16:>                                                         (0 + 0) / 4]processin
g lines: Message sent at Thu Mar 31 21:46:49 PDT 2016:  1
processing lines: Message sent at Thu Mar 31 21:46:50 PDT 2016: 9
processing lines: Message sent at Thu Mar 31 21:46:51 PDT 2016: 8
```

In the right window, to make it convenient to debug, I print out each message the Consumer has consumed.

```
Message sent at Thu Mar 31 21:46:37 PDT 2016:   1
Message sent at Thu Mar 31 21:46:38 PDT 2016:   9
Message sent at Thu Mar 31 21:46:39 PDT 2016:   2
Message sent at Thu Mar 31 21:46:40 PDT 2016:   5
Message sent at Thu Mar 31 21:46:41 PDT 2016:   6
Message sent at Thu Mar 31 21:46:42 PDT 2016:   1
Message sent at Thu Mar 31 21:46:43 PDT 2016:   8
Message sent at Thu Mar 31 21:46:44 PDT 2016:   1
Message sent at Thu Mar 31 21:46:45 PDT 2016:   7
Message sent at Thu Mar 31 21:46:46 PDT 2016:   6
Message sent at Thu Mar 31 21:46:47 PDT 2016:   5
Message sent at Thu Mar 31 21:46:48 PDT 2016:   6
Message sent at Thu Mar 31 21:46:49 PDT 2016:   1
Message sent at Thu Mar 31 21:46:50 PDT 2016:   9
Message sent at Thu Mar 31 21:46:51 PDT 2016:   8
Message sent at Thu Mar 31 21:46:52 PDT 2016:   8
Message sent at Thu Mar 31 21:46:53 PDT 2016:   2
Message sent at Thu Mar 31 21:46:54 PDT 2016:   7
Message sent at Thu Mar 31 21:46:55 PDT 2016:   10
Message sent at Thu Mar 31 21:46:56 PDT 2016:   8
Message sent at Thu Mar 31 21:46:57 PDT 2016:   5
Message sent at Thu Mar 31 21:46:58 PDT 2016:   1
Message sent at Thu Mar 31 21:46:59 PDT 2016:   9
Message sent at Thu Mar 31 21:47:00 PDT 2016:   10
Message sent at Thu Mar 31 21:47:01 PDT 2016:   9
Message sent at Thu Mar 31 21:47:02 PDT 2016:   10
Message sent at Thu Mar 31 21:47:03 PDT 2016:   5
Message sent at Thu Mar 31 21:47:04 PDT 2016:   10
Message sent at Thu Mar 31 21:47:05 PDT 2016:   9
Message sent at Thu Mar 31 21:47:06 PDT 2016:   2
Message sent at Thu Mar 31 21:47:07 PDT 2016:   8
Message sent at Thu Mar 31 21:47:08 PDT 2016:   9
Message sent at Thu Mar 31 21:47:09 PDT 2016:   10
Message sent at Thu Mar 31 21:47:10 PDT 2016:   10
Message sent at Thu Mar 31 21:47:11 PDT 2016:   5
Message sent at Thu Mar 31 21:47:12 PDT 2016:   1
Message sent at Thu Mar 31 21:47:13 PDT 2016:   10
Message sent at Thu Mar 31 21:47:14 PDT 2016:   4
Message sent at Thu Mar 31 21:47:15 PDT 2016:   7
```

```
[Stage 1:>                                                          (0 + 0) / 4]processin
g lines: Message sent at Thu Mar 31 21:46:44 PDT 2016:  1
processing lines: Message sent at Thu Mar 31 21:46:45 PDT 2016: 7
processing lines: Message sent at Thu Mar 31 21:46:46 PDT 2016: 6
processing lines: Message sent at Thu Mar 31 21:46:47 PDT 2016: 5
processing lines: Message sent at Thu Mar 31 21:46:48 PDT 2016: 6
-------------------------------------------
Time: 1459486009000 ms
-------------------------------------------
(7,1)
(5,1)
(6,2)
(1,1)

[Stage 16:>                                                         (0 + 0) / 4]processin
g lines: Message sent at Thu Mar 31 21:46:49 PDT 2016:  1
processing lines: Message sent at Thu Mar 31 21:46:50 PDT 2016: 9
processing lines: Message sent at Thu Mar 31 21:46:51 PDT 2016: 8
processing lines: Message sent at Thu Mar 31 21:46:52 PDT 2016: 8
processing lines: Message sent at Thu Mar 31 21:46:53 PDT 2016: 2
-------------------------------------------
Time: 1459486014000 ms
-------------------------------------------
(7,1)
(2,1)
(8,2)
(9,1)
(5,1)
(6,2)
(1,2)

processing lines: Message sent at Thu Mar 31 21:46:54 PDT 2016: 7
processing lines: Message sent at Thu Mar 31 21:46:55 PDT 2016: 10
processing lines: Message sent at Thu Mar 31 21:46:56 PDT 2016: 8
processing lines: Message sent at Thu Mar 31 21:46:57 PDT 2016: 5
processing lines: Message sent at Thu Mar 31 21:46:58 PDT 2016: 1
-------------------------------------------
Time: 1459486019000 ms
-------------------------------------------
```

I will attach the text file for analyzing.

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaRandIntProducer --master local[5] spark-kafka-stream-0.0.1.jar
localhost:9092 spark-topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-
1.7.5.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]
Message sent at Thu Mar 31 21:46:37 PDT 2016:        1
Message sent at Thu Mar 31 21:46:38 PDT 2016:        9
Message sent at Thu Mar 31 21:46:39 PDT 2016:        2
Message sent at Thu Mar 31 21:46:40 PDT 2016:        5
Message sent at Thu Mar 31 21:46:41 PDT 2016:        6
Message sent at Thu Mar 31 21:46:42 PDT 2016:        1
Message sent at Thu Mar 31 21:46:43 PDT 2016:        8
Message sent at Thu Mar 31 21:46:44 PDT 2016:        1
Message sent at Thu Mar 31 21:46:45 PDT 2016:        7
Message sent at Thu Mar 31 21:46:46 PDT 2016:        6
Message sent at Thu Mar 31 21:46:47 PDT 2016:        5
Message sent at Thu Mar 31 21:46:48 PDT 2016:        6
Message sent at Thu Mar 31 21:46:49 PDT 2016:        1
Message sent at Thu Mar 31 21:46:50 PDT 2016:        9
Message sent at Thu Mar 31 21:46:51 PDT 2016:        8
Message sent at Thu Mar 31 21:46:52 PDT 2016:        8
Message sent at Thu Mar 31 21:46:53 PDT 2016:        2
Message sent at Thu Mar 31 21:46:54 PDT 2016:        7
Message sent at Thu Mar 31 21:46:55 PDT 2016:        10
```

```
Message sent at Thu Mar 31 21:46:56 PDT 2016:       8
Message sent at Thu Mar 31 21:46:57 PDT 2016:       5
Message sent at Thu Mar 31 21:46:58 PDT 2016:       1
Message sent at Thu Mar 31 21:46:59 PDT 2016:       9
Message sent at Thu Mar 31 21:47:00 PDT 2016:       10
Message sent at Thu Mar 31 21:47:01 PDT 2016:       9
Message sent at Thu Mar 31 21:47:02 PDT 2016:       10
Message sent at Thu Mar 31 21:47:03 PDT 2016:       5
Message sent at Thu Mar 31 21:47:04 PDT 2016:       10  Start of the 30-second period
Message sent at Thu Mar 31 21:47:05 PDT 2016:       9
Message sent at Thu Mar 31 21:47:06 PDT 2016:       2
Message sent at Thu Mar 31 21:47:07 PDT 2016:       8
Message sent at Thu Mar 31 21:47:08 PDT 2016:       9
Message sent at Thu Mar 31 21:47:09 PDT 2016:       10
Message sent at Thu Mar 31 21:47:10 PDT 2016:       10
Message sent at Thu Mar 31 21:47:11 PDT 2016:       5
Message sent at Thu Mar 31 21:47:12 PDT 2016:       1
Message sent at Thu Mar 31 21:47:13 PDT 2016:       10
Message sent at Thu Mar 31 21:47:14 PDT 2016:       4
Message sent at Thu Mar 31 21:47:15 PDT 2016:       7
Message sent at Thu Mar 31 21:47:16 PDT 2016:       8
Message sent at Thu Mar 31 21:47:17 PDT 2016:       1
Message sent at Thu Mar 31 21:47:18 PDT 2016:       6
Message sent at Thu Mar 31 21:47:19 PDT 2016:       2
Message sent at Thu Mar 31 21:47:20 PDT 2016:       1
Message sent at Thu Mar 31 21:47:21 PDT 2016:       2
Message sent at Thu Mar 31 21:47:22 PDT 2016:       1
Message sent at Thu Mar 31 21:47:23 PDT 2016:       7
Message sent at Thu Mar 31 21:47:24 PDT 2016:       4
Message sent at Thu Mar 31 21:47:25 PDT 2016:       7
Message sent at Thu Mar 31 21:47:26 PDT 2016:       5
Message sent at Thu Mar 31 21:47:27 PDT 2016:       5
Message sent at Thu Mar 31 21:47:28 PDT 2016:       8
Message sent at Thu Mar 31 21:47:29 PDT 2016:       9
Message sent at Thu Mar 31 21:47:30 PDT 2016:       1
Message sent at Thu Mar 31 21:47:31 PDT 2016:       8
Message sent at Thu Mar 31 21:47:32 PDT 2016:       5
Message sent at Thu Mar 31 21:47:33 PDT 2016:       10  End of the 30-second period
Message sent at Thu Mar 31 21:47:34 PDT 2016:       6
Message sent at Thu Mar 31 21:47:35 PDT 2016:       5
Message sent at Thu Mar 31 21:47:36 PDT 2016:       1
Message sent at Thu Mar 31 21:47:37 PDT 2016:       8
Message sent at Thu Mar 31 21:47:38 PDT 2016:       9
Message sent at Thu Mar 31 21:47:39 PDT 2016:       6
Message sent at Thu Mar 31 21:47:40 PDT 2016:       6
Message sent at Thu Mar 31 21:47:41 PDT 2016:       4
Message sent at Thu Mar 31 21:47:42 PDT 2016:       5
Message sent at Thu Mar 31 21:47:43 PDT 2016:       1
Message sent at Thu Mar 31 21:47:44 PDT 2016:       3
Message sent at Thu Mar 31 21:47:45 PDT 2016:       3
Message sent at Thu Mar 31 21:47:46 PDT 2016:       4
Message sent at Thu Mar 31 21:47:47 PDT 2016:       3
Message sent at Thu Mar 31 21:47:48 PDT 2016:       1
Message sent at Thu Mar 31 21:47:49 PDT 2016:       1
Message sent at Thu Mar 31 21:47:50 PDT 2016:       6
Message sent at Thu Mar 31 21:47:51 PDT 2016:       9
Message sent at Thu Mar 31 21:47:52 PDT 2016:       7
Message sent at Thu Mar 31 21:47:53 PDT 2016:       7
Message sent at Thu Mar 31 21:47:54 PDT 2016:       6
Message sent at Thu Mar 31 21:47:55 PDT 2016:       1
```

```
Message sent at Thu Mar 31 21:47:56 PDT 2016:       6
Message sent at Thu Mar 31 21:47:57 PDT 2016:       5
Message sent at Thu Mar 31 21:47:58 PDT 2016:       1
Message sent at Thu Mar 31 21:47:59 PDT 2016:       10
Message sent at Thu Mar 31 21:48:00 PDT 2016:       3
Message sent at Thu Mar 31 21:48:01 PDT 2016:       6
Message sent at Thu Mar 31 21:48:02 PDT 2016:       5
```

```
[cloudera@localhost target]$ $SPARK_HOME/bin/spark-submit --class
kafka.streaming.KafkaWindowRandIntConsumer --master local[5] spark-kafka-stream-0.0.1.jar
localhost:9092 spark-topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-
1.7.5.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]
16/03/31 21:46:40 WARN NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
16/03/31 21:46:40 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback
address: 127.0.0.1; using 192.168.80.191 instead (on interface eth0)
16/03/31 21:46:40 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
16/03/31 21:46:42 WARN MetricsSystem: Using default name DAGScheduler for source because
spark.app.id is not set.
Kafka parameters: {zookeeper.connect=localhost:2181, group.id=spark-app,
metadata.broker.list=localhost:9092}
KafkaWindowLogCount parameters: topics=spark-topic; batchIntervalInSeconds=1;
windowDurationInSeconds=30; slidingWindowDurationInSeconds=5
[Stage 1:>                                                          (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:46:44 PDT 2016:   1
processing lines: Message sent at Thu Mar 31 21:46:45 PDT 2016:     7
processing lines: Message sent at Thu Mar 31 21:46:46 PDT 2016:     6
processing lines: Message sent at Thu Mar 31 21:46:47 PDT 2016:     5
processing lines: Message sent at Thu Mar 31 21:46:48 PDT 2016:     6
-------------------------------------------
Time: 1459486009000 ms
-------------------------------------------
(7,1)
(5,1)
(6,2)
(1,1)

[Stage 16:>                                                         (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:46:49 PDT 2016:   1
processing lines: Message sent at Thu Mar 31 21:46:50 PDT 2016:     9
processing lines: Message sent at Thu Mar 31 21:46:51 PDT 2016:     8
processing lines: Message sent at Thu Mar 31 21:46:52 PDT 2016:     8
processing lines: Message sent at Thu Mar 31 21:46:53 PDT 2016:     2
-------------------------------------------
Time: 1459486014000 ms
-------------------------------------------
(7,1)
(2,1)
(8,2)
(9,1)
(5,1)
(6,2)
```

```
(1,2)

processing lines: Message sent at Thu Mar 31 21:46:54 PDT 2016:    7
processing lines: Message sent at Thu Mar 31 21:46:55 PDT 2016:    10
processing lines: Message sent at Thu Mar 31 21:46:56 PDT 2016:    8
processing lines: Message sent at Thu Mar 31 21:46:57 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:46:58 PDT 2016:    1
-------------------------------------------
Time: 1459486019000 ms   Contains the previous results.
-------------------------------------------
(7,2)
(2,1)
(8,3)
(9,1)
(10,1)
(5,2)
(6,2)
(1,3)

[Stage 67:>                                                         (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:46:59 PDT 2016:  9
processing lines: Message sent at Thu Mar 31 21:47:00 PDT 2016:    10
processing lines: Message sent at Thu Mar 31 21:47:01 PDT 2016:    9
processing lines: Message sent at Thu Mar 31 21:47:02 PDT 2016:    10
processing lines: Message sent at Thu Mar 31 21:47:03 PDT 2016:    5
-------------------------------------------
Time: 1459486024000 ms
-------------------------------------------
(7,2)
(2,1)
(8,3)
(9,3)
(10,3)
(5,3)
(6,2)
(1,3)

processing lines: Message sent at Thu Mar 31 21:47:04 PDT 2016:    10
processing lines: Message sent at Thu Mar 31 21:47:05 PDT 2016:    9
processing lines: Message sent at Thu Mar 31 21:47:06 PDT 2016:    2
processing lines: Message sent at Thu Mar 31 21:47:07 PDT 2016:    8
processing lines: Message sent at Thu Mar 31 21:47:08 PDT 2016:    9
-------------------------------------------
Time: 1459486029000 ms
-------------------------------------------
(7,2)
(2,2)
(8,4)
(9,5)
(10,4)
(5,3)
(6,2)
(1,3)

processing lines: Message sent at Thu Mar 31 21:47:09 PDT 2016:    10
processing lines: Message sent at Thu Mar 31 21:47:10 PDT 2016:    10
processing lines: Message sent at Thu Mar 31 21:47:11 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:12 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:13 PDT 2016:    10
-------------------------------------------
```

```
Time: 1459486034000 ms
-------------------------------------------
(7,2)
(2,2)
(8,4)
(9,5)
(10,7)
(5,4)
(6,2)
(1,4)

processing lines: Message sent at Thu Mar 31 21:47:14 PDT 2016:    4
processing lines: Message sent at Thu Mar 31 21:47:15 PDT 2016:    7
processing lines: Message sent at Thu Mar 31 21:47:16 PDT 2016:    8
processing lines: Message sent at Thu Mar 31 21:47:17 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:18 PDT 2016:    6
-------------------------------------------
Time: 1459486039000 ms  First 30 seconds arrives.
-------------------------------------------
(7,2)
(2,2)
(8,5)
(4,1)
(9,5)
(10,7)
(5,3)
(6,1)
(1,4)

processing lines: Message sent at Thu Mar 31 21:47:19 PDT 2016:    2
processing lines: Message sent at Thu Mar 31 21:47:20 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:21 PDT 2016:    2
processing lines: Message sent at Thu Mar 31 21:47:22 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:23 PDT 2016:    7
-------------------------------------------
Time: 1459486044000 ms  Numbers begin rolling.
-------------------------------------------
(7,3)
(2,3)
(8,3)
(4,1)
(9,4)
(10,7)
(5,3)
(6,1)
(1,5)

processing lines: Message sent at Thu Mar 31 21:47:24 PDT 2016:    4
processing lines: Message sent at Thu Mar 31 21:47:25 PDT 2016:    7
processing lines: Message sent at Thu Mar 31 21:47:26 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:27 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:28 PDT 2016:    8
-------------------------------------------
Time: 1459486049000 ms
-------------------------------------------
(7,3)
(2,3)
(8,3)
(4,2)
(9,4)
```

```
(10,6)
(5,4)
(6,1)
(1,4)

[Stage 409:>                                                        (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:47:29 PDT 2016:  9
processing lines: Message sent at Thu Mar 31 21:47:30 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:31 PDT 2016:    8
processing lines: Message sent at Thu Mar 31 21:47:32 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:33 PDT 2016:    10
-------------------------------------------
Time: 1459486054000 ms  Numbers from second 4 ~ 33 are correct!
-------------------------------------------
(7,3)
(2,3)
(8,4)
(4,2)
(9,3)
(10,5)
(5,4)
(6,1)
(1,5)

[Stage 471:>                                                        (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:47:34 PDT 2016:  6
processing lines: Message sent at Thu Mar 31 21:47:35 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:36 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:37 PDT 2016:    8
processing lines: Message sent at Thu Mar 31 21:47:38 PDT 2016:    9
-------------------------------------------
Time: 1459486059000 ms
-------------------------------------------
(7,3)
(2,2)
(8,4)
(4,2)
(9,2)
(10,4)
(5,5)
(6,2)
(1,6)

[Stage 533:>                                                        (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:47:39 PDT 2016:  6
processing lines: Message sent at Thu Mar 31 21:47:40 PDT 2016:    6
processing lines: Message sent at Thu Mar 31 21:47:41 PDT 2016:    4
processing lines: Message sent at Thu Mar 31 21:47:42 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:43 PDT 2016:    1
-------------------------------------------
Time: 1459486064000 ms
-------------------------------------------
(7,3)
(2,2)
(8,4)
(4,3)
(9,2)
(10,1)
(5,5)
(6,4)
```

```
(1,6)

[Stage 597:>                                                          (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:47:44 PDT 2016:  3
processing lines: Message sent at Thu Mar 31 21:47:45 PDT 2016:    3
processing lines: Message sent at Thu Mar 31 21:47:46 PDT 2016:    4
processing lines: Message sent at Thu Mar 31 21:47:47 PDT 2016:    3
processing lines: Message sent at Thu Mar 31 21:47:48 PDT 2016:    1
-------------------------------------------
Time: 1459486069000 ms
-------------------------------------------
(7,2)
(2,2)
(8,3)
(3,3)
(4,3)
(9,2)
(10,1)
(5,5)
(6,3)
(1,6)

[Stage 660:>                                                          (0 + 0) /
4]processing lines: Message sent at Thu Mar 31 21:47:49 PDT 2016:  1
processing lines: Message sent at Thu Mar 31 21:47:50 PDT 2016:    6
processing lines: Message sent at Thu Mar 31 21:47:51 PDT 2016:    9
processing lines: Message sent at Thu Mar 31 21:47:52 PDT 2016:    7
processing lines: Message sent at Thu Mar 31 21:47:53 PDT 2016:    7
-------------------------------------------
Time: 1459486074000 ms
-------------------------------------------
(7,3)
(8,3)
(3,3)
(4,3)
(9,3)
(10,1)
(5,5)
(6,4)
(1,5)

processing lines: Message sent at Thu Mar 31 21:47:54 PDT 2016:    6
processing lines: Message sent at Thu Mar 31 21:47:55 PDT 2016:    1
processing lines: Message sent at Thu Mar 31 21:47:56 PDT 2016:    6
processing lines: Message sent at Thu Mar 31 21:47:57 PDT 2016:    5
processing lines: Message sent at Thu Mar 31 21:47:58 PDT 2016:    1
-------------------------------------------
Time: 1459486079000 ms
-------------------------------------------
(7,2)
(8,2)
(3,3)
(4,2)
(9,3)
(10,1)
(5,4)
(6,6)
(1,7)

processing lines: Message sent at Thu Mar 31 21:47:59 PDT 2016:    10
```

```
processing lines: Message sent at Thu Mar 31 21:48:00 PDT 2016:     3
processing lines: Message sent at Thu Mar 31 21:48:01 PDT 2016:     6
processing lines: Message sent at Thu Mar 31 21:48:02 PDT 2016:     5
-------------------------------------------
Time: 1459486084000 ms
-------------------------------------------
(7,2)
(8,1)
(3,4)
(4,2)
(9,2)
(10,1)
(5,4)
(6,7)
(1,6)

-------------------------------------------
Time: 1459486089000 ms  No incoming messages, numbers keep decreasing.
-------------------------------------------
(7,2)
(3,4)
(4,2)
(9,1)
(10,1)
(5,3)
(6,6)
(1,5)

-------------------------------------------
Time: 1459486094000 ms
-------------------------------------------
(7,2)
(3,4)
(4,1)
(9,1)
(10,1)
(5,2)
(6,4)
(1,4)

-------------------------------------------
Time: 1459486099000 ms
-------------------------------------------
(7,2)
(3,1)
(9,1)
(10,1)
(5,2)
(6,4)
(1,3)

-------------------------------------------
Time: 1459486104000 ms
-------------------------------------------
(3,1)
(10,1)
(5,2)
(6,3)
(1,2)
```

```
-------------------------------------------
Time: 1459486109000 ms
-------------------------------------------
(3,1)
(10,1)
(5,1)
(6,1)

-------------------------------------------
Time: 1459486114000 ms
-------------------------------------------

-------------------------------------------
Time: 1459486119000 ms
-------------------------------------------
```

2. Solve it in Python.

kafka_window_consumer.py:

```
from __future__ import print_function

import sys

from pyspark import SparkContext
from pyspark.streaming import StreamingContext
from pyspark.streaming.kafka import KafkaUtils

if __name__ == "__main__":
    if len(sys.argv) != 3:
        print("Usage: direct_kafka_wordcount.py <broker_list> <topic>",
file=sys.stderr)
        exit(-1)

    sc = SparkContext(appName="PythonStreamingDirectKafkaWordCount")
    ssc = StreamingContext(sc, 1)
    ssc.checkpoint("file:////home/cloudera/Documents/hw08/checkpoint")

    brokers, topic = sys.argv[1:]
    kvs = KafkaUtils.createDirectStream(ssc, [topic], {"metadata.broker.list":
brokers})
    lines = kvs.map(lambda x: x[1].split("\t")[1] + " total")
    lines.pprint()

    counts = lines.flatMap(lambda line: str(line).split(" ")) \
        .map(lambda word: (word, 1)) \
        .reduceByKey(lambda a, b: a+b) \
        .reduceByKeyAndWindow(lambda x, y: x + y, lambda x, y: x - y, 30, 5)
    counts.pprint()

    ssc.start()
    ssc.awaitTermination()
```

We should set path for the checkpoint before executing the code. The checkpoint directory is provided to allow for periodic RDD checkpointing. Details about checkpointing can be found:
http://spark.apache.org/docs/latest/streaming-programming-guide.html#checkpointing

There's a trick here. After splitting the string and get the second number, I attach the string with " total". That's because the flatMap() in the next step flat the string aggressively. Like this:

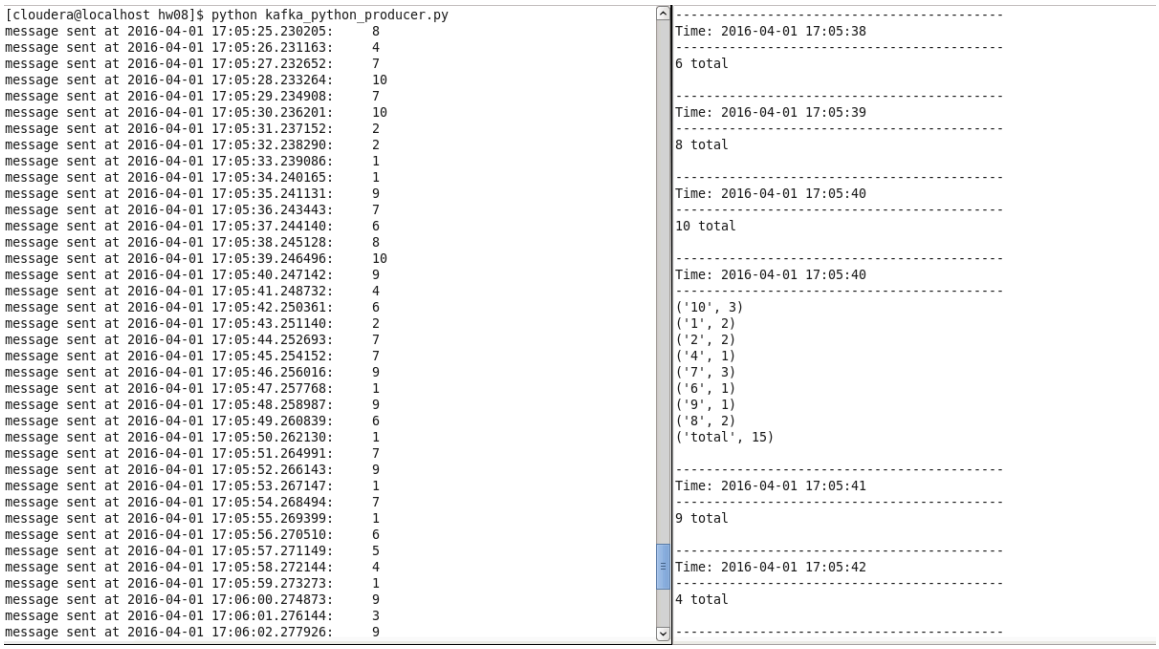
```
>>> lines.take(3)
[u'message sent at 2016-04-01 15:11:47.600825:\\t10', u'message sent at 2016-
04-01 15:11:47.600825:\\t10', u'message sent at 2016-04-01
15:11:47.600825:\\t7']

>>> words = lines.flatMap(lambda line: line.split()[1])
>>> words.take(5)
['1', '0', '1', '0', '7']
```

10 will be split to '1' and '0' here. So I add " total" to make sure there are two elements in each map element. Thus number '10' won't be split into parts. We can also use 'total' to calculate total number of numbers in each window.

In the right side window, I print out each message received too for easily debugging.

```
[cloudera@localhost hw08]$ python kafka_python_producer.py
message sent at 2016-04-01 17:05:25.230205:     8
message sent at 2016-04-01 17:05:26.231163:     4
message sent at 2016-04-01 17:05:27.232652:     7
message sent at 2016-04-01 17:05:28.233264:     10
message sent at 2016-04-01 17:05:29.234908:     7
message sent at 2016-04-01 17:05:30.236201:     10
message sent at 2016-04-01 17:05:31.237152:     2
message sent at 2016-04-01 17:05:32.238290:     2
message sent at 2016-04-01 17:05:33.239086:     1
message sent at 2016-04-01 17:05:34.240165:     1
message sent at 2016-04-01 17:05:35.241131:     9
message sent at 2016-04-01 17:05:36.243443:     7
message sent at 2016-04-01 17:05:37.244140:     6
message sent at 2016-04-01 17:05:38.245128:     8
message sent at 2016-04-01 17:05:39.246496:     10
message sent at 2016-04-01 17:05:40.247142:     9
message sent at 2016-04-01 17:05:41.248732:     4
message sent at 2016-04-01 17:05:42.250361:     6
message sent at 2016-04-01 17:05:43.251140:     2
message sent at 2016-04-01 17:05:44.252693:     7
message sent at 2016-04-01 17:05:45.254152:     7
message sent at 2016-04-01 17:05:46.256016:     9
message sent at 2016-04-01 17:05:47.257768:     1
message sent at 2016-04-01 17:05:48.258987:     9
message sent at 2016-04-01 17:05:49.260839:     6
message sent at 2016-04-01 17:05:50.262130:     1
message sent at 2016-04-01 17:05:51.264991:     7
message sent at 2016-04-01 17:05:52.266143:     9
message sent at 2016-04-01 17:05:53.267147:     1
message sent at 2016-04-01 17:05:54.268494:     7
message sent at 2016-04-01 17:05:55.269399:     1
message sent at 2016-04-01 17:05:56.270510:     6
message sent at 2016-04-01 17:05:57.271149:     5
message sent at 2016-04-01 17:05:58.272144:     4
message sent at 2016-04-01 17:05:59.273273:     1
message sent at 2016-04-01 17:06:00.274873:     9
message sent at 2016-04-01 17:06:01.276144:     3
message sent at 2016-04-01 17:06:02.277926:     9
```

```
[cloudera@localhost hw08]$ spark-submit kafka_window_consumer.py localhost:9092 spark-t
opic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-1.7.5.jar!/org/s
lf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-1.7.5.jar!/org/sl
f4j/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]
16/04/01 17:05:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
16/04/01 17:05:21 WARN Utils: Your hostname, localhost.localdomain resolves to a loopba
ck address: 127.0.0.1; using 192.168.80.194 instead (on interface eth0)
16/04/01 17:05:21 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
16/04/01 17:05:23 WARN MetricsSystem: Using default name DAGScheduler for source becaus
e spark.app.id is not set.
-------------------------------------------
Time: 2016-04-01 17:05:26
-------------------------------------------
8 total

-------------------------------------------
Time: 2016-04-01 17:05:27
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:05:28
-------------------------------------------
7 total

-------------------------------------------
Time: 2016-04-01 17:05:29
-------------------------------------------
10 total

-------------------------------------------
Time: 2016-04-01 17:05:30
-------------------------------------------
```




```
[cloudera@localhost hw08]$ python kafka_python_producer.py
message sent at 2016-04-01 17:05:25.230205:  8
message sent at 2016-04-01 17:05:26.231163:  4
message sent at 2016-04-01 17:05:27.232652:  7
message sent at 2016-04-01 17:05:28.233264:  10
message sent at 2016-04-01 17:05:29.234908:  7
message sent at 2016-04-01 17:05:30.236201:  10
message sent at 2016-04-01 17:05:31.237152:  2
message sent at 2016-04-01 17:05:32.238290:  2
message sent at 2016-04-01 17:05:33.239086:  1
message sent at 2016-04-01 17:05:34.240165:  1
message sent at 2016-04-01 17:05:35.241131:  9
message sent at 2016-04-01 17:05:36.243443:  7   Start of the 30-second period
message sent at 2016-04-01 17:05:37.244140:  6
message sent at 2016-04-01 17:05:38.245128:  8
message sent at 2016-04-01 17:05:39.246496:  10
message sent at 2016-04-01 17:05:40.247142:  9
message sent at 2016-04-01 17:05:41.248732:  4
message sent at 2016-04-01 17:05:42.250361:  6
message sent at 2016-04-01 17:05:43.251140:  2
message sent at 2016-04-01 17:05:44.252693:  7
message sent at 2016-04-01 17:05:45.254152:  7
message sent at 2016-04-01 17:05:46.256016:  9
message sent at 2016-04-01 17:05:47.257768:  1
message sent at 2016-04-01 17:05:48.258987:  9
message sent at 2016-04-01 17:05:49.260839:  6
message sent at 2016-04-01 17:05:50.262130:  1
message sent at 2016-04-01 17:05:51.264991:  7
message sent at 2016-04-01 17:05:52.266143:  9
message sent at 2016-04-01 17:05:53.267147:  1
message sent at 2016-04-01 17:05:54.268494:  7
message sent at 2016-04-01 17:05:55.269399:  1
message sent at 2016-04-01 17:05:56.270510:  6
message sent at 2016-04-01 17:05:57.271149:  5
message sent at 2016-04-01 17:05:58.272144:  4
message sent at 2016-04-01 17:05:59.273273:  1
```

```
message sent at 2016-04-01 17:06:00.274873:  9
message sent at 2016-04-01 17:06:01.276144:  3
message sent at 2016-04-01 17:06:02.277926:  9
message sent at 2016-04-01 17:06:03.279141:  8
message sent at 2016-04-01 17:06:04.280130:  4
message sent at 2016-04-01 17:06:05.281139:  3  End of the 30-second period
message sent at 2016-04-01 17:06:06.282353:  4
message sent at 2016-04-01 17:06:07.283629:  10
message sent at 2016-04-01 17:06:08.284883:  4
message sent at 2016-04-01 17:06:09.286564:  1
message sent at 2016-04-01 17:06:10.287134:  4
message sent at 2016-04-01 17:06:11.288153:  2
message sent at 2016-04-01 17:06:12.289167:  8
message sent at 2016-04-01 17:06:13.290837:  3
message sent at 2016-04-01 17:06:14.292264:  2
message sent at 2016-04-01 17:06:15.293133:  8
message sent at 2016-04-01 17:06:16.294165:  5
message sent at 2016-04-01 17:06:17.295139:  9

[cloudera@localhost hw08]$ spark-submit kafka_window_consumer.py localhost:9092 spark-
topic
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/lib/zookeeper/lib/slf4j-log4j12-
1.7.5.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/lib/flume-ng/lib/slf4j-log4j12-
1.7.5.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]
16/04/01 17:05:21 WARN NativeCodeLoader: Unable to load native-hadoop library for your
platform... using builtin-java classes where applicable
16/04/01 17:05:21 WARN Utils: Your hostname, localhost.localdomain resolves to a loopback
address: 127.0.0.1; using 192.168.80.194 instead (on interface eth0)
16/04/01 17:05:21 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
16/04/01 17:05:23 WARN MetricsSystem: Using default name DAGScheduler for source because
spark.app.id is not set.
-------------------------------------------
Time: 2016-04-01 17:05:26
-------------------------------------------
8 total

-------------------------------------------
Time: 2016-04-01 17:05:27
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:05:28
-------------------------------------------
7 total

-------------------------------------------
Time: 2016-04-01 17:05:29
-------------------------------------------
10 total

-------------------------------------------
Time: 2016-04-01 17:05:30
-------------------------------------------
7 total
```

```
-----------------------------------------
Time: 2016-04-01 17:05:30  Correct!
-----------------------------------------
('8', 1)
('total', 5)
('4', 1)
('7', 2)
('10', 1)

-----------------------------------------
Time: 2016-04-01 17:05:31
-----------------------------------------
10 total

-----------------------------------------
Time: 2016-04-01 17:05:32
-----------------------------------------
2 total

-----------------------------------------
Time: 2016-04-01 17:05:33
-----------------------------------------
2 total

-----------------------------------------
Time: 2016-04-01 17:05:34
-----------------------------------------
1 total

-----------------------------------------
Time: 2016-04-01 17:05:35
-----------------------------------------
1 total

-----------------------------------------
Time: 2016-04-01 17:05:35  Correct!
-----------------------------------------
('10', 2)
('1', 2)
('2', 2)
('4', 1)
('7', 2)
('8', 1)
('total', 10)

-----------------------------------------
Time: 2016-04-01 17:05:36
-----------------------------------------
9 total

-----------------------------------------
Time: 2016-04-01 17:05:37
-----------------------------------------
7 total

-----------------------------------------
Time: 2016-04-01 17:05:38
-----------------------------------------
6 total
```

```
-------------------------------------------
Time: 2016-04-01 17:05:39
-------------------------------------------
8 total

-------------------------------------------
Time: 2016-04-01 17:05:40
-------------------------------------------
10 total

-------------------------------------------
Time: 2016-04-01 17:05:40
-------------------------------------------
('10', 3)
('1', 2)
('2', 2)
('4', 1)
('7', 3)
('6', 1)
('9', 1)
('8', 2)
('total', 15)

-------------------------------------------
Time: 2016-04-01 17:05:41
-------------------------------------------
9 total

-------------------------------------------
Time: 2016-04-01 17:05:42
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:05:43
-------------------------------------------
6 total

-------------------------------------------
Time: 2016-04-01 17:05:44
-------------------------------------------
2 total

-------------------------------------------
Time: 2016-04-01 17:05:45
-------------------------------------------
7 total

-------------------------------------------
Time: 2016-04-01 17:05:45
-------------------------------------------
('10', 3)
('1', 2)
('2', 3)
('4', 2)
('7', 4)
('6', 2)
('9', 2)
('8', 2)
('total', 20)
```

```
-------------------------------------------
Time: 2016-04-01 17:05:46
-------------------------------------------
7 total

-------------------------------------------
Time: 2016-04-01 17:05:47
-------------------------------------------
9 total

-------------------------------------------
Time: 2016-04-01 17:05:48
-------------------------------------------
1 total

-------------------------------------------
Time: 2016-04-01 17:05:49
-------------------------------------------
9 total

-------------------------------------------
Time: 2016-04-01 17:05:50
-------------------------------------------
6 total

-------------------------------------------
Time: 2016-04-01 17:05:50
-------------------------------------------
('10', 3)
('1', 3)
('2', 3)
('4', 2)
('7', 5)
('6', 3)
('9', 4)
('8', 2)
('total', 25)

-------------------------------------------
Time: 2016-04-01 17:05:51
-------------------------------------------
1 total

-------------------------------------------
Time: 2016-04-01 17:05:52
-------------------------------------------
7 total

-------------------------------------------
Time: 2016-04-01 17:05:53
-------------------------------------------
9 total

-------------------------------------------
Time: 2016-04-01 17:05:54
-------------------------------------------
1 total

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

```
Time: 2016-04-01 17:05:55
-------------------------------------------
7 total

-------------------------------------------
Time: 2016-04-01 17:05:55
-------------------------------------------
('10', 3)
('1', 5)
('2', 3)
('4', 2)
('7', 7)
('6', 3)
('9', 5)
('8', 2)
('total', 30)

-------------------------------------------
Time: 2016-04-01 17:05:56
-------------------------------------------
1 total

-------------------------------------------
Time: 2016-04-01 17:05:57
-------------------------------------------
6 total

-------------------------------------------
Time: 2016-04-01 17:05:58
-------------------------------------------
5 total

-------------------------------------------
Time: 2016-04-01 17:05:59
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:06:00
-------------------------------------------
1 total

-------------------------------------------
Time: 2016-04-01 17:06:00
-------------------------------------------
('10', 2)
('1', 7)
('2', 3)
('5', 1)
('4', 2)
('7', 5)
('6', 4)
('9', 5)
('8', 1)
('total', 30)

-------------------------------------------
Time: 2016-04-01 17:06:01
-------------------------------------------
9 total
```

```
-------------------------------------------
Time: 2016-04-01 17:06:02
-------------------------------------------
3 total

-------------------------------------------
Time: 2016-04-01 17:06:03
-------------------------------------------
9 total

-------------------------------------------
Time: 2016-04-01 17:06:04
-------------------------------------------
8 total

-------------------------------------------
Time: 2016-04-01 17:06:05
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:06:05  Correct from 17:05:36 ~ 17:06:05
-------------------------------------------
('10', 1)
('1', 5)
('3', 1)
('2', 1)
('5', 1)
('4', 3)
('7', 5)
('6', 4)
('9', 7)
('8', 2)
...

-------------------------------------------
Time: 2016-04-01 17:06:06
-------------------------------------------
3 total

-------------------------------------------
Time: 2016-04-01 17:06:07
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:06:08
-------------------------------------------
10 total

-------------------------------------------
Time: 2016-04-01 17:06:09
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:06:10
-------------------------------------------
1 total
```

```
-------------------------------------------
Time: 2016-04-01 17:06:10
-------------------------------------------
('10', 1)
('1', 6)
('3', 2)
('2', 1)
('5', 1)
('4', 5)
('7', 4)
('6', 3)
('9', 6)
('8', 1)
...

-------------------------------------------
Time: 2016-04-01 17:06:11
-------------------------------------------
4 total

-------------------------------------------
Time: 2016-04-01 17:06:12
-------------------------------------------
2 total

-------------------------------------------
Time: 2016-04-01 17:06:13
-------------------------------------------
8 total

-------------------------------------------
Time: 2016-04-01 17:06:14
-------------------------------------------
3 total

-------------------------------------------
Time: 2016-04-01 17:06:15
-------------------------------------------
2 total

-------------------------------------------
Time: 2016-04-01 17:06:15
-------------------------------------------
('10', 1)
('1', 6)
('3', 3)
('2', 2)
('5', 1)
('4', 5)
('7', 3)
('6', 2)
('9', 5)
('8', 2)
...
```

Check the checkpoint:

```
[cloudera@localhost checkpoint]$ ls -l checkpoint*
-rw-r--r--. 1 cloudera cloudera 10698 Apr  1 17:06 checkpoint-1459555573000
-rw-r--r--. 1 cloudera cloudera 10770 Apr  1 17:06 checkpoint-1459555574000
-rw-r--r--. 1 cloudera cloudera 10773 Apr  1 17:06 checkpoint-1459555574000.bk
-rw-r--r--. 1 cloudera cloudera 10996 Apr  1 17:06 checkpoint-1459555575000
-rw-r--r--. 1 cloudera cloudera 10831 Apr  1 17:06 checkpoint-1459555575000.bk
-rw-r--r--. 1 cloudera cloudera 10996 Apr  1 17:06 checkpoint-1459555576000
-rw-r--r--. 1 cloudera cloudera 10921 Apr  1 17:06 checkpoint-1459555576000.bk
-rw-r--r--. 1 cloudera cloudera 10903 Apr  1 17:06 checkpoint-1459555577000
-rw-r--r--. 1 cloudera cloudera 11000 Apr  1 17:06 checkpoint-1459555577000.bk
-rw-r--r--. 1 cloudera cloudera 11084 Apr  1 17:06 checkpoint-1459555578000
```

Check the log files from 4 partitions:

```
offset: 367 position: 27937 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 1931142582 payload: message sent at 2016-04-01 17:05:34.240165: 1
offset: 368 position: 28008 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 1572512215 payload: message sent at 2016-04-01 17:05:35.241131: 9
offset: 369 position: 28079 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 1427021423 payload: message sent at 2016-04-01 17:05:38.245128: 8
offset: 370 position: 28150 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2423513208 payload: message sent at 2016-04-01 17:05:43.251140: 2
offset: 371 position: 28221 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3043930848 payload: message sent at 2016-04-01 17:05:49.260839: 6
offset: 372 position: 28292 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3797415885 payload: message sent at 2016-04-01 17:05:53.267147: 1
offset: 373 position: 28363 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 257578074 payload: message sent at 2016-04-01 17:05:55.269399:  1
```

```
offset: 443 position: 32561 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 144059516 payload: message sent at 2016-04-01 17:05:32.238290:  2
offset: 444 position: 32632 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2745532117 payload: message sent at 2016-04-01 17:05:33.239086: 1
offset: 445 position: 32703 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3452801365 payload: message sent at 2016-04-01 17:05:37.244140: 6
offset: 446 position: 32774 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 267010062 payload: message sent at 2016-04-01 17:05:44.252693:  7
offset: 447 position: 32845 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3455914024 payload: message sent at 2016-04-01 17:05:45.254152: 7
offset: 448 position: 32916 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 91008178 payload: message sent at 2016-04-01 17:05:48.258987:   9
offset: 449 position: 32987 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3402565091 payload: message sent at 2016-04-01 17:05:56.270510: 6
```

```
offset: 365 position: 27815 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 4238048976 payload: message sent at 2016-04-01 17:05:36.243443: 7
offset: 366 position: 27886 isvalid: true payloadsize: 46 magic: 0 compresscodec: NoCompressionCod
ec crc: 3509526809 payload: message sent at 2016-04-01 17:05:39.246496: 10
offset: 367 position: 27958 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2508630069 payload: message sent at 2016-04-01 17:05:46.256016: 9
offset: 368 position: 28029 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 108681922 payload: message sent at 2016-04-01 17:05:50.262130:  1
offset: 369 position: 28100 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 1394141348 payload: message sent at 2016-04-01 17:05:51.264991: 7
offset: 370 position: 28171 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 68631772 payload: message sent at 2016-04-01 17:05:52.266143:   9
```

```
offset: 309 position: 22726 isvalid: true payloadsize: 46 magic: 0 compresscodec: NoCompressionCod
ec crc: 1406862289 payload: message sent at 2016-04-01 17:05:28.233264: 10
offset: 310 position: 22798 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2780546033 payload: message sent at 2016-04-01 17:05:29.234908: 7
offset: 311 position: 22869 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2387701168 payload: message sent at 2016-04-01 17:05:31.237152: 2
offset: 312 position: 22940 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2603655902 payload: message sent at 2016-04-01 17:05:40.247142: 9
offset: 313 position: 23011 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3057536583 payload: message sent at 2016-04-01 17:05:41.248732: 4
offset: 314 position: 23082 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 2136171803 payload: message sent at 2016-04-01 17:05:42.250361: 6
offset: 315 position: 23153 isvalid: true payloadsize: 45 magic: 0 compresscodec: NoCompressionCod
ec crc: 3009310696 payload: message sent at 2016-04-01 17:05:47.257768: 1
```

Messages are automatically distributed to different partitions randomly.