Project: Kafka + Spark Streaming + PySpark

Student :

Presented by MANICKAM RAVISEKAR ,
Master of Science in Computer Science, 19599 , Fall Semester 2022

Professor : Dr Henry Chung
TA : Liang

SAN FRANCISCO BAY
UNIVERSITY
47671 WestingHouse Dr.,
Fremont, CA 94539

ACKNOWLEDGEMENT

One of our master's degree Project for Spark Streaming using Scala , Pyspark and Kafka streaming,

Is an Interesting, which made me to learn new things, it is useful in designing and applying using Scala, Pyspark, Kafka stream programming.

For deploying this project, I would like to thank Dr. Henry Chang an TA Liang for providing all the required input .

Also, for all I would like to always pray to Almighty for giving us wisdom and power to understand things.

Content

Index :

Abstract

Event streaming is the digital equivalent of the human body's central nervous system. It is the technological foundation for the 'always-on' world where businesses are increasingly software-defined and automated, and where the user of software is more software.

Technically speaking, event streaming is the practice of capturing data in real-time from event sources like databases, sensors, mobile devices, cloud services, and software applications in the form of streams of events; storing these event streams durably for later retrieval; manipulating, processing, and reacting to the event streams in real-time as well as retrospectively; and routing the event streams to different destination technologies as needed. Event streaming thus ensures a continuous flow and interpretation of data so that the right information is at the right place, at the right time.

Kafka Installation download required version

CS570: last homework: Exec | Inbox (10) - manickam@stu | Untitled presentation - Goo | Assignments: CS570 - Big D | CS570 Big Data Processing | Apache Kafka

https://kafka.apache.org/downloads

kafka

GET STARTED DOCS POWERED BY COMMUNITY APACHE DOWNLOAD KAFKA

Kafka 3.2.1 fixes 13 issues since the 3.2.0 release. For more information, please read the detailed Release Notes.

3.2.0

- Released May 17, 2022
- Release Notes
- Source download: kafka-3.2.0-src.tgz (asc, sha512)
- Binary downloads:
  - Scala 2.12 - kafka_2.12-3.2.0.tgz (asc, sha512)
  - Scala 2.13 - kafka_2.13-3.2.0.tgz (asc, sha512)

  We build for multiple versions of Scala. This only matters if you are using Scala and you want a version built for the same Scala version you use. Otherwise any version should work (2.13 is recommended).

Kafka 3.2.0 includes a number of significant new features. Here is a summary of some notable changes:

- log4j 1.x is replaced with reload4j
- StandardAuthorizer for KRaft (KIP-801)
- Send a hint to the partition leader to recover the partition (KIP-704)
- Top-level error code field in DescribeLogDirsResponse (KIP-784)
- kafka-console-producer writes headers and null values (KIP-798 and KIP-810)
- JoinGroupRequest and LeaveGroupRequest have a reason attached (KIP-800)
- Static membership protocol lets the leader skip assignment (KIP-814)

Unarchive the downloaded file

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~$ ls -l kafka_2.12-3.2.0.tgz
-rw-r--r-- 1 hduser hduser 104000785 Dec  5 01:22 kafka_2.12-3.2.0.tgz
hduser@cs570bigdata:~$ tar -xzf kafka_2.12-3.2.0.tgz
hduser@cs570bigdata:~$ cd kafka_2.12-3.2.0
hduser@cs570bigdata:~/kafka_2.12-3.2.0$
```

Start the ZooKeeper service

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ echo "Starting Zookeeper"
Starting Zookeeper
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ bin/zookeeper-server-start.sh config/zookeeper.properties
```

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
[2022-12-05 01:28:17,893] INFO Server environment:os.name=Linux (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,893] INFO Server environment:os.arch=amd64 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,893] INFO Server environment:os.version=5.15.0-56-generic (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO Server environment:user.name=hduser (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO Server environment:user.home=/home/hduser (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO Server environment:user.dir=/home/hduser/kafka_2.12-3.2.0 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO Server environment:os.memory.free=490MB (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO Server environment:os.memory.max=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO Server environment:os.memory.total=512MB (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.enableEagerACLCheck = false (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.digest.enabled = true (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.closeSessionTxn.enabled = true (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.flushDelay=0 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.maxWriteQueuePollTime=0 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.maxBatchSize=1000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,894] INFO zookeeper.intBufferStartingSizeBytes = 1024 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,895] INFO Weighed connection throttling is disabled (org.apache.zookeeper.server.BlueThrottle)
[2022-12-05 01:28:17,915] INFO minSessionTimeout set to 6000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,915] INFO maxSessionTimeout set to 60000 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,916] INFO Response cache size is initialized with value 400. (org.apache.zookeeper.server.ResponseCache)
[2022-12-05 01:28:17,917] INFO Response cache size is initialized with value 400. (org.apache.zookeeper.server.ResponseCache)
[2022-12-05 01:28:17,917] INFO zookeeper.pathStats.slotCapacity = 60 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-12-05 01:28:17,917] INFO zookeeper.pathStats.slotDuration = 15 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-12-05 01:28:17,917] INFO zookeeper.pathStats.maxDepth = 6 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-12-05 01:28:17,918] INFO zookeeper.pathStats.initialDelay = 5 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-12-05 01:28:17,918] INFO zookeeper.pathStats.delay = 5 (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-12-05 01:28:17,918] INFO zookeeper.pathStats.enabled = false (org.apache.zookeeper.server.util.RequestPathMetricsCollector)
[2022-12-05 01:28:17,939] INFO The max bytes for all large requests are set to 104857600 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,942] INFO The large request threshold is set to -1 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:17,943] INFO Created server with tickTime 3000 minSessionTimeout 6000 maxSessionTimeout 60000 clientPortListenBacklog -1 datadir /tmp/zookeeper/version-2 snapdir /tmp/zook
eeper/version-2 (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:18,031] INFO Using org.apache.zookeeper.server.NIOServerCnxnFactory as server connection factory (org.apache.zookeeper.server.ServerCnxnFactory)
[2022-12-05 01:28:18,061] WARN maxCnxns is not configured, using default value 0. (org.apache.zookeeper.server.ServerCnxnFactory)
[2022-12-05 01:28:18,064] INFO Configuring NIO connection handler with 10s sessionless connection timeout, 1 selector thread(s), 4 worker threads, and 64 kB direct buffers. (org.apache.zook
eeper.server.NIOServerCnxnFactory)
[2022-12-05 01:28:18,084] INFO binding to port 0.0.0.0/0.0.0.0:2181 (org.apache.zookeeper.server.NIOServerCnxnFactory)
[2022-12-05 01:28:18,134] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2022-12-05 01:28:18,136] INFO Using org.apache.zookeeper.server.watch.WatchManager as watch manager (org.apache.zookeeper.server.watch.WatchManagerFactory)
[2022-12-05 01:28:18,143] INFO zookeeper.snapshotSizeFactor = 0.33 (org.apache.zookeeper.server.ZKDatabase)
[2022-12-05 01:28:18,146] INFO zookeeper.commitLogCount=500 (org.apache.zookeeper.server.ZKDatabase)
[2022-12-05 01:28:18,224] INFO zookeeper.snapshot.compression.method = CHECKED (org.apache.zookeeper.server.persistence.SnapStream)
[2022-12-05 01:28:18,227] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-12-05 01:28:18,232] INFO Snapshot loaded in 87 ms, highest zxid is 0x0, digest is 1371985504 (org.apache.zookeeper.server.ZKDatabase)
[2022-12-05 01:28:18,233] INFO Snapshotting: 0x0 to /tmp/zookeeper/version-2/snapshot.0 (org.apache.zookeeper.server.persistence.FileTxnSnapLog)
[2022-12-05 01:28:18,234] INFO Snapshot taken in 1 ms (org.apache.zookeeper.server.ZooKeeperServer)
[2022-12-05 01:28:18,290] INFO zookeeper.request_throttler.shutdownTimeout = 10000 (org.apache.zookeeper.server.RequestThrottler)
[2022-12-05 01:28:18,292] INFO PrepRequestProcessor (sid:0) started, reconfigEnabled=false (org.apache.zookeeper.server.PrepRequestProcessor)
[2022-12-05 01:28:18,360] INFO Using checkIntervalMs=60000 maxPerMinute=10000 maxNeverUsedIntervalMs=0 (org.apache.zookeeper.server.ContainerManager)
[2022-12-05 01:28:18,362] INFO ZooKeeper audit is disabled. (org.apache.zookeeper.audit.ZKAuditProvider)
```

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ bin/kafka-server-start.sh config/server.properties
[2022-12-05 01:31:03,362] INFO Registered kafka:type=kafka.Log4jController MBean (kafka.utils.Log4jControllerRegistration$)
```

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
[2022-12-05 01:31:11,220] INFO [GroupCoordinator 0]: Starting up. (kafka.coordinator.group.GroupCoordinator)
[2022-12-05 01:31:11,317] INFO [GroupCoordinator 0]: Startup complete. (kafka.coordinator.group.GroupCoordinator)
[2022-12-05 01:31:11,337] INFO Successfully created /controller_epoch with initial epoch 0 (kafka.zk.KafkaZkClient)
[2022-12-05 01:31:11,383] INFO [TransactionCoordinator id=0] Starting up. (kafka.coordinator.transaction.TransactionCoordinator)
[2022-12-05 01:31:11,411] INFO [TransactionCoordinator id=0] Startup complete. (kafka.coordinator.transaction.TransactionCoordinator)
[2022-12-05 01:31:11,412] INFO [Transaction Marker Channel Manager 0]: Starting (kafka.coordinator.transaction.TransactionMarkerChannelManager)
[2022-12-05 01:31:11,429] INFO Feature ZK node created at path: /feature (kafka.server.FinalizedFeatureChangeListener)
[2022-12-05 01:31:11,523] INFO [ExpirationReaper-0-AlterAcls]: Starting (kafka.server.DelayedOperationPurgatory$ExpiredOperationReaper)
[2022-12-05 01:31:11,862] INFO [/config/changes-event-process-thread]: Starting (kafka.common.ZkNodeChangeNotificationListener$ChangeEventProcessThread)
[2022-12-05 01:31:12,027] INFO Updated cache from existing <empty> to latest FinalizedFeaturesAndEpoch(features=Features{}, epoch=0). (kafka.server.Finalize
dFeatureCache)
[2022-12-05 01:31:12,187] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Starting socket server acceptors and processors (kafka.network.SocketServer)
[2022-12-05 01:31:12,207] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started data-plane acceptor and processor(s) for endpoint : ListenerName(PLAI
NTEXT) (kafka.network.SocketServer)
[2022-12-05 01:31:12,209] INFO [SocketServer listenerType=ZK_BROKER, nodeId=0] Started socket server acceptors and processors (kafka.network.SocketServer)
[2022-12-05 01:31:12,378] INFO Kafka version: 3.2.0 (org.apache.kafka.common.utils.AppInfoParser)
[2022-12-05 01:31:12,383] INFO Kafka commitId: 38103ffaa962ef50 (org.apache.kafka.common.utils.AppInfoParser)
[2022-12-05 01:31:12,384] INFO Kafka startTimeMs: 1670184072210 (org.apache.kafka.common.utils.AppInfoParser)
[2022-12-05 01:31:12,387] INFO [KafkaServer id=0] started (kafka.server.KafkaServer)
[2022-12-05 01:31:13,231] INFO [BrokerToControllerChannelManager broker=0 name=forwarding]: Recorded new controller, from now on will use broker cs570bigdat
a:9092 (id: 0 rack: null) (kafka.server.BrokerToControllerRequestThread)
[2022-12-05 01:31:13,234] INFO [BrokerToControllerChannelManager broker=0 name=alterPartition]: Recorded new controller, from now on will use broker cs570bi
gdata:9092 (id: 0 rack: null) (kafka.server.BrokerToControllerRequestThread)
```

Create topic using the given command

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ bin/kafka-topics.sh --create --topic input_recommend_product --zookeeper localhost:2181 --partitions 3 --replication-factor 1
Exception in thread "main" joptsimple.UnrecognizedOptionException: zookeeper is not a recognized option
        at joptsimple.OptionException.unrecognizedOption(OptionException.java:108)
        at joptsimple.OptionParser.handleLongOptionToken(OptionParser.java:510)
        at joptsimple.OptionParserState$2.handleArgument(OptionParserState.java:56)
        at joptsimple.OptionParser.parse(OptionParser.java:396)
        at kafka.admin.TopicCommand$TopicCommandOptions.<init>(TopicCommand.scala:567)
        at kafka.admin.TopicCommand$.main(TopicCommand.scala:47)
        at kafka.admin.TopicCommand.main(TopicCommand.scala)
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ bin/kafka-topics.sh --create --topic input_recommend_product --bootstrap-server localhost:9092 --partitions 3 --replication-factor 1
WARNING: Due to limitations in metric names, topics with a period ('.') or underscore ('_') could collide. To avoid issues it is best to use either, but not both.
Created topic input_recommend_product.
hduser@cs570bigdata:~/kafka_2.12-3.2.0$
```

pip3 install msgpack
pip3 install kafka-python

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ python3 producer.py
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ pip3 install msgpack
Collecting msgpack
  Downloading msgpack-1.0.4-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (322 kB)
     |[redacted]| 322 kB 553 kB/s
Installing collected packages: msgpack
Successfully installed msgpack-1.0.4
hduser@cs570bigdata:~/kafka_2.12-3.2.0$
```

Create producer.py and consumer.py and run in 2 terminals

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ cat producer.py
from kafka import KafkaProducer

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

producer.send('input_recommend_product', b'(1, Main Menu), (2, Phone) , (3, Smart Phone), (4, iPhone)')

producer.close()
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ cat consumer.py
from kafka import KafkaConsumer

consumer = KafkaConsumer('input_recommend_product',
bootstrap_servers=['localhost:9092'])
for msg in consumer:
   print(msg)
hduser@cs570bigdata:~/kafka_2.12-3.2.0$
```

Run producer.py and consumer.py in 2 different terminals

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ python3 producer.py
hduser@cs570bigdata:~/kafka_2.12-3.2.0$
```

hduser@cs570bigdata: ~/kafka_2.12-3.2.0

```
hduser@cs570bigdata:~/kafka_2.12-3.2.0$ python3 consumer.py
ConsumerRecord(topic='input_recommend_product', partition=0, offset=0, timestamp=1670186311427, timestamp_type=0, key=None, value=b'(1
 Main Menu), (2, Phone) , (3, Smart Phone), (4, iPhone)', headers=[], checksum=None, serialized_key_size=-1, serialized_value_size=58,
serialized_header_size=-1)
```

Wordcount Scala Program Run To Count Words

hduser@cs570bigdata: ~

```
ssc: org.apache.spark.streaming.StreamingContext = org.apache.spark.streaming.StreamingContex
t@7546033d

scala> // Create a DStream that will connect to hostname:port, like localhost:9999

scala> val lines = ssc.socketTextStream("10.0.0.50", 9999)
lines: org.apache.spark.streaming.dstream.ReceiverInputDStream[String] = org.apache.spark.str
eaming.dstream.SocketInputDStream@1e060d4b

scala> // Split each line in each batch into words

scala> val words = lines.flatMap(_.split(" "))
words: org.apache.spark.streaming.dstream.DStream[String] = org.apache.spark.streaming.dstrea
m.FlatMappedDStream@11fcf35

scala> // Count each word in each batch

scala> val pairs = words.map(word => (word, 1))
pairs: org.apache.spark.streaming.dstream.DStream[(String, Int)] = org.apache.spark.streaming
.dstream.MappedDStream@2fb3113a

scala> val wordCounts = pairs.reduceByKey(_ + _)
wordCounts: org.apache.spark.streaming.dstream.DStream[(String, Int)] = org.apache.spark.stre
aming.dstream.ShuffledDStream@33ba63e4

scala> // Print the elements of each RDD generated in this DStream to the console

scala> wordCounts.print()

scala> // Start the computation

scala> ssc.start()
```

Netcat listening on port 9999 with input data

hduser@cs570bigdata: ~

```
hduser@cs570bigdata:~$ nc -lk 9999
Hello-World
Hello-World
Hello-World
Big-Data
Big-Data
Big-Data
```

Output of Streaming Data

hduser@cs570bigdata: ~

```
-------------------------------------------
Time: 1669955571000 ms
-------------------------------------------

-------------------------------------------
Time: 1669955572000 ms
-------------------------------------------
(Hello-World,3)
(Big-Data,3)

-------------------------------------------
Time: 1669955573000 ms
-------------------------------------------

-------------------------------------------
Time: 1669955574000 ms
-------------------------------------------

-------------------------------------------
Time: 1669955575000 ms
-------------------------------------------
```

Netcat listening on port 9999 with input data

hduser@cs570bigdata: ~

```
hduser@cs570bigdata:~$ nc -lk 9999
Hello-World
Hello-World
Apple Apple
Apple Apple
CS575
CS575
```

Output of Streaming Data

hduser@cs570bigdata: ~/gweek12

```
scala>

scala> // Wait for the computation to terminate

22/12/02 09:47:08 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with only 0 peer/s.
22/12/02 09:47:08 WARN BlockManager: Block input-0-1669954628600 replicated to only 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 1669954628000 ms
-------------------------------------------

-------------------------------------------
Time: 1669954630000 ms
-------------------------------------------
(Apple,4)
(CS575,2)
(Hello-World,2)

-------------------------------------------
Time: 1669954632000 ms
-------------------------------------------

-------------------------------------------
Time: 1669954634000 ms
-------------------------------------------

-------------------------------------------
Time: 1669954636000 ms
-------------------------------------------

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

- Hint 1: use the netcat server to send the text through a TCP connection. For example, nc -lk 9999 can be used to transfer any text that you type in the terminal through port 9999.

- Hint 2: if you are using local as the master URL when creating StreamingContext, you have to give at least one more core as the number of input streams, because each input stream would create a receiver that occupies one core. If you use local, it only gives one core to the context, which is used by the socket stream's receiver, leaving no core available for processing the data, thus, you should use local[2]

Scala Program NetworkWordCount

```
 To run this on your local machine, you need to first run a Netcat server
 *    `$ nc -lk 9999`
 * and then run the example
 *    `$ bin/run-example org.apache.spark.examples.streaming.NetworkWordCount localhost 9999`
package org.apache.spark.examples.streaming
import org.apache.spark.SparkConf
import org.apache.spark.storage.StorageLevel
import org.apache.spark.streaming.{Seconds, StreamingContext}
object NetworkWordCount {
  def main(args: Array[String]): Unit = {
    if (args.length < 2) {
      System.err.println("Usage: NetworkWordCount <hostname> <port>")
      System.exit(1)
    }
    StreamingExamples.setStreamingLogLevels()
    // Create the context with a 1 second batch size
    val sparkConf = new SparkConf().setAppName("NetworkWordCount")
    val ssc = new StreamingContext(sparkConf, Seconds(1))
    // Create a socket stream on target ip:port and count the
    // words in input stream of \n delimited text (e.g. generated by 'nc')
    // Note that no duplication in storage level only for running locally.
    // Replication necessary in distributed scenario for fault tolerance.
    val lines = ssc.socketTextStream(args(0), args(1).toInt, StorageLevel.MEMORY_AND_DISK_SER)
    val words = lines.flatMap(_.split(" "))
    val wordCounts = words.map(x => (x, 1)).reduceByKey(_ + _)
    wordCounts.print()
    ssc.start()
    ssc.awaitTermination()
  }
}
```

hduser@cs570bigdata:~$ nc -lk 9999

hduser@cs570bigdata: ~

```
PROJECT Kafka SPARK StreamingPROJECT Kafka SPARK StreamingPROJECT Kafka SPARK St
reamingPROJECT Kafka SPARK Streaming STREAMING STREAMING STREAMINGSTREAMINGSTREA
MING STREAMINGSTREAMING
```

Scala Streaming NetworkWordCount localhost 9999

hduser@cs570bigdata: /stage

```
22/12/05 04:24:33 INFO TaskSchedulerImpl: Removed TaskSet 2552.0, whose tasks have all completed, from pool
22/12/05 04:24:33 INFO BlockManagerInfo: Removed broadcast_1288_piece0 on cs570bigdata:45789 in memory (size: 2.9 KiB, free: 434.3
MiB)
22/12/05 04:24:33 INFO DAGScheduler: ResultStage 2552 (print at NetworkWordCount.scala:56) finished in 0.175 s
22/12/05 04:24:33 INFO DAGScheduler: Job 1276 is finished. Cancelling potential speculative or zombie tasks for this job
22/12/05 04:24:33 INFO TaskSchedulerImpl: Killing all running tasks in stage 2552: Stage finished
22/12/05 04:24:33 INFO DAGScheduler: Job 1276 finished: print at NetworkWordCount.scala:56, took 0.214097 s
-------------------------------------------
Time: 1670194473000 ms
-------------------------------------------
(STREAMING,2)
(Kafka,4)
(STREAMINGSTREAMINGSTREAMING,1)
(Streaming,1)
(STREAMINGSTREAMING,1)
(StreamingPROJECT,3)
(SPARK,4)
(PROJECT,1)

22/12/05 04:24:33 INFO JobScheduler: Finished job streaming job 1670194473000 ms.0 from job set of time 1670194473000 ms
22/12/05 04:24:33 INFO JobScheduler: Total delay: 0.396 s for time 1670194473000 ms (execution: 0.392 s)
22/12/05 04:24:33 INFO BlockManagerInfo: Removed broadcast_1289_piece0 on cs570bigdata:45789 in memory (size: 2.8 KiB, free: 434.4
MiB)
```

Conclusion

Spark streaming is extension of core spark api, when used with Kafka and Python / scala we can achieve numerous things, it can mainly used for real time data for example weather, share marketing there are many to mention.

Also, we can place all the data in filesystems for future reference.

Also, we can apply in machine learning algorithms and graph processing.

References

SFBU exercise materials