CSP554—Big Data Technologies (Ruchi Sharma)

Assignment #9

Worth: 10 points (5 points for each exercise) + 5 points extra credit

Exercise 1) Read and provide a half page summary and analysis of this article available on the blackboard in the 'Articles' section: Beyond Batch Processing: Towards Real-Time and Streaming Big Data

Review:

In this article, authors gave a brief survey with focus on two new aspects: real-time processing and stream processing solutions for big data.

Article focuses on three main sections which are as follows: 1.) Strength, features, and shortcomings of standard MapReduce. 2.) Real-time processing solutions and stream processing systems. 3.) A comparative study between both above.

Solutions which comes from real-time processing is further classified into two categories:

1.) Solutions that try to reduce the overhead of MapReduce and make it faster to enable execution of jobs in less than second.
2.) Solutions that focus on providing a means for real-time queries over structured and unstructured big data using new optimized approaches.

Real-Time Big Data Processing:

Article elaborates it in depth for the solutions. There can be many challenges also. As we know, MapReduce and its extensions are mostly designed for batch processing of fully staged big data and they are not appropriate for processing interactive workloads and streaming big data. To remove this some solutions have been introduced. While this type of system sounds like a game changer, the reality is that real-time systems can be extremely hard to implement using common software systems. As these systems take control over the program execution, it brings an entirely new level of abstraction. What this means is that the distinction between the control-flow of your program and the source code is no longer apparent because the real-time system chooses which task to execute at that moment. Another common challenge with real-time operating systems is that the tasks are not isolated entities. The system decides which to schedule and sends out higher priority tasks before lower priority ones, thereby delaying their execution until all the higher priority tasks are completed.

Streaming Big Data:

Nowadays Data Stream is very common. In Standard system before any computation is started, all the input data must be completely available on the input store. Today's applications need more stream-like demands in which the input data is not available completely at the beginning. But there are pros and cons associated with every kind of system. One of the biggest challenges that organizations face with stream processing is that the system's long-term data output rate must be just as fast or faster than the long-term data input rate otherwise the system will begin to have issues with storage and memory.

Exercise 2) Read and provide a half page summary and analysis of this article available on the blackboard in the 'Articles' section: Real-time stream processing for Big Data.

Review:

In this article, authors gave an overview over the state of the art of stream processors for low-latency Big Data analytics and conduct a qualitative comparison of the most popular contenders, namely Storm and its abstraction layer Trident, Samza and Spark Streaming. They described their respective underlying rationales, the guarantees they provide and discuss the trade-offs that come with selecting one of them for a task.

Storm is the first distributed stream processing system to gain traction throughout research and practice and was initially promoted as the "Hadoop of real-time". because its programming model provided an abstraction for stream-processing like the abstraction that the MapReduce paradigm provides for batch-processing. On top of the Java API, Storm is also Thrift-compatible and comes with adapters for numerous languages such as Perl, Python and Ruby. Storm Trident is an extension of Storm that makes it an easy-to-use distributed real-time analytics framework for Big Data. Instead of having to create a lot of Spouts and Bolts and take care of how messages are distributed, Trident comes with a lot of the work already done.

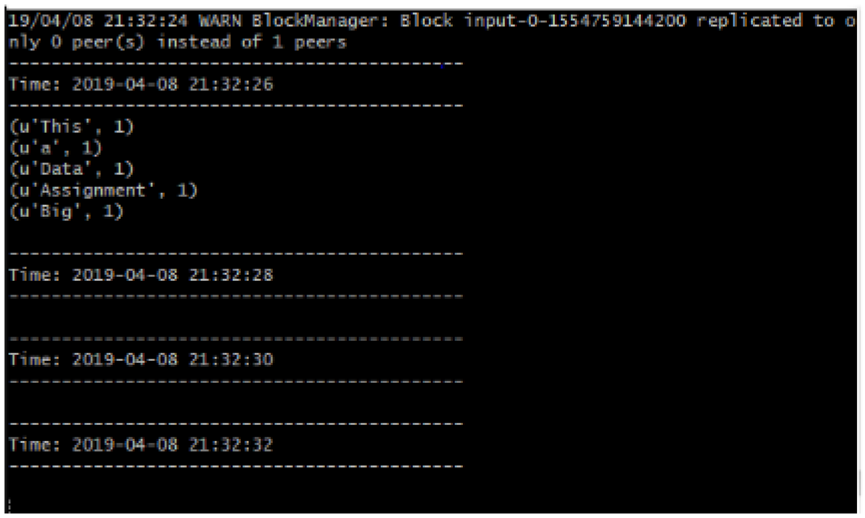
Samza is very similar to Storm in that it is a stream processor with a one-at-a time processing model and at-least-once processing semantics. It was co-developed with the queueing system Kafka. It uses Apache Kafka for messaging, and Apache Hadoop YARN to provide fault tolerance, processor isolation, security, and resource management. It has many pros such as Durability, Scalability, Fault Tolerance, Managed state, Simple API, Processor isolation, Pluggable.

Spark Streaming is a real-time processing tool, that has a high-level API, is fault tolerant, and is easy to integrate with SQL DataFrames and GraphX. It has an easy high level API Easy to integrate with other parts of Spark ecosystem like Spark SQL. In addition to these performance benefits, Spark provides a variety of machine learning algorithms out-of-the-box through the MLlib libraryExercise.

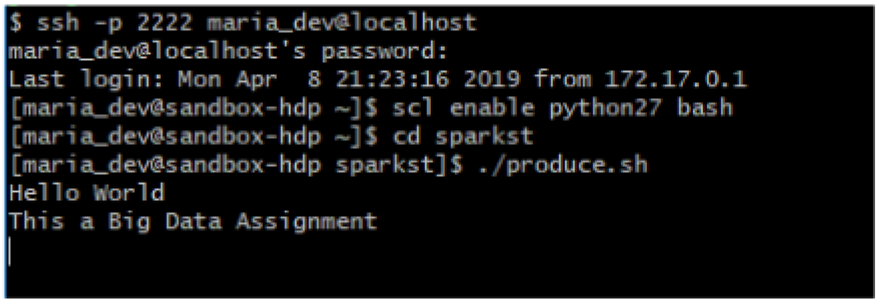
Apart from all these systems In the last couple of years, a great number of stream processors have emerged that all aim to provide high availability, fault-tolerance and horizontal scalability such as Flink, Apex, Twitter's Heron. The list of distributed stream processors goes on. These systems parts gap between batch and stream-processing in both research and practice.

3) 5 points extra credit

(1 point) Get the spark streaming demo code to work in your Hortonworks sandbox. Provide screen shots of output for various inputs

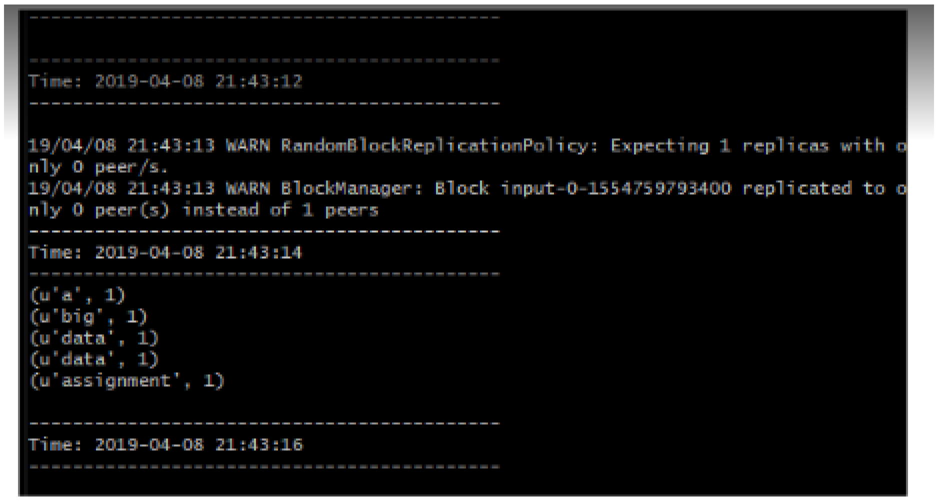


```
19/04/08 21:32:24 WARN BlockManager: Block input-0-1554759144200 replicated to o
nly 0 peer(s) instead of 1 peers
-------------------------------------------
Time: 2019-04-08 21:32:26
-------------------------------------------
(u'This', 1)
(u'a', 1)
(u'Data', 1)
(u'Assignment', 1)
(u'Big', 1)

-------------------------------------------
Time: 2019-04-08 21:32:28
-------------------------------------------

-------------------------------------------
Time: 2019-04-08 21:32:30
-------------------------------------------

-------------------------------------------
Time: 2019-04-08 21:32:32
-------------------------------------------
```

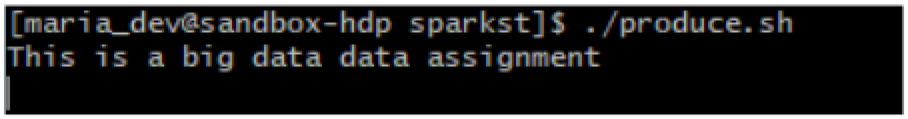


```
$ ssh -p 2222 maria_dev@localhost
maria_dev@localhost's password:
Last login: Mon Apr  8 21:23:16 2019 from 172.17.0.1
[maria_dev@sandbox-hdp ~]$ scl enable python27 bash
[maria_dev@sandbox-hdp ~]$ cd sparkst
[maria_dev@sandbox-hdp sparkst]$ ./produce.sh
Hello World
This a Big Data Assignment
```

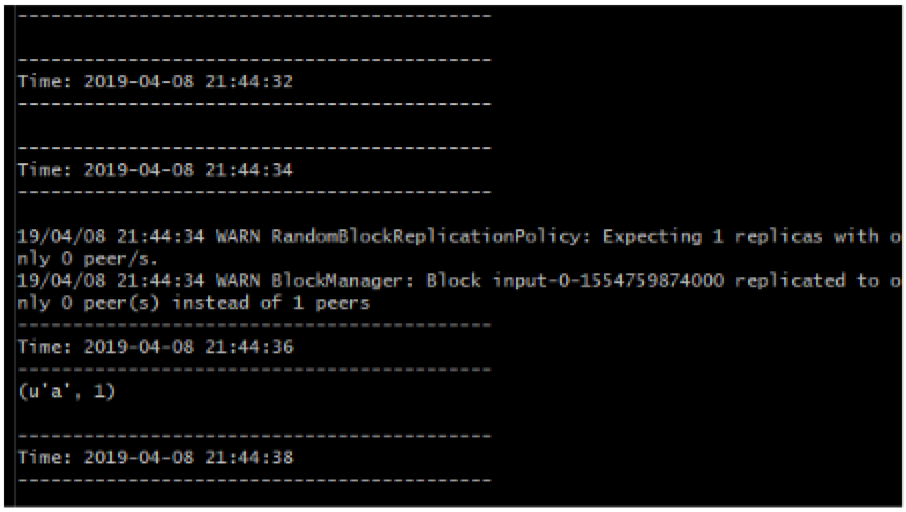
(4 points) Modify consume.py to output a count of words beginning with only the letters a through h inclusive using just RDD transformations and actions. Provide screen shots of output for various inputs.



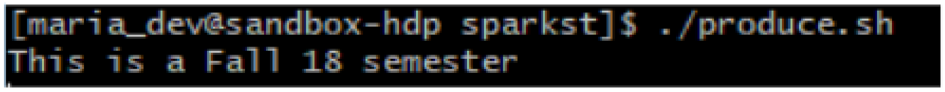
```
--------------------------------------------

--------------------------------------------
Time: 2019-04-08 21:43:12
--------------------------------------------

19/04/08 21:43:13 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with o
nly 0 peer/s.
19/04/08 21:43:13 WARN BlockManager: Block input-0-1554759793400 replicated to o
nly 0 peer(s) instead of 1 peers
--------------------------------------------
Time: 2019-04-08 21:43:14
--------------------------------------------
(u'a', 1)
(u'big', 1)
(u'data', 1)
(u'data', 1)
(u'assignment', 1)

--------------------------------------------
Time: 2019-04-08 21:43:16
--------------------------------------------
```



```
[maria_dev@sandbox-hdp sparkst]$ ./produce.sh
This is a big data data assignment
```



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

--------------------------------------------
Time: 2019-04-08 21:44:32
--------------------------------------------

--------------------------------------------
Time: 2019-04-08 21:44:34
--------------------------------------------

19/04/08 21:44:34 WARN RandomBlockReplicationPolicy: Expecting 1 replicas with o
nly 0 peer/s.
19/04/08 21:44:34 WARN BlockManager: Block input-0-1554759874000 replicated to o
nly 0 peer(s) instead of 1 peers
--------------------------------------------
Time: 2019-04-08 21:44:36
--------------------------------------------
(u'a', 1)

--------------------------------------------
Time: 2019-04-08 21:44:38
--------------------------------------------
```



```
[maria_dev@sandbox-hdp sparkst]$ ./produce.sh
This is a Fall 18 semester
```