# CSP554—Big Data Technologies

## Assignment #9

**Worth: 5 points + 5 points extra credit**

Exercise 1) 5 points
Read the article "Real-time stream processing for Big Data" available on the blackboard in the 'Articles' section and then answer the following questions:

a) (1.25 points) What is the Kappa architecture and how does it differ from the lambda architecture?
   - In streaming data processing, the Kappa architecture is used. The fundamental concept behind Kappa architecture is that you can conduct both real-time and batch processing with a single technology stack, which is particularly useful for analytics.
   - The Kappa architecture also supports historical analytics, which reads the stored streaming data from the messaging engine in a batch manner later to produce additional analyzable outputs for more types of analysis. The Kappa architecture also allows for real-time analytics since the data is read and converted immediately after being loaded into the messaging system.

b) (1.25 points) What are the advantages and drawbacks of pure streaming versus micro-batch real-time processing systems?
   - Micro-batch processing is a method for gathering and operating on data in small groups. In comparison, conventional batch processing often necessitates working with a broad data set. Micro batch processing is a variation of conventional batch processing in which data is processed more often to deal with smaller groups of new data.
   - The world has changed, and there are many applications where micro-batches are insufficient. Organizations commonly use micro-batch processing to make design decisions that resist stream processing. For example, an Apache Spark shop could be using Spark streaming, which is essentially a micro-batch processing extension of the Spark API with the added benefit of using in-memory computing resources. Stream processing allows systems to respond to new data events as they occur. Rather than grouping and gathering data at preset times, stream processing systems capture and process data as it is produced.
   - In some cases, micro-batch processing has sufficed, but companies are rapidly learning that stream processing based on in-memory technology –

whether in the cloud or on-premises – is the better choice. Current applications are rapidly using stream processing technology. As data has exploded over the past decade, businesses have switched to real-time analysis to respond to data closer to the time it was generated to solve for a range of use cases and applications.

c) (1.25 points) In few sentences describe the data processing pipeline in Storm.
   - The Apache Storm is looking for a Zookeeper cluster. At least two distinct components make up a Storm cluster: Storm Nimbus and Storm Supervisor. Storm Nimbus is a close master node to Hadoop Job Tracker. It is in charge of distributing technology across the cluster, assigning tasks to subordinates, and ensuring that there are no mistakes. Storm Supervisor is a node of the workplace. It executes the code provided to it by Storm Nimbus. Storm clusters are stateless and fail-safe. Since Nimbus distributes the code to supervisors if there is a Zookeeper and at least one supervisor node, a Storm cluster will operate without a Nimbus node.
   - The Storm also provides Storm DRPC, Storm Log reader, and Storm UI. Hadoop and other similar systems process data using the idea of a job, which is a batch process that will ultimately produce a result and end. Topologies, which are execution graphs in which each node includes computing logic and each edge defines how data is distributed between nodes, are used instead by Hurricane.
   - Spouts and bolts are the two streams that make up a storm topology. The source stream is represented by a spout. A bolt is a stream data processing agent with the ability to generate new sources. For eg, a spout could connect to the Twitter API and output a tweet stream, which a bolt could then consume and output as a stream of trending topics.

d) (1.25 points) How does Spark streaming shift the Spark batch processing approach to work on real-time data streams?
   - Modifications to the streaming in order to meet real-time requirements, Spark divides the stream of incoming data objects into small batches, converts them to RDDs, and processes them normally. It also takes care of data flow and distribution on its own.

Exercise 2) 5 points extra credit

Follow the document “Instructions for setting up a VM with Kafka” included with this assignment and execute the demo code. Provide enough screen shots to indicate you have completed the document through section 4. Then remember to terminate your VM.