MIDS W205

Exercise #	2	**Title**	Introduction to the Elements of a Streaming application.
Related Module(s)	8,9	**Goal**	Implement an end to end streaming app.
Last Updated	10/23/15	**Expected duration**	15-25 hours.

Introduction

Streaming applications may seem complex but understand how they operate will be critical for a data scientist. In this Exercise we will explore an extensive streaming application analyzing Twitter data. In order to allow you to explore a more complex implementation in a limited amount of time you will be using an existing code base. You will install it, run it, show that you understand it and finally enhance it.

Scope and Goal

In this Exercise, you will capture and process live streaming Twitter data covering the following features:

- Data Streaming
- Capturing the live data
- Processing to get insights

In order to get exposed to a more substantial example of stream processing we will ask you to understand deploy an existing example rather than developing an application from scratch. You will be provided with a Storm application and you will be asked to understand, document, deploy and execute the code.

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

Use Case:

Catching and analyzing live twitter data around your business interest area can give you a deeper understanding of current social trends and demands. Older data can give you information on the mainstream trends over a certain period of time, but live data can give you exact and accurate insights real time. For example, say there is manager who manages live TV ads during a popular TV program, which is broadcasted every week. Basing on the Twitter trends at the time of the show live, the manager can decide which ad would be more contextual and engage viewers even more. This will ensure the viewer's interest in not only in the show but in the ads as well.

So, in this exercise, you will exactly capture live social tweets to see people's live interests, process it real time and actually summarize or aggregate to get insights.

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

Overall Guideline - for all steps

Here is the detail guideline for each of the steps for implementation. You would use the same Amazon AMI for creating your own EC2 server for this exercise. You must have a github account if you wish to store your scripts, data, etc., which is recommended. You may not want to keep your EC2 server live all the time as you will run out of credit that way. So, you could save your work in github as you progress and when you make your sever alive, you can re -pull the code and use. This is optional as applies while you work.

Phase 1
Step 1: Clone Code Tree onto your Server
Step 2: Code Directory & Files Walkthrough
Step 3: Documenting the Code Structure and inter relationships
Step 4: Create a twitter Account if you don't have one

Phase 2
Step 1: Spout & Bolt - review code
Step 2: Spout & Bolt - Commenting Code at line level

Phase 3
Step 1: Bolt & Spout – Execute End to End
Step 2: Collect your observation
Step 3: Collect output results

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

Instructions, Resources and Prerequisites

Resource	What
http://storm.apache.org/documentation.html	Apache Storm Documentation
https://streamparse.readthedocs.org/en/latest/api.html	Stream Parse Documentation

Additional Readings for your self studies:
1. Create an Architecture diagram of covering each component of Apache Storm including the functions of each component.
2. Read further to get to know what is a Storm Tuple.
3. Compare Storm over Spark streaming
4. Where can u output the bolt data?

Infrastructure:

Amazon EC2, AMI, S3, Github
You will be using the Amazon EC2 student's account which is provided to you by UCB. You will be accessing the AMI provided as well to create your own server and work on this.

Here is the AMI Name : **UCB W205 Base** - ami-98848cf0

You can fetch these to your local filesystem using the **wget** program

The Github Repository for the same is :
https://github.com/UC-Berkeley-I-School/data-science-w205/tree/master/exercise_2

Technology:

Apache Storm, Amazon EC2, github, python, Twitter API

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

Exercise Execution Guideline: Week by Week

Here is an overall guideline of implementing a real time system using live twitter data. There could be many variations in real life based on your business case in future. For this exercise, you can follow these following steps:

Week 9
Step 1: Clone Code Tree onto your Server
Step 2: Code Directory & Files Walkthrough
Step 3: Documenting the Code Structure and inter relationships
Step 4: Create a twitter Account if you don't have one

Week 10
Step 1: Spout & Bolt - review code
Step 2: Spout & Bolt - Commenting Code at line level

Week 11
Step 1: Bolt & Spout – Execute End to End
Step 2: Collect your observation
Step 3: Collect output results

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

10. Data Set:

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

http://hortonworks.com/hadoop-tutorial/hello-world-an-introduction-to-hadoop-hcatalog-hive-and-pig/

12. Recommended Readings:

1) https://en.wikipedia.org/wiki/Storm_(event_processor)
2) http://hortonworks.com/hadoop/storm/
3) https://github.com/apache/storm/

4) https://storm.apache.org/documentation/Tutorial.html
5)
6)

KARTHIK, PLEASE ADD A FEW MORE LINES IN HERE

Submissions, Timeline, Assessment Criteria:

Submission 1: 15 Points
Submission Week: Week 10
Submission Items:

a. Spreadsheet with column headings
    i. Folder name
    ii. File name
    iii. Description/Purpose
    iv. Dependency on other files
b. Status on the twitter account creation

Submission 2: 15 Points
Submission Week: Week 11
Submission Items:

a. Spout code
    i. With Line by Line comments
b. Bolt code
    i. With Line by Line comments

Submission 3: 40 Points
Submission Week: Week 12
Submission Items:

a. End to End Run: Process Run - Screen shots (At least 3 of your choice)
b. Top 10 words with popularity – Results
c. processing