CSE4/587 Data-intensive Computing Spring 2018
Due Date: 4/6/2018 by Midnight.

# LAB2: DATA AGGREGATION, BIG DATA ANALYSIS AND VISUALIZATION: B. RAMAMURTHY (VER. 1)

## OVERVIEW:

In this lab, we will expand our skills in data exploration developed in Lab1 and enhance them by adding big data analytics and visualization skills. This document describes Lab2: Data Aggregation, Big Data Analysis and Visualization, involves (i) data aggregation from more than one source using the APIs (Application programming interface) exposed by data sources, (ii) Applying classical big data analytic method of MapReduce to the unstructured data collected, and (iii) building a visualization data product.

We will leverage the data collection and exploratory data analysis skills developed in Lab1 to accomplish the goals of Lab2.

## LEARNING OUTCOMES:

- ✓ Automate data collection from multiple sources using the APIs offered by the businesses
- ✓ Explain the importance of evaluating the reliability of data (for example: social media vs news media)
- ✓ Apply classical big data analytical methods: MapReduce for word count and related family of algorithms
- ✓ Work on Hadoop 2.x, and HDFS and process the data using big data algorithms
- ✓ Learn a high level language-based data analysis by exploring Python as data processing language
- ✓ Apply modern visualization methods and disseminate results using the web/mobile interface

## OBJECTIVES:

The lab goals will be accomplished through these specific objectives:

1. **Explore** a high level programming language for data collection and analysis. This will be Python with its popular libraries such as “pandas”.
2. **Choose** a topic of interest to you. It could be “sports”, “weather” or anything of current interest. Make sure you will get enough data from your data sources on the topic you have chosen.
3. **Aggregate** data from multiple sources to corroborate any findings and outcomes of data analysis.
4. **Install** a virtual machine (VM) image for data storage in HDFS and Hadoop infrastructure.
5. **Familiarize** yourself with **MapReduce (MR)** model and programming using this model.
6. **Code** solutions in Java or Python to process data in <key,value> format using MR model.
7. **Visualize** the outcome of the MR analysis using “wordcloud” visualization tool.
8. **Compare** the outcomes of the same analysis for at least two different sources: first an opinion social media source such a twitter and the second one, a reliable researched source such as NYTimes.
9. **Create** a responsive web interface (web tool) for visualizing the outcome of your analysis.
10. **Document** the complete development process as a README in your submission.

11. **(optionally) Extend** the work to collect "real" big data on a topic and apply sophisticated methods such as Latent Dirichlet Allocation (LDA) and a generate conference poster or paper.

## LAB DESCRIPTION:

Introduction: An important and critical phase of the data-science process is data collection. Several organizations including the federal government (data.gov) have their data available to the public for various purposes. Social network applications such as Twitter and Facebook collect enormous amount of data contributed by their numerous and prolific user. For other businesses such as Amazon and NYTimes data is a significant and valuable byproduct of their main business. Nowadays everybody has data. Most of these data generator businesses make subset of their data available for use by registered users for free. Some of them as downloadable data files (.csv, .xlsx) as a database (.db, .db3). Sometimes the data that needs to be collected is not in a specific format but is available as a web page content. In this case, typically a web crawler is used to crawl the web (pages) and scrap the data from these web pages and extract the information needed. Data generating organizations have realized the need to share at least a subset of their data with users interested in developing applications. Entire data sets are sold as products.

Recall from Lab 1, that an API or application programming interface is a standard, secure and programmatic access to data by an organization that owns the data. An API offers a method for one or two way communication among software (as well as hardware) components as long as they carry the right credentials. These credentials for authentication for programmatic access is defined by another standard OAuth (Open Authentication) delegation protocol [6] or API key in some case as in NYTimes data access [7].

We will collect data about from at least two sources, one opinion-based social media in twitter, and research data in New York Times, for the same topic or key phrase. Process the two data sets collected individually using classical big data methods. Compare the outcomes using popular visualization methods.

### LAB 2: WHAT TO DO?

PAIR PROGRAMMING: We are going to allow pair programming for this lab. You will work in groups of one or two. **(No groups >2)**. You will get an F for the course if your group plagiarizes or copies somebody else's work or some other group's work. You can discuss anything ONLY with your pair team member. **Members in the pair have to work on the entire problem and submit your own notebooks, and submission.**

Preparation: Here are the preliminary requirements for the lab.

1. Development language: We plan to use python. If you do not know the language we are giving an opportunity to do in the part 1 of the lab. You will work with examples in Chapters 3-5 of your text book. We will leave two copies of your text as reference in the Lockwood library.
2. For twitter and NYTimes data you will need to get the appropriate Oauth and AI keys. You do have the Oauth keys from Lab1 and you can reuse it for twitter search API. For NYTimes or any

other API, you will have to apply and get the API keys ready. Now you know how to get access to many other data sources using the standard APIs the data organizations provide.

3. For data analytics, you will need to either use the Hadoop VM we have provided or use a Hadoop installation you are familiar with. You may install it from scratch if you have prior experience with this. Many organizations such as Cloudera provide their bundle. You can also use cloud offerings by aws (amazon), and Google cloud, if you are familiar with them. You have too many choice. Cannot decide? Just use the VM, easy for you and for use to grade.
4. Now for the visualization of the results. We want you to use the d3.js, a very popular javascript library for visualization. We have chosen to introduce d3.js for you understand origins of d3.js and how it came about from NYTimes need for complex visualizations [8].

Part 1: (15%) Complete the python code expositions discussed in Chapter 3-5 of your text book. Keep all the source code in a three directories: Lab2→Part1→Ch3, Lab2→Part1→Ch4, Lab2→Part1→Ch5. This looks like a lot, but it is a good way to learn a language, by doing it, and the code is available to you. If you find any bugs please do write to the author. Sometimes they have bug bounty to offer you. Be a good software citizen. Now you are ready to do big data analytics using python programming language. Due date 3/14/2018

Part 2: (85%) Now that you armed with the language to process your data, gather the data. The second part of project involves (i) aggregating data from multiple sources (ii) process using bug data methods and (iii) visual rending for review and decision making.

a. (5%) Choose a topic of current interest to people in the USA. Something that is in the news. Use the topic as the key word or phrase to aggregate tweets and news articles about the topic for the same period. For the initial prototype just use 1 day, later you can collect these two set of data for the same period from the different sources you have identified. You may have to tweak the phrase to get a good yield of tweets and news articles.
b. (5%) Now import the VM appliance for Hadoop infrastructure and test the basic commands with the sample data provided.
c. (5%) Load the data aggregated in step (a) into the VM, two directories: TwitterData and NewsData. Each directory can have many files of data.
d. (20%) Code and execute MapReduce word count on each of the data sets. Map will clean and parse the data sets into words, remove stop words, and reduce will count the useful words. Twitterdata→TwitterWords and NewsData→NewsWords. Review and visually compare the output for representative words about the topic. You may have to change the search word, obtain new sets of data that may comparable sets of output words. You can use Python or java for your coding language.
e. (10%) Visualize each of the outputs using d3,js and on a simple web page that you create for this lab.
f. (5%) Now repeat the steps c) to e) for larger data set collected over week. May be you will see some convergence in your output.
g. (5%) Now design a web page and feed the results by embedding d3.js code (with replaceable worldclouds) in it, finalizing the display of results. In fact, you should be able to create interactive data product! Input a search topic, we will return the word cloud associated with that topic!
h. (20%) We want to drill deeper into our analysis. Using the smallest data sets you collected in step a), analyze each set (Twitter and News) word co-occurrence for only the top ten words. Assume

context for co-occurrence is the "tweet" in the case of TwitterData, and the paragraph of the news article in the NewsData. Your "map" function emits <word, co-occurring word> and your "reduce" function should collate the co-occurrences for the top ten words and output them in a suitable format.

i. (5%) Document all the activities and how we can use your explorations and repeat them with some other data. Use block diagrams where needed. A well-organized directory structure is a requirement.
j. (5 points) A short video that explains your data analysis and visualization process.

**Infrastructure**: We will provide a virtual image that will run on any Virtual Box. You can also install Hadoop from the scratch if you are good at installing and managing software.

**Submission:**

1. You will create a folder in timberlake named lab2. (Timberlake is a cse server).
2. Every file should have your name only at the top of the notebook and your team member's name in the second line.
3. Store or transfer all the file to lab2 folder on timberlake: yourLastNamePart1.ipynb, yourLastNamePart2.ipyn, **all the data used including curated tweet**s;
4. On timberlake tar the lab2 files into yourLastNameLab2.tar
5. Submit using submit_cse487 filename.tar or submit_cse587 filename.tar

## DUE DATE: 4/6/2018 BY MIDNIGHT. ONLINE SUBMISSION AND DEMO REQUIRED.

## HOW CAN DO WELL IN THIS LAB?

- Start working on it today.
- Please install Virtual Box and download the virtual image from UBox. Make sure it works with the sample data we have provided with in it.
- Leverage your data acquisition knowledge from Lab1. Strat collecting data about the topic of your choice.
- You may not get the data you want in the last minute. You cannot copy data from others.
- Plan, design, prototype, test and iterate through these steps.
- Choose a partner so that you can complement each other in skills and learn from each other.
- Attend TA office hours and recitations every week. Attend any number of office hours by any TA until your questions are answered.
- Enroll in Piazza (CSS4/587) and ask questions. Don't post code. Be civil. This is a public forum.
- Login into timberlake.cse.buffalo.edu and make sure you have an account on cse servers. If not send mail to cse-consult@cse.buffalo.edu to get an account.
- Create a lab2 folder with dummy files, tar/zip the file, submit the zip file and check it out it goes without any problem.
- Finally, no cheating. Do not copy or get the code from somebody. By this you are building a disadvantage. You are missing a golden opportunity to learn. The lab, the languages and tools may be hard for non-programmers, but that is no substitute for hard work. Of course, we will make sure people who cheat are appropriately penalized.

REFERENCES:


1. Twitter API. Twitter Developer https://dev.twitter.com/, last viewed 2017.
2. New York Time Developer Network. https://developer.nytimes.com/, last viewed 2018.
3. D3.js, https://en.wikipedia.org/wiki/Mike_Bostock, last viewed 2018.
4. J. Lin and C. Dyer. Data-Intensive Text Processing with MapReduce, Synthesis Lectures on Human Language Technologies, 2010, Vol. 3, No. 1, Pages 1-177, (doi: 10.2200/S00274ED1V01Y201006HLT007). Morgan & Claypool Publishers. An online version of this text is also available through UB Libraries since UB subscribes to Morgan and Claypool Publishers. Online version available. Use it for coding word count and word co-occurrence.
5. M.Knoll, MR in Python. http://www.michael-noll.com/tutorials/writing-an-hadoop-mapreduce-program-in-python/