CSE4/587 Data-intensive Computing Spring 2017

LAB4: LARGE SCALE DATA (TEXT) PROCESSING WITH HADOOP MAPREDUCE: B. RAMAMURTHY

OVERVIEW:

The hands-on practical learning components of the course comprises two types of activities: labs covering one or two knowledge units (skills, competencies) of data-intensive computing and a single term project serving as a capstone covering the entire data pipeline. In the first half of the course we learned data analytics and quantitative reasoning using R language. In the second half we will focus on big data approaches for data analytics with Hadoop MapReduce and Apache Spark.

In Lab1 we wrote a data client and very simple information server. In Lab2 we worked on data cleaning and data munging. In lab (Lab 3) we applied machine learning algorithms and statistical models to data with the ultimate goal of being able to predict the outcome for a certain query or classify data according to certain criteria. More specifically, we explored algorithms discussed in Chapter 3 of Doing Data Science textbook [1]: linear regression, k-nearest neighbors, k-means. In this lab (lab4), we will be exploring approaches that deal with big data, especially text data, using the Google's MapReduce algorithm [2].

GOALS:

Major goals of the lab4 are to:

1. **Identify** problems solvable using MR approach.
2. **Design MR Algorithms** for solving big data problems involving Write Once Read Many (WORM) data such as historical text, health care data.
3. **Understand and learn** to apply MapReduce (MR) algorithm for processing large data sets.
4. **Store and retrieve** text data in Hadoop Distributed File System (HDFS) as <key,value>.
5. **Implement** the MR solutions designed in steps 2 and 3 on a stand-alone virtual machine or on the cloud (Amazon AWS, Google or cloud service providers).
6. **Interpret the results** to enable decision making.

OBJECTIVES:

The lab goals will be accomplished through these specific objectives:

1. You will be learn problem solving using MR as discussed in the text by Lin and Dyer [3].
2. You will follow the pedagogical pattern: (i) Learn the big data file system through installation of a stand-alone VM, (ii) solve worked out MR problems such as "wordcount" on the VM and using data sources created in the first labs (iii) solve real-world problem using MR and (iv) present the results and interpret the results.

LAB DESCRIPTION:

Introduction: In this age of analytics, data science process plays a critical role for many organizations. 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. Very often data collected is not in the format required for the downstream processes such as EDA, analysis, prediction and visualization. The data needs to be cleaned, curated and munged before modeling and algorithmic processing.

In Lab1 we acquired data from twitter using its REST API and processed it. In Lab2 we prepared the data for (i) question answering (ii) change format to accommodate EDA (iii) understand data by plotting it, and then standardized and attempted to normalize data for supporting further analysis. In Lab3 we analyzed the data with specific algorithms (linear regression, k-nn, k-means), evaluate the error rates of the model (fit) and interpret the results. In Lab 4 we will scale up the data processing with Hadoop file system [8, 9] and MapReduce algorithm [2].

Also we will follow the pedagogical pattern

- Preparation before lab (pre-lab)
- Learn from working on some of the solved problems on Hadoop VM and MapReduce.
- Design MR algorithms and use Hadoop and MR to solve real-world data problems.
- Interpret and document the results.

PREPARATION: Here are the preliminary requirements for the lab. **Time needed: 3 to 4 hours (Day 1)**

1. Work environment: Many options are available for the work environment for this lab. We highly recommend the virtual image that we have provided on the UBBox [4]. Alternatively you can work on Amazon AWS [5] or Google Cloud Platform [6]. Both cost money but you do get a certain allowance as a UB students ($100 cloud services credit from AWS and $300 credit from Google for first time users). If you are using cloud resources make sure you cleanup and not let it run beyond the time you need it.
2. Download the VM and install it on your laptop. It works on Linux, Windows and Mac.
3. Make sure you are able to login as per the instructions given in the simple instruction manual that accompanies this VM.
4. Run the basic “wordcount” with the data given there.
5. Moving data in and out of the VM: Add a different set of data (texts) and check if the “wordcount” is working for the new data set.

LAB 3: WHAT TO DO?

1. **(5 points) What's trending?: Wordcount on tweets (Day 2, Time needed: 3-4 hours)**
   Collect tweets using the approach you used in Lab1. However this time you will collect tweets about a certain domain, say, soccer or economy. Run "wordcount" on the tweets (may be on the @word and #tags) and visualize the output using "tag cloud" or "word cloud'. (This can be dynamic and realtime too!) Use the VM you installed in the preparation for this lab. **Input**: Tweets for a given domain **Output**: Word-cloud for the Input **Processing**: MR on HDFS

2. **(10 points) Word co-occurrence on tweets (Day 3, Time needed: 4-5 hours)**
   First step in sentiment analysis is the co-occurrence of the topic of interests with words representing good or bad sentiments. We will not perform sentiment analysis. We will do just word co-occurrence. Perform word co-occurrence as described in Lin and Dyer's text [3], with pairs and stripes methods for the tweets collected for the step above. Of course, use MapReduce approach with the data stored on the HDFS of the VM you installed. **Input**: Tweets **Output**: Co-occurrence pairs and stripes **Processing**: MR on HDFS

3. **(15 points) Featured Activity 1: Wordcount on Classical Latin text (Day 4,5: Time needed 4-5 hours each day)**
   This problem was provided by researchers in the Classics department at UB. They have provided two classical texts and a lemmatization file to convert words from one form to a standard or normal form. In this case you will use several passes through the documents. The documents needed for this process are available in in the UBbox [7].
   Pass 1: Lemmetization using the lemmas.csv file
   Pass 2: Identify the words in the texts by <word <docid, [chapter#, line#]> for two documents.
   Pass 3: Repeat this for multiple documents.

   Here is a rough algorithm (non-MR version):

```
for each word in the text

normalize the word spelling by replacing j with i and v with u throughout

check lemmatizer for the normalized spelling of the word

if the word appears in the lemmatizer
        obtain the list of lemmas for this word
        for each lemma, create a key/value pair from the lemma and the location where the
        word was found
else
    create a key/value pair from the normalized spelling and
                    the location where the word was found
```

4. **(20 points) Featured Activity 2: Word co-occurrence among multiple documents. (Day 6, 7: 4-5 hours each day).**
   a. In this activity you are required to "scale up" the word co-occurrence by increasing the number of documents processed from 2 to n. Record the performance of the MR infrastructure and plot it as discussed in Chapter 3 of Lin and Dyer's text[3]. Also see the performance evaluation charts on p.56 (60) of the same text.
   b. From word co-occurrence that deals with just 2-grams (or two words co-occurring) increase the co-occurrence to n=3 or 3 words co-occurring. Discuss the results and plot the performance and scalability.

5. **You have to work on your own. This is an individual lab. You will get an F for the course if you plagiarize or copy somebody else's work or share your work with somebody.**

DUE DATE: 4/16/2017 BY 11.59PM. ONLINE SUBMISSION.

REFERENCES:

[1] C. ONeil and R. Schutt, Doing Data Science, ISBN:978-1-4493-5865-5. Oreilly Media, Doing Data Science, http://shop.oreilly.com/product/0636920028529.do, 2013.

[2] Dean, J. and Ghemawat, S. 2008. **MapReduce: simplified data processing on large clusters.** *Communication of ACM* 51, 1 (Jan. 2008), 107-113.

[3] J. Lin & C. Dyer. Data-intensive text processing with MapReduce, https://lintool.github.io/MapReduceAlgorithms/

[4] Virtual Machine for Hadoop MapReduce on UBbox: https://buffalo.box.com/s/52did77hn2vjoje7iguf19btgs6vhvsc, last viewed 2017.

[5] Amazon AWS Elastic MapReduce. https://aws.amazon.com/emr/ , Last Viewed April 2017.

[6] MapReduce on Google App Engine. https://cloud.google.com/appengine/docs/standard/python/dataprocessing/, last viewed April 2017.

[7] UBBox data set for Lab4: https://buffalo.box.com/s/s4v5zsod0djle5l8xa5ml9imeei28n71 last viewed 2017.

[8] Apache Hadoop: http://hadoop.apache.org/, last viewed 2017.

[9] Hadoop: The Definitive Guide, by Tom White, 2nd edition, Oreilly's , 2010.