CS8803 BDS / CS4365

Homework Assignment 1

(Programming Category)

Student Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Student Session: cs8803 or CS4365 (circle one)

You are given three types of programming problems in the first homework assignment. You only need to choose one of them as your first homework. Feel free to choose any of your favorite programming language Java, C, Perl, Python, …

Post Date: Before or on Monday of Week 3 (Aug. 31)

Due Date: midnight on Friday of Week 4 (Sept 11)

**Problem 1. Learning HDFS and Hadoop MapReduce.**

*Suitable for students who are the beginner of Hadoop MapReduce Platform*

Install HDFS and Hadoop MapReduce on your laptop, and run the word count map-reduce program, and report the runtime for two different sizes of datasets.

You may use excel file to generate your runtime statistics plot or organize the performance measurement data in a tabular format.

You are encouraged to learn by observing the runtime performance of Hadoop MapReduce program through different ways of programming the same problem and show their impact on the runtime performance of the MapReduce job.

Deliverable.

1. Source code
2. screen shots of your execution process.
3. Runtime statistics in excel plots or tabular format.

**Problem 2. Learning Data Collection Challenges imposed by big data in volume and variety by writing and running a Web Crawler**

*Suitable for students who are interested in understanding the volume challenge of big data through a strawman program.*

You are asked to write a focused web crawling, a strawman version by either modifying an open source Web Crawler package or written from scratch by following the five steps below:

1. Choose a domain of interest, such as cc.gatech.edu.
2. Choose topic specific rules to narrow down your search scope, say faculty home pages, course pages, student home pages.
3. Follow the robot exclusion standard
4. Collect crawling statistics, including #pages/min crawled, the number of keywords and URLs encountered, and display your statistics in curve or tabular plots. If you do not remove duplicates, then it would be great for you to display the amount of duplicates crawled.
5. You are encouraged to only crawl a portion of the content, e.g., the top 100 lines of each page encountered.

Optional Functionalities:

* HTML pages are notoriously error‐prone􏰁browsers go to great length to tolerate imperfect HTML pages; you may find it not easy to do. Feel free to skip them but it would be great to keep a statistics.
* Not to visit the same URL twice
* URL normalization
* Crawl at a determined slow rate (e.g., sleep a second between consecutive page downloads)
* How to store the crawled pages/content

Deliverable:

1. Source code and executable with readme.
2. Crawl Statistics (Crawl speed🡪 #pages/minute, ratio of #URL crawled / #URL to be crawled, etc.)
3. ScreenShots of your Web Crawler’s Command lines or GUIs

Feel free to use any open source crawler as the code base.

**Problem 3.**

**Learning Data Analytics with Hadoop MapReduce and/or SPARK**

This problem has a number of options. Option 1 and Option 2 are designed for those students who are familiar with Hadoop MapReduce programming and would like to try something more challenging. Option 3 is designed for students who are familiar with both Hadoop MapReduce and Spark and interested in hand-on comparison of them through example data problems and configuration tuning.

**Option 1.**

Write a MapReduce program to solve a data analytics problem of your own interest.

**Option 2.**

The White House Visitor Log dataset available at

<http://www.whitehouse.gov/files/disclosures/visitors/WhiteHouse-WAVES-Released-0827.csv> . The attributes in this dataset are described at:

<http://www.whitehouse.gov/files/disclosures/visitors/WhiteHouse-WAVES-Key-1209.txt> . A spreadsheet of the data can be found at:

<http://www.whitehouse.gov/briefing-room/disclosures/visitor-records> .

You are required to write an efficient MapReduce programs to find the following information:

(i) The 10 most frequent visitors (NAMELAST, NAMEFIRST, NAMEMID) to the White House.

(ii) The 10 most frequently visited people (visitee\_namelast, visitee\_namefirst) in the White House.

(iii) The 10 most frequent visitor-visitee combinations.

(iv) Some other interesting statistics that you can think of.

Throughout this programming assignment, do not limit your programs to run with a single reduce task only.

**Option 3.**

Compare Hadoop MapReduce and SPARK using two common analytic problems: a simple one like sort or word count, and a complex one like Clustering or k nearest neighbor search.

Deliverable (for any one of the three options):

1. Source code and executable with readme.
2. ScreenShots of your execution process/evvironments
3. Input and output of your program
4. Runtime measurements