CSE 4/587 - Practice Midterm Exam #2

Below you will find questions taken from the FA22 midterm and final exams that relate to the content we've covered in SP23 thus far. These questions should be used to familiarize yourself with the styles of questions asked in this class. **It is by no means an exhaustive list of content that may show up on the midterm, nor is it indicative of the length of the midterm.** Additionally, some questions may contain content that has not been covered yet in this semester's offering of the course.

Name: _____ Person #: _____

UBIT: _____ Seat #: _____

Academic Integrity

My signature on this cover sheet indicates that I agree to abide by the academic integrity policies of this course, the department, and university, and that this exam is my own work.

Signature: _____ Date: _____

Instructions

1. This exam contains 9 total pages (including this cover sheet). Be sure you have all the pages before you begin.
2. Clearly write your name, UBIT name, person number, and seat number above. **Additionally, write your UBIT name at the top of every page now.**
3. You have 50 minutes to complete this exam. Show all work where appropriate, but keep your answers concise and to the point.
4. After completing the exam, sign the academic integrity statement above. Be prepared to present your UB card upon submission of the exam paper.
5. You must turn in all of your work. No part of this exam booklet may leave the classroom.

DO NOT WRITE BELOW

Q1	Q2	Q3	Q4	Total
20	15	35	10	70

Question 1 - Hadoop and HDFS [20 Points]

a. List two major differences between Hadoop1.x and Hadoop2.x versions. [4 points]

b. How is an HDFS block replicated? Where are map and reduce tasks executed? [4 points]

c. In HDFS, what is a (i) heartbeat (ii) BlockReport? Explain. [4 points]

d. List two functions of a NameNode. List two functions of a DataNode. [4 points]

e. What is the primary data type of the MapReduce model? Why are Maps able to run in parallel over the data? [4 points]

Question 2 - MapReduce [20 Points]

SETI@home is a long-running project searching for extra-terrestrial life by analyzing radio frequency signals recorded by various telescopes. The radio signal intensity was scaled and “printed” (stored) as integers from 0-35 inclusive, with digits from 0-9, and a-z representing 10-35. We want to configure a Hadoop-MapReduce infrastructure to analyze this voluminous repository for any significant contact from extraterrestrials. Consider a Hadoop-MapReduce configuration as given below:.

- We use the word count algorithm from class. Here a "word" is a digit or character representing the SETI signal. Our reducer class is also used as our combiner class.
- Assume that the input has a total of **G = 40Tbyte data**. (1T = 10^{12} bytes, 1M= 10^{6} bytes)
- Input corpus is split equally into **S** sites, each running a MR cluster.
- Assume you plan to configure **M = 200** mappers per site. There are **R** reducers.

Answer the following questions for the configuration listed above:

a) What is the: [6 points]

(i) Input keyspace of the mappers?

(ii) Size of the input processed by each *site*?

(iii) Workload of each mapper in bytes?

b) Assume that mappers suppress the range of values (0-15) and emit only the radio signals of values (16-35) inclusive. Assume that all combiners will run right before the shuffle and sort step. [6 points]

(i) What is the maximum number of <key,value> pairs that will be shuffled and sorted?

(ii) How many distinct keys will each reducer have to reduce?

(iii) How many <key,value> pairs will be in the final output?

d) Draw a diagram that shows how the data flows in your MapReduce application, starting from input and resulting in output. Include mappers, reducers, and combiners. Label your diagram with the expressions you have derived above. [8 points]

Question 3 - Word Co-Occurrence [20 Points]

Computing word Co-Occurrence is an important problem in a number of different domains that involves counting the number of times one word appears in the same context as another. Sequentially, this can be accomplished by creating an $\boldsymbol{N}$ x $\boldsymbol{N}$ matrix $\boldsymbol{M}$, where $\boldsymbol{N}$ is the number of words in our vocabulary, and $\boldsymbol{M}_{ij}$ is the number of times that word $\boldsymbol{w}_i$ appears in the same context as word $\boldsymbol{w}_j$.

a) Write pseudocode for a mapper and a reducer to compute word co-occurrence using the pairs approach. You can assume that the function ***Neighbors*(*w*)** is already defined for you, and returns a list of words in the same context as $\boldsymbol{w}$. [12 points]

b) The other approach for computing word co-occurrence is using stripes. How do pairs and stripes relate to our original sequential formulation using matrix $\boldsymbol{M}$. [4 points]

c) Describe one advantage and one disadvantage that the stripes approach has compared to the pairs approach. [4 points]

Scrap Paper