香 港 中 文 大 學
The Chinese University of Hong Kong

Course Examinations 2013-14 (1st term)

Course Code & Title : **CSCI4180 Introduction to Cloud Computing**

Time allowed : **2** **hours** **0** **minutes**

Student I.D. No. : .... **Seat No. :** ....

You have two hours to complete the exam. All questions are to be completed. The full score is **100 points**. This is an open-book, open-note exam. You are allowed to use an electronic calculator approved by the University. Other electronic equipments are prohibited. Write down your **student ID** and **seat number** in the answer book. Write **all** the answers in the answer book. Write **neatly**. Anything that is unreadable will receive zero point.

Questions

1. **(32%) Short questions.** Please try to limit the answer to each question within **20 words**. Precise and neat answers are preferred.

   (a) (4%) State one similarity and one difference between cloud computing and grid computing.

   (b) (4%) State one similarity and one difference between a public cloud and a virtual private cloud.

   (c) (4%) Explain one solution that can properly handle the namenode failure in HDFS, without requiring any changes to HDFS.

   (d) (4%) Explain one type of input data where in-mapper combining can improve the performance of the default combiner in the wordcount problem.

   (e) (4%) State two reasons why relational databases may give poor performance to Google products.

   (f) (4%) Load balancing may not help improve the performance of BigTable. Explain why. Give two reasons.

   (g) (4%) Explain the reason why Haystack stores photos in the log-structured format instead of storing photos as individual files.

   (h) (4%) State one advantage and one disadvantage of type-2 virtualization over type-1 virtualization.

2. **(9%) Design Trade-offs of Amazon Dynamo.**

   Amazon Dynamo uses three parameters (N, R, W) to determine the read performance, write performance, availability, and consistency. By default, it uses the configuration (3,2,2). Explain how the following parameters will affect (1) read performance, (2) write performance, (3) availability, and (4) consistency, compared to the default configuration (upgraded, degraded, or unchanged). Make sure you provide justifications for your answers.

   (a) (2,1,1)?

   (b) (4,3,3)?

   (c) What are the default parameters (N, R, W) in HDFS? Explain your answers.

3. **(15%) Analytics using MapReduce.**

   Suppose that we are given a table of employee records in the following format: `(name, age, country, salary)`. Our goal is to compute the *variance* of the salaries of all employees for each age group in each country. We define three age groups: "less than 25", "between 25 and 45", and "over 45". We also want to improve the performance of the MapReduce program via the *default combiner*. Write the pseudo-code of the MapReduce program for the corresponding functions.

   Hint: For a sequence of numbers $\langle x_1, x_2, \cdots, x_n \rangle$, we can compute the variance as follows

$$v = \frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n}(\sum_{i=1}^{n} x_i^2) - \bar{x}^2,$$

where $\bar{x}$ is the mean of the sequence of numbers.

4. **(16%) Single-source Shortest Path**

   (a) (10%) Consider the graph in Figure 1. Show how you can use MapReduce to find the shortest paths from node A to all other nodes in the graph. Edges are undirected (e.g., both links A-B and B-A have weight 8). Show (i) how you initialize the shortest-path values, (ii) the emitted outputs of the Map and Reduce functions of the *first two iterations*. Feel free to define the notation that is necessary.

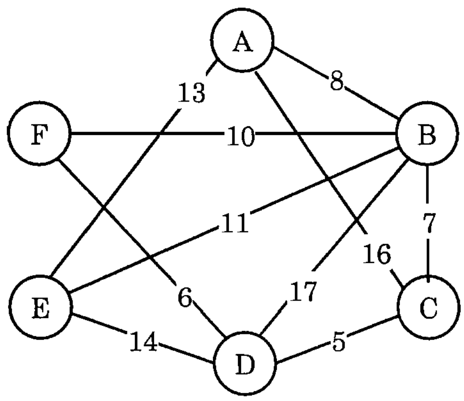

Figure 1: Graph for Question 4.

   (b) (6%) The parallel BFS algorithm that we discussed in class only finds the shortest distances, but doesn't find the shortest paths. Use the Parallel BFS pseudo-code as a starting point (see the lecture notes), explain how you change the implementation (including the data structure definition and the map and reduce functions) to trace the *predecessor* (i.e., the previous node along the shortest path) of each node.

5. **(22%) Deduplication and cloud storage**

   Suppose that we develop a deduplication system for cloud storage. We use Rabin fingerprinting as the chunking algorithm. Let us assume that we store a file with the following byte sequence $\{t_1, t_2, \cdots\}$, where t_i denotes the i^{th} byte. Then we operate on the byte sequence and generate Rabin fingerprints (RFPs) based on the following formula:

$$p_s(d, q) = \begin{cases} (\sum_{i=1}^{m} t_i \times d^{m-i}) \ mod\ q, & s = 0 \\ (d \times (p_{s-1} - d^{m-1} \times t_s) + t_{s+m}) \ mod\ q, & s > 0 \end{cases}$$

   for some constant parameters m, d, and q. We pick $q = 2^k$ for some integer k, and an anchor point is generated if $p_s == 0$. Here, we do *not* bound the maximum chunk size and do not specifically handle the long runs of zeroes.

   (a) (3%) Explain one scenario where the average chunk size is greater than $q = 2^k$.

   (b) (3%) Explain one scenario where the chunking results are equivalent to fixed-size chunking.

(c) (4%) Explain the drawback if (i) d is too small and (ii) d is too large.

(d) (6%) Explain how the following metrics are affected (upgraded, degraded, or unchanged) by increasing the value of q: (i) storage efficiency (i.e., the amount of space saving), (ii) memory usage for indexing, and (iii) read performance.

(e) (3%) Suppose that we modify our chunking implementation such that an anchor point is generated if $p_s == 0$ or $p_s == 1$. Give one advantage and one disadvantage. Explain your answers.

(f) (3%) For security reasons, we want to make sure that only the data chunks belonging to the same user can be deduplicated with each other. In other words, even if user A and user B have the same chunk, we store two copies of the chunk instead of one, so different users don't share the same physical chunk. Explain how you would revise the deduplication algorithm to achieve this goal.

6. **(6%) Zookeeper**

(a) (4%) Herd effect is no longer an issue if the number of clients is small. Suppose that we only have a small number of clients and we want to implement a distributed lock. Write the pseudo-code of the Zookeeper implementation of the distributed lock, such that any pending client can obtain the lock with the equal probability.

(b) (2%) Zookeeper is used in HBase. Explain one scenario how Zookeeper is used in the HBase implementation.

Hope you have a fruitful winter break!

-End-