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

Course Examinations 2012-13 (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%) Explain two reasons why enterprises may hesitate to use cloud computing.

   (b) (4%) A service provider claims to provide public cloud computing services. Users can send an email request to the helpdesk 30 minutes in advance in order to access the services for seven days. Is it a real cloud computing model? Explain two reasons to justify your answer.

   (c) (4%) Explain two types of applications whose performance can become worse when MapReduce is used.

   (d) (4%) Explain one scenario where in-mapper combining hurts the MapReduce performance.

   (e) (4%) What is the function of Chubby in BigTable during normal operations?

   (f) (4%) In the old NFS-based design of Facebook Haystack, how many I/Os are required to store one photo? Briefly explain how you derive the answer.

   (g) (4%) Explain the difference between the EPHEMERAL and EPHEMERAL_SEQUENTIAL flags in Zookeeper.

   (h) (4%) Explain one advantage of hardware-assisted virtualization over paravirtualization.

2. **(12%) On the performance of HDFS.**

   There are many different ways to improve the read/write performance of HDFS. For each of the following changes made to a Hadoop platform, explain how it affects *both* read throughput and write throughput (i.e., upgrade, downgrade, or no change).

   (a) Increase the replication factor.

   (b) Increase the block size.

   (c) Employ inline deduplication in HDFS.

3. **(10%) 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 in each country. We also want to improve the performance of the MapReduce program via *in-mapper combining*. Write the pseudo-code of the MapReduce program for the corresponding mapper and reducer 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%) PageRank**

Suppose that we have a web graph containing only four webpages denoted by n_1, n_2, n_3, and n_4. Our goal is to compute the PageRank of each page using MapReduce. The webpages contain URL links to other webpages, as defined below:

- n_1 contains links to n_2, n_3, and n_4;
- n_2 contains a link to n_4;
- n_3 contains links to n_1 and n_4; and
- n_4 contains links to n_2 and n_3.

(a) (12%) Show how you can solve for the PageRank values for all webpages using MapReduce. We assume that there is no random jump. In your answers, you need to show (i) how you initialize the PageRank 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.

(b) (4%) Derive the final PageRank value of each webpage. (Hint: No MapReduce is required here. You can use simple mathematics to derive the final answers.)

5. **(30%) 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:

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

for some constant parameters m, d, and q.

(a) (4%) Suppose that we store a file of size F and $q = 2^k$ for some integer k. The anchor points are generated based on the following pseudo-code:

```
if ((RFP & q) == 0) {
    put one anchor point;
}
```

How many RFPs are generated on average? Explain your answer. State any of your assumptions.

(b) (4%) Suppose that the file (of size F) to be stored contains a very long range of zeroes. How many RFPs will be generated? Explain your answer. State any of your assumptions.

(c) (4%) Referring to the situation in Part (b). Explain how you would modify the pseudo-code of the anchor point generation in Part (a) to reduce the number of RFPs being generated.

(d) (4%) In practice, we also bound the maximum chunk size when Rabin fingerprinting is used. Explain why it is necessary. If we don't bound the maximum chunk size, explain a possible file content pattern where the chunk size can be unbounded.

(e) (4%) Suppose that we store a large number of files, and we apply the chunking algorithm to each file independently. It is found that both Rabin fingerprinting and fixed-size chunking achieves the *same* storage saving. What is the possible reason? Explain your answers.

(f) (4%) Suppose now that we use fixed-size chunking of size 8KB. We store data on a harddisk of size 160GB. We generate fingerprints using SHA-1 (160 bits long). To reduce the memory usage for indexing, we design an in-memory indexing structure as a Bloom filter with a false positive probability 0.001. What is the minimum size of the indexing structure? Explain your answer.

(g) (3%) Explain how you would combine sparse indexing with the Bloom filter to further reduce the memory usage for indexing.

(h) (3%) To upload data to the cloud with minimum bandwidth, we can apply deduplication on the client side and upload only modified chunks. Patrick argues that it poses new security risks. Explain one possible security attack.

Hope you have a fruitful winter break!

-End-