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

Course Examinations 2011-12 (2nd 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. 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. **(30%) Short questions.** The answer to each question must be at most **20 words**. Any answer in excess of 20 words will receive zero point.

   (a) (4%) Name two differences between public clouds and traditional data centers.

   (b) (3%) Multitenancy is a necessary component for cloud computing, but could be problematic. Explain one potential problem due to multitenancy.

   (c) (3%) Cloud clients need to avoid the lock-in problem when using cloud computing services. Explain one solution that can relieve the lock-in problem.

   (d) (3%) Describe one type of application whose performance can become worse when MapReduce is used.

   (e) (3%) Explain the straggler problem in MapReduce.

   (f) (4%) Name one similarity and one difference between BigTable and MapReduce.

   (g) (3%) The Amazon Dynamo system uses consistent hashing, instead of simple hash functions such as "`SHA256(x) mod n`", where x is the data item and n is the number of server. Why?

   (h) (4%) In the Amazon Dynamo system, for a particular piece of data, let:

   - the number of replicas involved be N.
   - the number of replicas that a client reads from be R.
   - the number of replicas that a client waits for the write respond be W.

   What are the pros and the cons for a system to have a configuration of (N, W, R) = (3, 1, 1)?

   (i) (3%) The Facebook Haystack system stores photos into a big log-structured file, instead of storing each photo as one file. Why?

2. **(9%) Boosting MapReduce performance.**

   There are many different ways to improve the throughput performance of a MapReduce job. For each of the following changes made to a Hadoop platform, explain how it affects the MapReduce throughput (i.e., upgrade, downgrade, or no change).

   (a) Increase the split size.

   (b) Increase the replication factor.

   (c) Connect all computing nodes with a high-speed switch.

3. **(9%) 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 by implementing the combiner function. Write the pseudo-code of the MapReduce program for the corresponding Map, Combiner, and Reduce 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.

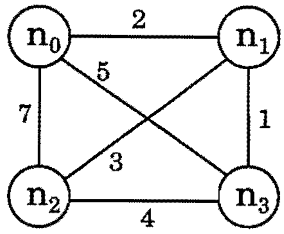

Figure 1: Network for Q4.

4. **(12%) Shortest-path algorithm.**

   Consider the network shown in Figure 1. Using the extended version of the parallel BFS algorithm based on MapReduce, show how you can solve for the shortest path distance from node n_0 to each of other nodes. You may define N_i (where $0 \leq i \leq 3$) as the node structure of node n_i that contains the adjacency list of node n_i and the current shortest path distance from node n_0. In your answers, you need to show (i) the initialized shortest path distances of all node structures, (ii) the emitted outputs of the Map and Reduce functions in each MapReduce iteration, and (iii) the stopping criterion.

5. (28%) **Rabin Fingerprinting and Deduplication.**

   (a) Let us consider the Rabin fingerprinting algorithm.

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

   i. (2%) Which parameter is controlling the minimum chunk size?

   ii. (3%) Generally speaking, it is not favorable for a deduplication storage system to have a large value for the minimum chunk size, e.g., 100K. Why?

   iii. (3%) Generally speaking, it is not favorable for a deduplication storage system to have a small value for the minimum chunk size, e.g., 2. Why?

   iv. (3%) Although there is no restriction in choosing its value, why is the parameter "q" usually set to be 2^i where $i > 0$?

   v. (3%) When there is a long stream of zeros in a file? How will the variable-size chunking scheme react?

   (b) A Bloom filter is usually deployed before you start searching in the indexing structure.

   i. (4%) Why is deploying the Bloom filter beneficial? Please explain your answer in terms of operations that the Bloom filter can save.

   ii. (4%) Is it beneficial to deploy the Bloom filter with a lot of hashing function? Explain your answer.

iii. (2%) It is not beneficial to deploy the Bloom filter with a lot of data items expected to be inserted. Explain your answer.

iv. (4%) To solve the above problem, Dr. Wong suggests that when the number of items inserted in the Bloom filter exceeds a certain threshold, say T, we can double the number of bitmaps so as to make the Bloom filter deployment beneficial again.

An example implementation is:

- Clear all bitmaps, and double the number of bitmaps.
- Say currently there are 2^i bitmaps, where $i > 0$. Name the bitmaps from 0 to $2^i - 1$ inclusively.
- Re-insert all the keys in the indexing structure.
- If the least significant i bits of the key is k, where $0 \leq k < 2^i$, re-insert the key into the k bitmap.

Why does Dr. Wong's suggestion can "*make the Bloom filter deployment beneficial again*"?

6. (12%) **Zookeeper.**

(a) (2%) In the leader election implementation for zookeeper, unlike the classical election algorithm, the zookeeper version chooses the one with the **smallest znode sequential number**. Why?

(b) (10%) Do you still remember the reader-writer problem that you learnt in the OS course? We are going to write a set of pseudocodes using the Zookeeper.

- We are not going to distinguish whether a process is a reader or a writer. Rather, a process is trying to obtain a **read lock** or a **write lock**, but not both.
- A process is granted a read lock if there is no process granted a write lock. Hence, there can be more than one process sharing the read lock.
- A process is granted a write lock if there is no process granted a read lock nor a write lock.
- If a process is not granted a lock, it should be waiting for the lock until it is granted the required lock.
- When there are more than one process waiting for a lock, you can allow any one of the waiting process to be granted the lock.

Obtaining Read lock.	**Obtaining Write lock.**
Step 1. Create a znode ...	Step 1. Create a znode ...

Hope you have a bright future!!

-End-