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

Course Examinations 2015-16 (1st term)

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

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**. You are allowed to bring two A4-sized cheat sheets, with both sides printed or written, and 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. **(40%, 5% each) Quick Questions.** Keep your answers short (e.g., 20 words) and precise.

   (a) Explain what "resource pooling" and "elasticity" mean in cloud computing.

   (b) MapReduce emphasizes locality. Explain what it means and why it is necessary.

   (c) What is a straggler in MapReduce? How can we eliminate the straggler problem from MapReduce?

   (d) In MapReduce, why does the stripe approach improve the performance of pair approach in general? Give two reasons.

   (e) In the BigTable implementation of WebTable, we store the reverse URL as the row key. What is a reverse URL? Why is it being used?

   (f) Explain one advantage and one disadvantage for Amazon Dynamo to have a configuration $(N, R, W) = (3, 1, 1)$.

   (g) Facebook's Haystack avoids relying on CDN and NFS to keep photos. State one drawback for using each of the approaches.

   (h) Zookeeper uses atomic broadcast. Explain what atomic broadcast is.

2. **(16%) MapReduce.**

   Suppose that we are given the employee records, each of which is represented as a string `record` = "name, age, country, salary". Each record is identified by the employee ID (denoted by an integer `employeeID`). Our goal is to compute the *variance* of the salaries of all employees for each age in each country. We use MapReduce to solve this problem.

   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.

   We use the *default combiner* to improve the performance.

   Write the pseudo-code of the map, combine, and reduce functions, based on the following map and reduce interfaces:

   - map (**Integer** `employeeID`, **String** `record`)
   - combine (**Type1** k, **Type2** [v1, v2, $\cdots$])
   - reduce (**Type3** k, **Type4** [v1, v2, $\cdots$])

   You'll need to decide the correct data types in both the combine and reduce functions.

   Hint: For your reference, Figure 1 illustrates the MapReduce pseudo-code of the Mean Computation Version 2 in Lecture 4 (note: the pseudo-code may have some problems itself).

```
1: class Mapper
2:     method Map(string t, integer r)
3:         Emit(string t, integer r)

1: class Combiner
2:     method Combine(string t, integers [r_1, r_2, ...])
3:         sum ← 0
4:         cnt ← 0
5:         for all integer r ∈ integers [r_1, r_2, ...] do
6:             sum ← sum + r
7:             cnt ← cnt + 1
8:         Emit(string t, pair (sum, cnt))          ▷ Separate sum and count

1: class Reducer
2:     method Reduce(string t, pairs [(s_1, c_1), (s_2, c_2) ...])
3:         sum ← 0
4:         cnt ← 0
5:         for all pair (s, c) ∈ pairs [(s_1, c_1), (s_2, c_2) ...] do
6:             sum ← sum + s
7:             cnt ← cnt + c
8:         r_avg ← sum/cnt
9:         Emit(string t, integer r_avg)
```

Figure 1: MapReduce Pseudo-code of Mean Computation Version 2.

3. **(12%) Shortest Path Algorithm.**

Consider the network shown in Figure 2. Using the parallel Dijkstra's 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, and (ii) the emitted outputs of the Map and Reduce functions in each MapReduce iteration. You only need to show the *first two* iterations.

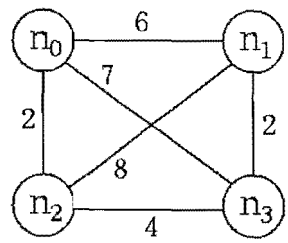

Figure 2: Network for Q3.

4. **(20%) Cloud storage.**

We apply inline deduplication to streams of fixed-size data chunks. We also store the chunks in units of containers. Each container can store at most four chunks.

Suppose that we have already stored four containers, with a total of 16 chunks. The containers are denoted by:

- $C1 : (D1, D2, D3, D4)$,
- $C2 : (D5, D6, D7, D8)$,
- $C3 : (D9, D10, D11, D12)$, and
- $C4 : (D13, D14, D15, D16)$,

where $C1 - C4$ are containers, and $D1 - D16$ are chunks.

Now, we have a new stream of chunks to be stored:
S: $(D1, D3, D5, D7, D8, D11, D16, D17, D18, D19, D20, D21, D22)$.

We assume that if chunks have identical content, they will have the same identifier (e.g., the $D1$ in S has the same content as the already stored $D1$ in $C1$).

(a) (2%) Show all containers that will be used to store the chunks of stream S after inline deduplication.

(b) (8%) Suppose that we apply capping to limit the number of containers. Suppose we set $T = 2$. Show all containers that will be used to store the chunks of stream S after inline deduplication. Explain your steps. Also, explain how capping improves read performance in this case.

(c) (4%) Explain how you can use the out-of-order deduplication to maximize the read performance of stream S. Show all containers that will finally be stored.

(d) (6%) Consider a cloud storage service that follows a tiered pricing model. For storage pricing, the service charges \$0.30/GB/month for the first 1,000GB and \$0.28/GB/month for the next 49,000GB. We assume that both transfer pricing and request pricing are free of charge. Suppose that we use the cloud storage service starting on January 1, 2015. We generate 300GB of backup data per month. No deduplication is applied to our backup data. Suppose that we pay bills at the end of each month. How much money in total do we have to pay from January 1, 2015 to May 31, 2015? State your assumptions if needed.

5. **(12%) Zookeeper.**

The following pseudo-code lists the implementation of the distributed lock using Zookeeper (only the lock function is shown).

```
1. n = create(l + /lock-, EPHEMERAL|SEQUENTIAL)
2. C = getChildren(l, false)
3. if n is lowest znode in C, exit
4. p = znode in C ordered just before n
5. if exists(p, true) wait for watch event
6. goto 2
```

(a) (4%) What does "EPHEMERAL|SEQUENTIAL" mean?

(b) (4%) What happens if we assign p to be the lowest znode in C in Line 4 instead? Explain your answers.

(c) (4%) If we replace Lines 5-6 with the following code:

```
if exists(p, true)
  wait for watch event
else
  goto 2
```

In other words, the process directly acquires the lock after it is triggered by the watch event. However, it doesn't work. State a scenario where it doesn't work. Explain your answers.

Hope you have a fruitful winter break!

-End-