CSIT6000O Advanced Cloud Computing

##### Spring Semester 2022

**Midterm Examination**

**Date: Mar 25, 2022** **Time: 7:30 – 9:30 pm**

#### **Name**: \_Yuan Fangxu\_\_\_\_\_\_\_\_ **Student ID**: \_\_20799126\_\_ **Email**: \_\_fyuanad@connect.ust.hk\_\_\_\_\_\_\_

# Instructions

1. Please write your name, student ID and email address on this page.
2. Please answer all questions within the space provided on the examination paper. You may use back of the pages for your rough work.
3. Please be **concise**, and this is **NOT** an essay contest.
4. This paper consists of **4** questions and **14** pages.
5. Please read each question very carefully and answer the questions clearly to the point. Make sure your answers are neatly written, legible, and readable.
6. Show all the steps used in deriving your answer, wherever appropriate.

|  |  |  |
| --- | --- | --- |
| **Question** | **Points** | **Score** |
| 1 | 20 |  |
| 2 | 20 |  |
| 3 | 25 |  |
| 4 | 35 |  |
| **Total** | **100** |  |

**Question 1 (20 marks): Cloud concepts, characteristics, and service models**

Avalon Inc. is a startup company that has been asked by a city to monitor and record near misses and accidents between cars, bicyclists and pedestrians 24 hours per day, 7 days per week, 365 days per year. Avalon is planning to deploy video cameras and dedicated computing at each busy intersection in the city.

Avalon plans to capture video at a sustained frame grab rate of 30 frames per second, 24 hours per day. They will provide privacy to drivers, cyclists and pedestrians by denaturing the video at the intersection, before it is stored. Since the bandwidth from the intersection to the cloud is limited, Avalon plans to process the streams of data at the intersection to identify near misses and accidents. They will then send the data for just the “interesting” cases to the cloud for storage and potential further analysis.

1. **(6 marks)** We know that cloud computing offers various benefits and features to users. For each benefit listed below, indicate whether you believe it would be important for Avalon Inc. and explain why.

|  |  |  |
| --- | --- | --- |
| **Cloud Advantages** | **Beneficial or not to Avalon**  **(Yes/No)** | **Why?** |
| Resource Elasticity | Yes | Because Avalon is a startup company, obviously it does not have the power or money to build any datacenter or computing machine by itself, so the cloud computing is the best choice. |
| On-demand computing with no upfront infrastructure cost | Yes | Because the possibility of misses and accidents are different in different days and different time in one day, so the demands of resources are different in different time. |
| Reduced IT maintenance | Yes | Because it is a city-wide project, it is a huge effort to manage so many machines in one space. Besides, different places in a city could be far from each other, so Cloud is beneficial for Avalon |

1. **(4 marks)** Cloud computing offers various service models, including IaaS, PaaS, FaaS, and SaaS. What service model do you believe best-fit Avalon’s needs? Explain your answers.

IaaS is needed. Because Avalon need to storage “interesting” cases and use them for potential further analysis. So Avalon need a full control of OS, storage, applications and some networking components due to the secrets.

But I think FaaS and SaaS could be used together too. Avalon could use Faas to store the cases like S3 in AWS, and use SaaS like Lambda in AWS to analysis.

1. **(4 marks)** Avalon has grown rapidly and gone multi-national, soon to cover 50% of the world’s major cities. In this case, should they build their own private cloud? Explain your answer.

Yes.

1. It needs to protect the data of secretary from the public.
2. There are a lot of data, it will be treated by the network bandwidth and availability issues if it does not build their own private cloud.
3. The honesty is the most important for a big company, it needs to make sure that it has a greater control of the service and resilience.
4. **(6 marks)** Cloud experts often say that "in order to experience elastic cloud computing, there needs to be **many users** who each has **small computing needs** and has **unrelated and different workloads**."Explain why these three features are important to enable elastic cloud computing.

Here are the reasons:

1. Due the huge amount of need from many users. Everyone needs the computing resources at different time on different worloads. So it is possible to make use of the pieces of computing time and computing resources from the machines when people could not get his or her own machine at any time.
2. Computing needs are different, so we cannot provide only one or two kind of resources service, when we get different type of computing resources at datacenter, we could provide people with the resource they need.

**Question 2 (20 marks): Virtualization**

1. **(3 marks)** As a cloud service provider, which of the following describes a benefit of virtual machines? (*Note*: there may be more than one correct answer)
2. Shared hardware for multiple users
3. Dedicated hardware for each individual user
4. Virtual machines can be quickly provisioned in a few seconds
5. Virtual machines occupy less storage space than physical machines
6. Virtual machine allows users to flexibly customize the runtime environments
7. Virtual machines can perfectly replace physical machines: any user program that can run on a physical machine can also run in a VM

ACD

1. **(3 marks)** OSs use privileged instructions to manage hardware resources like page tables and I/O devices. When executed with a VM on a modern hypervisor providing full virtualization, what happens when a guest OS executes a privileged instruction of this sort? Explain your answer. Note that in full virtualization, the CPU runs in two modes, the user mode and the kernel mode.

If the guest OS is in user mode, guest could not run the instructions. If guest OS is in virtual kernel mode, the hypervisor will examine every instruction. The instructions will be translated into new instructions perform equivalent task in hardware.

1. **(4 marks)** In Intel x86 architecture, a CPU implements protection rings in hardware in order to enforce the separation of the kernel space from user space. There are four rings, Ring 0, 1, 2, and 3. In para-virtualization, (1) which ring does the guest OS runs in, and (2) which ring does the host OS kernel run in? In hardware-assisted virtualization, (3) how many rings are there, and (4) which ring does the host OS kernel or the hypervisor run in?
2. 3
3. 0
4. 4
5. Ring -1
6. **(2 marks)** Compare the difference between para-virtualization and full virtualization and explain why the former outperforms the latter in terms of performance.

Because the full virtualization do not has a guest OS modification but para does. Full type doesn’t change the OS, and para type made some minimal changes to the OS.

1. **(3 marks)** While para-virtualization generally outperforms full virtualization, the latter is still widely used in many systems. What is the advantage of full virtualization over paravirtualization? Explain your answer.
2. Keep the guest OS unmodified
3. VM portability
4. Prevent an unstable VMs from influencing system performance

For example, sometimes we need to make sure that the hardware is protected from the VM.

1. **(3 marks)** Compared to VMs, containers are more lightweight and efficient. Explain the reasons that make them so efficient.
2. Configure once, run anything
3. Containers are isolated
4. There is no binary translation and no emulation in contains but VM does.
5. **(2 marks)** Because containers are more efficient than VMs, someone suggest to replace VMs by containers in cloud environments. Explain two reasons to counterargue this suggestion.
6. Sometimes we need full OS capabilities.
7. A container can pollute other containers which share the host OS, but VMs have barrier between them.
8. VM supports heterogeneous environment

**Question 3 (25 marks): Distributed storage systems**

1. **(4 marks)** Suppose in AWS cloud, an application can have access to three storage options, Hadoop Distributed File System (HDFS), EC2 Elastic Block Service (EBS), and AWS Simple Storage Service (S3). EBS is the default storage option of EC2 instances that works just like a disk, while S3 is a global key-value store. Answer the following two questions with justifications.
2. **(2 marks)** Which of the three options would you expect to be the *easiest* for a traditional enterprise application that requires extensive disk I/O to be modified to use?

EBS, because they are work just like a disk, and it is used for disk I/O.

1. **(2 marks)** Which of the three storage options can provide the highest total data bandwidth, and what design feature of that storage option enables such a high total bandwidth?

HDFS, because it is the only distributed file system. There are many nodes. We just could get data from one of the datanodes

1. **(3 marks)** HDFS configures a large data block (chunk) in its design. What is the default block size? Why does it choose such a large block size?

128MB. Because it needs to reduce the number of blocks, at the same time the NameNodes just need to store less metadata of blocks

1. **(4 marks)** Justify if HDFS is suitable for the following two cases:
2. A large streaming of log data (**2 marks**)
3. A large image dataset like ImageNet consisting of tens of millions of images (**2 marks**)

A no

B yes

Because the data in image dataset are independent but data in log data are dependent

1. **(4 marks)** Assume that a big company has built a Hadoop cluster with 10,000 machines, and for each machine, the monthly failure rate is 10%. Suppose a large file consisting of 100 HDFS blocks is stored in the cluster. To achieve 99% storage reliability for that file, what is the *minimum number of block replicas* the cluster operator should configure? (*Note*: for simplicity, you can assume that no two blocks and their replicas are placed on the same machine.)

2000

1. **(4 marks)** Consider a Hadoop cluster that runs a NameNode and a secondary NameNode to provide high availability. What happens when the NameNode goes down during file read operation?

If there is no answer in the first NameNode, then it will search in the secondary NameNode, if there is still no answer, give up, if there is answer, then update it in the first NameNode. If there is an answer in the first NameNode, there is no need to search the secondary NameNode.

1. **(6 marks)** Imagine that your company uses a distributed data processing system based on Hadoop, deployed on Amazon EC2 instances with all data stored in an HDFS file system running on those same instances. Your colleague argues that, because HDFS maintains three replicas of each block, it is safe to use Spot instances instead of on-demand instances. Do you agree? Explain your answer.

I don’t agree. Because if we use spot instances, it is possible that we could find the date we need in Hadoop due to distribute data may be lost. However, we could get back-up data form the on-demand instance to avoid the situation.

**Question 4 (35 marks): MapReduce**

(a) **(11 marks, 1 mark each) Multiple choice**

(*Note*: only **one** answer is correct)

(1) The MapReduce programming model is *most* suitable for the following scenarios:

1. A large cluster of commodity machines
2. A small number of high-performance servers
3. A supercomputer
4. A single PC

A

(2) Which one of the following correctly describes the map function?

1. It converts a relational database into key-value pairs
2. It indexes the data to list all the words occurring in it
3. It processes data to create a list of key-value pairs
4. It tracks data across multiple tables and clusters in Hadoop

C

(3) Combiners increase the efficiency of a MapReduce program because:

1. They provide a mechanism for different mappers to communicate with each other, thereby reducing synchronization overhead.
2. They provide an optimization and reduce the total number of computations that are needed to execute an algorithm by a factor of *r*, where *r* is the number of reducer.
3. They aggregate intermediate map output locally on each individual machine and therefore reduce the amount of data that needs to be shuffled across the network to the reducers.
4. They aggregate intermediate map output in a small number of nearby (i.e., rack-local) machines and therefore reduce the amount of data that needs to be shuffled across the network to the reducers.

C

(4) You use the hadoop fs put command to write a 300 MB file using an HDFS block size of

64 MB. Just after this command has finished writing 200 MB of this file, what would another

user see when trying to access this file?

1. They would see the content of the file through the last completed block.
2. They would see the current state of the file, up to the last bit written by the command.
3. They would see Hadoop throw an concurrentFileAccessException when they try to access this file.
4. They would see no content until the whole file is written and closed.

C

(5) If you run the word count MapReduce program with *m* mappers and *r* reducers, how many output files will you get at the end of the job? And how many key-value pairs will there be in each file? Assume *k* is the number of unique words in the input files.

1. There will be *r* files, each with exactly *k*/*r* key-value pairs.
2. There will be *r* files, each with exactly *k*/*m* key value pairs.
3. There will be *m* files, each with exactly *k*/*m* key-value pairs.
4. There will be *m \* r* files, each with exactly *k*/(*m \* r*) key-value pairs.

A

(6) Does the MapReduce programming model provide a way for reducers to communicate with

each other?

1. Yes, all reducers can communicate with each other by passing information through the jobconf object.
2. Yes, reducers can communicate with each other by dispatching intermediate key value pairs that get shuffled to another reduce.
3. Yes, reducers running on the same machine can communicate with each other through shared memory, but not reducers on different machines.
4. No, each reducer runs independently and in isolation.

D

(7) Jason has a Hadoop cluster with 50 machines under default setup (replication factor 3, block size 128 MB). Each machine has 100 GB of HDFS disk space. The cluster is currently empty (no job, no data). Jason intends to upload 1 TB of plain text (in 5 files of approximately 200 GB each), followed by running Hadoop’s standard WordCount job. What is going to happen?

1. WordCount fails: too many input splits to process
2. WordCount runs successfully
3. The data upload fails at the last file: due to replication, all disks are full
4. The data upload fails at the first file: it is too large to fit onto a node

C

(8) When is the reduce method first called in a MapReduce job?

1. Reduce methods and map methods all start at the beginning of a job, in order to provide optimal performance for map-only or reduce-only jobs.
2. Reducers start copying intermediate key value pairs from each Mapper as soon as it has completed. The reduce method is called as soon as the intermediate key-value pairs start to arrive.
3. Reducers start copying intermediate key-value pairs from each Mapper as soon as it has completed. The reduce method is called only after all intermediate data has been copied and sorted.
4. Reducers start copying intermediate key-value pairs from each Mapper as soon as it has completed. The programmer can configure in the job what percentage of the intermediate data should arrive before the reduce method begins.

C

(9) Given a Mapper, Reducer, and Driver class packaged into a jar, which is the correct way of

submitting the job to the cluster?

1. jar MyJar.jar
2. jar MyJar.jar MyDriverClass inputdir outputdir
3. hadoop jar MyJar.jar MyDriverClass inputdir outputdir
4. hadoop jar class MyJar.jar MyDriverClass inputdir outputdir

C

(10) Which of the following application is not well suited for MapReduce?

1. Text mining on a large collections of unstructured documents.
2. Analysis of large amounts of Web logs (queries, clicks, etc.).
3. Online transaction processing (OLTP) for an e-commerce Website.
4. Graph mining on a large social network (e.g., Facebook friends network).

C

(11) How can a distributed filesystem such as HDFS provide opportunities for optimization of a MapReduce operation?

1. Data represented in a distributed filesystem is already sorted
2. Distributed filesystems must always be resident in memory, which is much faster than disk.
3. Data storage and processing can be co-located on the same node, so that most input data relevant to Map or Reduce will be present on local disks or cache.
4. A distributed filesystem makes random access faster because of the presence of a dedicated node serving file metadata.

C

(b) **(6 marks)** Consider the Hadoop pseudo code shown below. It computes, for each key, the log average across all values associated with that key. The formula for the log average is:

map(string t, int r):

emit(string t, int r)

reduce(string t, ints [r1, r2, r3, ...]):

float sum = 0.0;

int count = 0;

foreach int r in ints:

sum += log(r);

count++;

float logAvg = exp( 1/(float)count \* sum );

emit(string t, float logAvg);

Rewrite the *pseudo-code* above and **include a combiner** to speed up the job. Make sure that both versions of the code, with and without the combiner, give the same final results.

map(string t, int r):

**# Your code goes here**

emit(string t, pair(r,1))

combiner( string t, pairs[(s1,c1),(s2,c2)..]):

**# Your code goes here**

float sum = 0.0;

int count = 0;

for auto (s,c) in pairs[(s1,c1),(s2,c2)..]

{

sum += log(s);

count += c

}

emit(string t, pair(sum,count));

reduce( string t, pairs[(s1,c1),(s2,c2)..] ):

**# Your code goes here**

float sum = 0.0;

int count = 0;

for auto (s,c) in pairs[(s1,c1),(s2,c2)..]

{

sum += log(s);

count += c

}

float ans = exp(1/(float) count\*sum)

emit(string t, ans);

(c) **(6 marks)** Given a large dataset of text files, we want to compute the relative frequencies of co-occurrences defined as follows:

where *N*(*A, B*) tallies up the co-occurrences of word-*A* and word-*B* in the text, and *N*(*A*) is the number of word-*A* appeared in the text. The following MapReduce algorithm is based on the “pairs” approach. However, the solution has a problem and does not give the correct answer. (1) Find out and explain the problem (**2 marks**), and (2) fix it (**4 marks**).

class MAPPER

map(docid a, doc d):

foreach term w in doc d:

foreach term u in neighbors(w):

emit(pair(w, “”), 1)

emit(pair(w, u), 1)

class REDUCER

float marginal = 0

reduce(pair (w, u), counts[c1, c2, ...]):

int s = 0

foreach c in counts[c1, c2, ...]:

s = s + c

if u = “”

marginal = s

else

emit(pair (w, u), float s/marginal)

**# Your answer goes here**

(d) **(6 marks)** Given a customer-product graph where customers and products are nodes, and an edge is formed if a customer buys a product, design a MapReduce algorithm to compute the frequency distribution of the number of customers by the number of products they bought. The following gives an example input and the expected output.

*Input:*

(c1, [p1, p2]) // customer-1 bought product-1 and product-2

(c2, [p1, p2]) // customer-2 bought product-1 and product-2

(c3, [p1]) // customer-3 bought product-1

(c4, [p1, p3, p4]) // customer-4 bought product-1, product-3, and product-4

*Output: // (product\_count, customer\_count)*

(1, 1) // there is one customer (c3) who ordered one product

(2, 2) // there are two customers (c1 and c2) who ordered two products

(3, 1) // there is one customer (c4) who ordered three products

Your MapReduce pseudo-code:

map(string customer, products [p1, p2, ...]):

**# Your code goes here**

int count = 0;

for p in products [p1,p2,…]

count ++

emit(count,1)

reduce( int index, nums [num1, num2, ...] ):

**# Your code goes here**

int count = 0;

for num in nums[num1,num,2,…]

count += num

emit(index,count)

(e) **(6 marks)** Given the same input data as the previous question, design a MapReduce algorithm to count, for each pair of two customers, the number of products they bought both. *Please explain your algorithm.* (**3 marks out of 6**)

*Input:*

(c1, [p1, p2]) // c1 bought p1 and p2

(c2, [p1, p2]) // c2 bought p1 and p2

(c3, [p1]) // c3 bought p1

*Output:*

((c1, c2), 2) // both c1 and c2 purchase the same two products (p1 and p2)

((c1, c3), 1) // both c1 and c3 purchase one same product (p1)

((c2, c3), 1) // both c2 and c3 purchase one same product (p1)

*Note:* Duplicate pairs should be avoided, e.g., (c2, c1) duplicates with (c1, c2), and should not be emitted. Use a pseudo code to describe your MapReduce algorithm.