Full Name:
Andrew ID:

15-719: Advanced Cloud Computing
Midterm Exam Answers
March 8th, 2017

Total Time: 80 minutes

Instructions:

- Write your Andrew id on the top of every page in case they get separated.
- Make sure your exam is not missing any sheets. There should be 14 sheets.
- Write your answers in the spaces provided below each problem. If you make a mess, clearly indicate your final answer.
- If you have to make an assumption, write it down clearly.
- The exam has a maximum score of 75 points (plus 2 bonus points).
- This is a closed book exam. You are allowed one sheet of notes.
- No electronic devices are allowed (this means **no** laptops/computers/smart-phones)
- Manage your time wisely.

Question No	Max. Points	Earned Points
1	10	
2	12	
3	12	
4	8	
5	8	
6	4	
7	12	
8	9	
9	2 (bonus)	
Total	**75**	

1. Definitions and Use Cases (Points: 10)

Kayvon Inc. (KayvonInc.com) is a local startup that has been asked by the 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. Kayvon Inc. (KI) is planning to deploy video cameras and dedicated computing at each busy intersection in the city.

KI plans to capture video at a sustained frame grab rate of 20 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, KI 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.

i. 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 to Kayvon Inc and describe why. (**Points: 6**)

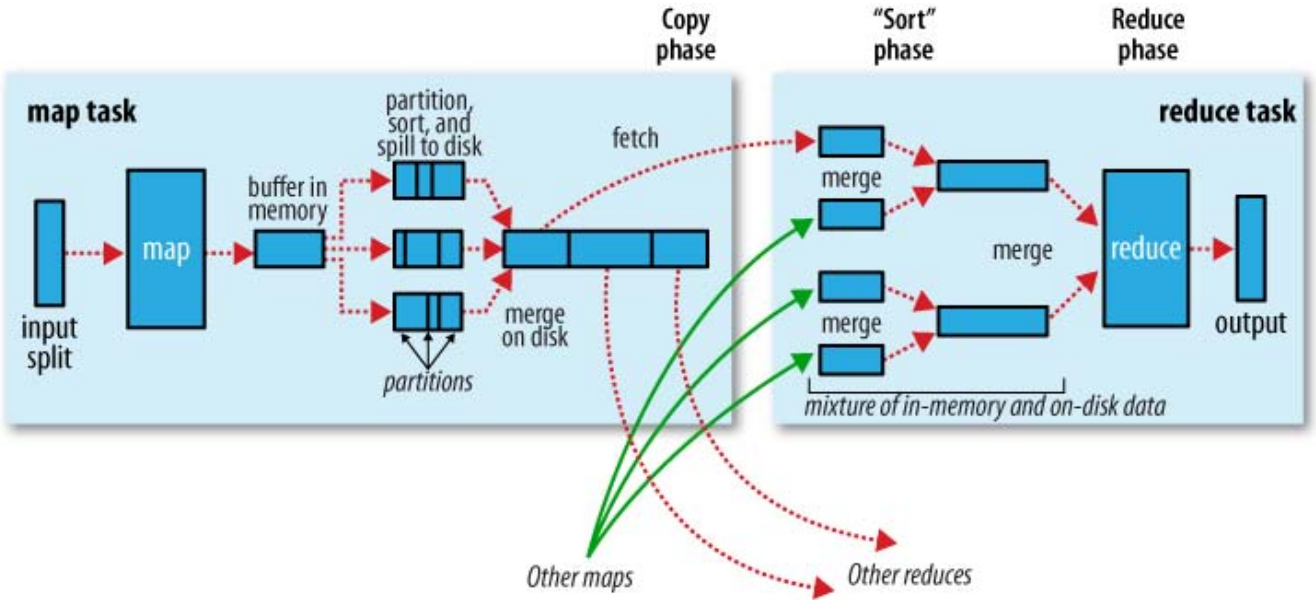
Cloud Advantages	Beneficial or not beneficial to Kayvon Inc.	Why?
VM Elasticity	No or Yes	No: Since KI is using the cloud for storage and batch processing (not latency sensitive), the cloud EC2 usage can be adjusted very slowly to the long term average Yes: Since KI is only uploading interesting cases for further analysis, it's cloud has variable compute needs and it can save money by shutting off VMs when idle.
No upfront cost	Yes	KI does not have to invest in building storage and processing clusters as well as backup. Ki can direct its initial investments to improve its technology rather than build infrastructure.
Reduced IT maintenance	Yes	As a startup, Ki can direct its initial investments to improve its technology rather than build infrastructure.

ii. KI has grown and gone international, soon to cover 75% of the world's big cities! Should they build their own private cloud? Explain your answer. (**Points: 4**)

- Yes! At this scale, KI has grown large enough that it can realize enough of the economies of scale that cloud providers enjoy, so it may be more cost-effective to have a private cloud that exactly matches its needs (and avoids the profit margin cloud providers seek).

2. Frameworks (Points: 12)

i. The figure below illustrates the significant data processing actions of map and reduce tasks. This question asks you to explain, in words, the steps for a portion of it: how each mapper processes and prepares intermediate data for use by reducers. Be sure to clarify where data is put (into a memory buffer vs. in a file) **(Points: 4)**



- Each mapper writes its output to a memory buffer (buffer size can be configured), when the buffer reaches a threshold, it spills to local disk.
- Before writing to disk, the data is divided into partitions depending on the number of reducers. Within each partition, the data is sorted in memory.
- For each spill to disk, a new spill file is created for each partition.
- Before the task is finished the spill files are merged and sorted.

ii. Explain why Spark is considered "in-memory" compared to Hadoop. **(Points: 4)**

- Spark allows the application program to cache frequently used RDDs in application memory so future computation (jobs) may reuse data from application memory, while Hadoop jobs always start by reading data from the file system and end by writing output to the file system.

iii. Spark keeps track of the lineage of each RDD. What is the lineage information used for? **(Points: 4)**

- When an action is invoked, Spark computes the needed RDDs according to their lineage;
- When a previous cached RDD partition is lost, Spark may recover the lost RDD partition from its parent according to its lineage.

3. Scheduling (Points: 12)

i. A MapReduce Job consists of many tasks that are distributed among TaskTrackers for execution. Sometimes, even when the machines and the tasks are identical, a few of the tasks will take much longer to complete than the others. But, a map or reduce stage cannot complete until its constituent tasks all complete. Describe a technique used by both Google's MapReduce and Hadoop MapReduce to mitigate this problem. Explain why it is safe and effective. **(Points: 4)**

- Speculative replication of tasks. Any map or reduce task can be executed multiple times, because each task is idempotent and saves its output in a file with a predetermined name. So, multiple instances of the same task can be executed and the results can be safely used from first one to finish. By only replicating tasks that run for longer than the norm, the scheduler can avoid too much waste of resources in addressing the straggler problem. This approach is effective, because it is unlikely that both the original and the replica tasks will be stragglers.

ii. Imagine that a user submits a job to the Mesos scheduler that manages their company's cluster. The job needs to run on one of a particular subset of machines that have particular software pre-loaded and configured on them. If all of those machines are currently fully allocated to other equal-priority jobs, what do you believe will happen with the new job? **(Points: 4)**

- The new job will receive offers of whatever machines are available, until it gets an offer it likes. Until one or more of those equal-priority jobs finishes, the offers will not include the needed machines. In addition, if the new job is unlucky, a different job that doesn't care which machine it gets might be offered one of the needed machines first, because there is no machine-specific arbitration in the Mesos meta-scheduler.

iii. It is common for a modern cluster scheduler to intermix jobs that have different priorities, such as "business critical" and "best effort". Naturally, when the scheduler considers allocating resources to jobs in its pending queue, higher priority jobs will "win" (get the resources they need) before lower-priority jobs. But, when there are no more high-priority jobs, resources will be allocated to lower-priority jobs. What should the scheduler do, though, if a high-priority job is submitted at a time when all resources are already allocated to lower-priority jobs? **(Points: 4)**

- Preempt running low-priority jobs/tasks, from lowest to highest priority, until enough resources are freed. Depending on the system, this might be done by killing jobs, saving jobs to later restart them, or migrating them.

4. **Compute encapsulation (Points: 8)**

   i. 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 hardware virtualization, what happens when a guest OS executes a privileged instruction of this sort? Explain your answer. **(Points: 4)**

   - The most common approach is called "trap-and-emulate". The hardware "traps" when the instruction is attempted, transferring control to the hypervisor, which then emulates the change that the instruction was attempting and returns control to the guest OS as though the instruction had been executed normally.

   ii. Greg believes paravirtualization generally outperforms full hardware-level virtualization. What is one advantage of full hardware virtualization over paravirtualization? Explain your answer. **(Points: 4)**

   - Ability to run unmodified OSs within guest VMs. With paravirtualization, the guest OSs must be modified to interact properly with the hypervisor. With full hardware virtualization, the guest OSs need not be aware that they are not running on real hardware.

5. **Storage (Points: 8)**

   During a course project in AWS or a similar cloud services, imagine running an application that could have access to a Hadoop Distributed File System (HDFS), an Elastic Block Service (EBS) and a Simple Storage Service (S3).

   i. Which of the three storage options would you expect to be the easiest for a traditional enterprise application to be modified to use, and why? **(Points: 4)**

   - EBS because it supports a simple disk abstraction, simple operating systems can mount it like a disk, install a file system on it, and use it just like they did real disks in the enterprise data centers. HDFS is neither a raw disk interface nor POSIX (distributed) file system; instead applications have to link in a library can call into it. S3 is effectively a key-value store for "large" values, and enterprise systems generally do not have key-value stores, and especially not stores that use http for access.

   ii. Which of the three storage options is the most likely to provide the highest total data bandwidth, and what design feature is most likely to provide this performance advantage? **(Points: 4)**

   - HDFS is designed for high total bandwidth; the key design feature providing high bandwidth is the way HDFS metadata servers give a client opening a flle the map of where the chunks of the file are stored on chunk servers so that clients can access the server holding each chunk directly without further interaction with the metadata server, at least until the file is closed.

6. Applications of Mobile and the Cloud (Points: 4)

i. Technology trends continue to increase the processing capabilities of mobile devices. The latest smartphones include up to 8 processing cores and often have fairly powerful GPUs as well. In spite of this, it may make sense to offload computation to the cloud or a cloudlet. Describe two (2) circumstances under which it might be beneficial to offload application functionality from such a powerful smartphone? **(Points: 4)**

- Answer: any two of several possible reasons: a) Battery limitations of device; b) Servers have been getting faster too, so it may still be faster to offload; c) There may be large databases in the cloud that are needed; d) Peak performance may be good, but thermal issues may limit how long it can be sustained in a mobile device

7. Elasticity (Points: 12)

i. At your internship job, an older programmer, Kamal, comments that cloud computing was invented because of special shopping days like Cyber Monday in the US, during which Amazon does over $3B in online sales, or Single's Day in China, during which Alibaba does over $14B in online sales.

a) Give one reason why this might be false (and explain). That is, why might Cyber Monday and Single's Day not be great examples for cloud computing? **(Points: 2)**

- Cloud computing has elasticity to accommodate bursty load when the bursts of each use of the cloud are small compared to the total and independent of each other. On Cyber Monday and Single's Day these retailers experience huge bursts, all on the same day. It is highly unlikely that the total of all online retail business on these days is small and independent of the rest of the work in the cloud.

b) Give one reason why this might be true, and explain? **(Points: 2)**

- The computing infrastructure built to handle these rare, massively busy, days will not be fully utilized during the rest of the year. This is a good reason for an online retailer to also offer to rent computers to outside companies.

ii. Kamal's job is to convert an in-house web application into an elastic cloud application. Kamal's design has a load balancer machine on the network path of each http request immediately before the web server machines he plans to dynamically rent. Alicia, a CMU student interviewing the day of Kamal's design review, comments that this load balancer machine might not be needed. Give at least one reason that Alicia is correct and briefly explain how this machine might be eliminated (that is, how load balancing could be accomplished without the machine). **(Points: 4)**

- The application may not need to change the number of servers quickly, so it might use DNS to distribute lists of servers to client machines, so that each client selects a possibly different server. This eliminates a load balancing machine, by doing it in the client.
- Load balancing might be simple enough to do in the network router device coming into the data center, and not need a dedicated machine.

iii. Alicia also mentions that if Kamal wants to have a simple cloud application, then his web application might have problems implementing its backend database. Give at least one reason that the backend database might be a difficult part of Kamal's elastic cloud application. **(Points: 4)**

- If the application is read-only with a full copy of all catalogue data in each server, but places an order in the common backend database, the order taking application may update the same inventory records, which could be serialized under a single lock. This database might therefore be limited to the number of updates a single thread can do, limiting the elastic scaling of this application.

8. Project 2 (Points: 9)

i. During the execution of a Spark program, which of the statements below best describes what the number of tasks in a stage depends on (select the single most important option below). Explain your answer. **(Points: 3)**

a. the number of records in the RDDs that that stage operates on;
b. the number of partitions of the RDDs that that stage operates on;
c. the number of CPU cores in your Spark cluster;
d. the memory size previsioned for each Spark executor.

- B, Spark schedules a task to operate on a partition the way MapReduce schedules a map to operate on an input file chunk.

ii. Kevin wrote a Spark driver program that operates on an RDD of key-value pairs. To make sure each RDD partition is not too large, Kevin tried using the *partitionBy* operator to repartition that RDD by the key, but found that some partitions always contain a huge number of records no matter how many partitions the RDD is instructed to have. What is one possible reason why such a problem occurs? Explain your answer. **(Points: 4)**

- The key distribution might be highly skewed and there might be a large number of records share the same key being partitioned into the same partition. Alternatively, Kevin might have provided his own hash function which did not distribute values very well, so a different hash function could help.

iii. Mike wrote the code below to compute the sum of the numbers stored in the array *data* using Spark. Assuming Mike made no mistakes in other parts of his program, which of the multiple choice answers listed below best describes what will happen if he runs this program on Spark (in client mode)? Select one option and explain. **(Points: 2)**

```
counter = 0
rdd = sc.parallelize(data)
def increment_counter(x):
    global counter
    counter += x
rdd.foreach(increment_counter)
print("computed sum: ", counter)
```

a. The sum is correctly computed;
b. The program throw exception because *variable* is read before initialized;
c. The program runs without exception but the sum is not computed correctly;
d. None of the above

- C, because the counter is a local and is not delivered back to the driver

9. Bonus (Points: up to 2)

i. Which topic did Prof. Ganger talk about when the projector refused to work? **(Points: 1)**

- Building a cloud or OpenNebula

ii. The 15-719 staff members need suggestions for a place to take a relaxing vacation during Spring Break. Where should they go and why? **(Points: 1)**

- Anywhere

iii. Which of our awesome TAs gave the most talks in class so far and on what topic? **(Points: 1)**

- Jinliang Wei
- Project2/Spark

iv. What type of food should the instructors have during the exam grading "party" ? **(Points: 1)**

- Something with chocolate, so they are happy ?
- … or anything else ;)