Problem Set 3

15-440/15-640 Distributed Systems Spring 2025

Assigned: Thursday, March 20, 2025

Due: Thursday, March 27, 2025 at 11.59 pm

Submission procedure:

- Create a .pdf of your answers and upload to Gradescope.
- Here are some tips to make your submission easy to read and to grade. Remember, the easier you make this, the less likely we are to make grading errors. Following these guidelines will help us to focus on the technical content of your answers rather than trying to understand what you have written.
 - Don't hand write your answers. Use Latex or Google Docs or some similar input mechanism. If you use Latex, a template can be found on the course web page.
 - Put the answer to each question on a separate page.
 - Carefully tag your pdf pages to each question on gradescope. You can use the SHIFT key to select multiple pages and associate them with a single question.
- Assume SI notation
 - \circ 1 KB = 10³ bytes, 1 MB = 10⁶ bytes, 1 GB = 10⁹ bytes
 - \circ 1 Kbps = 10³ bits per second (bps), 1 Mbps = 10⁶ bps, 1 Gbps = 10⁹ bps
 - but a byte is still 8 bits (not 10 bits) :-)
- Remember that you have a limit of 2 grace days totaled over all four problem sets. You can use at most one of those grace days per problem set. Although Gradescope does not track grace days, we will. Exceeding your grace days will result in a zero grade for that problem set.

Question 1 (22 points)

Ruiqi's online board game has gained such popularity that her original server frequently crashes due to traffic overload. Michael and Zara proposed redesigning it for scalability in the cloud as a multistage web application that delivers dynamic content. The application has three stages:

- 1. Frontend Server: This processes incoming client requests and renders dynamic content through lightweight **independent** operations.
- Backend Server: This executes core game logic. Unfortunately, this stage is only partly scale-out compatible. 50% of the computation cycles must run on a single node (parallelization on multiple cores is fine, though). The remaining 50% of the computation cycles as well as all memory and storage needs can be fully distributed across multiple nodes..
- 3. Database: This manages sharing-intensive critical data, and **must run on a single node** to ensure data integrity.

The resources used (CPU, memory, disk) for each stage generally scales with the number of users, subject to a minimum for each instance as summarized below. So a frontend server instance with 100 users will use 20 GB of RAM and Disk, while one with 25 users will use 5 GB RAM and 10 GB disk, while one with just 5 users will use 4 GB RAM and 10 GB disk.

	Resources per concurrent user			Minimum resources per instance of stage	
Stage	CPU Ops (cycles) per second	RAM (GB)	Disk (GB)	RAM (GB)	Disk (GB)
Frontend Server	0.5 Gops	0.2 GB	0.2 GB	4 GB	10 GB
Backend Server	0.5 Gops	2 GB	1 GB	32 GB	120 GB
Database	50 Mops	0.5 GB	2 GB	16 GB	300 GB

The elastic cloud service provides several different machine types with different resources and costs:

Instance Type	CPU (GHz, Cores)	RAM (GB)	Disk (GB)	Hourly Cost (\$)
Small	2.5 GHz, 2 cores	8	100	0.08
Medium	3.2 GHz, 4 cores	32	250	0.25
Large	3.8 GHz, 8 cores	64	500	0.60
X-Large	5.0 Ghz, 20 cores	256	2000	3.25

- A. Assuming a constant load of 64 concurrent users playing the game, what is the most cost-effective configuration of server types for the three stages of this application? Consider each stage separately. Show your work.
- B. As loyal clients of the cloud service provider, Michael and Zara were presented with the option to purchase the server instances at the following prices. Should they buy or rent if they anticipate a sustained workload of 64 concurrent users over three years? How much more / less does it cost to rent than buy?

Instance Type	Hourly Cost (\$)	Buy Cost (\$)
Small	0.08	2000
Medium	0.25	5500
Large	0.60	10000
X-Large	3.25	55000

- C. The game grows in popularity. During the day (12 hours), the load is a sustained 256 concurrent users. At night the load drops to 160 concurrent users. With this level of load, what is the cost difference between renting and buying over a 3 year period? (Assume the same pricing as in B above)
- D. What is the maximum number of concurrent users that can be supported? Which is the bottleneck resource and processing stage?

Question 2 (20 points)

CloudPets is a cloud-based virtual pet platform where users can adopt, train, and interact with AI-powered virtual creatures that persist across devices. Each pet is housed in a distributed ecosystem, ensuring high availability and scalability. Users can access their CloudPets from any device.

- A. CloudPets wants to implement some nifty features through virtualization. For each of the following scenarios, describe which encapsulation method you would use: VM (must specify Type 1 or Type 2), containers, or processes. Justify your answer in 1-2 sentences.
 - a. Mahika is developing a multiplayer feature where users' pets can interact and play games with other pets in the virtual world. When a user enters a multiplayer room, Mahika's program launches a dedicated instance of the server that can only be accessed by invited users. These interactions / games typically last between a few seconds and one minute. After all users leave the room, the server will shut down.
 - b. The CloudPets CEO wants to develop a customizable "personality" module that users can run on their personal machines. The module takes parameters from the user and spawns a single-threaded AI worker that finetunes their pet's personality. CloudPets wants to ensure that the local CPU usage is as efficient as possible.
 - c. Roy, a developer of the system, wants to test out some new features without affecting the production service. To do his testing, he runs copies of the entire server stack on some local machines (including his Linux desktop machine and his MacOS laptop).
- B. CloudPets has a service-level agreement (SLA) with Awazon Cloud (a cloud service provider) that 90% of requests will have a response time of less than 200 ms. Awazon Cloud uses VMs to scale to different request rates from CloudPets. Each VM can service 500 requests per second, and takes 30 seconds to boot.

Initially, CloudPet runs on just a single VM, which is fast enough to handle the arriving requests. At time T_x , CloudPets goes viral on social media, and the number of requests jumps immediately to 1800 per second. Awazon Cloud launches more VMs to scale to this demand.

- a. How many additional VMs does Awazon Cloud need to launch?
- b. Is the SLA violated at any time? Explain with calculations.
- C. The Awazon Cloud engineers notice that once a VM is booted, it is terminated within a few seconds, and then a new one is booted again. Thus, the total VM time is high, yet requests must wait in long queues with slow response times. No errors are reported, and their cloud has sufficient resources. Why is this problem occurring? What is a possible fix for this?

Question 3 (20 points)

You have built a large scale-out web application that incorporates caching strategies. To evaluate how well the caching system works, you need to analyze how many times the system fetches data from various external sites. Each of the hundreds of VMs in your system produces a simple log file, listing each access to an external site in chronological order, one per line. You need to write analysis code that computes the frequency (count) of accesses per site, and generates output files based on the frequency. You will use MapReduce to perform this analysis.

Example input data (from all of the log files)

google.com amazon.com apple.com google.com adobe.com google.com amazon.com

Expected Output:

N files, where each file contains a set of unique site names (one per line), where file "1" contains all the sites that were accessed once, file "2" all sites accessed twice, ...

The MapReduce library:

For this problem, assume a MapReduce library that requires you to write a map function and a reduce function for a map-reduce job. The library runs multiple mapper processes on the input files, calling your map function on each data item (line) in the input files. The map function returns a key-value pair. All of these are grouped and sorted by key and then fed to the reducers.

The library runs multiple reducer processes over which the key space is partitioned. Each reducer calls your reduce function for each unique key in its portion of the key space. The arguments to the reduce function are the key and a list of all values that were produced by the mappers with that key. The reduce function should return a tuple of a file name and a list of lines to put in the file. If more than one reducer process tries to create a file with the same name, the underscore character ("_") and the reducer id are appended to the name and multiple files are created.

Here is an example of how you use this library to write "identity" map and reduce functions:

def map(input):

key = input value = input return (key, value)

- A. Using the MapReduce system described above, write pseudo code to perform this analysis and produce the desired output. Note that this will require two Map-Reduce jobs to be run one after the other. Write out the pseudo code for the four functions (Map and Reduce functions for each of the MR jobs) needed. Ensure it looks like the example functions above.
- B. Use the Example Input Data above to write the output key-value pairs for each map and reduce functions you defined above, in the order that they should be run.
- C. For the first MapReduce job, assume the mapper function takes 2ms per call, and that the reduce function takes 3ms per value. If you have 15 mappers and 10 reducers, what are the best and worst case scenarios and resulting execution times for a dataset with 30,000 log entries? (Consider only the first MapReduce job).
- D. Consider the situation in part C, but now assume that the 15 mappers each have a 1% chance of being executed on a slow machine that runs 10 times slower than a normal one. Assuming the best case scenario from part C, what would be the *expected time* it takes to finish this job?

Question 4 (18 points)

For each of the following, determine if the situation / code is fail-fast or not. Explain why/why not, and how to make it fail fast if it is not.

- A. A bus tracker application developed by a local Pittsburgh bus company is supposed to show its users where the buses are in real time. When a bus cannot be tracked, the application displays the last known location of the bus.
- B. Consider the following piece of Java code.

```
try {
    foo(); // can throw exceptions
} catch (Exception e) {
    // do nothing.
}
```

C. Consider the following piece of C code.

```
char s[] = { `h', 'e', 'l', 'l', 'o' };
int fd = open("foo", "w");
write(fd, s, sizeof(s));
```

- D. To purchase an item in an online store, the customers enter their payment details and click the "Pay" button. Each time the customer clicks "Pay," the system checks with the server to ensure the transaction has not yet been completed before processing the payment. If the transaction has already been completed, an error message will appear.
- E. When a printer's ink cartridge is empty, the printer still shows a "printing" status on the screen and moves the paper through the printer without printing anything.
- F. In a restaurant, customers place their orders by typing up their orders on a tablet. The system then sends the order to the kitchen. If an item listed cannot be properly interpreted (due to a typo or formatting issue), it automatically substitutes the item with the most ordered item on the menu.

Question 5 (20 points)

Carnegie Computer Service is expanding its Andrew File System infrastructure to meet the growing demand for distributed system storage. Two server companies are competing in the bidding:

- Pear, Inc. has a monolithic server design with all components permanently soldered together. If any component fails, the entire server has to be repaired or replaced. The system has a MTBF of 300 hours and an MTTR of 4 hours.
- SoftWork System's modular server is composed of four separate parts (modules) and has failure independence between the modules. When a system component fails, only the failed module has to be repaired. The MTBF and MTTR statistics of the four modules are as follows:

	MTBF (hours)	MTTR (hours)
Storage Module	1000	4
Networking Module	2000	2
Metrics Monitoring Module	3000	1
Processing Module	6000	4

Note that we define the term MTBF as strictly the time that a system is operational, not including the time for repairs. That is, the MTBF of a component is measured from the start of its operation to the beginning of the next failure.

- A. Suppose availability is defined as the percentage of time that the server is actually capable of work. What is the availability of Pear's monolithic server? Show your calculations.
- B. Under the same conditions, what is the availability of SoftWork's modular server? Show your calculations.
- C. Suppose Carnegie Computer Service needs to deploy 10 distributed storage servers to serve all of the users across its many campuses. Because each server stores different files with no replication, availability here is defined as <u>the fraction of time / probability of all of the servers</u> <u>being up</u>. I.e., the whole system is down if even one of the servers is down.
 - a. What is the availability of this storage service if using Pear's servers? Show your calculations.
 - b. What is the availability of this storage service if using SoftWork's servers? Show your calculations.

- D. Carnegie Computer Service also wants to deploy an email service. This service will use replication across multiple servers to increase availability. The service is available as long as <u>at least one server is operational</u>. Carnegie Computer Service wants to provide *four nines* of availability, i.e., ensure that the overall system availability is at least 0.9999.
 - a. What is the minimum number of Pear servers needed to achieve this availability? Show your calculations.
 - b. What is the minimum number of SoftWorks servers needed to achieve this availability? Show your calculations.