Problem Set 3

15-440/15-640 Distributed Systems Spring 2024

Assigned: Thursday, March 21, 2024

Due: Thursday, April 4, 2024 at 11.59 pm

Submission procedure:

- Create a **.pdf** of your answers and upload to `Gradescope`. All enrolled students should have Gradescope accounts. Ask **well in advance of the deadline** to be added to Gradescope if you don't have access; it is unacceptable to ask to be added immediately before the deadline.
- 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
    - 1 KB = 10^3 bytes, 1 MB = 10^6 bytes, 1 GB = 10^9 bytes
    - 1 Kbps = 10^3 bits per second (bps), 1 Mbps = 10^6 bps, 1 Gbps = 10^9 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 (12 points)

For each of the following scenarios, describe which encapsulation method you would use: VM, container, or UNIX processes. Justify your answer in one or two sentences.

1. Jason is working on a data miner which does a deep dive to collect various data. He first starts working on his personal 32-core computer. He spawns multiple single-threaded data miners and wants to optimize the entire data mining system. Note that only Jason needs access to this system and he is not worried about this system crashing or failing.

2. Jason now wants to switch his data miner to a multi-tenant cloud-based environment. Each node (all running on the same OS) has assigned workers that perform the deep dive to collect various data. If a node failure is recognized, a new node is spawned. While Jason is concerned about isolation from competitors, he wants to also ensure that the restart time when spawning a new node is minimal.

3. Jason wants to explore other options as well to find the best solution for his data miner. Instead of restarting whenever a node fails, Jason now intends to modify his system to implement checkpointing to recover the full system state in case of failure. His idea is to resume all nodes from the recovered state of the latest checkpoint (wherein failed nodes will resume running from the checkpoint on new nodes).

Question 2 (32 points)

SlideChat is a hot new app across universities where students can post blogs and browse anonymously in university-specific forums. However, amid its meteoric rise in popularity, developers of SlideChat are being notified of major latency issues in the app and disgruntled users are leaving by the hundreds. SlideChat currently relies on a fleet of frontend nodes distributed across the universities that the platform is available to. Students from University A cannot access content from University B, and vice versa. SlideChat's backend consists of a MySQL database, which acts as a master store for all content.

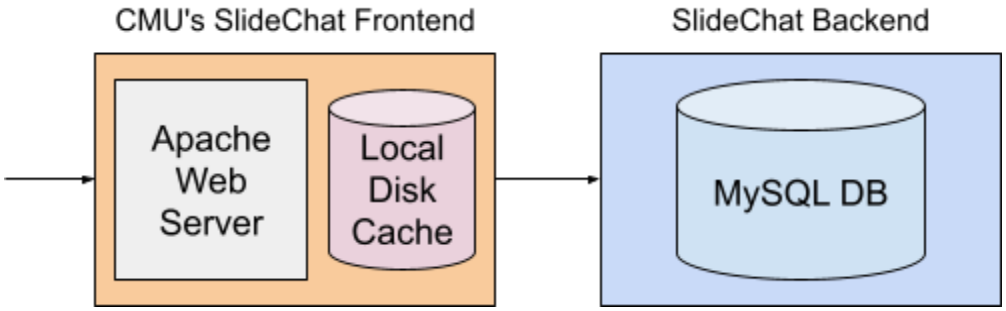

Image of CMU's specific frontend server connected to SlideChat's backend using a 2-tier system

A. Assume that posts are static content and each frontend delivers the same feed to all students at a specific university.

(i) What observations/symptoms would make you think the Frontend is the bottleneck?

(ii) What observations/symptoms would make you think the Backend is the bottleneck?

B. Explain under what circumstances and how each of the following scaling strategies can help in alleviating system bottlenecks.

a. Frontend Scale-up
b. Frontend Scale-out
c. Backend Scale-up
d. Backend Scale-out

C. SlideChat v2.0 plans to increase user engagement by adding comments/upvotes on blog posts and personalizing each user's feed based on a highly-sophisticated personalization algorithm. Since blog posts are now dynamic, they introduce a scaled-out app server level, expanding to a 3-tier system as shown below.

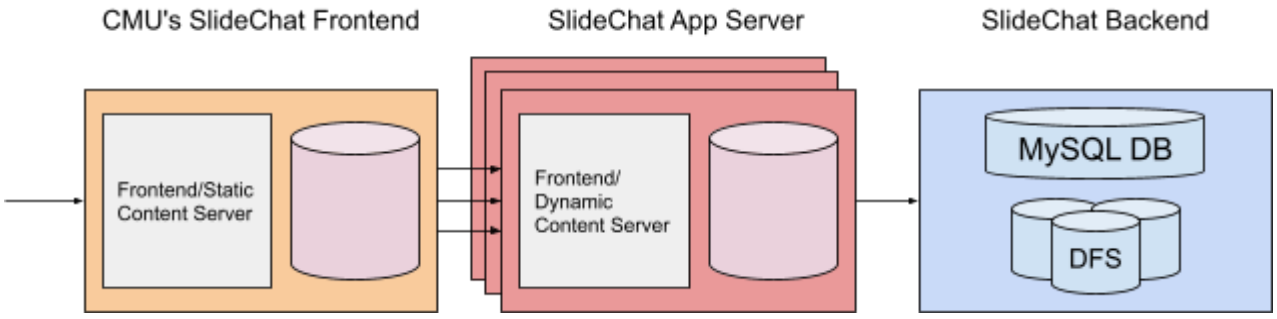

Image of CMU's frontend server connected to rest of 3-tier system

Note that they have reconfigured the backend/storage tier using a distributed file system for content along with a MySQL database used only for login information. As a consequence, the backend is not the bottleneck. The app is still suffering from latency issues with SlideChat v2.0 users reporting that as they scroll to the bottom of their feed, their content takes longer and longer to load. Identify the most likely tier that is acting as a bottleneck in this scenario and explain why.

D. For your chosen tier (in part C), choose one of the following strategies that might help alleviate said bottleneck. Again, more than one option may be correct, but choose only one to explain.

a. Increasing network bandwidth
b. Increasing memory (RAM)
c. Increasing CPU speed
d. Increasing server nodes

E. Suppose the service uses a fixed queue length threshold for determining when to scale out at tier 2. However, you notice that this results in the number of tier-2 servers fluctuating very quickly. Why is this a problem and how could it be fixed? Explain your proposed solution in detail.

Question 3 (16 points)

A. You stored some data on a disk and monitored its performance across some time duration. Here's what you observed about the disk:

Time Stamp	Observation
0	Start the disk;
5	Failure; start to repair;
6	Finished repairing; continue;
17	Failure; start to repair;
20	Finished repairing; continue;
27	Failure; start to repair;
39	Finished repairing; continue;
44	Failure; start to repair;
46	Finished repairing; continue;
50	Failure.

What is the MTBF and the MTTR of this disk?

B. Availability between time T_start and T_end is defined by T_available / T_total where T_available is the time duration where the disk is running (i.e. no failure and not under repair) and T_total is (T_end-T_start).

Calculate the availability of the disk from the table above between each of the following time periods:

- (T_start=0, T_end=50)
- (T_start=22, T_end=47)

C. Now suppose the data you are trying to store is large and thus you decide to split the data (not necessarily evenly) and store it in two different disks A and B. A failure happens for your storage system as long as any one of the disks fails, and you must fix the disk that fails

immediately (during which the other disk is turned off). Calculate the MTBF and MTTR of your system given the following stats:

a. MTBF_A = 30, MTBF_B = 30, and both disks have a deterministic repair time of 5
b. MTBF_A = 30, MTBF_B = 45, both disks have a deterministic repair time of 5

Recall from the lecture that the time before failure is a random variable. MTBF is the mean of the distribution of this random variable. You can assume that this random variable follows an exponential distribution with mean equal to the MTBF.

Hint: The following two properties of exponential distributions can be useful here:

1. The minimum of two independent exponential random variables with means $\mu 1$ and $\mu 2$ has mean $1/(1/\mu 1 + 1/\mu 2)$.
2. Memorylessness property: The exponential distribution is memoryless, that is future probabilities do not depend on any past information. Mathematically, if X is exponentially distributed, it says that P(X > x + k|X > x) = P(X > k).

Question 4 (16 points)

In each of the following cases, state with brief explanation whether the scenario exhibits fail-fast or Byzantine behavior. If it is Byzantine, suggest a fix that could make it fail-fast.

A. A blockchain network where (unverified) miners communicate with each other through TCP on a public network and work together to agree on the next block to be added to the chain.

B. A payment app that helps you pay for your meals at a campus food court. If you click the payment button multiple times, it processes the payment sequentially. On the first attempt, the payment will successfully go through. Any subsequent request will poll the server to check whether the payment has been processed already, refusing the request if previously completed or if an error occurs.

C. A build-your-own frozen yogurt shop uses a smart scale to charge its customers the appropriate amount based on the weight of the frozen yogurt with the appropriate toppings placed on it. When the weight sensor fails, the device uses the latest weight it successfully received from its weight sensor to compute the price.

D. A digital platform tracks scores during live football games in the CMU Fitness Center. The platform communicates with a server physically located at the game to retrieve live data every 30 seconds. When there is a network failure, the platform will display an error stating that it was unable to retrieve the latest scores.

Question 5 (24 points)

A banking system uses write-ahead logging (WAL) based on a no-undo/redo algorithm to implement local atomic transactions. The account database is implemented as a key-value store on disk. The system supports these operations:

- get(K) obtains the value of key K
- put(K, V) sets the value of key K to V

For any transaction Tx, there will be three types of log entries:

- <Tx, start> is written to the log when Tx has just started
- <Tx, K, V> is written to the log when transaction Tx sets the value of key K to value V
- <Tx, commit> is written to the log when transaction Tx is committed, after which all unflushed trailing part of the log is flushed

(a) The system starts running at Time 0. From Time 0 to Time 15, three transactions T1, T2 and T3 execute interleaved as shown below:

Time	**T1**	**T2**	**T3**
0	begin (T1)		
1	X = get(A)		
2	X = X + 150		
3	put(A, X)		
4	X = get(B)		
5	X = X − 150		
6	put(B, X)		
7	end(T1)		
8		begin(T2)	
9		Y = get(D)	begin(T3)
10		Y = Y − 50	W = get(A)
11		put(D, Y)	Z = get(B)
12		Y = get(C)	put(A, Z)
13		Y = Y − 200	put(B, W)
14		put(C, Y)	
15		end(T2)	

Suppose accounts A, B, C, and D have initial balances of 100, 200, 300, and 400 respectively. Show the contents of the write-ahead log after Time 7.

(b) Suppose a log truncation occurs after Time 7. The system continues to run until the server crashes after Time 15. (i) What would the log contents be before crash recovery begins? (ii) Upon crash recovery, which transaction(s) should be redone? Explain your answer in one sentence.

(c) From (b), assume that the crash recovery completes successfully. What are the final values of A, B, C, and D after the recovery? (i.e. If another transaction performs get(A), get(B), get(C), get(D), what are the return values respectively?)