Problem Set 4

15-440/15-640 Distributed Systems Spring 2024

Assigned: Monday, April 15, 2024

Due: Thursday, April 25, 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 of the ACID properties is violated. Justify your answer and propose a fix, in one or two sentences an.

A. Farnam is trying to finish 15-440 Project 4, but in a haste, he has forgotten to unlock images after aborted transactions.

B. Erin owes Max $15,440 because she lost a bet. She promised Max that she would transfer the money to his bank account. Erin uses the Bank of David and Max uses the Citizens Bank. Erin clicks “send money” on her computer, and she sees money leave her account. A message pops up announcing that the transfer is complete. Unbeknownst to her, the Citizens Bank server has crashed, and was unable to connect to the network. The next day, Max questions Erin about why he has not received his money yet, as the transfer has not been reflected in his account.

C. Monica pays $12 for her Rev Noodles. However, immediately after she finishes tapping her card and sees “Payment complete”, the payment system crashes. Upon rebooting, Monica is forced to pay again.

D. The online news section of The Tartan allows users to add comments to an article offline after they download the article. Their comments will later be pushed to the database as a plaintext string once the users go back online. John and Hanson downloaded the same article and added one comment each when they were offline. Then both of them connected back to WiFi. After this, some viewers see John’s comment only and some viewers only see Hanson’s comment only.

Question 2 (27 points)

It's the end of the semester and the 15440 TAs need to grade the final exam. Each TA has been assigned a group of scanned, ungraded finals on a distributed grading software called GradeScore. Each TA downloads the exams assigned to them before grading. GradeScore has a built-in communication system implemented with 2PC to agree on any grading changes. For example, suppose one of the TAs wants to propose a change to a rubric item. It is required that every other TA must agree with the change and that no TA has graded that problem yet. A TA can either vote YES or NO if they think the change is fair or not. If a TA has already started or finished grading a problem, then by default, they will vote NO to the proposal.

A. When changes to rubric items for one or more questions is proposed by a TA, what resources are necessary and sufficient to be locked by GradeScore for correct implementation of 2PC? Be specific (explain the resource to be locked, where it's located, and when to lock) and explain why.

B. GradeScore engineers are thinking about making a change to the traditional 2PC protocol. State yes or no to each of the scenarios that ***could*** happen for the given setup. Explain your answer briefly. If the answer is yes, give an example of how that scenario could occur. Unless specified otherwise, assume that all other aspects of the system follow the traditional 2PC.

Proposed change: The coordinator node stops resending decisions to nodes after 30 seconds of not receiving an ACK from them.

a. Some TAs end up grading the same problem with different rubrics

b. A rubric item has changed for a problem that's already been graded

c. A problem becomes permanently locked so a TA can never grade it

d. Once the proposing TA commits the decision, some nodes may not implement the decision

C. After realizing the potential negative effects of their proposed change to the 2PC protocol, GradeScore engineers abandon that change and instead are thinking about making a different change to the traditional 2PC protocol. State yes or no to each of the scenarios that ***could*** happen for the given setup. Explain your answer briefly. If the answer is yes, give an example of how that scenario could occur. Unless specified otherwise, assume that all other aspects of the system follow the traditional 2PC.

Proposed change: Nodes release resource locks after 60 seconds of not hearing a decision from the coordinator.

a. Some TAs end up grading the same problem with different rubrics

b. A rubric item has changed for a problem that's already been graded

c. A problem becomes permanently locked so a TA can never grade it

d. Once the proposing TA commits the decision, some nodes may not implement the decision

Question 3 (28 points)

This question uses the same Paxos notation as in class. "I➔J" denotes a successfully delivered message from server I to server J. There are 4 types of messages:

- Prepare(n), where n is a unique round number
- Promise(n, already_accepted(n_k, a_k)), where n_k and a_k are the last sequence number and value, respectively accepted by Acceptor k and stored at its stable storage. If Acceptor k has not accepted anything, the value is null.
- PleaseAccept(n, v), where v = a_k of highest n_k seen among the promise responses, or any value if no promise response contained a past accepted proposal
- Accept_OK()

Each of the sub-questions below shows a sequence of messages from Paxos running on 3 servers (S1, S2, S3). In each case, state whether the implementation of Paxos appears buggy or correct. If buggy, identify the first step in the sequence that is incorrect and briefly explain your reasoning. Give an example of what could go wrong due to this bug.

A.
1. S1➔S1: Prepare(101)
2. S1➔S1: Promise(101, null)
3. S1➔S2: Prepare(101)
4. S2➔S3: Prepare(102)
5. S1➔S3: Prepare(101)
6. S3➔S2: Promise(102, null)
7. S2➔S1: Prepare(102)
8. S3➔S1: Promise(101, null)
9. S1➔S2: Promise(102, null)

B.
1. S1➔S1: Prepare(101)
2. S1➔S1: Promise(101, null)
3. S1➔S2: Prepare(101)
4. S2➔S1: Promise(101, null)
5. S3➔S3: Prepare(103)
6. S3➔S3: Promise(103, null)
7. S3➔S2: PleaseAccept(103, "v3")
8. S2➔S3: Accept_OK()
9. S1➔S2: PleaseAccept(101, "v1")
10. S2➔S1: Accept_OK()

C.

1. S1➔S1: Prepare(101)
2. S1➔S1: Promise(101, null)
3. S1➔S2: Prepare(101)
4. S2➔S1: Promise(101, null)
5. S3➔S3: Prepare(101)
6. S3➔S3: Promise(101, null)
7. S3➔S2: Prepare(101)
8. S2➔S3: Promise(101, null)
9. S1➔S1: PleaseAccept(101, “v1”)
10. S1➔S1: Accept_OK()
11. S1➔S2: PleaseAccept(101, “v1”)
12. S2➔S1: Accept_OK()
13. S3➔S1: PleaseAccept(101, “v3”)

D.

1. S1➔S1: Prepare(101)
2. S1➔S1: Promise(101, null)
3. S1➔S2: Prepare(101)
4. S2➔S1: Promise(101, null)
5. S1➔S1: PleaseAccept(101, “v1”)
6. S1➔S1: Accept_OK()
7. S1➔S2: PleaseAccept(101, “v1”)
8. S2➔S1: Accept_OK()
9. S3➔S3: Prepare(103)
10. S3➔S3: Promise(103, null)
11. S3➔S2: Prepare(103)
12. S2➔S3: Promise(103, null)
13. S3➔S3: PleaseAccept(103, “v3”)
14. S3➔S3: Accept_OK()
15. S3➔S2: PleaseAccept(103, “v3”)
16. S2➔S3: Accept_OK()

Question 4 (18 points)

Answer the below questions about GFS briefly in a few sentences.

A. [Architecture]
GFS was designed with a master-worker architecture. The master node holds metadata, while multiple chunkservers hold data. When a client makes a read/write request to the master node, the master node replies to the client with the chunkserver name(s) to contact. The data transfer happens directly between the client and a chunkserver. Please explain advantages of this architecture in enabling low latency and a large number of concurrent requests.

Hint 1: Consider the difference in hardware that is used for the master node and chunkservers during a read/write request.

Hint 2: Think about the role of multiple chunkservers.

B. [Use Case]
Explain why GFS is NOT a good fit for a large number of small files (smaller than the default chunk size).

C. [Replication]
Why do you think GFS stores the first two replicas of a chunk on the same rack?
Why do you think GFS stores the third replica on a different rack?

D. [Chunk Migration]
Recall that GFS can enable the master server to perform chunk migrations between chunk servers. Provide one benefit of using this feature and one disadvantage.

E. [Pros/cons of GFS]

Your supervisor is interested in launching a new cloud storage service named CloudVault. CloudVault has several crucial requirements:

- It must support concurrent reading and writing by multiple users.
- Users should not have to worry about data loss after uploading their files to CloudVault.
- CloudVault needs to be able to support a vast number of users.
- CloudVault must ensure high availability, allowing users to access it around the clock without concerns about CloudVault undergoing any downtime.

Provide one argument in favor of using GFS and one argument against using GFS.

F. [GFS Append]

GFS supports the APPEND operation, which simplifies concurrency management through atomic execution without synchronization. Discuss which RPC semantics (at least once, at most once, exactly once) are applied to the APPEND operation and provide your reasoning.

Question 5 (15 points)

For each of the following use cases, state and explain (1-2 sentences) which is a better fit: MapReduce, MPI, or neither.

A. Mahika needs to run a large-scale quantum simulation to validate a new theory she has come up with. The simulation involves performing a large number of matrix vector multiplication on quantum state vectors and Hermitian matrices of large sizes.

B. CMU student services department has 1 petabyte of student records and wants to sort them by student ID.

C. Aditya wants to train a deep neural network to classify handwritten symbols using TensorFlow and a very large dataset. The backpropagation step required while training is an iterative process that updates model parameters in each step.

D. You want to find the number of connected components in a graph created from the social network Facesmash, where each node is a person and an edge exists between two nodes if the people are “friends”.

E. You have a distributed file system that needs to scale to petabytes of data. You want to process all the files periodically to check for data corruptions and report them. The reported corruptions need to be grouped by the type of hard disk they occurred on.