28.04.2025 Week 10 exercises: Key-value stores and CAP theorem

Exercise 1:

What is the CAP theorem?
It states that in a distributed data store, it is impossible to provide more than two out of the three following properties: consistency, availability and partition tolerance.

Exercise 2:

What is the definition of consistency in the CAP theorem?
Reads return either the value of the most recent write or an error.

Exercise 3:

Describe the simplest system you can think of that provides consistency and partition tolerance, in a distributed setting.
Hint: It is a very dumb system.
All reads return errors. There is no availability.

Exercise 4:

Describe the simplest system you can think of that provides availability and partition tolerance, in a distributed setting.
Each server maintains its own data store and they lazily synchronize whenever they want. They answer queries from the client based on the state of their local store. There is no consistency guarantee.

Exercise 5:

Explain the difference between the C in ACID and the C in CAP.
Consistency in ACID requires to only accept transactions that leave the database in a valid state. Consistency in CAP is the guarantee that you read the most recent write or an error.

Exercise 6:

Among the following propositions, identify which of the CAP theorem guarantees is/are violated.

- You matched on Tinder with your soul mate. Whenever you try to send a message, the application tells you to retry.

  Availability.

- You create a slack channel to discuss about the next PUBG gaming session you will have with your Swiss and American friends. After a while, you only see messages from your Swiss friends. Worried, you call Bob who lives in the US, and learn that he only sees messages from your American friends.

  Consistency, if one considers the chat room as a the global shared object.

  ~~Partition tolerance, if one considers the independent systems as consistent and available.~~

  Here is a good read for partition tolerance in CAP: https://codahale.com/you-cant-sacrifice-partition-tolerance/

- You open Facebook messenger on both your phone and computer and start talking with Mark Zuckerberg about privacy issues. You do not see the same content on your phone and on your laptop, making the conversation really confusing. All your private data is sold to Cambridge Analytica.

  Consistency.

- You want to find a torrent of the latest Smurfs movie. The website keeps telling you that there is no matching result. The same thing happens for any other movie you try to find. Time to get a streaming account.

  Availability.

- You play an online video game. Suddenly, everybody freezes, and none of the players is able to move. Apparently, a cat played with the plug of one of the servers in the data-center, disconnecting it from the other machines.

  Availability.

Exercise 7:

What are Quorums? How do they relate to consistency levels in Cassandra?

Quorums correspond to a number of nodes ($> 50\%$) that contain replicas of a given key-value pair. In a read operation, the coordinator waits for a number of replicas before sending the result to a client. Conversely, a client writes new values to a number of replicas.

A client can choose a consistency level for each operation, and some of them involve a quorum of replicas.

Exercise 8:

How does Cassandra compare to MySQL in terms of speed? Why?

Cassandra is faster at the expense of weaker consistency guarantees.

Exercise 9:

To ensure serializability (*i.e.*, avoid overlapping of writes or writes and reads) in quorum-based protocols, one must obtain the permission of a quorum in order to read or write a data item. Concerning quorums, answer the following questions:

(a) What property of majorities make them required in write quorums (*i.e.*, $W > \frac{N}{2}$)?

The intersection between two majorities contains at least one element. In the event of concurrent writes to a shared object, such intersecting element will be aware that one of them happened before. Consequently, it will not grant its permission for the latest overlapping write, therefore precluding the second writer from obtaining permission from a quorum.

(b) Why $R + W$ must be strictly greater than N?

Because the intersection of the two quorums R and W will then have at least one element. Assuming timestamp-based quorums, this intersecting node will be able to return the result of most recent successful write.

(c) What are the implications of choosing between $\langle R = \lceil \frac{N}{2} \rceil, W = \lceil \frac{N+1}{2} \rceil \rangle$ and $\langle R = 1, W = N \rangle$?

where R is the read quorum, W is the write quorum and N is total number of replicas in a distributed database.

They will have an impact on the latency of reads and writes. In the first case, reads and writes will take about the same amount of time. In the second, writes will be way more expensive than reads. Choosing between these options (or any other one in between) depends mostly on the application.