Log In Join

Back To Module Home

Distributed Systems

0% completed

Search Module

Introduction to Distributed Systems

Basic Concepts and Theorems

Conclusion

Mark Module as Completed

CAP Theorem's Consistency Model

In this lesson, we will explore the consistency model that is used by the CAP theorem.

We'll cover the following

In the previous lesson, we saw the four fundamental consistency models. When explaining the CAP theorem, we encountered the term consistency. Let's explore that now.

Consistency model used by the CAP theorem#

The "C" property in the CAP theorem refers to the linearizability model we previously described. This means it's impossible to build a system that will be *available* during a network partition while also being *linearizable*.

In fact, there is research that shows that even some weaker forms of consistency, such as sequential consistency, cannot be supported in tandem with availability under a network partition.

This vast number of different consistency models creates a significant amount of complexity.

As we explained previously, modeling consistency is supposed to help us reason about these systems. However, the explosion of consistency models can have the opposite effect.

Categorization of consistency models#

The CAP theorem can conceptually draw a line between all these consistency models and separate them into two major categories:

- Strong consistency models
- Weak consistency models

Strong consistency models#

Strong consistency models correspond to the “C” in the CAP theorem and cannot be supported in systems that need to be *available* during *network partitions*.

Weak consistency models#

Weak consistency models are the ones that can be supported while also preserving *availability* during a network partition.

Two commonly supported consistency models#

Considering the guarantees provided by several popular distributed systems nowadays (i.e., Apache Cassandra, DynamoDB, etc.), two models are commonly supported.

- Strong consistency, specifically *linearizability*

- Weak consistency, specifically *eventual consistency*

Reasons#

Most probably, the reasons most of the systems converged to the above two models are the following:

- **Linearizability** was selected amongst the available strong consistency models because a system needs to give up availability as part of the CAP theorem to support a strong consistency model. It then seems reasonable to provide the strongest model amongst the available ones, facilitating the software engineers' work with it.
- **Eventual Consistency** was selected amongst the available weak consistency models thanks to its simplicity and performance. Along the same lines, given the application relinquishes the strict guarantees of strong consistency for increased performance, it might as well accept the weakest guarantees possible to get the biggest performance boost it can. This makes it much easier for people to design and build applications on top of distributed systems, and to decide which side of the CAP theorem they prefer to build their application on.

Consistency Models

Next

Isolation Levels and Anomalies

Mark as Completed