A Review of 'CAP Twelve Years Later: How the "Rules" Have Changed'

Prashanth R Duggirala

Summary

This article tries to address some of the misconceptions around the CAP theorem which was proposed by the author more than a decade earlier. CAP theorem states that any database shared over a network can have at most two of three desirable properties between Consistency (C), High Availability of data (A) and tolerance towards network partitions (P). Though the expression of CAP served great purpose by initiating a lively debate about these tradeoffs in database systems, the author expands on the argument that the "2 of 3" formulation has misled people by oversimplifying the relations between these properties and that all three properties are more continuous than binary. The article also talks about how CAP is not a single choice for a whole system rather than a set of different choices for the many subsystems which should maximise the combination of consistency and availability for the system as a whole to make sense for the productivity of the application.

The article states that partitions are rare but there is a deep relation with latency, which are more common and during a timeout condition the program must make a decision to favor either consistency or availability. A partition resilient system is which detects a partition and enters a partition mode that either limits or allows operations to preserve or risk consistency depending on what the application requires. The article also says designers can also choose to selectively limit the application during partition so that the system can merge state during recovery, delaying risky operations is another implementation of this strategy. After the partition is resolved and communication restores, there should be a strategy for partition recovery where all the earlier partitions must be made consistent and the article discusses strategies towards achieving it. The first is to use version vectors and commutative replicated data types (CRDTs), they are a class of data structures that converge to a consistent state after a partition is resolved. Another strategy that is discussed is using compensating transactions to fix inconsistencies by reconciling the view of the system that exists on each side of the partition.

Strong points

1. The article sheds light on the misleading interpretation of the CAP theorem arising from the oversimplification of the theorem and clarifies that it is not the case of which property to sacrifice but partitions are inevitable and the goal is to maximise combinations of consistency and availability that makes sense for the specific application.

2. The article clearly defines consistency in different contexts used in database systems. Consistency, as interpreted in CAP, is equivalent to having a single up-to-date instance of the data whereas consistency as defined in ACID usually refers to the capability of maintaining the database in a consistent state at all times.

3. The article draws an important relationship between network partitions and latency which addresses one of the important criticisms the classical interpretation of CAP faced.

Weak points

1. The definition of the term “availability (A)” in CAP seems ambiguous in the article.

2. While the article recognizes latency can be used to model partitions in practice, it could have rewritten CAP abbreviation to include it so as to avoid any further debate.

3. The article focuses solely on network partitions and latency issues while it could have thrown some light on some other increasingly critical network issues.

Comments expanding upon weak points

Though the article states that “availability is obviously continuous from 0 to 100 percent”, there are different ways in which we can interpret “availability”. One can interpret it as an observed metric about how often a database system returns a successful response during operation. Or it can also be interpreted as a property of the algorithm the system is running. If the algorithm guarantees successful responses in all cases, the algorithm can be also referred to as “available”, the distinction is unclear.

Second, the author actually cites another article which addresses this minor issue where they mention that CAP can be rewritten as PACELC: if there is a partition (P) how does the system tradeoff between availability and consistency (A and C); else (E) when the system is running as normal in the absence of partitions, how does the system tradeoff between latency (L) and consistency (C)? This includes latency as another factor which seems more intuitive and worth mentioning in the article.

Finally, The CAP Theorem focuses on network partitions and latency, in practice there are many other failures that can occur. Individual servers can crash, network links can go down, Data on servers can be corrupted. Additionally, on networks DDoS attacks, or even more complex hacks are becoming more common. Such attacks, cannot simply be modeled as network partitions or latency. This article, being written after more than a decade after the initial article should have addressed these issues to some extent.