Report

The goal of this paper is to situate CAP Theorem in the broader context of distributed computing theory. The CAP Theorem depicts:
Consistency: Consistency, informally, simply means that each server returns the right response to each request,
Availability: The second requirement of the CAP Theorem is that the service guarantee availability. It simply means that each request eventually receives a response. Obviously, a fast response is better than a slow response, but for the purpose of CAP, it turns out that even requiring an eventual response is sufficient to create problems,
Partition-tolerance: The third requirement of the CAP theorem is that the service is partition tolerant. Unlike the other two requirements, this property can be seen as a statement regarding the underlying system: communication among the servers is not reliable, and the servers may be partitioned into multiple groups that cannot communicate with each other.

Advantages: It helps us to think about the effective way to choose design database systems. NoSQL (non-relational) databases are ideal for distributed network applications. NoSQL databases are classified based on the two CAP characteristics they support: 1) CP Database 2) AP Database 3)CA Database

Disadvantages: The CAP theorem only offers us to choose between Consistency and Availability. It does not provide a way that is balanced Consistency and Availability. High consistency comes at the cost of lower availability. High availability comes at the cost of lower consistency. CAP theorem fails to encompass problems like node failure, loss or delay of messages, and restart time-lapse of nodes other than partition. Fair link loss is possessed by a link if it has a nonzero probability of packet loss. In such a link the lost packets will be delivered by a limited number of repeated attempts ensuring packets reach the destination.

Limitations: Outages can be caused by a variety of factors that the CAP theorem doesn't consider, such as single-node hardware failure, application bugs, or operator error. And when the system is considered as a whole, even network partitions can be handled in ways that increase availability without sacrificing consistency.

Applications: The applications of CAP Theorem are such as search engines, e-commerce, online music services, or cloud-based data storage. Some applications are organized hierarchically, partitioning along these different dimension multiple times.

Conclusion: The CAP Theorem is one example of a fundamental trade-off between safety and liveness in fault-prone systems. Examining this inherent trade-off yields some insights into how systems can be designed to meet an application’s needs, despite unreliable networks. CAP theorem also provides a base for designing and modeling databases, but it fails to meet the current needs. The problems are self-imposing, majorly because definitions of CAP theorem are open to interpretation.