CAP Theorem and Google's Spanner

Contents

Executive Summary

This report delves into the CAP theorem—often called Brewer's theorem—and explains its three guarantees (Consistency, Availability, Partition Tolerance), the unavoidable trade-off when a network partition occurs, and the special case when no partition exists. It then examines Google Spanner, a globally distributed SQL database that leverages Google's private network and the TrueTime API to deliver strong consistency and high availability by dramatically reducing the likelihood and impact of partitions. Finally, it contrasts ACID and BASE philosophies and reviews Brewer's 2012 clarification that the C/A trade-off is only relevant during partitions.

1 Introduction

Distributed data stores underpin modern cloud applications, but network failures are inevitable at scale. The CAP theorem formalizes the limits on what a distributed system can guarantee when nodes lose communication, and understanding CAP is essential for system architects making consistency vs. availability decisions in the face of partitions [1].

2 CAP Theorem

2.1 Definition and Origins

The CAP theorem, conjectured by Eric Brewer in 2000 and proven by Gilbert and Lynch in 2002, asserts that a distributed system can simultaneously provide at most two of three properties: Consistency, Availability, and Partition Tolerance [1, 2]. Brewer originally presented this as a "principle," later clarified to highlight that the C/A choice only emerges under actual network partition events [1].

2.2 Components

- **Consistency (C)**: Every read receives the most recent write or an error—distinct from ACID's notion of database state consistency [1].
- **Availability (A)**: Every request processed by a non-failed node returns a non-error response, though it may not reflect the latest write [3].
- **Partition Tolerance (P)**: The system continues to operate despite an arbitrary number of lost or delayed messages between nodes [1].

2.3 Trade-offs under Network Partition

In the event of a network partition—when subsets of nodes can't communicate—designers must choose between:

1. **Consistency over availability (CP)**: Cancel or delay operations until the partition heals, ensuring reads reflect the latest writes but risking unavailability [1].
2. **Availability over consistency (AP)**: Serve reads and writes immediately from whichever partition, accepting possible stale data [4].

2.4 Implications of No Partition

When no partition exists (e.g., within a tightly controlled LAN), both consistency and availability can be satisfied simultaneously (CA system). However, realistic WAN-scale deployments must tolerate partitions, making the CP vs. AP choice unavoidable under failure [1].

3 Google Spanner

3.1 Overview

Google Spanner is a globally distributed, strongly consistent SQL database built to span data across multiple continents while maintaining transactional semantics comparable to a single-machine database [5].

3.2 TrueTime and Global Timestamps

Spanner's key innovation is the *TrueTime* API, which provides tight bounds on clock uncertainty using GPS and atomic clocks. Every transaction is stamped with a globally consistent timestamp, enabling lock-free, externally consistent reads across any replica without blocking writes [6].

3.3 Infrastructure and Partition Minimization

Rather than "breaking" CAP, Spanner minimizes partition risk by running on Google's private fiber network with redundant paths, specialized hardware, and rigorous operational controls. This infrastructure investment makes partitions so rare that practitioners can operate under CA assumptions in practice [7].

3.4 Operational Reliability

Years of SRE-driven optimizations have reduced network-related outages to under 10 % of Spanner's total downtime causes, shifting the availability focus away from partitions toward software and hardware reliability [7].

3.5 Pragmatic Approach to CAP

While no system can guarantee 100 % availability under partitions, Spanner's approach effectively raises the bar: by drastically lowering the probability of partitions, it allows applications to benefit from strong consistency and high availability in nearly all practical scenarios [8].

4 Comparisons and Philosophies

4.1 ACID vs. BASE

- **ACID systems** (e.g., traditional RDBMS) typically choose CP: they favor consistency and will sacrifice availability under partitions (e.g., transaction rollbacks)

[1].

- **BASE systems** (common in NoSQL solutions) choose AP: they favor availability, allowing eventual consistency and conflict-resolution techniques like CRDTs to reconcile divergent replicas later [9].

4.2 Brewer's Clarification

In 2012, Brewer emphasized that the "two out of three" CAP summary is misleading because consistency and availability can both be met *unless* a partition occurs. Thus, the real trade-off point is *only* when messages are dropped or delayed between nodes [1].

5 Conclusion

Understanding the CAP theorem's constraints helps architects make informed design decisions for distributed systems. Google Spanner illustrates a pragmatic path: invest heavily in minimizing partitions (via private networks and clock synchronization) so that strong consistency and high availability can coexist in practice.

References

[1] Wikipedia contributors, "CAP theorem," *Wikipedia, The Free Encyclopedia*, Mar. 2025. https://en.wikipedia.org/wiki/CAP_theorem

[2] S. Gilbert and N. Lynch, "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services," *ACM SIGACT News*, vol. 33, no. 2, 2002.

[3] IBM Cloud Docs, "What Is the CAP Theorem?" IBM, accessed May 2025. https://cloud.ibm.com/docs/Cloudant?topic=Cloudant-cap-theorem

[4] J. Johnson, "CAP Theorem Explained: Consistency, Availability Partition Tolerance," BMC Blogs, Oct. 30 2024. https://www.bmc.com/blogs/cap-theorem/

[5] Wikipedia contributors, "Spanner (database)," *Wikipedia, The Free Encyclopedia*, Feb. 2025. https://en.wikipedia.org/wiki/Spanner_(database)

[6] Google Cloud Blog, "Strict Serializability and External Consistency in Spanner," Mar. 2023. https://cloud.google.com/blog/products/databases/strict-serializability-and-external-consistency-in-spanner

[7] T. Corbett et al., "Spanner, TrueTime The CAP Theorem," Google Research, 2016. https://research.google.com/pubs/archive/45855.pdf

[8] J. Corbett et al., "Spanner: Google's Globally Distributed Database," in *Proceedings of OSDI '12*, 2012. https://dl.acm.org/doi/10.1145/2491245

[9] J. Corbett et al., "Spanner: Google's Globally-Distributed Database," USENIX, 2012. https://www.usenix.org/system/files/conference/osdi12/osdi12-final-16.pdf

[10] J. Brownlee, "Spanner, the Google Database That Mastered Time, Is Now Open to Everyone," *Wired*, Feb. 2017. https://www.wired.com/2017/02/spanner-google-database-harnessed-time-now-open-everyone/