WIKIPEDIA
The Free Encyclopedia

CAP theorem

In theoretical computer science, the **CAP theorem**, also named **Brewer's theorem** after computer scientist Eric Brewer, states that any distributed data store can provide only two of the following three guarantees:[1][2][3]

Consistency
: Every read receives the most recent write or an error.

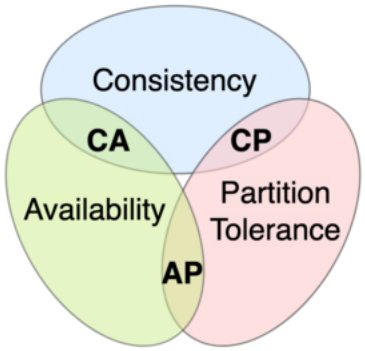
Availability
: Every request receives a (non-error) response, without the guarantee that it contains the most recent write.

Partition tolerance
: The system continues to operate despite an arbitrary number of messages being dropped (or delayed) by the network between nodes.

When a network partition failure happens, it must be decided whether to do one of the following:

- cancel the operation and thus decrease the availability but ensure consistency
- proceed with the operation and thus provide availability but risk inconsistency.

Thus, if there is a network partition, one has to choose between consistency or availability. Note that consistency as defined in the CAP theorem is quite different from the consistency guaranteed in ACID database transactions.[4]



CAP Theorem Venn Diagram

Explanation

No distributed system is safe from network failures, thus network partitioning generally has to be tolerated.[5][6] In the presence of a partition, one is then left with two options: consistency or availability. When choosing consistency over availability, the system will return an error or a time out if particular information cannot be guaranteed to be up to date due to network partitioning. When choosing availability over consistency, the system will always process the query and try to return the most recent available version of the information, even if it cannot guarantee it is up to date due to network partitioning.

In the absence of a partition, both availability and consistency can be satisfied.[7]

Database systems designed with traditional ACID guarantees in mind such as RDBMS choose consistency over availability, whereas systems designed around the BASE philosophy, common in the NoSQL movement for example, choose availability over consistency.[8]

History

According to University of California, Berkeley computer scientist Eric Brewer, the theorem first appeared in autumn 1998.[8] It was published as the CAP principle in 1999[9] and presented as a conjecture by Brewer at the 2000 Symposium on Principles of Distributed Computing (PODC).[10] In 2002, Seth Gilbert and Nancy Lynch of MIT published a formal proof of Brewer's conjecture, rendering it a theorem.[1]

In 2012, Brewer clarified some of his positions, including why the often-used "two out of three" concept can be somewhat misleading because system designers only need to sacrifice consistency or availability in the presence of partitions; partition management and recovery techniques exist. Brewer also noted the different definition of consistency used in the CAP theorem relative to the definition used in ACID.[8][11]

A similar theorem stating the trade-off between consistency and availability in distributed systems was published by Birman and Friedman in 1996.[12] Birman and Friedman's result restricted this lower bound to non-commuting operations.

The PACELC theorem, introduced in 2010,[7] builds on CAP by stating that even in the absence of partitioning, there is another trade-off between latency and consistency. PACELC means, if partition (P) happens, the trade-off is between availability (A) and consistency (C); Else (E), the trade-off is between latency (L) and consistency (C).

See also

- Fallacies of distributed computing
- PACELC theorem
- Paxos (computer science)
- Raft (computer science)
- Zooko's triangle
- Inconsistent triad

References

1. Gilbert, Seth; Lynch, Nancy (2002). "Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services". *ACM SIGACT News*. Association for Computing Machinery (ACM). **33** (2): 51–59. doi:10.1145/564585.564601 (https://doi.org/10.1145%2F564585.564601). ISSN 0163-5700 (https://www.worldcat.org/issn/0163-5700). S2CID 15892169 (https://api.semanticscholar.org/CorpusID:15892169).
2. "Brewer's CAP Theorem" (https://www.julianbrowne.com/article/brewers-cap-theorem/). *julianbrowne.com*. 2009-01-11.
3. "Brewers CAP Theorem on distributed systems" (https://www.royans.net/2010/02/brewers-cap-theorem-on-distributed.html). *royans.net*. 2010-02-14.
4. Liochon, Nicolas. "The confusing CAP and ACID wording" (http://blog.thislongrun.com/2015/03/the-confusing-cap-and-acid-wording.html). *This long run*. Retrieved 1 February 2019.
5. Kleppmann, Martin (2015-09-18). A Critique of the CAP Theorem (https://www.repository.cam.ac.uk/handle/1810/267054) (Report). Apollo - University of Cambridge Repository. arXiv:1509.05393 (https://arxiv.org/abs/1509.05393). Bibcode:2015arXiv150905393K (https://ui.adsabs.harvard.edu/abs/2015arXiv150905393K). doi:10.17863/CAM.13083 (https://doi.org/10.17863%2FCAM.13083). S2CID 1991487 (https://api.semanticscholar.org/CorpusID:1991487). Retrieved 24 November 2019.

6. Martin, Kleppmann. "Please stop calling databases CP or AP" (https://martin.kleppmann.com/2015/05/11/please-stop-calling-databases-cp-or-ap.html). *Martin Kleppmann's Blog*. Retrieved 24 November 2019.
7. Abadi, Daniel (2010-04-23). "DBMS Musings: Problems with CAP, and Yahoo's little known NoSQL system" (http://dbmsmusings.blogspot.com/2010/04/problems-with-cap-and-yahoos-little.html). *DBMS Musings*. Retrieved 2018-01-23.
8. Brewer, Eric (2012). "CAP twelve years later: How the "rules" have changed" (http://www.infoq.com/articles/cap-twelve-years-later-how-the-rules-have-changed). *Computer*. Institute of Electrical and Electronics Engineers (IEEE). **45** (2): 23–29. doi:10.1109/mc.2012.37 (https://doi.org/10.1109%2Fmc.2012.37). ISSN 0018-9162 (https://www.worldcat.org/issn/0018-9162). S2CID 890105 (https://api.semanticscholar.org/CorpusID:890105).
9. Armando Fox; Eric Brewer (1999). *Harvest, Yield and Scalable Tolerant Systems*. Proc. 7th Workshop Hot Topics in Operating Systems (HotOS 99). IEEE CS. pp. 174–178. doi:10.1109/HOTOS.1999.798396 (https://doi.org/10.1109%2FHOTOS.1999.798396).
10. Eric Brewer. "Towards Robust Distributed Systems" (http://www.cs.berkeley.edu/~brewer/cs262b-2004/PODC-keynote.pdf) (PDF).
11. Carpenter, Jeff; Hewitt, Eben (July 2016). *Cassandra: The Definitive Guide* (https://www.oreilly.com/library/view/cassandra-the-definitive/9781491933657/) (2nd ed.). Oreilly. ISBN 9781491933657. "In February 2012, Eric Brewer provided an updated perspective on his CAP theorem [..] Brewer now describes the "2 out of 3" axiom as somewhat misleading. He notes that designers only need sacrifice consistency or availability in the presence of partitions, and that advances in partition recovery techniques have made it possible for designers to achieve high levels of both consistency and availability."
12. Ken Birman; Roy Friedman (April 1996). "Trading Consistency for Availability in Distributed Systems" (https://ecommons.cornell.edu/handle/1813/7235). hdl:1813/7235 (https://hdl.handle.net/1813%2F7235).

External links

- Spanner, TrueTime and the CAP Theorem (https://research.google.com/pubs/pub45855.html)
- Perspectives on the CAP theorem (http://groups.csail.mit.edu/tds/papers/Gilbert/Brewer2.pdf): An update from Gilbert and Lynch in 2012
