CAP and Database System

Wenliang ZHANG
https://zedware.github.io

Outline

- A brief history
- ACID vs. BASE vs. CAP
- 2 out of 3?
- More On Databases
- Concrete examples
- Summary

A brief history

- Eric Brewer's background
  - Professor of UC Berkeley
  - Co-founder and Chief Scientist of Inktomi, acquired by Yahoo!
  - VP of Infrastructure at Google

https://en.wikipedia.org/wiki/Eric_Brewer_(scientist)

A brief history

- Related research fields
  - [Distributed] Systems
    - SOSP, OSDI, FAST
  - Database systems
    - SIGMOD, VLDB, ICDE
  - Gaps between Systems and Databases
    - In the good old days, less gap
  - Gaps between research and industry
    - Always there
    - Research leads the design and implementations before the Internet era
    - Industry leads due to the infrastructure, big data and requirements

A brief history

- 1997 BASE coined but not so popular
- 1998, 1999 developed CAP
- 2000 CAP introduced to public in the keynote
- 2002 CAP theorem proven formally
- 2006 Bigtable: A Distributed Storage System for Structured Data
- 2007 Dynamo: Amazon’s highly available key-value store
- 2008 PNUTS: Yahoo’s hosted data serving platform
- 2009 Cassandra - A Decentralized Structured Storage System
- 2012 Spanner: Google’s Globally-Distributed Database

A brief history – what is learnt?

- Various compromise of CAP
- Replication state machine
- Multiple replica (Paxos, Raft and Quorum)
- Distributed transactions
- Snapshot isolation
- Multiple tenant
- SQL is pervasive

Outline

- A brief history
- ACID vs. BASE vs. CAP
- 2 out of 3?
- More On Databases
- Concrete examples
- Summary

Definition

- ACID first
  - Atomic, Consistency, Isolation, Durability
- BASE coined to take the opposite meaning of ACID
  - Basic Availability, Soft-state, Eventual Consistency
- CAP
  - Consistency, Availability, Tolerance to network Partitions
- BASE is not popular as CAP
  - CAP is the Winner of the era

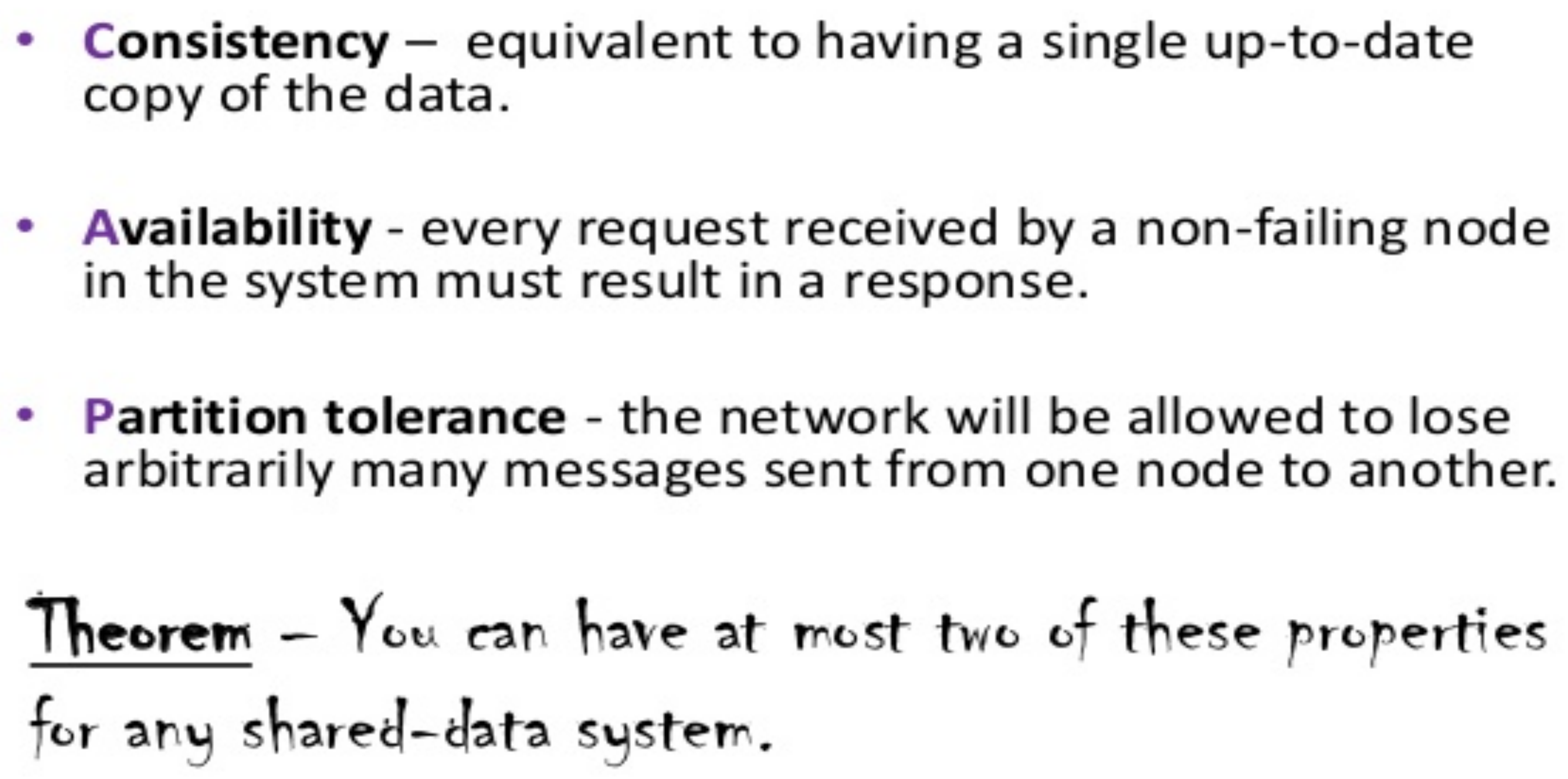
ACID

- Skip this slide for the database guys

BASE

- Basically available
  - The system guarantee availability in terms of the CAP theorem
- Soft state
  - State of the system may change over time, even without input
  - Due to the eventual consistency model
- Eventual consistency
  - The system will become consistent over time, given that it doesn't receive input during that time.

CAP Theorem



- **Consistency** – equivalent to having a single up-to-date copy of the data.
- **Availability** - every request received by a non-failing node in the system must result in a response.
- **Partition tolerance** - the network will be allowed to lose arbitrarily many messages sent from one node to another.

Theorem – You can have at most two of these properties for any shared-data system.

https://image.slidesharecdn.com/cap-131117230434-phpapp02/95/dynamo-and-bigtable-in-light-of-the-cap-theorem-11-638.jpg?cb=1384729712

ACID vs. BASE [vs. CAP]

- ACID
  - Strong consistency
  - Isolation
  - Focus on commit
  - Nested TXes
  - Availability?
  - Conservative
  - Difficult evolution (e.g. schema)

- BASE
  - Weak consistency (stale data)
  - Availability first
  - Best effort
  - Approximate answer is ok
  - Aggressive
  - Simpler, faster, and easier to evolution

ACID vs. CAP

- A
    - Atomic vs.
    - Availability
- C
    - General consistency vs.
    - Single-copy consistency

Retrospect – General ideas

- Terms are evolving
    - Marketing vs. Technical
    - Convergence
        - Database system is a black hole
        - Converged with many technologies
    - Used in different scenarios
        - Database vs. Non-database
- Example
    - Big Data

Retrospect - Things CAP does NOT say

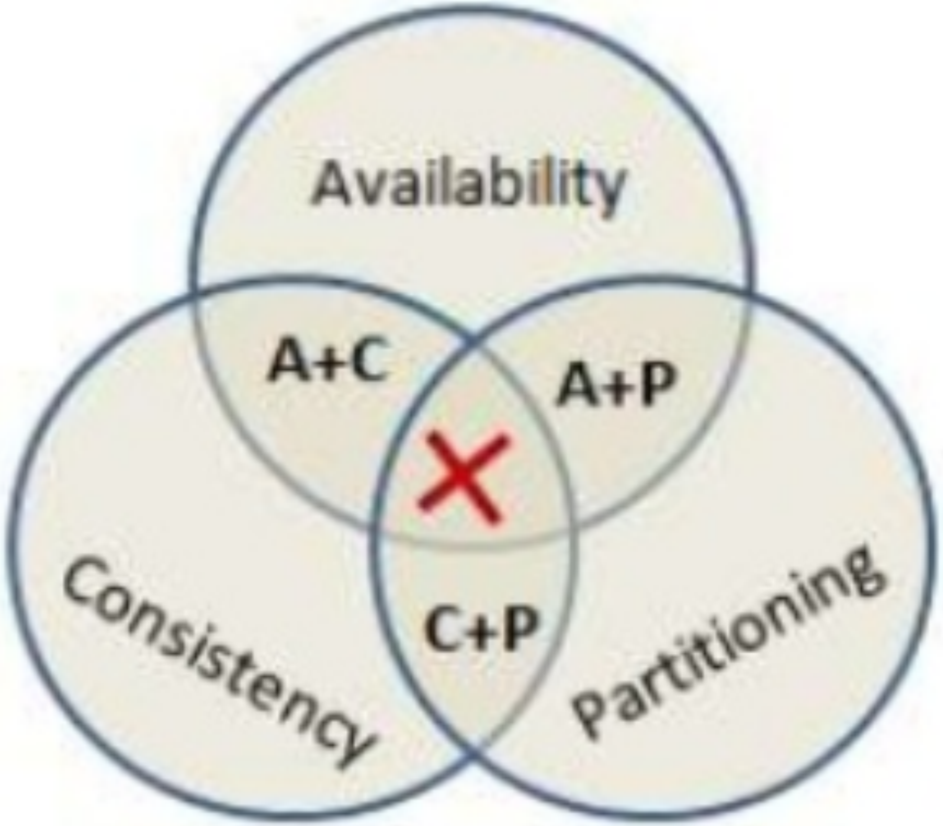
- Give up on consistency (in the wide area)
  - Inconsistency should be the exception
  - Many projects give up more than needed
- Give up on transactions (ACID)
  - Need to adjust "C" and "I" expectations (only)
- Don't use SQL
  - SQL is appearing in "NoSQL" systems
  - Declarative languages fit well with CAP

Source: EricBrewer_NoSQLPastPresentFuture.pdf

Outline

- A brief history
- ACID vs. BASE vs. CAP
- 2 out of 3?
- More On Databases
- Concrete examples
- Summary

2 out of 3?



2 out of 3? - AC

- Examples
    - Single site databases
    - Cluster databases
    - LDAP
    - xFS file system
- Traits
    - 2-phase commit
    - Cache validation protocols

2 out of 3? - CP

- Examples
  - Distributed databases
  - Distributed locking
  - Majority protocols
- Traits
  - Pessimistic locking
  - Making minority partitions unavailable

2 out of 3? - AP

- Examples
  - Web caching
  - DNS
  - GFS, Dynamo etc.
- Traits
  - Expirations lease
  - Conflict resolution
  - Optimistic

Outline

- A brief history
- ACID vs. BASE vs. CAP
- 2 out of 3?
- More On Databases
- Concrete examples
- Summary

More On Databases

- Typical systems
  - Oracle – single-site, primary/backup, cluster
  - Spanner – majority protocol
    - Azure storage also claimed to be no compromise in CAP
- Traditional storage vs. GFS
- Traditional database vs. Spanner

More On Databases

- What does partition mean
  - Oracle in traditional local site
  - Oracle and its friends in different organizations
  - Spanner in Google’s private-owned global network
- Bank systems
  - Many sub-systems in real world application
  - Are `partitioned/disconnected` from the very beginning
  - Rely on the similar idea as WAL
  - XA is not widely used
  - Application solve the distributed transactions

Outline

- A brief history
- ACID vs. BASE vs. CAP
- 2 out of 3?
- More On Databases
- Concrete examples
- Summary

Nuances

- Not 1 or 0
  - There are weak isolations in databases
  - Consistency in databases may be weak due to crash, bugs etc.
  - Single site database can have `partition`s
  - Visiting secondary replica may lead to better availability but hurt consistency (think about the 3 modes in Oracle database)
  - Actions after Partition occurs
- Case by case
  - Must make tradeoffs in industry
  - Different systems have different choices

Concrete examples

- Application
  - ATM can withdraw offline
  - Check kiting
  - Flight tickets and onboard
  - e-Commerce overbooked
- Compensation TX
  - Most cases can be compensated
  - Some can’t: external actions
  - Causal consistency?

Outline

- A brief history
- ACID vs. BASE vs. CAP
- 2 out of 3?
- More On Databases
- Concrete examples
- Summary

Summary

- CAP leads to exploration of systems
- Convergence of distributed system & database
- The devil is in the detail
- Database is different
  - Think about Spanner again
  - Spanner, TrueTime and the CAP Theorem

Those who cannot remember the past are condemned to repeat it.

- George Santayana