Data Architecture and Database Technologies (I)

Dr. Karl-Heinrich Anders

ACID versus BASE

CONCISTENCY IN DISTRIBUTED SYSTEMS

Consistency

- Consistency means that in the case of redundant storage of data (possibly distributed across several nodes) (replication), all replicas have the same value.
- In distributed databases (often also in NoSQL systems), consistency is often only implemented in a weakened form for performance reasons, in which a write operation is not immediately propagated to all replicas. This concept is called eventual consistency.
- The concept of consistency is unfortunately overloaded and is not used uniformly today. One reason for this is the emergence of NoSQL systems with their underlying theories, for example the CAP theorem.
- Although both C's stand for 'consistency', that in CAP means something different than that in ACID. **Consistency in CAP means 'same values in all replicas'** (sometimes called 'structural consistency'), **while consistency in ACID stands for logical agreement with the modeled section of the real world.** The latter is sometimes also called 'semantic consistency' or better integrity.

Eventual consistency

- Eventual consistency is a weakened variant of consistency that is often used for distributed databases. For performance reasons, data is not immediately distributed to all servers/partitions during write operations.

- Instead, algorithms are used to ensure that the data is made consistent after the write operations have been completed, usually without any indication of the time period in which the process will be completed. In the meantime, different data sets are on each server. This can cause identical, simultaneous queries from multiple users to produce different results. All that can be trusted is that the data will eventually be consistent, hence the name of this concept.

- Eventual consistency is one of those concepts used in NoSQL databases to achieve high performance and scalability, and it is one of the basic concepts of **BASE**. These advantages are bought with the fact that the stricter **ACID** consistency concept common in relational database management systems is no longer guaranteed.

With Eventual Consistency, different consistency models are followed:

- **Monotonic Read Consistency:**
  - once a client has read an object x, it will never read an older version of the object thereafter.

- **Monotonic Write Consistency:**
  - if a client has set an attribute of object x to the value '1' and then changed it to '2', it is guaranteed that in no replica the value '2' is replaced by '1'.

- **Read Your Own Writes (sometimes abbreviated to RYOW):**
  - if a client has written an object x, it will never read an older version of the object.

- **It must be weighed up for the respective application which of the various consistency models is most suitable in each case.**