UNIVERSITÀ
CATTOLICA
del Sacro Cuore

NoSQL databases

Definition

> A **NoSQL** (originally referring to "non-SQL" or "non-relational“, now “Not Only” SQL) database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.
>
> Motivations for this approach include: simplicity of design, simpler "horizontal" scaling to clusters of machines (which is a problem for relational databases), finer control over availability and limiting the object-relational impedance mismatch.

source: NoSQL – Wikipedia

Traditional RDBMS

In the computing system (web and business applications), there are enormous data that comes out every day from the web. A large section of these data is handled by Relational database management systems (RDBMS). The idea of relational model came with E.F.Codd’s 1970 paper "*A relational model of data for large shared data banks*" which made data modeling and application programming much easier. Beyond the intended benefits, the relational model is well-suited to client-server programming and today it is predominant technology for storing **structured data** in web and business applications.

ACID rules

A database transaction, must be atomic, consistent, isolated and durable. Below we have discussed these four points.

- **Atomic :** A transaction is a logical unit of work which must be either completed with all of its data modifications, or none of them is performed.
- **Consistent :** At the end of the transaction, all data must be left in a consistent state.
- **Isolated :** Modifications of data performed by a transaction must be independent of another transaction. Unless this happens, the outcome of a transaction may be erroneous.
- **Durable :** When the transaction is completed, effects of the modifications performed by the transaction must be permanent in the system.

Often these four properties of a transaction are acronymed as **ACID**.

CAP theorem

In theoretical computer science, the **CAP theorem**, also named **Brewer's theorem** after computer scientist Eric Brewer, states that it is impossible for a distributed data store to simultaneously provide more than two out of the following three guarantees:

- *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

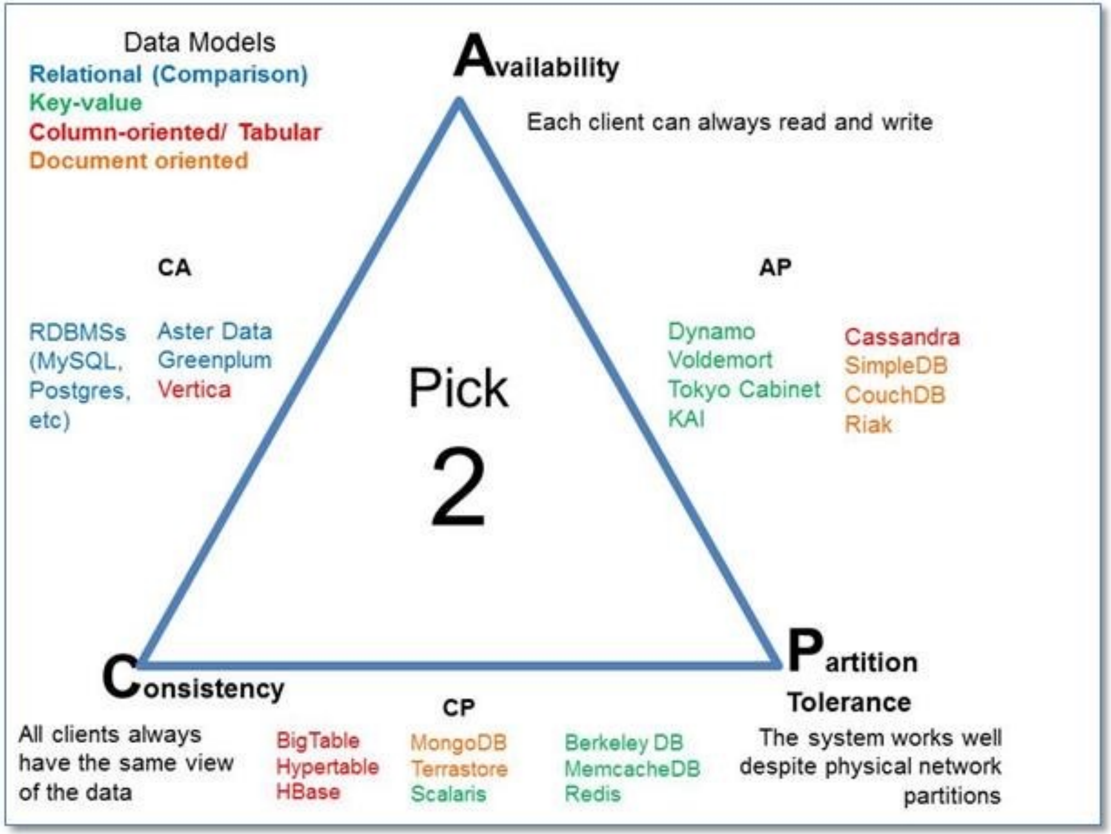
When a network partition failure happens should we decide to

- Cancel the operation and thus decrease the availability but ensure consistency
- Proceed with the operation and thus provide availability but risk inconsistency

The CAP theorem implies that in the presence of a network partition, one **has to choose between consistency and availability**.

Note that consistency as defined in the CAP theorem is quite different from the consistency guaranteed in ACID database transactions.

CAP theorem visualization



BASE system

The CAP theorem states that a distributed computer system cannot guarantee all of the following three properties at the same time: Consistency, Availability, Partition tolerance.

A BASE system gives up on consistency.

- **B**asically **A**vailable indicates that the system *does* guarantee availability, in terms of the CAP theorem.
- **S**oft state indicates that the state of the system may change over time, even without input. This is because of the eventual consistency model.
- **E**ventual consistency indicates that the system will become consistent over time, given that the system doesn't receive input during that time.

NoSQL categories (data models)

There are four most common categories of NoSQL databases:

- Key-value stores
- Column-oriented
- Graph
- Document oriented

Each of these categories has its own specific attributes and limitations. There is not a single solutions which is better than all the others, however there are some databases that are better to solve specific problems.

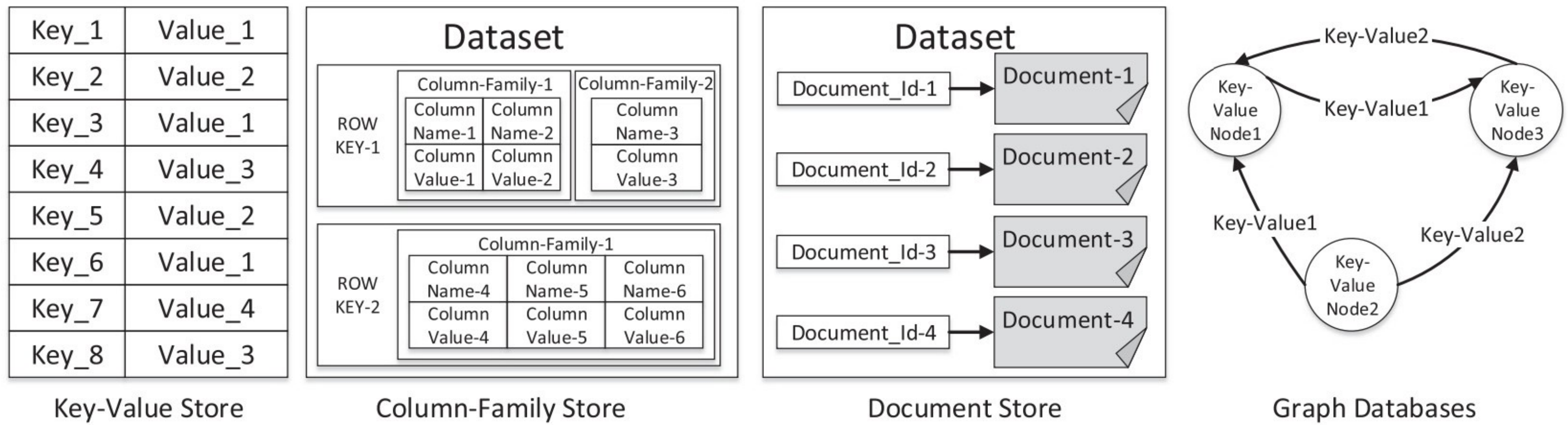


Figure 1 Different types of NoSQL data models.

source: Grolinger et al., [Data management in cloud environments: NoSQL and NewSQL data stores]

Key-value stores

- Key-value stores are most basic types of NoSQL databases.
- Designed to handle huge amounts of data.
- Based on Amazon’s Dynamo paper.
- Key value stores allow developer to store schema-less data.
- In the key-value storage, database stores data as hash table where each key is unique and the value can be string, JSON, BLOB (Binary Large OBjec) etc.
- A key may be strings, hashes, lists, sets, sorted sets and values are stored against these keys.
- For example a key-value pair might consist of a key like "Name" that is associated with a value like "Robin".
- Key-Value stores can be used as collections, dictionaries, associative arrays etc.
- Key-Value stores follow the 'Availability' and 'Partition' aspects of CAP theorem.
- Key-Values stores would work well for shopping cart contents, or individual values like color schemes, a landing page URI, or a default account number.

Example of Key-value store DataBase : Redis, Dynamo, Riak. etc.

Column-oriented

- Column-oriented databases primarily work on columns and every column is treated individually.
- Values of a single column are stored contiguously.
- Column stores data in column specific files.
- In Column stores, query processors work on columns too.
- All data within each column datafile have the same type which makes it ideal for compression.
- Column stores can improve the performance of queries as it can access specific column data.
- High performance on aggregation queries (e.g. COUNT, SUM, AVG, MIN, MAX).
- Works on data warehouses and business intelligence, customer relationship management (CRM), Library card catalogs etc.

Example of Column-oriented databases : BigTable, Cassandra, SimpleDB etc.

Graph databases

- A graph database stores data in a graph.
- It is capable of elegantly representing any kind of data in a highly accessible way.
- A graph database is a collection of nodes and edges
- Each node represents an entity (such as a student or business) and each edge represents a connection or relationship between two nodes.
- Every node and edge are defined by a unique identifier.
- Each node knows its adjacent nodes.
- As the number of nodes increases, the cost of a local step (or hop) remains the same.
- Index for lookups.
- **Example of Graph databases :** OrientDB, Neo4J, Titan.etc.

Document-oriented

- A collection of documents
- Data in this model is stored inside documents.
- A document is a key value collection where the key allows access to its value.
- Documents are not typically forced to have a schema and therefore are flexible and easy to change.
- Documents are stored into collections in order to group different kinds of data.
- Documents can contain many different key-value pairs, or key-array pairs, or even nested documents.

Example of Document Oriented databases : MongoDB, CouchDB etc.

Performance

Data model	Performance	Scalability	Flexibility	Complexity	Functionality
Key–value store	high	high	high	none	variable (none)
Column-oriented store	high	high	moderate	low	minimal
Document-oriented store	high	variable (high)	high	low	variable (low)
Graph database	variable	variable	high	high	graph theory
Relational database	variable	variable	low	moderate	relational algebra

source: NoSQL – Wikipedia

References

- NoSQL introduction - w3resource
- What is NoSQL? NoSQL Databases Explained | MongoDB
- What is NoSQL? | Nonrelational Databases, Flexible Schema Data Models | AWS (amazon.com)
- CAP theorem – Wikipedia

Reading list

[K. Grolinger, W.A. Higashino, A. Tiwari and M. Capretz, Data management in cloud environments: NoSQL and NewSQL data stores, *Journal of Cloud Computing*, 2013.]

(documents available on Blackboard)

Questions, comments?