NoSQL Database technologies

A brief introduction for NoSQL and comparison with RDBMS.

M Hithesh Chowdary

What is NoSQL ?

- NoSQL database, also called Not Only SQL, is an approach to data management and database design that's useful for very large sets of distributed data.
- NoSQL is especially useful when an enterprise needs to access and analyse massive amounts of unstructured data.
- Companies that use NoSQL include Netflix, LinkedIn, Twitter, and Facebook.

Why NoSQL ?

- Ironically, relational databases deal poorly with relationships. Relationships do exist in relational databases, but only as a means of joining tables.
- Simply put, NoSQL databases are a better choice when you have data that doesn't fit well into tables(i.e., the Relational model) or what is often called big data.
- But most NoSQL stores compromise on true ACID transactions and compromise consistency (in the sense of the CAP theorem) in favour of availability and partition tolerance.

Storage in NoSQL

- A **key/value database** is ideal when data is accessed by using a key and value is just a blob of unstructured data.
- A **document store database** manages and stores data at the document level. It's similar to a key/value database, but a structure is imposed on the data.
- A **column DBMS** also known as a column store or wide column store reorients the focus of the data from the row to the column, storing data as sections of columns rather than as rows.
- A **graph database** focuses on relationships between values, and stores data using the mathematical concept of a graph

My project

- It is about designing a system which suggests to a user what kind of DBMS technology is suitable for his/her application or product based on its data. For e.g.,
- Calculating average income? Ask a relational database.
- Building a shopping cart? Use a key-value Store.
- Storing structured product information? Store as a document.
- Describing how a user got from point A to point B? Follow a graph.
- The above are pretty basic but need to build an algorithm for different kinds of data and technologies

Some comparisons

	NoSQL	Relational
Scalability	Yes	No
ACID	Not always	Yes
SQL	Yes(some extent)	Yes
Market share	Growing	High
Ease of Management	High	Low

- There is actually no single winner.
- It always depends on the kind of data the enterprise is dealing with.
- The user needs help in deciding what DB technology to use for his/her application or product.

Next generation DB

Problems

- The first question that arises is whether a common query language is feasible at all for NoSQL, and if yes why not just SQL.
- How to deal with ACID properties in NoSQL?
- Lack of tool support even for a rich dataset?

Solutions

- SQL cannot be used because it mainly depends on the join mechanism which NoSQL DB's try to avoid. We need a language that is adapting.
- NewSQL emerging which retains the scalability of NoSQL also confirms to the ACID properties.
- As most of them are open source more contribution and usage from developers to help tool set and authenticity improvement.

Conclusions

- While NoSQL has come aboard, the relational world hasn't been sitting still. It's been rising to meet the challenge in a lot of ways. As a result, it's a mistake to ignore the merits of relational databases.
- Materialized view can help significantly in RDBMS.
- It's really a matter of both succeeding.
- Examples for NoSQL DB's include Mongo DB, Cassandra, Neo4J, Couch DB, HBase..etc.

Thank you ☺