NoSQL databases

Seminar 2
Md Sahil
BCSEIV
Roll - 001710501029

Abstract:

Relational Databases dominated the industry for almost 20 years. It had its downsides, which led to people innovating new forms of databases, most of which didn't become much popular. But with the introduction of the Internet and the big companies like Amazon and Google, relational databases slowly became inadequate. WIth the huge amount of data and incoming traffic, the requirement of huge data stores became necessary. Scaling became a problem in relational databases because it was not designed to scale horizontally. It was made to run on a single huge box. As a result companies started to build their own database systems that were quite different from relational databases. Google made their Bigtable and Amazon made Dynamo. This led to the "NoSQL" movement.

NoSQL databases are hard to define.
Common characteristics include

- Non - relational
- Most of them are open-source
- Cluster friendly
- Designed for 21st century web
- Schema-less

Types of NoSQL databases:

- Document
- Column family
- Graph
- Key-value

Consistency Problems

Relational Databases are ACID (Atomic, Consistent, Isolated, Durable).
Among the NoSQL databases Graph databases are ACID

Other aggregate oriented databases (Document, Column family, Key-value) don't need transactions as much. The design of these databases are supposed to be made as such each entity in the database is aggregated on transaction boundaries. But still, there are several methods through which we can address consistency problems in aggregate oriented databases. Consistency problems can be of two types:

- Logical -
  This happens when data is divided into shards and distributed into a number of small stores.
- Replication -
  This occurs when data is replicated in order to provide greater availability and fault tolerance.

Most distribution systems need to come up with business specific choices in order to deal with these consistency problems. In most systems the designer has to choose between Consistency or Availability.

When to choose NoSQL

There are mainly two drivers that push for the use of NoSQL databases. One of them is the ability to handle huge amounts of data. The second factor is ease of development.
NoSQL databases can also be used for analytics.

In real world applications the choice of type of database depends on the needs of the application. And most of the time a single application uses multiple types of databases.