7

NO-SQL

7.1 NO-SQL: INTRODUCTION

NoSQL is a non-relational database management systems, different from traditional relational database management systems in some significant ways. It is designed for distributed data stores where very large scale of data storing needs (for example Google or Facebook which collects terabits of data every day for their users). These type of data storing may not require fixed schema, avoid join operations and typically scale horizontally.

RDBMS vs NoSQL

RDBMS

- Structured and organized data
- Structured query language (SQL)
- Data and its relationships are stored in separate tables.
- Data Manipulation Language, Data Definition Language
- Tight Consistency

NoSQL

- Stands for Not Only SQL
- No declarative query language
- No predefined schema
- Key-Value pair storage, Column Store, Document Store, Graph databases
- Eventual consistency rather ACID property
- Unstructured and unpredictable data
- CAP Theorem
- Prioritizes high performance, high availability and scalability
- BASE Transaction

Advantages of NoSQL

There are many advantages of working with NoSQL databases such as MongoDB and Cassand main advantages are high scalability and high availability.

1. **High scalability:** NoSQL database use sharding for horizontal scaling. Partitioning of data and placing it on multiple machines in such a way that the order of the data is preserved is sharding. Vertical scaling means adding more resources to the existing machine whereas horizontal scaling means adding more machines to handle the data. Vertical scaling is not that easy to implement but horizontal scaling is easy to implement. Examples of horizontal scaling databases are MongoDB, Cassandra etc. NoSQL can handle huge amount of data because of scalability, as the data grows NoSQL scale itself to handle that data in efficient manner.
2. **High availability:** Auto replication feature in NoSQL databases makes it highly available because in case of any failure data replicates itself to the previous consistent state.

Disadvantages of NoSQL:

NoSQL has the following disadvantages.

1. **Narrow focus:** NoSQL databases have very narrow focus as it is mainly designed for storage but it provides very little functionality. Relational databases are a better choice in the field of Transaction Management than NoSQL.
2. **Open-source:** NoSQL is open-source database. There is no reliable standard for NoSQL yet. In other words two database systems are likely to be unequal.
3. **Management challenge:** The purpose of big data tools is to make management of a large amount of data as simple as possible. But it is not so easy. Data management in NoSQL is much more complex than a relational database. NoSQL, in particular, has a reputation for being challenging to install and even more hectic to manage on a daily basis.
4. **GUI is not available:** GUI mode tools to access the database is not flexibly available in the market.
5. **Backup:** Backup is a great weak point for some NoSQL databases like MongoDB. MongoDB has no approach for the backup of data in a consistent manner.
6. **Large document size:** Some database systems like MongoDB and CouchDB store data in JSON format. Which means that documents are quite large (BigData, network bandwidth, speed), and having descriptive key names actually hurts, since they increase the document size.

Types of NoSQL Database

Types of NoSQL databases and the name of the databases system that falls in that category are:

1. MongoDB falls in the category of NoSQL document based database.
2. **Key value store:** Memcached, Redis, Coherence
3. **Tabular:** Hbase, Big Table, Accumulo
4. **Document based:** MongoDB, CouchDB, Cloudant

When Should NoSQL be Used

1. When huge amount of data need to be stored and retrieved .
2. The relationship between the data you store is not that important
3. The data changing over time and is not structured.
4. Support of Constraints and Joins is not required at database level.
5. The data is growing continuously and you need to scale the database regular to handle the data.

Types of NoSQL Databases

There are mainly four categories of NoSQL databases. Each of these categories has its unique attributes and limitations. No specific database is better to solve all problems. You should select a database based on your product needs.

- Key-value Pair Based
- Column-oriented Graph
- Graphs based
- Document-oriented

Key Value Pair Based

Data is stored in key/value pairs. It is designed in such a way to handle lots of data and heavy load.

Key-value pair storage databases store data as a hash table where each key is unique, and the value can be a JSON, BLOB(Binary Large Objects), string, etc.

Column-based

Column-oriented databases work on columns and are based on BigTable paper by Google. Every column is treated separately. Values of single column databases are stored contiguously.

They deliver high performance on aggregation queries like SUM, COUNT, AVG, MIN etc. as the data is readily available in a column.

Column-based NoSQL databases are widely used to manage data warehouses, business intelligence, CRM, Library card catalogs,

HBase, Cassandra, HBase, Hypertable are examples of column based database.

Document-Oriented

Document-Oriented NoSQL DB stores and retrieves data as a key value pair but the value part is stored as a document. The document is stored in JSON or XML formats. The value is understood by the DB and can be queried.

The document type is mostly used for CMS systems, blogging platforms, real-time analytics & e-commerce applications. It should not use for complex transactions which require multiple operations or queries against varying aggregate structures.

Graph-Based

A graph type database stores entities as well the relations amongst those entities. The entity is stored as a node with the relationship as edges. An edge gives a relationship between nodes. Every node and edge has a unique identifier.

Compared to a relational database where tables are loosely connected, a Graph database is a multi-relational in nature. Traversing relationship is fast as they are already captured into the DB, and there is no need to calculate them.

Graph base database mostly used for social networks, logistics, spatial data.

Neo4J, Infinite Graph, OrientDB, FlockDB are some popular graph-based databases.

7.2 USAGES

Use of NoSQL in Industry

1. *Session Store*

- Managing session data using relational database is very difficult, especially in case where applications are grown very much.
- In such cases the right approach is to use a global session store, which manages session information for every user who visits the site.
- NOSQL is suitable for storing such web application session information very is large in size.
- Since the session data is unstructured in form, so it is easy to store it in schema less documents rather than in relation database record.

2. *User Profile Store*

- To enable online transactions, user preferences, authentication of user and more, it is required to store the user profile by web and mobile application.
- In recent time users of web and mobile application are grown very rapidly. The relational database could not handle such large volume of user profile data which growing rapidly, as it is limited to single server.
- Using NOSQL capacity can be easily increased by adding server, which makes scaling cost effective

3. *Content and Metadata Store*

- Many companies like publication houses require a place where they can store large amount of data, which include articles, digital content and e-books, in order to merge various tools for learning in single platform
- The applications which are content based, for such application metadata is very frequently accessed data which need less response times.
- For building applications based on content, use of NoSQL provide flexibility in faster access to data and to store different types of contents

4. *Mobile Applications*

- Since the smartphone users are increasing very rapidly, mobile applications face problems related to growth and volume.
- Using NoSQL database mobile application development can be started with small size and can be easily expanded as the number of user increases, which is very difficult if you consider relational databases.
- Since NoSQL database store the data in schema-less for the application developer can update the apps without having to do major modification in database.
- The mobile app companies like Kobo and Playtika, uses NOSQL and serving millions of users across the world.

5. *Third-Party Data Aggregation*

- Frequently a business require to access data produced by third party. For instance, a consumer packaged goods company may require to get sales data from stores as well as shopper's purchase history.

- In such scenarios, NoSQL databases are suitable, since NoSQL databases can manage huge amount of data which is generating at high speed from various data sources.

6. *Internet of Things*

- Today, billions of devices are connected to internet, such as smartphones, tablets, home appliances, systems installed in hospitals, cars and warehouses. For such devices large volume and variety of data is generated and keep on generating.
- Relational databases are unable to store such data. The NOSQL permits organizations to expand concurrent access to data from billions of devices and systems which are connected, store huge amount of data and meet the required performance.

7. *E-Commerce*

- E-commerce companies use NoSQL for store huge volume of data and large amount of request from user.

8. *Social Gaming*

- Data-intensive applications such as social games which can grow users to millions. Such a growth in number of users as well as amount of data requires a database system which can store such data and can be scaled to incorporate number of growing users NOSQL is suitable for such applications.
- NOSQL has been used by some of the mobile gaming companies like, electronic arts, zynga and tencent.

9. *Ad Targeting*

- Displaying ads or offers on the current web page is a decision with direct income To determine what group of users to target, on web page where to display ads, the platforms gathers behavioral and demographic characteristics of users.
- A NoSQL database enables ad companies to track user details and also place the very quickly and increases the probability of clicks.
- AOL, Mediamind and PayPal are some of the ad targeting companies which uses NoSQL

7.3 APPLICATION

Often people purchase a particular platform because of the apps that run on it. Many NoSQL-based applications fall into the app category. These applications could not have become a reality using existing relational database technologies.

Facebook messaging platform

Apache Cassandra was created by Facebook to power their Inbox. It did this for a number of years. Cassandra worked by doing the following:

- Cassandra indexed users' messages and the terms (words, and so on) in the messages and drove a search over all the content in those messages. The user ID was the primary key. Each term became a super column, and the message IDs was the column names.
- Cassandra provided the ability to list all messages sent to and from a particular user. Here the user id was the primary key, the recipient IDs were the super columns, and the message IDs were the column names.

The original Facebook **Cassandra paper** is annotated with recent information and is maintained by DataStax, the commercial company promoting Cassandra today.

Amazon DynamoDB

Amazon originally published the Dynamo paper, thereby launching the concept of NoSQL key-value stores. Since then, Amazon has created a separate database called **DynamoDB** as a service offered on the Amazon Web Services marketplace site.

Although DynamoDB gets its name from the original Dynamo, it has a different approach: DynamoDB provides worldwide synchronous replication in order to guarantee consistency and durability essential in enterprise applications.

With DynamoDB, you pay only for the hourly throughput capacity you use, as you use it, rather than for the amount of data you store, which is an interesting model that new application developers will find appealing. You also get as of writing a 'free tier' option that includes 25GB of storage and a number of write and read capacity units.

Google Mail

Google's Bigtable was created to provide wide-column storage for a range of Google's applications, including Orkut, Google Earth, web indexing, Google Maps, Google Books, YouTube, blogger.com, Google Code and **Google Mail.**

Bigtable clones provide index lookup tables for very large sets of information

LinkedIn

LinkedIn has used Hadoop to churn information about relationships overnight and to push the latest graph information to the Voldemort key-value NoSQL store for query the next day. In this way, LinkedIn maintained a rolling view of all data in the service.

BBC iPlayer online media catalog

The British Broadcasting Corporation has an online service to provide UK citizens with a free catchup service called the iPlayer for BBC television and radio shows.

The information for episodes, series, and brands is updated by a different team from that responsible for scheduling episodes for TV.

The BBC moved multiple MySQL systems to a single MarkLogic Server 6 repository to provide access to program metadata. This operation included creating a data services API called **Nitro and** embedding it in MarkLogic Server.

Nitro now powers an increasing number of BBC services. Nitro started by replacing functionality in iPlayer to help stabilize the performance of that platform. In the future, Nitro will include feeds to partner organizations and have a public-facing API.

BBC Sport and Olympics platforms

In 2011, the BBC realized that its journalists were spending a lot of time deciding where to publish stories on the BBC Sport website. This cost a lot of time and money and stories weren't consistently available to users in different areas of the sports website.

The BBC created an entirely new type of solution called Dynamic Semantic Publishing (DSP) to automate much of this process. By using a combination of MarkLogic Server 6 (the version without a triple store) and Ontotext's GraphDB (formerly BigOWLIM), the BBC was able to suggest topics on stories to their journalists.

This approach also allowed the BBC to use the relationships inherent in the subjects mentioned in the stories to determine where to publish the data, rather than rely on the journalists.

By going to the BBC Sport home page and clicking on the link for the England football team, you see not only stories about the England football team, but also any players who happen to play for England, or stories about the players' spouses, even though they aren't explicitly mentioning the England football team in the stories.

HealthCare.gov

Healthcare.gov has been called the most complex IT system implementation of all time. Building it required several systems, with the most visible one being the HealthCare.gov marketplace.

Behind the scenes, many other systems provide supporting functions, including stores for information from other agencies, such as IRS data and information about coverage that states already offer to their residents. Also, insurers submit the policies they want to offer to citiznes on the federal marketplace website.

Communication between the various systems also requires storage of messages for safety (so they're not lost) and later delivery. Although HealthCare.gov provides coverage to citizens in thirty-four states, the back-end systems support all fifty states through the database, and feeds the states' own marketplaces.

The Centers for Medicare & Medicaid Services (CMS) selected MarkLogic to provide the back-end database for all these systems' data. MarkLogic Server stores an anonymized version of all the XML content flowing between these systems and provides the capability to match requirements of citizens with insurance coverage available.

The subsystem that tracks and analyzes all message traffic in real time has proven to be the most visible success of a NoSQL system that affects citizens lives directly. Although the project experienced public difficulties, the level of complexity that was handled and the now successful rollout to more than seven million newly covered Americans resulted in a great success.

Secure Information Sharing

In many situations, you need to provide access to information while also maintaining its security. Here are several examples:

- A book publisher providing access to summaries so that you can verify the relevance of a book before purchase, but only view the full book after purchase
- A multiagency social care application with different access rights for child protection officers, medical staff, educators, and law enforcement agencies
- An intelligence-sharing application where high-level information on an intelligence report is shared for discovery, but where all access must be applied for and granted on a case-by-case basis

These situations share a common approach: they require security set at the record level as a minimum, so that you can show or hide a record to different users of the system.

Also, to provide secure access to specific sections within a record, you will also require either denormalization, or cell-based, or label-based access control (LBAC). LBAC enforces record security based on the content of that record rather than explicit permissions set for that record.

In these scenarios, NoSQL databases that support record or cell/element/triple level security, such as Accumulo, MarkLogic Server, and AllegroGraph, are good options.

Citizen Engagement

Governments use NoSQL databases to empower citizens with information about how their country is governed. A good example is Fairfax County in Virginia, which uses MarkLogic Server to provide geospatial information through an online browse and search interface to government agencies and residents. The service covers a range of information — for example, **geographic points in the county and police-related events.**

In the UK, the award-winning **legislation.gov.uk website** provides information on UK laws dating back to more than one thousand years! If you want to know the laws about theft of property in Wales in 1542, just visit the website!

You can also find laws currently being debated by Parliament, and upcoming legal clause activations are available as annotations for current legislation. This service provides citizens as well as lawmakers with a very rich reference on legal matters throughout the UK.

EXERCISES

1. What is NO-SQL?
2. Give advantages and disadvantages of NO-SQL.
3. Compare NO-SQL and RDBMS.
4. Explain the types of NO-SQL databases.
5. Give the uses of NO-SQL.
6. Explain any 5 applications of NO-SQL.

❑❑❑