When to Use MongoDB Rather than MySQL

MySQL and MongoDB represent two sides of an argument that has been raging recently concerning data storage – the tried and tested relational database vs. non-relational or NoSQL database. They are both open-source products distributed under a version of the GNU GPL, and both are also available as commercial versions offering many more features and corporate support.

Overview

We all know that in the database technology world, it comes down to two main database types – SQL (relational) and NoSQL (non-relational). The differences between them are rooted in the way they are designed, which data types they support, and how they store them.

Relational Database Management System (RDBMS)

RDBMS Database is a relational database. It is the standard language for relational database management systems. Data is stored in the form of rows and columns in RDBMS. The relations among tables are also stored in the form of the table SQL (Structured query Language) is a programming language used to perform tasks such as update data on a database, or to retrieve data from a database. Some common relational database management systems that use SQL are: Oracle, Sybase, Microsoft SQL Server, Access, etc.

Features of RDBMS

1. SQL databases are table based databases
2. Data store in rows and columns
3. Each row contains a unique instance of data for the categories defined by the columns.
4. Provide facility primary key, to uniquely identify the rows.

Limitations for SQL database

Scalability: Users have to scale relational database on powerful servers that are expensive and difficult to handle. To scale relational database, it has to be distributed on to multiple servers. Handling tables across different servers is difficult.

Complexity: In SQL server's data has to fit into tables anyhow. If your data doesn't fit into tables, then you need to design your database structure that will be complex and again difficult to handle.

Popular SQL databases and RDBMS's

- **MySQL**—the most popular open-source database, excellent for Enterprise Applications.
- **Oracle**—an object-relational DBMS written in the C++ language. If you have the budget, this is a full-service option with great customer service and reliability. Oracle has also released an Oracle NoSQL database.
- **IMB DB2**—a family of database server products from IBM that are built to handle advanced "big data" analytics.
- **Sybase**—a relational model database server product for businesses primarily used on the Unix OS, which was the first enterprise-level DBMS for Linux.
- **MS SQL Server**—a Microsoft-developed RDBMS for enterprise-level databases that supports both SQL and NoSQL architectures.

NoSQL

NoSQL commonly referred to as “Not Only SQL”. With NoSQL, unstructured, schema less data can be stored in multiple collections and nodes and it does not require fixed table sachems, it supports limited join queries, and we scale it horizontally.

Benefits of NoSQL

highly and easily scalable

Relational database or RDBMS databases are vertically Scalable When load increase on RDBMS database then we scale database by increasing server hardware power, need to by expensive and bigger servers and NoSQL databases are designed to expand horizontally and in Horizontal scaling means that you scale by adding more machines into your pool of resources.

Maintaining NoSQL Servers is Less Expensive

Maintaining high-end RDBMS systems is expensive and need trained manpower for database management but NoSQL databases require less management. it supports many Features like automatic repair, easier data distribution, and simpler data models make administration and tuning requirements lesser in NoSQL.

Lesser Server Cost and Open-Source

NoSQL databases are cheap and open source. NoSQL database implementation is easy and typically uses cheap servers to manage the exploding data and transaction while RDBMS databases are expensive and it uses big servers and storage systems. So the storing and processing data cost per gigabyte in the case of NoSQL can be many times lesser than the cost of RDBMS.

No Schema or Fixed Data model

NoSQL database is schema less so Data can be inserted in a NoSQL database without any predefined schema. So the format or data model can be changed any time, without application disruption. And change management is a big headache in SQL.

Support Integrated Caching

NoSQL database support caching in system memory so it increases data output performance and SQL database where this has to be done using separate infrastructure.

Limitations & disadvantage of NoSQL

1. NoSQL database is Open Source and Open Source at its greatest strength but at the same time its greatest weakness because there are not many defined standards for NoSQL databases, so no two NoSQL databases are equal
2. No Stored Procedures in mongo dB (NoSQL database).

Popular NoSQL Databases

- **MongoDB—**the most popular NoSQL system, especially among startups. A document-oriented database with JSON-like documents in dynamic schemas.
- **Apache's Couch DB—**a true DB for the web, it uses the JSON data exchange format to store its documents; JavaScript for indexing, combining and transforming documents; and, HTTP for its API.

- **HBase—**another Apache project, developed as a part of Hadoop, this open-source, non-relational "column store" NoSQL DB is written in Java, and provides Big Table-like capabilities.

SQL vs NoSQL: MySQL vs MongoDB

Now that we've established the key structural differences between SQL and NoSQL databases, let's delve into the key functional differences between the two, looking specifically at MySQL and MongoDB as examples.

What is MySQL?

MySQL is a popular open-source relational database management system (RDBMS) that is developed, distributed and supported by Oracle Corporation. Like other relational systems, MySQL stores data in tables and uses structured query language (SQL) for database access. In MySQL, you pre-define your database schema based on your requirements and set up rules to govern the relationships between fields in your tables. Any changes in schema necessitates a migration procedure that can take the database offline or significantly reduce application performance.

MySQL: The SQL Relational Database

The following are some MySQL benefits and strengths:

- **Maturity**: MySQL is an extremely established database, meaning that there's a huge community, extensive testing and quite a bit of stability.
- **Compatibility:** MySQL is available for all major platforms, including Linux, Windows, Mac, BSD and Solaris. It also has connectors to languages like

Node.js, Ruby, C#, C++, Java, Perl, Python and PHP, meaning that it's not limited to SQL query language.

- **Cost-effective:** The database is open source and free.
- **Replicable:** The MySQL database can be replicated across multiple nodes, meaning that the workload can be reduced and the scalability and availability of the application can be increased.
- **Data Sharding:** While sharding cannot be done on most SQL databases, it can be done on MySQL servers. This is both cost-effective and good for business.

What is MongoDB?

MongoDB is an open-source, non-relational database developed by MongoDB, Inc. MongoDB stores data as documents in a binary representation called BSON (Binary JSON). Related information is stored together for fast query access through the MongoDB query language. Fields can vary from document to document; there is no need to declare the structure of documents to the system – documents are self-describing. If a new field needs to be added to a document, then the field can be created without affecting all other documents in the collection, without updating a central system catalog, and without taking the system offline.

MongoDB: The NoSQL Non-Relational Database

The following are some of MongoDB benefits and strengths:

- **Dynamic schema:** As mentioned, this gives you flexibility to change your data schema without modifying any of your existing data.
- **Scalability:** MongoDB is horizontally scalable, which helps reduce the workload and scale your business with ease.

- **Manageability:** The database doesn't require a database administrator. Since it is fairly user-friendly in this way, it can be used by both developers and administrators.
- **Speed:** It's high-performing for simple queries.
- **Flexibility:** You can add new columns or fields on MongoDB without affecting existing rows or application performance.

Reasons to Use a SQL Database

Not every database fits every business need. That's why **many companies rely on both relational and non-relational databases for different tasks**. Although NoSQL databases have gained popularity for their speed and scalability, there are still situations in which a highly structured SQL database might be preferable. Two reasons why you might consider a SQL database are:

1.You need ACID compliancy (Atomicity, Consistency, Isolation, Durability). ACID compliancy reduces anomalies and protects the integrity of your database. It does this by defining exactly how transactions interact with the database, which is not the case with NoSQL databases, which have a primary goal of flexibility and speed, rather than 100% data integrity.

2.Your data is structured and unchanging. If your business is not growing exponentially, there may be no reason to use a system designed to support a variety of data types and high traffic volume.

Reasons to Use a NoSQL Database

To prevent the database from becoming a system-wide bottleneck, especially in high volume environments, NoSQL databases perform in a way that relational databases cannot.

The following features are driving the popularity of NoSQL databases like MongoDB, Couch DB, Cassandra, and HBase:

1. Storing large volumes of data without structure. A NoSQL database doesn’t limit storable data types. Plus, you can add new types as business needs change.

2. Using cloud computing and storage. Cloud-based storage is a great solution, but it requires data to be easily spread across multiple servers for scaling. Using affordable hardware on-site for testing and then for production in the cloud is what NoSQL databases are designed for.

3. Rapid development. If you are developing using modern agile methodologies, a relational database will slow you down. A NoSQL database doesn’t require the level of preparation typically needed for relational databases.

When to use?

MongoDB	MySQL
HIGH AVAILABILITY When you need high availability of data with automatic, fast and instant data recovery.	**LOW-MAINTENANCE** If you're just starting and your database is not going to scale much, MySQL will help you in easy and low-maintenance setup.
IN-BUILT SHARDING In future, if you're going to grow big as MongoDB has in-built sharding solution.	**LIMITED BUDGET** If you want high performance on a limited budget.
UNSTABLE SCHEMA If you have an unstable schema and you want to reduce your schema migration cost.	**FIXED SCHEMA** If you've fixed schema and data structure isn't going to change over the time like WikiPedia
NO DBA If you don't have a Database Administrator (but you'll have to hire one if you're going to go BIG).	**HIGH TRANSACTION** If high transaction rate is going to be your requirement (like BBC around 30,000 inserts/minute, 4000 selects/hour)
CLOUD COMPUTING If most of your services are cloud-based, MongoDB is best suitable for you	**DATA SECURITY** If data security is your top priority, MySQL is the most secure DBMS.

Conclusion

To answer the main question: “when to use MongoDB instead of MySQL?” you need to take into account your project requirements and further goals. MySQL is well-recognized for its high performance, flexibility, reliable data protection, high availability, and management ease. Proper data indexing can solve the issue with performance, facilitate interaction and ensure robustness. But if your “data is unstructured and complex, or if you can’t pre-define your schema, you’d better opt for MongoDB.” And what is more, if you need to handle a large volume of data and store it as documents—MongoDB will help you to meet the challenges.

R Sai Krishna

Application Developer