

Research into Databases

Research into best fit databases for Microservices



Name: Victoria C. A. Fong

Student number: 488384

Introduction

This document is based on research into databases that are to be best used in each or all microservices.

This research touches down on debating between relational and non-SQL databases for using in microservices, looking into some of the most common databases used for microservices and lastly, seeing which database or databases is best to use for the services.

Table of Contents

Introduction	2
SQL or NoSQL database	4
Conclusion.....	4
Best know databases to use in microservices.....	5
MySQL.....	Error! Bookmark not defined.
Redis	5
MongoDB	Error! Bookmark not defined.
Database Conclusion (Schema)	Error! Bookmark not defined.
Conclusion.....	7
Sources	8

SQL or NoSQL database

After doing research into debating between choosing SQL or Non-SQL database to use, I have come to a reasoning that I will be using NoSQL database(s) for the project.

The reasons for choosing a NoSQL database are:

1. The Development progress with using a NoSQL database is much faster than using a SQL database.
 - a. It allows developers to be in control of the structure of the data where quick iterations and frequent code pushes can be made.
2. The data that is more easily handles with a NoSQL database.
 - a. The data that is stored in the databases are similarly in a form of objects used in applications. This reduces the need for translating the form from the data stored to the form the data handles in the code.
3. NoSQL handles bigger data in comparison to SQL database.
 - a. NoSQL was designed to handle big data as part of their fundamental architecture. When applications (the microservice) grow bigger with more users, the database must handle this large amount of data.
 - b. NoSQL does not require engineering to handle web-scale applications in comparison to SQL databases. Further, NoSQL databases are often based on scale-out strategy. For this reason, it makes it cheaper to scale out large amount of data.
 - c.
4. New paradigms designed for applications can be more easily supported.

Conclusion

For the reason that the project is based on a large audience and fast development where scaling, frequent code update and implementations are needs, the NoSQL databases are better to use for the microservices.

Best know databases to use in microservices

MongoDB

One of the most popular NoSQL databases in the world is MongoDB. It's a document-based, general-purpose database. The data that is stored is in the format of JSON-like documents in MongoDB.

Advantages

1. **Flexibility:** due to the database having no schemas, it allows you to store any type of data.
2. **Speed:** one of mongo's most important advantages is its speed. The database can index documents which is faster access to documents.
3. **Scalability:** MongoDB provides horizontal scaling. Here, you can divide big datasets over several servers.
4. **Querying capabilities.** MongoDB has powerful querying capabilities with its Ad-hoc queries.

Disadvantages

The majority of disadvantages lie for companies wanting to join other databases. When implementing this into a microservice and using something like a message broker to handle this, are these disadvantages to nothing.

Redis

Redis is an open-source NoSQL database. It is a data structure store that runs in memory. It can be used for a lot of purposes such as cache, a message broker, or a database.

Advantages

1. **Excellent for caching.**
2. **Advanced data structures:** supports a broad range of data structures.
3. **Flexibility:** allowing you to store key-value pairs up to 512 MB.
4. **Scalability:** Redis databases can be quickly scaled without experiencing any downtime or loss in performance.

Disadvantages

1. Large-scale cloud deployment can be hard.
2. "Role-Based-Account-Control" cannot be implemented.
3. Lacks in-built encryption.

Cassandra

Cassandra is an open-source NoSQL database. Cassandra popularity by developers is for its ability to handle large volumes of data and scalability.

Advantages

1. **Data storage flexibility:** the database allows to store structured, semi-structured and unstructured data.
2. **Performance:** the database is most popular for many write requests, due to it being able to handle them quickly and efficiently.
3. **Scalability:** the database can scale linear to assure rapid responsiveness.

Disadvantages

The READ operations can be slow where there is a risk of delay when there are too many READ requests sent out. Further, it has a limit of number of queries available for retrieval of data.

Conclusion

From researching these three databases, I can conclude that MongoDB is the winner for working with the microservices. This is because the database can handle large amount of data, scalable, speed and flexibility. Redis was mainly not chosen for its difficulty of scaling on cloud deployment and lack in in-built encryption. For Cassandra, there was much positive except the risk of there being a delay for too many READ requests. Each microservice majority of requests are READ requests where the consequences of speed were not an option.

Sources

<https://www.mongodb.com/nosql-explained/when-to-use-nosql>

<https://cloudinfrastructureservices.co.uk/cassandra-vs-mongodb-vs-redis-vs-mysql-vs-postgresql-pros-and-cons/>