



RESEARCH DOCUMENT

Database Choice



Trifonov,Dobri D.V.

Contents

Which is the better suited database for our application	2
What are the structural differences?	2
SQL (Column-Oriented)	2
NoSQL	2
SQL (Row-Oriented)	2
Advantages and Disadvantage	2
ClickHouse	2
MongoDB	2
Postgres	2
Conclusion	3
References	4

Which is the better suited database for our application

What are the structural differences?

SQL (Column-Oriented) – A column-orientated database is largely optimized for reading enormous amounts of data. In them you can only read the data columns you need without going through the entire database like you would with a standard row-oriented database. This results in database performance increase and data that is easier to compress. The data is stored such that each row of a column will be next to other rows from that same column.

NoSQL - NoSQL databases on the other hand do not have keys that create relationships between data tables. Because of that we lose the convenience of SQL's ability to join tables together using simple queries. The data in a NoSQL database is unstructured, meaning that every data row does not need to match the set columns. This allows for a slightly better flexibility because we can create data without structure, then add to it as we go on with the project. Data can be stored in plethora of ways such as Document, Column, Graph or even Key-Value store.

SQL (Row-Oriented) - All SQL database share the same SQL language. Some database have additional features on top of the default ones, but they more or less have the same common functionality. What keeps them all so similar is the structure. The data is laid out in a table with columns and rows, the data is connected with keys (Primary, Foreign Keys, etc). This makes the SQL data structure very relational and have a very rigid structure.

Advantages and Disadvantage

ClickHouse – ClickHouse is a Column-orientated database, meaning it was designed to read through massive databases in the fastest way possible. Column oriented databases store information vertically, meaning the individual values are stored contiguously by column. The advantage of that is that reads are much more efficient because a lower volume of data is loaded due to reading only the relevant data instead of the whole record. Clickhouse is typically involved in highly complex queries over all data (possibly petabytes). Assuming a table continues for millions of rows and we wanted all values from one column in a column oriented database we can just take that column instead of accessing each row and reading all other available columns.

MongoDB – MongoDB is a NoSQL database, meaning it was designed to handle a large number of requests. Where it particularly shines is its fast write speeds. This is a result of it being a document database that does not have many strict checks. This makes MongoDB the default choice for applications that need to ingest large amounts of data, making it ideal for applications that write in real time. MongoDB is also horizontally scalable database that can be expanded by adding more nodes to it. It is an ideal choice for application that have large amount of users that are attempting to connect at the same time. It is scalable and on a large enough scale replica databases could be run in parallel with the original ones.

Postgres – Postgres is generally used when we have planned out, structured data and we need to keep that data in strict check. The set columns are unchangeable, tables relate to each other using different keys allowing for easy joins between tables. This allows for the creation of quicker and more complex queries. In addition to that Postgres also examine higher read speeds than most NoSQL databases and better write speeds than columnar-store databases. This makes Postgres ideal for applications that work with complex data and flexible workflows.

Conclusion

Clickhouse largely used for analysing large data volumes. While we do plan to read a lot of information from our database it is not nearly enough to justify using Clickhouse. We decided that it would be better to use a database that would be more optimized towards handling writing operations. MongoDB would be our preferred choice based on its great writing performance, but because of the necessity of our application to use relational database features, we have instead opted for Postgres.

Currently we are using both MongoDB and Postgres, from which the former is used as a raw data backup, while the latter contains all formatted, enriched and user information.

References

C. (n.d.). *Overview* / *ClickHouse Documentation*. ClickHouse.

<https://clickhouse.com/docs/en/>

Wikipedia contributors. (2021, October 13). *Column-oriented DBMS*. Wikipedia.

https://en.wikipedia.org/wiki/Column-oriented_DBMS#Benefits

What is MongoDB Atlas? — *MongoDB Atlas*. (n.d.). MongoDB.

[https://docs.atlas.mongodb.com/#:%7E:text=MongoDB%20Atlas%20is%20a%20mul](https://docs.atlas.mongodb.com/#:%7E:text=MongoDB%20Atlas%20is%20a%20multi,cloud%20providers%20of%20your%20choice.)
[ti,cloud%20providers%20of%20your%20choice.](https://docs.atlas.mongodb.com/#:%7E:text=MongoDB%20Atlas%20is%20a%20multi,cloud%20providers%20of%20your%20choice.)

MongoDB vs. Postgres Benchmarks / *Álvaro Hernández (OnGres)*. (2020, April 14).

YouTube. <https://www.youtube.com/watch?v=-AIjKrWi0x0&t=2091s>

Taylor, D. (2021, November 1). *What is MongoDB? Introduction, Architecture, Features & Example*. Guru99. <https://www.guru99.com/what-is-mongodb.html>