

Research Paper 1

HUGGO BASTOS | P&P II
E: HBASTOS@STUDENT.FULLSAIL.COM

THESIS 1

RELATIONAL VS NOSQL

One main difference between Relational databases and NoSQL databases, also known as non-relational databases, is that relational databases represent data in forms of tables while NoSQL databases are a key-value pair and do not have standard schema definitions which it needs to adhered to. Because of this NoSQL databases are able to handle unstructured data while relational databases can not. For complex queries SQL (relational) databases are much better and faster overall than NoSQL databases. NoSQL databases however, are better for hierarchal data storage and can handle much larger data sets much better than relational databases.

THREE NOSQL TWITTER FEATURES

1. Large amounts of data storage
2. People Search
3. Trying to read large amounts of data at a time

TWITTER FEATURES HOW TO

Large amounts of data storage isn't a "feature" of Twitter that twitter directly has but is something that they use NoSQL for. According to Kevin Weil, (Analytics lead at Twitter) Twitter has to store more data than it can write to a single hard drive. He states that MySQL is efficient at keeping up with the size that Twitter needs so they store their data in clusters which is powered through

Hadoop (a software that runs on NoSQL).

Twitter uses HBase for its people searching (a NoSQL database). It is much better for handling large amounts of data and searching through it if the data is unstructured. It is designed for low-latency and data-mutability (changing data).

Twitter is constantly being updated and every individuals' Twitter feed is unique to them. Twitter constantly has to get your feed and read large amounts of data from many different locations at any given time.

TWITTER FEATURES PRO/CON

1. Large amounts of data storage

One of the pros to using NoSQL for data storage is that it can store and much more easily handle large amounts of data. Storing a lot of data however means that there is a lot going on and you may even have to access the data as well constantly (loading someone's twitter feed, searching for a user, etc.). For this a Relational database is much faster and more efficient than using NoSQL.

2. People search

The Pros of using NoSQL for this is that that data can be left unorganized/structured and NoSQL can still work with it. One of the cons of this being done in NoSQL is that for higher amount of transactions constantly happening Relational databases are much faster and can perform these actions quicker.

3. Reading large amounts of data

In one of the links below there is a video of Kevin Weil doing a presentation of how

Twitter uses NoSQL. They primarily use Hadoop which solves their problems for reading and writing large amounts of data but does not do it efficiently. The pro would be that currently they are able to do everything they need by using NoSQL but it isn't optimal or efficient (he mentions that it does not do it fast enough/could be faster).

One Relational Facebook Feature

ONE RELATIONAL FACEBOOK FEATURE

1. Storing structured data.

FACEBOOK FEATURE HOW TO

1. Storing structured data

Facebook uses MySQL for structured data storage. Wall posts, user information, people's timelines and many other things are all done with MySQL. When working with structured data or large amounts of data (which Facebook is constantly doing) using a relational database insures speeds being much higher (which is very important when you get to a company that size of Facebook).

Facebook Feature Pros/Cons

FACEBOOK FEATURE PRO/CONS

One of the cons to using a relational database, and MySQL for Facebook in particular, is that it can only work with structured data. A pro to using MySQL for Facebook is that if working with structured data SQL is a very fast query language and it can run through operations much faster than if it were to use a non-relational database.

FIVE NOSQL DATABASES

1. MongoDB
2. Cassandra
3. Redis
4. HBase
5. Neo4j

FIVE NOSQL DATABASES PRO/CONS

MongoDB is a type of a non-relational database that is very fast and easy to use. It supports JSON and is good with high performance for simple queries. MongoDB can also work both with structured and unstructured data however takes long to set up. Another con is that It does not use SQL however as a query language.

Cassandra is both a massive and an open source non-relational database. It has linear scaling and works very fast at any level (even when working with large amounts of data). It can handle large amounts of information without a problem and can manage any type of data. It also has constant uptime. One of the downsides is that it has unreliable performance as many of its different jobs that are done are not scheduled by the user. This makes troubleshooting hard and it runs off of CQL which gets mistaken easily for SQL, in turn making troubleshooting very difficult at times.

WEATHER APP (TWO NOSQL SOLUTIONS)

A weather app would need to pull a lot of data from a lot of different places. Working with a weather app in NoSQL could be done by using a key-value pair system where you tie the current temperature at the coordinate to place. If someone were to search for example on the weather app at Orlando, FL, the app would connect to a database that gets the temperature that it is where Orlando is located and ties the two together. Cassandra would work well because it works very fast with any levels of data (which isn't always true with non-relational databases).

REFERENCES

SECTION

A Brief Introduction to Apache Cassandra. (2016, May 02). Retrieved from <https://academy.datastax.com/resources/brief-introduction-apache-cassandra>

Arsenault, C. (2017, April 20). The Pros and Cons of 8 Popular Databases. Retrieved from <https://www.keycdn.com/blog/popular-databases/>

Finley, K. (n.d.). How Twitter Uses NoSQL. Retrieved from <https://readwrite.com/2011/01/02/how-twitter-uses-nosql/>

Gajani, A. (2017, June 30). The key differences between SQL and NoSQL DBs. Retrieved from <http://www.monitis.com/blog/cc-in-review-the-key-differences-between-sql-and-nosql-dbs/>

Issac, L. P. (2014, January 14). SQL vs NoSQL Database Differences Explained with few Example DB. Retrieved from https://www.thegeekstuff.com/2014/01/sql-vs-nosql-db/?utm_source=tuicool

Lo, F. (n.d.). What is Hadoop and NoSQL? Retrieved from <https://datajobs.com/what-is-hadoop-and-nosql>

Mayo, M. (n.d.). KDnuggets. Retrieved from <https://www.kdnuggets.com/2016/06/top-nosql-database-engines.html>

NoSQL at Twitter. (n.d.). Retrieved from <https://www.infoq.com/presentations/NoSQL-at-Twitter>

HUGGO BASTOS | DVP 2
E: HBASTOS@STUDENT.FULLSAIL.COM