8

References .............................................................................................................................

7

Conns Riak Research...............................................................................................................

6

Conns neo4j Research............................................................................................................

5

Pauls Redis Research...........................................................................................................

4

Pauls Riak research ................................................................................................................

2

Shared 1&2...................................................................................................................................

**CONTENTS**

(**Shared 1 & 2)**

NoSQL is a non-relational Data Management system. It is a great substitute for SQL as it boasts numerous features which are exclusive to it. NoSQL compared to SQL, does not need as much developed hardware compared to the former to run a server as it can allocate memory which inadvertently does not strain your computer. Instead of just allocating all the memory to one server it can distribute that strain to multiple servers.

In addition to that, NoSQL has various types of formats to view and store your data. It can come in document-based or even graph-based databases. Having flexibility in your documents can make each one unique and better to display.

NoSQL is also great for big data databases as existing SQL databases were unable to withstand the increasing processing and memory requirements for bigger databases. Instead of being vertically stable, they are horizontally scalable which allows them to run bigger projects.

Instead of following the ACID property, the NoSQL database follows the Brewers CAP theory which stands for consistency, availability, and partition tolerance. NoSQL utilizes the standard of possible consistency. This implies that assuming there are no new updates for a specific information thing for a specific timeframe, in the long run all get to it will return the last refreshed worth. While this methodology incredibly builds access time and adaptability, it might prompt information misfortune the seriousness of the issue relies upon data set server support and the nature of utilization code.

NoSQL is also open-source, which means that it does not require expensive licensing fees and can run on standard hardware making it very cost-effective compared to its predecessor.

For such a relatively new database, NoSql has a very strong and thriving community and data sets have a solid local area of designers encompassing them, which means there is a lot of information online and designers which can help you better create and structure your database.

# Although there are numerous advantages of using NoSQL there are some disadvantages of using it compared to SQL. For example, there are not many tools for backup, Albeit some NoSQL information bases like MongoDB give a few apparatuses to reinforce your data, these instruments are not developed enough to guarantee legitimate total information reinforcement arrangement.

There is also a lack of standardization, this implies that there is no real standard that characterizes rules and jobs of NoSQL data. The plan and question dialects of NoSQL information bases shift generally between various NoSQL items significantly more broadly than they do among customary SQL data sets

Lastly, NoSQL puts a versatility and execution first yet with regards to a consistency of the information NoSQL does not take a lot of thought so it makes it unreliable when contrasted with the social data. For example, if you add in a duplicate set of data in NoSQL it will put it in with no issues whereas SQL prevents duplicated data from being displayed.

In conclusion, NoSQL is a great substitute for SQL as its databases are effectively adaptable, adaptable and easy to use as they have no unbending pattern. They are great for applications with no particular outline definitions, for example, content administration frameworks, huge information applications, ongoing investigation, and so on.

**RIAK Research (Paul)**

Riak is a conveyed data set intended to convey the most extreme information accessibility by dispersing information across various servers. However long your Riak server can arrive at one Riak server, it ought to have the option to compose information.

**Scalability:** Via naturally appropriating information across hubs in a bunch, Riak KV yields a close direct exhibition increment as the limit is added. Riak KV consequently reallocates information as hubs are added or eliminated to adjust information across hubs. Riak KV makes it altogether simpler for applications to scale by wiping out manual sharding and by duplicating information to diminish problem areas. Riak KV can deal with a great many keys, petabytes of capacity, a large number of clients, and billions of items. When you need scalability, Riak KV is a great option.

**Recoverability**: Riak KV is a clustered system that can withstand an expansive number of failed scenarios which includes the loss of nodes and keys due to failure of network or hardware. Its survivability is one of its core strengths and reasons to use it compared to other databases.

**Concurrency**: Riak supports concurrent data manipulation. Riak is great at handling large volumes of requests whilst also keeping response times low.

**Acid/Base Properties**: Solid consistency may likewise infer ACID exchanges, in which numerous particular items can be refreshed together and the consequences of the updates are destined to be predictable. This segment portrays the highlights accessible for solid consistency in Riak.

**Transactions:** Riak does not support ACID transactions however it does support durable transactions.

**Redis Research (Paul)**

**Scalability:** Redis can be scaled vertically and horizontally. It implements sharding for horizontal scaling. Although you can implement sharding for horizontal scaling it will become quite expensive.

**Recoverability:** Redis Enterprise permits you to reinforcement a depiction of your data set (across all shards) to one of the significant public distributed storage arrangements (Amazon S3, Azure Blob Storage, or Google Cloud Storage) just as FTP or Swift administrations just in-case of any errors or mishaps when coding your database.

**Concurrency:** Redis supports concurrent data manipulation. It can also provide concurrency at the I/O by using an I/O multiplexing mechanism and an event loop

**Acid/Base Properties:** Redis on the other hand provides partial ACID compliance by configuration because it is single strung and it is fully compliant if applied with appendfsync .

**Transactions:** MULTI, EXEC, DISCARD and WATCH are the establishment of exchanges in Redis. They permit the execution of a gathering of orders in a solitary advance, with two significant assurances. A Redis exchange is entered utilizing the MULTI order.

**Neo4j(Conn)**

**Neo4j transactions**

A DBMS-level exchange is a holder for data set exchanges. An information base exchange is begun when the primary question to a particular data set is given. moved back when the DBMS-level exchange is submitted or moved back.

**Recoverability**

Neo4j has a restore command that will restore the database to a previous saved point

**ACID/BASE Properties**

To completely keep up with data integrity and good transactional behaviour, Neo4j supports the ACID properties: Atomicity, in case any piece of an exchange fails, the database state is left unaltered. Consistency, any exchange will leave the information in a consistent state.

**Concurrency**

Neo4j is an ACID compliant database this is the same as some other ACID agreeable data set. For a write to happen locks need to initially happen on the items affected. One of the exchanges will be run first, lock related hubs and connections, perform said makes or refreshes, and afterward discharge said locks. The second exchange won't happen until the locks have been delivered and the second exchange would then be able to procure said locks.

**Scalability**

Neo4j scales with your data and your business needs, limiting expense and hardware while having great performance across connected databases. Relational databases scale, but with limited connections across tables

**Riak (Conn)**

**Transactions:** Riak doesn't uphold ACID exchanges anyway it upholds durable exchanges.

**Recoverability**

Data on disk can become corrupted by hardware failure, the Riak storage backends are intended to deal with instances of corruption in individual files or entries and can fix them automatically or just overlook the adulterated parts. Any other way, bunches can recuperate from information defilement in generally the same manner that they recuperate from information misfortune.

**ACID/BASE Properties:**

ACID properties and simultaneously giving the advantages that propelled the NoSQL development through a center layer. The proposed center layer utilizes a four stage submit convention to guarantee: the utilization of ongoing information, the utilization of the Pessimistic method to deny others managing information while it is utilized and the information refreshes living in numerous **areas to stay away from the deficiency of information and disillusionment.**

**Concurrency:** Riak upholds simultaneous information control. Riak is incredible at dealing with enormous volumes of solicitations while likewise keeping reaction times low.

**Scalability:** Via naturally circulating information across hubs in a bunch, Riak KV yields a close direct exhibition increment as limit is added. Riak KV naturally rearranges information as hubs are added or eliminated to adjust information across hubs.

References

<https://neo4j.com/docs/cypher-manual/current/introduction/transactions/>

<https://neo4j.com/docs/java-reference/current/transaction-management/>

<https://neo4j.com/docs/java-reference/current/transaction-management/>

https://www.sciencedirect.com/science/article/pii/S1319157815001044

References

Page title: Redis is single-threaded, then how does it do concurrent I/O?

Website name:

Publisher:

URL: <https://stackoverflow.com/questions/10489298/redis-is-single-threaded-then-how-does-it-do-concurrent-i-o>

Access date: 25 November 2021

Page title: SQL vs NoSQL: Which one is better to use? - GeeksforGeeks

Website name: GeeksforGeeks

URL: <https://www.geeksforgeeks.org/sql-vs-nosql-which-one-is-better-to-use/?ref=rp>

Access date: 25 November 2021

Page title: cap-theorem

Website name: Ibm.com

URL: <https://www.ibm.com/cloud/learn/cap-theorem>

Access date: 25 November 2021

Page title: Advantages and Disadvantages of NoSQL Databases - Technology Point

Website name: Technology Point

URL: <https://technologypoint.in/advantages-and-disadvantages-of-nosql-databases/>

Access date: 25 November 2021

Page title: SQL vs NoSQL Comparatives Advantages and Disadvantages

Website name: Bitnine.net

URL: <https://bitnine.net/blog-computing/sql-vs-nosql-comparative-advantages-and-disadvantages/?ckattempt=1>

Access date: 25 November 2021

Page title : Massive Scalability

Website name: Riak

URL: <https://riak.com/products/riak-kv/massive-scalability/index.html?p=10912.html>

Access date: 25 November 2021

Page title: Why Riak KV?

Website name: Docs.riak.com

URL: <https://docs.riak.com/riak/kv/latest/learn/why-riak-kv/index.html>

Access date: 25 November 2021

Page title: Backing Up

Website name: Docs.riak.com

URL: <https://docs.riak.com/riak/kv/latest/using/cluster-operations/backing-up/index.html>

Access date: 25 November 2021

Page title: Redis vs. Riak KV Comparison

Website name: Db-engines.com

URL: <https://db-engines.com/en/system/Redis%3BRiak+KV>

Access date: 25 November 2021

Page title: Redis is single-threaded, then how does it do concurrent I/O?

Website name: Stack Overflow

Access date: 25 November 2021

URL: <https://stackoverflow.com/questions/10489298/redis-is-single-threaded-then-how-does-it-do-concurrent-i-o>

Access date: 25 November 2021

Page title: A middle layer solution to support ACID properties for NoSQL databases

Website name: Science direct

Url: https://www.sciencedirect.com/science/article/pii/S1319157815001044

Access date: 25 November 2021

Page title: Transaction management - Neo4j Java Reference

Website name: Neo4j Graph Database Platform

Url: https://neo4j.com/docs/java-reference/current/transaction-management/

Access date: 25 November 2021