



GREEN UNIVERSITY OF BANGLADESH

COURSE TITLE:DATABASE

COURSE CODE :301

Database Assignment

TOPIC : NEWSQL

Submitted By :

Md Atikuzzaman

Id:151002019

Al Imran pappu

Id:151002045

Md.Asaduzzaman

Id:151002050

Tawhid Lubnan

Id:151002079

Dept.of CSE

Submitted To :

Biplab Chandra Debnath

Lecturer

Dept.of CSE

Green University Of

Bangladesh

Contents

1	What is NewSQL?	2
2	History	2
3	Systems	2
3.1	New architectures	3
3.2	SQL engines	3
3.3	Transparent sharding	3
4	Why NewSQL?	3
5	SQL vs NoSQL vs NewSQL	4
6	References	4

1 What is NewSQL?

NewSQL is a class of modern relational database management systems that seek to provide the same scalable performance of NoSQL systems for online transaction processing (OLTP) read-write workloads while still maintaining the ACID guarantees of a traditional database system.

2 History

The term was first used by 451 Group analyst Matthew Aslett in a 2011 research paper discussing the rise of new database systems as challengers to established vendors. Many enterprise systems that handle high-profile data (e.g., financial and order processing systems) also need to be able to scale but are unable to use NoSQL solutions because they cannot give up strong transactional and consistency requirements. The only options previously available for these organizations were to either purchase a more powerful single-node machine or develop custom middleware that distributes queries over traditional DBMS nodes. Both approaches are prohibitively expensive and thus are not an option for many. Thus, in this paper, Aslett discusses how NewSQL upstarts are poised to challenge the supremacy of commercial vendors, in particular, Oracle.

3 Systems

Although NewSQL systems vary greatly in their internal architectures, the two distinguishing features common amongst them is that they all support the relational data model and use SQL as their primary interface. The applications targeted by these NewSQL systems are characterized as having a large number of transactions that

- (1) are short-lived (i.e., no user stalls)
- (2) touch a small subset of data using index lockups
- (3) are repetitive (i.e. executing the same queries with different inputs).

These NewSQL systems achieve high performance and scalability by eschewing much of the legacy architecture of the original IBM System R design, such as heavyweight recovery or concurrency control algorithms. One of the first known NewSQL systems is the H-Store parallel database system.

NewSQL systems can be loosely grouped into three categories:

3.1 New architectures

This type of NewSQL systems are completely new database platforms. These are designed to operate in a distributed cluster of shared-nothing nodes, in which each node owns a subset of the data. These databases are often written from scratch with a distributed architecture in mind, and include components such as distributed concurrency control, flow control, and distributed query processing. Example systems in this category are Google Spanner, Clustrix, VoltDB, MemSQL, Pivotal's GemFire XD, SAP HANA, NuoDB, TiDB, and Trafodion.

3.2 SQL engines

This type of NewSQL systems are highly optimized storage engines for SQL. These systems provide the same programming interface as SQL, but scale better than built-in engines, such as InnoDB. Examples of these new storage engines include MySQL Cluster, Infobright, TokuDB and the now defunct InfiniDB.

3.3 Transparent sharding

These systems provide a sharding middleware layer to automatically split databases across multiple nodes. ScaleBase is an example of this type of system.

4 Why NewSQL?

SQL is a standard language used by developers, DBAs, and business analysts to query data sets. Traditional SQL databases offer rich transaction capabilities, ad-hoc query and reporting capability, and vast amounts of standards-based tooling. Where they fall short is in the ability to scale out to meet the needs of modern, high-performance applications.

NoSQL was a solution to issues of scale that emerged as organizations began dealing with immense volumes, velocity, and a variety of big data from a multitude of sources. NoSQL solutions offered availability, flexibility, and scale. However, they came with difficult tradeoffs - the sacrifice of data consistency and transactional

(ACID) guarantees, loss of ad-hoc query capability and increased application complexity. Trading off data consistency and ACID in many cases means application developers, who are not distributed data infrastructure experts, have had to try and fill the gap.

NewSQL attempted to do what NoSQL did without those tradeoffs. By removing just the parts of SQL that hampered scalability and performance, it was able to increase both while maintaining key advantages of SQL systems.

NewSQL technologies offer the best of both worlds: the scale of NoSQL with the ACID guarantees, strong consistency, minimized application complexity and interactivity of SQL.

5 SQL vs NoSQL vs NewSQL

Characteristic	SQL	NoSQL	NewSQL
Relational	Yes	No	Yes
SQL language support	Yes	No*	Yes
Fully ACID Compliant	Yes	No	Yes
Horizontal Scalability	No	Yes	Yes
High Performance	No	Yes	Yes
Schema-on-Write	Yes	No	Yes

Last, of all the unique in-memory scale-out architecture enables 100x faster performance than traditional alternatives. This enables NewSQL to convert live data into business value, analyze and act on streaming data, and make use of real-time intelligence. So, it combines streaming analytics, strong ACID guarantees, familiar SQL data interactivity, and native clustering.

6 References

- 1.<https://www.wikipedia.org>
- 2.<https://www.voltdb.com>