2023 IEEE 13th Symposium on Computer Applications & Industrial Electronics (ISCAIE) | 979-8-3503-4731-9/23/$31.00 ©2023 IEEE | DOI: 10.1109/ISCAIE57739.2023.10164996

Recent Advances and Benchmarking of NewSQL for OLTP and OLAP in the Big Data Age

John Zu An, Chung
Faculty of Engineering, Computing and Science
Swinburne University of Technology Sarawak Campus
Kuching, Malaysia
101233836@students.swinburne.edu.my

Miko May Lee, Chang
Faculty of Engineering, Computing and Science
Swinburne University of Technology Sarawak Campus
Kuching, Malaysia
mchang@swinburne.edu.my

Swee King, Phang
School of Engineering
Taylor's University
Kuala Lumpur, Malaysia
SweeKing.Phang@taylors.edu.my

***Abstract*—While the never-ending debate between relational and NoSQL database is ongoing, a new competitor, NewSQL, has quietly entered the field. Introduced in 2011, NewSQL is not as popular as relational or NoSQL database. The main selling point of NewSQL is its ability to scale horizontally while preserving ACID properties and thus preserving the support to handle OLTP workloads. Given the active research and evolving developments of NewSQL in the last decade, this research paper aims to identify how far has NewSQL advanced to and how comparable is it to existing database systems in terms of On-Line Transaction Processing (OLTP) and On-Line Analytics Processing (OLAP) to accommodate big data. Three research questions have been formulated as part of the systemic literature review (SLR) followed by an experimental benchmarking to validate the results from the SLR. The results show that while NewSQL still has room to improve, it is definitely ready to be used in productions, albeit having certain obstacles which may need to be addressed such as expertise in deployment and maintenance, as well as performance. One limitation of this research is that the testing was conducted on a single node and future research could include performance testing on multiple nodes.**

Keywords*—*NewSQL; OLAP; OLTP; Big Data; Data Management

I. INTRODUCTION

As the world reaches the peak of Industry 4.0, there has been more and more integration between existing technology and uprising technologies such as Internet of Things (IoT), Artificial Intelligence (AI), Machine Learning (ML) and Cloud Computing and Analytics [1]. With the integration of these technologies, a large amount of data is being generated and needs to be processed and analysed, which is where the buzzword "Big Data" comes into play. Big data refers to large amounts of data generated from day-to-day operations and can be further analysed to provide business insights and trends in a timely manner [2]. The traditional workflow is to use the Extract-Load-Transform (ELT) mechanism on On-Line Transaction Processing (OLTP) databases to transfer data into Data Warehouses (DW) for On-Line Analytics Processing (OLAP). Due to the increasing need for real-time analytics, such traditional workflow can no longer keep up with the trend [3].

Hence, this research investigates the use of emerging NewSQL databases along with Hybrid Transaction and Analytics Processing (HTAP) which are proposed and developed specifically to keep up with the trend of real-time analytics on big data. Section II describes the detailed research methodology for conducting the SLR to formulate the research questions. These research questions are further validated via experimental benchmarking in Section IV followed by detailed results and analyses addressing each RQ in Section V. Section VI discusses the findings and limitations while Section VII concludes the study with recommendations for future work.

II. METHODOLOGY

The overall research methodology for this research is split into two parts: systematic literature review based on the research questions followed by the experimental benchmarking and analysis in Section IV. As outlined in Figure 1, there are 7 steps taken for the SLR. Firstly, two databases are identified in the planning step including Scopus and IEEE with EndNote as the citation manager [4].

Secondly, the research scope is narrowed down and three research questions are defined using the PEO Framework, which stands for Population, Exposure and Outcome/Theme respectively:

(1) P (Population) – Computer Scientist / Data Scientist;

(2) E (Exposure) – NewSQL and HTAP adoption;

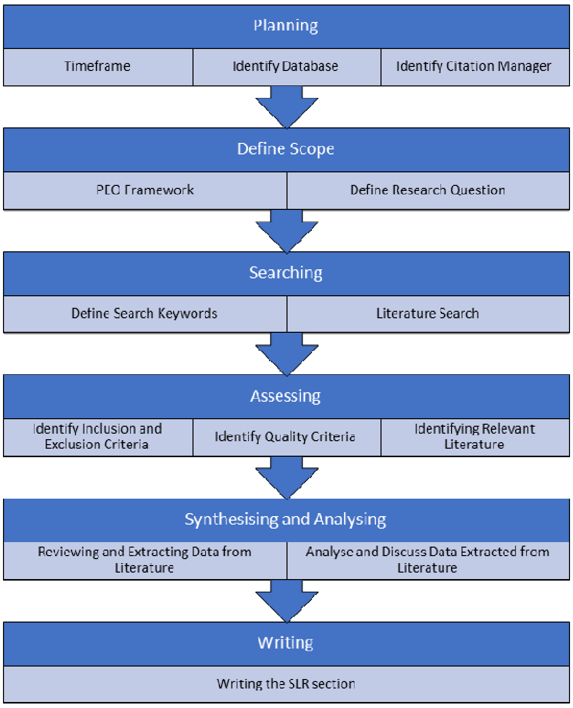
(3) O (Outcome/Theme) – Prevalence of NewSQL and HTAP in Big Data Processing.

Based these three aspects, the authors have defined the following three research questions (RQ) that reflect the objective of this research:

TABLE I. RESEARCH QUESTIONS

RQ1	What are the pros and cons of using NewSQL over RDBMS or NoSQL?
RQ2	What are the pros and cons of using HTAP NewSQL instead of OLTP with RDBMS and OLAP with Data Warehouse?
RQ3	How does NewSQL compare in terms of OLTP and OLAP performance over RDBMS and Data Warehouses?

Fig. 1. Systematic Literature Review (SLR)



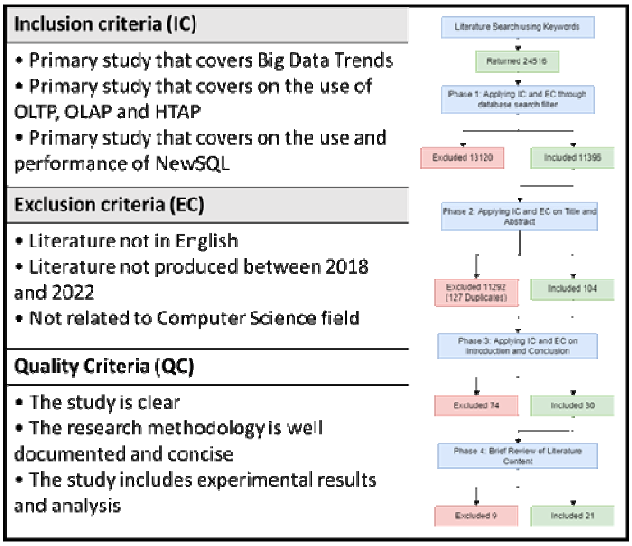
All RQs are first addressed using the findings from SLR followed by an experimental benchmarking for validation and finalisation with recommendations and future research directions. Based on the research scope identified, the following keywords are used as search strings in the literature databases: NewSQL; OLTP; OLAP; HTAP; Big Data AND (OLTP OR OLAP OR HTAP OR NewSQL OR Trends). A total of 24516 records are returned from Scopus (19391) and IEEE Explore (5125) databases.

Given the sheer number of literatures return from the search results, the search is further refined by using inclusion and exclusion criteria to identify relevant literatures from 2018 to 2022 for a comprehensive SLR to cover the recent advances [4]. After a series of literature assessment and selection, a total of 21 papers are selected based on the inclusion and exclusion criteria in Figure 2 and the grading is summarized in Table 2.

TABLE 2. GRADING LITERATURE BY QC

Ref	*QC1*	*QC2*	*QC3*	*Quality*
[13][14][17][19]	1	1	1	3
[8]	1	0.5	1	2.5
[12]	1	1	0.5	2.5
[2][5][6][7][15][18]	1	1	0	2
[3][9][10][11][16][20][21][22][23]	1	0.5	0	1.5

Fig. 2. Criteria for Literature Search and Selection



III. Literature Review

A. *RQ1: What are the pros and cons of using NewSQL over RDBMS or NoSQL*

The idea of relational database, also known as SQL databases, were first conceptualised in 1970, but the idea only came to fruition in 1979. Relational databases are the first type of database system to be developed, way before the Internet was created and computer systems advanced into today's scale, to accommodate storing data in a structured manner with strict attribute enforcement [24]. Relational database is still widely used in this day and age because of its strong support for ACID (atomicity, consistency, isolation and durability) transactions [21]. These are especially important for use cases such as the banking industry whereby the transactions need to be atomic (either the transactions go through, debiting and crediting the respective accounts or it fails and does not perform any updates), consistent (ensuring constraints set in place are not violated, for example if the transaction amount is more than the available amount in the account, the transaction should be rendered as fail), isolation (ensuring that transaction are processed accordingly so that it does not incur any irregularities) and durability (ensuring any transactions written to the database would persist in the case of database failure/outage).

However, with the rise of web applications, a new challenge was posed to relational database, which is the use of semi-structured data such as Extensible Mark-up Language (XML) and JavaScript Object Notation (JSON) data types. Due to the strict enforcement of attributes by relational database, such semi-structured data types were unsupported and called for the use of an alternative database system [6]. Additionally, in order to maintain its strong ACID compliance, relational database could only scale vertically (scaling by upgrading system hardware components such as storage, RAM and processor), which may be expensive [21]. This introduced a problem in dealing with large amounts of data as computing capabilities has limitations which pose as bottlenecks, in the form of connection bandwidth, disk read and write speed, processor clock rate, etc. [17]. Notable Relational Database Management Systems (RDBMS) are Oracle Database, MySQL, PostgreSQL, and Microsoft's SQL Server.

Since RDBMS is no longer an efficient database to accommodate the data being generated and stored due to the rise of Web 2.0, NoSQL was born and became a popular choice for software application by compensating the drawbacks in a relational database [6]. Web 2.0 consists mainly of semi-structured data in the form of webpage HTML and XML as well as HTTP responses in the form of JSON. NoSQL can accommodate this by supporting flexible schema and not enforcing strict attribute data type in contrast to relational database [6]. Furthermore, by relaxing ACID compliance, NoSQL is able to achieve higher performance and allows for horizontal scaling (scaling by increasing nodes or machines as an aggregated database). Instead of ACID compliance, NoSQL database adhere to the Consistency, Availability and Partition tolerance (CAP) theorem, which stipulates that a database can only provide for two of the three CAP properties. This subsequently results in most NoSQL databases supporting Basically Available, Soft-state and Eventual consistency (BASE) transactions. Applications such as social media and non-sensitive IoT can operate on Basically Available (prioritising performance over consistency), Soft-state (data being stored in the volatile RAM resulting in inconsistent dataset across nodes) and Eventual consistency transactions as data loss may be insignificant compared to performance. With horizontal scaling, this allows NoSQL databases to be fault tolerant and ensuring high availability of the database [21].

As NoSQL supports for BASE transactions, this means that the data in each database node may not be consistent with each other at any given time and will only synchronise all the data at intervals, thus the property of eventual consistency. This means that business sensitive applications such as banking and ticketing system cannot adopt the use of NoSQL as data loss may result in loss in revenue, which is not a fair trade-off over performance [6]. Unlike relational databases, NoSQL does not have any de facto standard for database developers to adhere to, which resulted in the different implementations and querying languages across different database solutions [9]. Notable NoSQL Database Management Systems are categorized according to their schema type viz. key-value stores (Redis, Voldemort), column-oriented (HBase, Cassandra), document-oriented (MongoDB, RavenDB) and graph-oriented (Neo4J, HypergraphDB) [21].

Comparing RDBMS and NoSQL, there is still a gap for applications such as big data that require strong ACID compliance and scalability. Given the need to accommodate a large OLTP workload without compromising ACID properties, especially for business-sensitive applications, Matthew Aslett coined the term NewSQL in 2011 to refer to a database that combines the best of RDBMS (strong ACID transactions and strict scheme enforcement) and NoSQL (horizontal scalability and performance) to accommodate the growing scale of such applications [15].

According to Murazzo, et al. [8], NewSQL are approximately 50 times faster than traditional RDBMS in terms of OLTP workload, which is ideal for most business who are already currently using RDBMS as part of their application. In addition to that, NewSQL also supports HTAP, the combination of OLTP and OLAP workload, which is a deal breaker if there is a need for real-time data analysis on data from OLTP transactions. NewSQL is built prepared for cloud computing, making it more cost effective when used on cloud architecture, which supports elastic scaling as well as a pay-for-what-you-use pricing. Unlike the manual sharding into multiple databases and replication required to scale traditional RDBMS into master and read-replicas, NewSQL automatically takes care of the process and has the added ability for the distributed databases to handle read-write transactions without compromising ACID [25]. However, NewSQL does not work as well as NoSQL when used in low-specification environment. In the study conducted by Pandya, et al. [26], VoltDB (a NewSQL database) which is an in-memory database failed in several tests because it ran out of memory. However, in the cases where it had sufficient memory, it outperformed MongoDB (a NoSQL database). Depending on the application's use case, migration from RDBMS to NewSQL may not be productive as the NewSQL database solution may not provide full support for all SQL commands. Since NewSQL's schema is based upon relational databases, it is not comparable for use case where graph-oriented schema NoSQL database is needed. The NewSQL Database Management Systems which are available commercially include SAP HANA, VoltDB, Google Spanner, TiDB, CockroachDB, Citus [15].

It should be noted that none of the database is a "one-size-fits-all" solution, and sometime a combination of multiple types is required depending on a case-by-case basis. Roy-Hubara, et al. proposed a method for users to identify which database is the most suitable, by specifying the data-related, functional and non-functional requirements, assigning qualitative points to each requirement, and calculating the weight of each point to ascertain which database is the most suitable [14]. Table 3 summarizes the differences between relational, NoSQL and NewSQL database in a concise manner based on the information gathered from the literatures.

TABLE 3. COMPARING RELATIONAL, NOSQL AND NEWSQL DATABASES [20]

Database	*Relational*	*NoSQL*	*NewSQL*
Schema	Table (Row-based)	Key-value stores; Column-oriented; Document-oriented; Graph-oriented	Table (Column and Row based)
Scaling capability	Vertical	Horizontal	Horizontal
Storage	On-disk	On-disk + In-memory	On-disk + In-memory
Supported property	ACID	BASE + CAP	ACID
Query Complexity	High	Low (except for Graph-oriented)	High
OLTP	Supported	Not supported	Supported
OLAP	Limited	Not supported	Supported
Machine Dependency	Single	Distributed	Distributed
Cloud Support	Supported	Supported	Supported

B. *RQ2: What are the pros and cons of using HTAP NewSQL instead of OLTP with RDBMS and OLAP*

OLTP refers to system focused on providing transactional processing for application which require strong ACID compliance while OLAP is a system focused on allowing analytical processing of data, which has grown into the scale of big data currently [17]. A data warehouse is a database developed to store big data and facilitate OLAP workload. However, it is not compared to RDBMS and NoSQL as it is not a database used for operational use cases but instead focused on analytics [6]. Data warehouses are multi-dimensional relational data stores; unlike the RDBMS mentioned previously, they are column-based instead of row-based [16]. According to an experiment by Muniswamaiah, et al. [12], columnar data can be analysed faster than RDBMS when paired with an in-memory database. While data warehouses provide real-time analytics once the data is loaded [6], it differs from the HTAP approach, which aims to provide real-time analysis over fresh data [23]. With the rise of cloud computing and lambda technology, there have also been efforts to support real-time analysis over fresh data, by automating the ETL process over big data. However, this approach still requires data to be passed through an ETL process that can be time-consuming, which HTAP with NewSQL aims to overcome through the elimination of the ETL process altogether [18].

While both OLTP and OLAP are crucial for a workflow, it was not feasible to facilitate both processes within the same database. As OLTP on its own is already performance intensive when dealing with big data, adding the OLAP workload will compromise the performance of OLAP which is sensitive. Moreover, since RDBMS which supports for OLTP is only able to scale vertically, merging both OLTP and OLAP workload would be overwhelmingly expensive, given the computing capabilities needed. However, with the emergence of NewSQL, which is able to scale horizontally, Hybrid Transactional Analytical Processing (HTAP) was able to be realised. The term HTAP was coined by Gartner in 2014, which refers to the combination OLTP and OLAP workload processing [11]. The reason HTAP was introduced is to allow for real-time analysis over fresh data, to facilitate the rising need for fast analysis on fresh datasets for decision-making. This is because HTAP eliminates the need for Extract-Load-Transform (ETL) process on OLTP database to OLAP data warehouse in traditional workflow. ETL processes are generally performed over large datasets and can be time-consuming and performance intensive. What's more, with HTAP, the storage needed to facilitate both OLTP and OLAP workloads can be greatly reduced as data is not replicated from the RDBMS to DW [18].

Although OLTP and OLAP workloads are distributed across multiple nodes, there are still interference between the two workloads. According to the experiment conducted by Sirin, et al. [19], while there was reduction in performance throughput of OLTP and OLAP workload, it did not compromise the ACID compliance within the OLTP workload. Nevertheless, the system requirements for a HTAP system are much higher compared to an OLTP-only or OLAP-only system, which may not be a concern for users who prioritise performance over cost. Although HTAP bridges OLTP and OLAP, each database solution has different implementation, which prioritise different aspects of data management.

IV. EXPERIMENTAL SETUP

Prior to this study, a benchmarking was done by Murazzo, et al. [8], which compared the performance between MySQL and Google Spanner, by using the Autonomous City of Buenos Aires's SUACI dataset for experimental analysis. From the results, it can be observed that Google Spanner has better performance when compared to MySQL. However, the specifications used for setting up both databases were not disclosed. Therefore, the results are questionable as the specifications used may not be equal. As such, there is a need to conduct a fair benchmark with proper disclosure which will be addressed in the next section.

Table 4 shows the software and hardware setup used in the experimental benchmarking. All the databases are started with the default configuration but with the RAM usage set to get the most out of the machine and the machine is cleared each time for a new scale factor benchmark.

TABLE 4. EXPERIMENTAL SETUP FOR BENCHMARKING

Software	• Operating System: Ubuntu 20.04 LTS • MySQL server 8.0.0 • PostgreSQL 14.5 • TiDB server v6.1.0 • Citus-11.0 • MonetDB v11.43.21 • sysbench 1.0.18 (OLTP Benchmarking)
Hardware	• Machine: Graphcore IPU POD-16 • CPU: 64 cores (240 vCPU) • RAM: 472GB • Disk: 485GB • Processor: AMD EPYC 7742

V. RESULTS AND ANALYSES

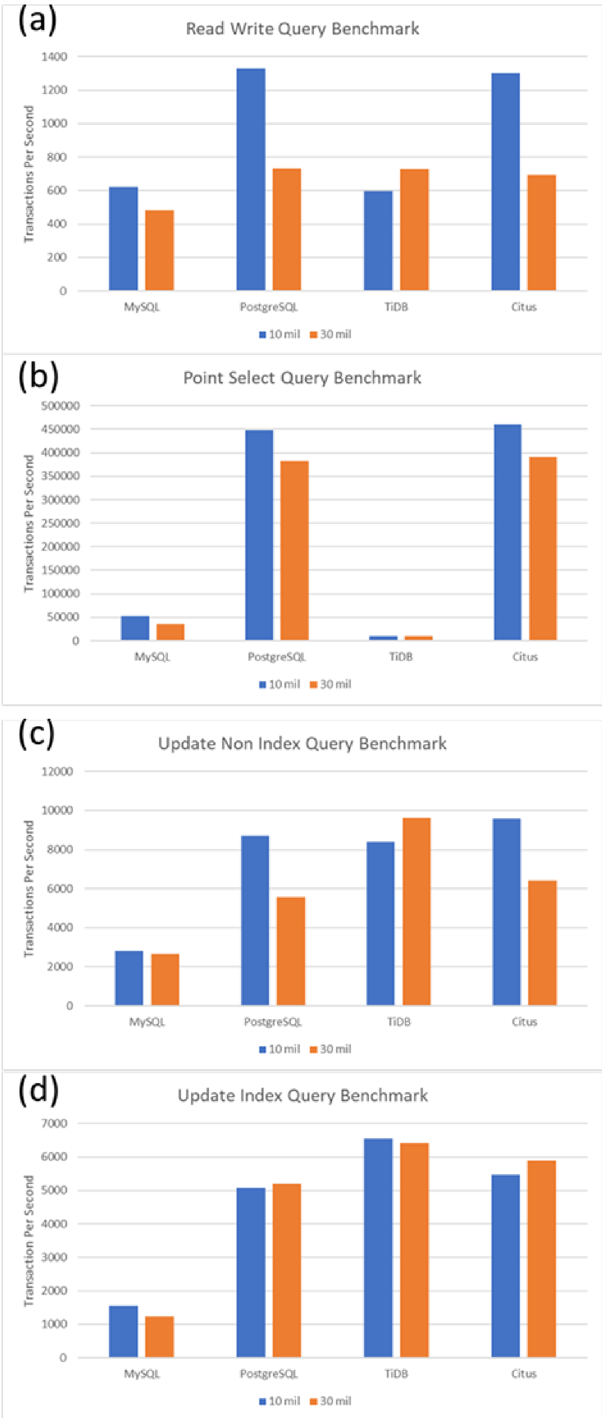
In an article by Yang [27], who is JD.com's Cloud Database Product Manager, he indicated that by switching over from MySQL to TiDB, it provided a significant boost of performance as well as cost-savings. Since TiDB is a HTAP database, it can provide for OLTP and OLAP workload, all while being under a single database architecture, which greatly improves integration as well as maintenance efficiency.

A. RQ3: How does NewSQL compare in terms of OLTP and OLAP performance over RDBMS and Data Warehouses

In order to substantiate the results from the available research, the researcher has decided to conduct an experimental benchmarking with TiDB and Citus as the NewSQL choice to be compared with MySQL and PostgreSQL. It should be noted that the tests are conducted on HTAP-based NewSQL databases, which may not reflect the full potential of NewSQL's OLTP ability. The reason MySQL and PostgreSQL were chosen for the comparison is because TiDB offers compatibility with MySQL [28] and Citus is an extension to PostgreSQL [29].

Additionally, to ensure that results are not skewed when conducting benchmarking on the databases, the databases are only hosted on a single machine. The benchmarks use a derivative of the TPC-C and TPC-DS benchmark, which is not comparable to published TPC-C or TPC-DS results as it does not fully comply to the TPC-C and TPC-DS benchmark specifications. Sysbench was used to conduct OLTP based workload testing on the databases and 4 tests (read write, point select, update index and update non-index) were used to benchmark the OLTP performance of the databases. In Figure 3(a), it shows that the NewSQL and its respective RDBMS counterparts have relatively comparable results while Figure 3(b) shows comparable results between Citus and PostgreSQL but TiDB performs significantly worse than MySQL. Figure 3(c) and 3(d) shows the results from the update operation on index and non-index attribute respectively, while Citus and PostgreSQL still achieved comparable results, it seems that TiDB performs significantly better than MySQL for the update category operations benchmark.

Fig. 3. Performance Benchmarking Results



As for the OLAP tests, the researcher has referenced the TPC-DS benchmark and conducted the tests on TiDB, Citus and MonetDB. Since the NewSQL database does not have any direct compatibilities with any data warehouse, the researcher has chosen to benchmark the NewSQL databases against only one data warehouse, which is MonetDB. However, there are queries processed by the NewSQL which either exceeded the time-limit of the benchmarking, exhausted the available resources on the machine or is not supported by the NewSQL database. While Citus only has issues with queries exceeding the time-limit of benchmarking, TiDB had issues with compatibility of queries and even exhausted the available resources on the machine while processing queries. This study has opted to not conduct the benchmarking on the RDBMS as they do not claim to support OLAP queries. Table 5 shows the time taken for the selected TPC-DS queries which has been executed by the databases.

TABLE 5. TIME TAKEN FOR OLAP QUERY EXECUTION

Query type	***MonetDB (HH:MM:SS)***	***TiDB (HH:MM:SS)***	***Citus (HH:MM:SS)***
Query 30 (Iterative Query)	0:0:0.0694	0:0:0.3406	0:25:42.7786
Query 40 (Reporting Query)	0:0:0.3938	0:0:0.3165	0:0:3.1907
Query 64 (Joining Multiple Fact Tables Query)	0:0:3.1907	0:0:5.2962	0:0:38.3575
Query 70 (CPU Intensive)	0:0:0.9101	Out of Memory	0:0:22.4536
Query 82 (Ad-hoc Input / Output Intensive Query)	0:0:0.1388	0:3:10.1584	0:0:3.2901

VI. DISCUSSION AND LIMITATION

A. Adoption of NewSQL

While NewSQL offers additional advantages over RDBMS, mainly the ability to scale horizontally, and some NewSQL database providing HTAP workload support, it is still far from being adopted widely. One notable reason could be that it requires users to change their workflow process. While there are NewSQL such as TiDB and Citus which maintains compatibility with MySQL and PostgreSQL as much as possible in order to facilitate an ease of transition, there are functionalities that may not be fully supported which can be evident in TiDB such as several missing functionalities from MySQL as outlined in their documentation. NewSQL, being a relatively new solution, may deter developers from adopting it in terms of support, pricing/cost-efficiency, availability, and performance, all of which are valid points for concern when adopting new technologies. The availability of expertise in NewSQL may also be scarce, which would also discourage businesses from taking the leap.

B. Performance of NewSQL in terms of OLTP and OLAP

NewSQL strives when there are multiple nodes. In terms of OLTP, the single node NewSQL is indeed comparable to its RDBMS counterparts. In Figures 3 which are showing the results from OLTP benchmarking, it shows that PostgreSQL and Citus being on par with one another. While TiDB and MySQL lacks behind PostgreSQL and Citus, TiDB seems to be on par with MySQL in read-write operations, has an edge over MySQL in update operations and perform worse than MySQL in terms of point-select operations. Since Citus is just an extension to PostgreSQL, it may be the reason why they seem equal in terms of OLTP workload. While TiDB is somewhat MySQL compatible, it should be noted that they are vastly different architectures. With that said, other OLTP-based NewSQL may outperform Citus and TiDB as they are HTAP-based databases.

In terms of OLAP, single node NewSQL seems to have fall short when compared to a Data Warehouse. While TiDB seem to fare quite well in iterative, reporting and joining multiple fact tables queries when compared to MonetDB, it seems that it ran out of resources when executing the CPU intensive query. On the other hand, Citus had a rather slow execution time when compared to MonetDB or even TiDB. However, Citus performed relatively well with the CPU intensive and ad-hoc queries when compared to TiDB. While TiDB seemed to have an edge over OLAP compared to Citus, it should be noted that the test is conducted on a single node machine which may not have all the relevant optimisations as the recommended distributed setting for production usage. In the test conducted by the Citus Lead Engineer, it also evidently shows that using Citus for analytical queries does yield results in terms of seconds and will only achieve results return in terms of milliseconds when paired with a rollup table. Unfortunately, since the TPC-DS benchmark queries are pre-set and to keep the benchmarking as fair as possible, only the standard queries were used.

C. Limitations

Since this benchmark was conducted on a single node database, it does not fully reflect the ability of NewSQL, which is OLTP performance or HTAP performance after horizontal scaling. Secondly, as mentioned in the section discussing the performance of NewSQL, a standard benchmarking is not directly translatable to a real-world scenario/use-case, which often has unpredictable or differing workloads when compared to a benchmarking scenario. Nevertheless, a standard benchmark can still provide a reference point as to the standard performance of NewSQL.

VII. CONCLUSION AND FUTURE WORK

This research has proved that an HTAP NewSQL database on a single node seems to offer comparable performance to RDMBS in terms of OLTP, but may not be on par with a data warehouse in terms of OLAP query speed. Nonetheless, using standard benchmarking may not fully reflect the performance of a NewSQL database. The researchers suggest that future works may opt to perform a real-world case study-based benchmarking in comparison to the business's existing architecture to better evaluate the cost-efficiency and performance of NewSQL compared to existing workflow. After all, there is no perfect solution when it comes to database selection, and different use cases may command an alternate approach to database choices. NewSQL database providers may also work with the TPC to publish a verified benchmarking result set to better aid developers in their decision-making to select an appropriate database for their use case. To start exploring on NewSQL, the researchers recommend looking through the documentation and tutorials provided by the database developers for the configurations available to fine tune the database and also the syntax which is used by the NewSQL database.

SOURCE CODE

All the scripts used for the benchmarking can be found on GitHub at https://github.com/JohnChung2002/5th-FECS-Future-Researcher

ACKNOWLEDGMENT

The authors would like to thank Swinburne University of Technology Sarawak Campus for the 5th FECS Future Researcher Internship Award.

REFERENCES

[1] IBM. "What is Industry 4.0?" IBM. https://www.ibm.com/my-en/topics/industry-4-0 (accessed Jul. 25, 2022).

[2] S. Jeble, S. Kumari, and Y. Patil, "Role of big data in decision making," (in English), Operations Supply Chain Manag., vol. 11, no. 1, pp. 36-44, 2018, doi: 10.31387/oscm0300198.

[3] S. D. Kuznetsov, P. E. Velikhov, and Q. Fu, "Real-time analytics, hybrid transactional/analytical processing, in-memory data management, and non-volatile memory," in 2020 Ivannikov Ispras Open Conference (ISPRAS), 10-11 Dec. 2020 2020, pp. 78-90, doi: 10.1109/ISPRAS51486.2020.00019.

[4] J. v. Brocke, A. Simons, B. Niehaves, K. Reimer, R. Plattfaut, and A. Cleven, "Reconstructing the giant: On the importance of rigour in documenting the literature search process," in ECIS 2009 Proceedings, 2009, no. 161. [Online]. Available: https://aisel.aisnet.org/ecis2009/161.

[5] K. T. Sridhar, "Big Data Analytics Using SQL: Quo Vadis?," Cham, 2018: Springer International Publishing, in Research and Practical Issues of Enterprise Information Systems, pp. 143-156.

[6] U. Aftab and G. F. Siddiqui, "Big Data Augmentation with Data Warehouse: A Survey," in 2018 IEEE International Conference on Big Data (Big Data), 10-13 Dec. 2018 2018, pp. 2785-2794, doi: 10.1109/BigData.2018.8622206.

[7] A. Nanda, S. Gupta, and M. Vijrania, "A Comprehensive Survey of OLAP: Recent Trends," in 2019 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA), 12-14 June 2019 2019, pp. 425-430, doi: 10.1109/ICECA.2019.8822203.

[8] M. Murazzo, P. Gómez, N. Rodríguez, and D. Medel, "Database NewSQL Performance Evaluation for Big Data in the Public Cloud," Cham, 2019: Springer International Publishing, in Cloud Computing and Big Data, pp. 110-121.

[9] P. Valduriez, R. Jimenez-Peris, and M. T. Özsu, "Distributed Database Systems: The Case for NewSQL," in Transactions on Large-Scale Data- and Knowledge-Centered Systems XLVIII: Special Issue In Memory of Univ. Prof. Dr. Roland Wagner, A. Hameurlain and A. M. Tjoa Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2021, pp. 1-15.

[10] R. Sharma, P. Agarwal, and R. P. Mahapatra, "Evolution in Big Data Analytics on Internet of Things: Applications and Future Plan," in Multimedia Big Data Computing for IoT Applications: Concepts, Paradigms and Solutions, S. Tanwar, S. Tyagi, and N. Kumar Eds. Singapore: Springer Singapore, 2020, pp. 453-477.

[11] G. Li and C. Zhang, "HTAP Databases: What is New and What is Next," presented at the Proceedings of the 2022 International Conference on Management of Data, Philadelphia, PA, USA, 2022. [Online]. Available: https://doi.org/10.1145/3514221.3522565.

[12] M. Muniswamaiah, T. Agerwala, and C. C. Tappert, "Integrating Polystore RDBMS with Common In-Memory Data," in 2020 IEEE International Conference on Big Data (Big Data), 10-13 Dec. 2020 2020, pp. 5762-5764, doi: 10.1109/BigData50022.2020.9377883.

[13] C. K. Leung, Y. Chen, C. S. H. Hoi, S. Shang, and A. Cuzzocrea, "Machine Learning and OLAP on Big COVID-19 Data," in 2020 IEEE International Conference on Big Data (Big Data), 10-13 Dec. 2020 2020, pp. 5118-5127, doi: 10.1109/BigData50022.2020.9378407.

[14] N. Roy-Hubara, P. Shoval, and A. Sturm, "A Method for Database Model Selection," Cham, 2019: Springer International Publishing, in Enterprise, Business-Process and Information Systems Modeling, pp. 261-275.

[15] I. Chereja, S. M. L. Hahn, O. Matei, and A. Avram, "Multidimensional Analysis of NewSQL Database Systems," Cham, 2021: Springer International Publishing, in Software Engineering and Algorithms, pp. 221-236.

[16] I. Kovačević and I. Mekterovic, "Novel BI data architectures," in 2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), 21-25 May 2018 2018, pp. 1191-1196, doi: 10.23919/MIPRO.2018.8400216.

[17] N. El-Sayed, Z. Sun, K. Sun, and R. Mayerhofer, "OLTP In Real Life: A Large-scale Study of Database Behavior in Modern Online Retail," in 2021 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 3-5 Nov. 2021 2021, pp. 1-8, doi: 10.1109/MASCOTS53633.2021.9614295.

[18] I. Chereja, S. M. L. Hahn, O. Matei, and A. Avram, "Operationalizing Analytics with NewSQL," Cham, 2021: Springer International Publishing, in Software Engineering and Algorithms, pp. 249-263.

[19] U. Sirin, S. Dwarkadas, and A. Ailamaki, "Performance Characterization of HTAP Workloads," in 2021 IEEE 37th International Conference on Data Engineering (ICDE), 19-22 April 2021 2021, pp. 1829-1834, doi: 10.1109/ICDE51399.2021.00162.

[20] N. Priya and E. Punithavathy, "A Review on Database and Transaction Models in Different Cloud Application Architectures," Singapore, 2022: Springer Nature Singapore, in Proceedings of Second International Conference on Sustainable Expert Systems, pp. 809-822.

[21] T. N. Khasawneh, M. H. A.-. Sahlee, and A. A. Safia, "SQL, NewSQL, and NOSQL Databases: A Comparative Survey," in 2020 11th International Conference on Information and Communication Systems (ICICS), 7-9 April 2020 2020, pp. 013-021, doi: 10.1109/ICICS49469.2020.239513.

[22] A. Wong et al., "A Survey of Natural Language Processing Implementation for Data Query Systems," in 2021 IEEE International Conference on Recent Advances in Systems Science and Engineering (RASSE), 12-14 Dec. 2021 2021, pp. 1-8, doi: 10.1109/RASSE53195.2021.9686815.

[23] A. Cuzzocrea and R. Moussa, "Towards Lambda-Based Near Real-Time OLAP over Big Data," in 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC), 23-27 July 2018 2018, vol. 01, pp. 438-441, doi: 10.1109/COMPSAC.2018.00067.

[24] D. Kelly. "A Brief History of Databases: From Relational, to NoSQL, to Distributed SQL." Cockroach Labs. https://www.cockroachlabs.com/blog/history-of-databases-distributed-sql/ (accessed Jul. 30, 2022).

[25] A. Pavlo and M. Aslett, "What's Really New with NewSQL?," SIGMOD Rec., vol. 45, no. 2, pp. 45–55, 2016, doi: 10.1145/3003665.3003674.

[26] A. Pandya, C. Kulkarni, K. Mali, and J. Wang, "An Open Source Cloud-Based NoSQL and NewSQL Database Benchmarking Platform for IoT Data," Cham, 2019: Springer International Publishing, in Benchmarking, Measuring, and Optimizing, pp. 65-77.

[27] M. Yang. "An 8x System Performance Boost: Why We Migrated from MySQL to a NewSQL Database." PingCAP. https://en.pingcap.com/case-study/an-8x-system-performance-boost-why-we-migrated-from-mysql-to-a-newsql-database/ (accessed Aug. 26, 2022).

[28] PingCAP. "MySQL Compatibility." PingCAP. https://docs.pingcap.com/tidb/dev/mysql-compatibility (accessed Jul. 30, 2022).

[29] Citus Data. "What is Citus?" Citus Data. https://docs.citusdata.com/en/v11.1/get_started/what_is_citus.html (accessed Jul. 30, 2022).