Database NewSQL performance evaluation for big data in the public cloud

María Murazzo[1]✉, Pablo Gómez[2], Nelson Rodríguez[1], Diego Medel[1]

[1] Departamento e Instituto de Informática - F.C.E.F. y N. - U.N.S.J.
[2] Alumno Avanzado Licenciatura en Cs. de la Computación - F.C.E.F. y N. - U.N.S.J.
Complejo Islas Malvinas. Ignacio de la Roza y Meglioli. C.P. 5402. Rivadavia. San Juan, 0264 4234129

marite@unsj-cuim.edu.ar, pablo.gomez.allende@gmail.com, nelson@iinfo.unsj.edu.ar, dmedel@iinfo.unsj.edu.ar

Abstract. For very years, relational databases have been the leading model for data storage, retrieval and management. However, due to increasing needs for scalability and performance, alternative systems have emerged, namely NewSQL technology. NewSQL is a class of modern relational database management systems (RDBMS) that 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. In this research paper, the performance of a NewSQL database is evaluated, compared to a MySQL database, both running in the cloud, in order to measure the response time against different configurations of workloads.

Keywords: Big Data, Cloud Computing, Spanner, Google Cloud Platform

1 Introduction

At present databases are not only expected to be flexible enough to handle different variety of data formats, they are expected to deliver extreme performance as well as easily scale to handle big data. According to an estimate there are 2.5 quintillion bytes of data created each day, but that pace is accelerating with the growth of the Internet of Things (IoT) (Gubbi, Buyya, Marusic, & Palaniswami, 2013).

Last two years 90 percent of the data in the world was generated. This growth of storage capacity, leading to the emergence of data management systems where data is stored in a distributed way, but accessed as well as analyzed as if it resides on a single machine. Different types of data as well as continuity in data availability has become more important than ever and expects data to be available 24×7 and from everywhere.

Structured Query Language (SQL) (McFadyen & Kanabar, 1991) became the standard of data processing because it contains such as data definition, data manipulation and data querying, all under one umbrella. RDBMS have always been distinguished by the ACID (Atomicity, Consistency, Isolation, Durability) principle set that ensures that

The final authenticated version is available online at http://doi.org/10.1007/978-3-030-27713-0

data integrity is preserved at all costs. RDBMS (Relational database management systems) can guarantee performance on the order of thousands of transactions per second, but in this time online transaction processing (OLTP) (Plattner & Hasso, 2009) in stages such as games, advertising, fraud detection and risk analysis involves more than million transactions per second that traditional RDBMS cannot easily handle. High availability without any point of failure as well as durability such as challenges have created a new wave processing database solutions, that manages data in structured unstructured ways.

New type of data management solutions are emerging to handle distributed content on open platforms. Unstructured data, non-relational databases, distributed architectures and big data analysis can change the way in which data is stored and analyzed, with the aim of obtaining useful information for making decisions in real time.

The objective of this paper is to address the problem of storage, recovery and analysis of large volumes of data using non-traditional databases.

The rest of this paper is organized as follows. Section 2 explains the generalities of big data, as well as the challenges that must be faced; Section 3 explains the Related technology and your convergence with big data; Section 4 addresses the generalities of the database administrators and in particular NewSQL solutions. In the following section the working environment and framework used is detailed. Finally, the developed tests, conclusions and future works are explained.

2 Big data characterization

Under the explosive increase of global data, the term of big data is mainly used to describe enormous datasets (Kacfah Emani, Cullot, & Nicolle, 2015). Big data is a term utilized refer to the increase in the volume of data that are difficult to store, process, and analyze through traditional database technologies. Compared with traditional datasets, big data often includes masses of unstructured data that need more real-time analysis. In addition, big data also brings new opportunities for discovering new values, helps us to gain an in depth understanding of the hidden values, and also incurs new challenges, ex. how to organize and manage such datasets (Oussous, Benjelloun, Ait Lahcen, & Belfkih, 2018) .

Nowadays, big data related to the service of Internet companies grow rapidly. For example, Google processes data of hundreds of Petabyte (PB), Facebook generates log data of over 10 PB per month, Baidu, a Chinese company, processes data of tens of PB, and Taobao, generates data of tens of Terabyte (TB) for online trading per day. While the amount of large datasets is drastically rising, it also brings many challenging problems demanding prompt solutions:

- The latest advances of information technology (IT) make it more easily to generate data. Therefore, we are confronted with the main challenge of collecting and integrating massive data from widely distributed data sources.
- The rapid growth of cloud computing and the Internet of Things (IoT) further promote the sharp growth of data. Cloud computing provides safeguarding, access sites and channels for data asset. In the paradigm of IoT, sensors all over the world are collecting and transmitting data to be stored and processed

in the cloud. Such data in both quantity and mutual relations will far surpass the capacities of the IT architectures and infrastructures of existing enterprises, and its real-time requirement will also greatly stress the available computing capacity. The increase growing data cause a problem of how to store and manage such huge heterogeneous datasets with moderate requirements on hardware and software infrastructure.

- In consideration of the heterogeneity, scalability, real-time, complexity, and privacy of big data, we shall the datasets at different levels during the analysis, modeling, visualization, and forecasting, so as to reveal its intrinsic property and improve the decisionmaking.

According to the analyzed, big data is an abstract concept. In general, big data shall mean the datasets that could not be perceived, acquired, managed, and processed by traditional IT and software/hardware tools within a tolerable time. Due to different concerns, scientific and technological enterprises, research scholars, data analysts, and technical practitioners have different definitions of big data. So, there have been considerable discussions between industry and academia about the definition of big data. In addition to developing a proper definition, the big data research should also focus on how to extract its value, how to use data, and how to transform "a bunch of data" into "big data" (Barrionuevo et al., 2018). In that sense, NIST defines big data (NIST, n.d.) as *"Big data shall mean the data of which the data volume, acquisition speed, or data representation limits the capacity of using traditional relational methods to conduct effective analysis or the data which may be effectively processed with important horizontal zoom technologies",* which focuses on the technological aspect of big data. It indicates that efficient methods or technologies need to be developed and used to analyze and process big data.

2.1 Big data challenges

The sharply increasing of data in the big data's age brings huge challenges on data acquisition, storage, management and analysis. Traditional data management and analysis systems are based on the relational database management system (RDBMS). However, such RDBMSs only apply to structured data, others semi-structured or unstructured data. In addition, RDBMSs are using more and more expensive hardware. It is apparently that the traditional RDBMSs could not handle the huge volume and heterogeneity of big data.

In (Mukherjee, Mishra, & Mishra, 2019) discuss obstacles in the development of big data applications. The key challenges are listed as follows:

- *Data representation:* many datasets have certain levels of heterogeneity in type, structure, semantics, organization, granularity, and accessibility. Data representation aims to make data more meaningful for computer analysis and user interpretation. However, an improper data representation will reduce the value of the original data and may even obstruct effective data analysis. Efficient data representation shall reflect data structure, class, and type, as well as integrated technologies, so as to enable efficient operations on different datasets.

- *Analytical mechanism:* the analytical system of big data shall process masses of heterogeneous data within a limited time. However, traditional RDBMSs are strictly designed with a lack of scalability and expandability, which could not meet the performance requirements. Non-relational (NoSQL) databases have shown their unique advantages in the processing of unstructured data and started to become mainstream in big data analysis. Even so, there are still some problems of non-relational databases in their performance and particular applications. We shall find a compromised solution between RDBMSs and non-relational databases.
- *Expendability and scalability:* the analytical system of big data must support present and future datasets. The analytical algorithm must be able to process expanding and more complex datasets.
- *Cooperation:* analysis of big data is an interdisciplinary research, which requires experts in different fields cooperate harvest the potential of big data. A comprehensive big data network architecture must be established to help scientists and engineers in various fields access different kinds of data and fully utilize their expertise, so as to cooperate to complete the analytical objectives.

3 Related technology

In order to gain a deep understanding of big data, is necessary introduce a fundamental technology that are closely related to big data: cloud computing. There is no a consensual definition of Cloud Computing yet. One of the most cited definition is the NIST's (Mell & Grance, 2011), where Cloud Computing is defined as being *"a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction"*.

Cloud computing is a technology to perform massive-scale and complex computing. It eliminates the need to maintain expensive computing hardware, dedicated space, and software. Big data is the object of the intensive computation operation and stresses the storage capacity of a cloud system. The main objective of cloud computing is to use huge computing and storage resources under concentrated management, so as to provide big data applications with fine grained computing capacity (Hashem et al., 2015). The development of cloud computing provides solutions for the storage and processing of big data. The distributed storage technology based on cloud computing can effectively manage big data; the parallel computing capacity by virtue of cloud computing can improve the efficiency of acquisition and analyzing big data.

Through the use of the cloud, access to software, hardware, and IaaS delivered over the Internet and remote data centers is possible. Cloud services have become a powerful architecture to perform complex large scale computing tasks and include a range of IT functions from storage and computation to database and application services. The need to store, process, and analyze large amounts of datasets has driven many organizations and people to adopt cloud computing (Liu, 2013).

Cloud computing and big data are conjoined. Big data provides to users the ability to use commodity computing to process distributed queries across multiple datasets and return resultant sets in a timely manner. Cloud computing provides the underlying engine through the use of distributed data-processing platforms. Thanks to cloud, big data are stored in a distributed fault-tolerant database and processed through a programming model for large datasets with a parallel algorithm distributed in a cluster.

Cloud computing has a leverage effect on Big Data, providing the computing and storage resources necessary to Big Data applications. The inherent characteristics of Cloud Computing, such as elasticity, scalability, automation, fault tolerance, and ubiquity offer an ideal environment for the development of Big Data applications.

4 Big Data database

One of the challenges that confront organizations dealing with Big Data is how and where to store the tremendous amount of data. In this context, the most widespread data management technology is relational database management systems (RDBMS). The data is stored in a structured way in form of tables or Relations. With advent of Big Data however, the structured approach falls short to serve the needs of Big Data systems which are primarily unstructured in nature. Increasing capacity of SQL although allows huge amount of data to be managed, it does not really count as a solution to Big Data needs, which expects fast reply and quick scalability (Pokorný & Jaroslav, 2015) (Madden, 2012).

To solve this problem a new kind of Database system called NoSQL (Jing Han, Haihong E, Guan Le, & Jian Du, 2011) was introduced to provide the scalability and unstructured platform for Big Data applications. NoSQL doesn't only stand SQL. NoSQL databases consist of a value pair key, documents, graph databases or wide column stores which do not have a standard schema which it needs to follow. It is also horizontally Scalable as opposed to vertical scaling in RDBMS. NoSQL provides great promises to be a perfect database system for Big Data applications; however that doesn't reach because of some major drawbacks like NoSQL does not guarantee ACID properties (Atomicity, Consistency, Isolation and Durability) of SQL systems. It is also not compatible with earlier versions of database.

This is where NewSQL (Kumar, Gupta, Maharwal, Charu, & Yadav, 2014) is a latest development in the world of database systems. NewSQL is a Relational Database with the scalability properties of NoSQL. You can define NewSql as a next generation scalable relational database management systems (RDBMS) for Online Transaction Processing (OLTP) that provide scalable performance of NoSQL systems for reading writing workloads, as well as maintaining the ACID guarantees of a traditional database system.

4.1 Architecture NewSQL

Traditional databases cannot deliver capacity on demand that application development might be hindered by all the work required to make the database scale. To overcome scalability challenges, developers add scaling like separating, sharding and clustering techniques. Another common approach is to add larger machines at more cost. An ideal DBMS should scale elastically, allowing new machines to be introduced to a running database and become effective immediately. Therefore, To adopt scale-out performance, DBMS that has been re-defined relational database technology and implement web-scale distributed database technology to tackle the multiple challenges associated with cloud computing and the rise of global application deployments (Moniruzzaman, 2014).

Important Characteristics of NewSQL Solutions:

- NewSQL provides feature SQL as the primary mechanism for application interaction.
- NewSQL support ACID properties for transactions.
- NewSQL controls a non-locking concurrency control mechanism which is helpful for the real-time reads will not conflict with writes.
- NewSQL (dbShards) architecture providing much higher per-node performance than available fromtraditional RDBMS solutions.
- NewSQL support a scale-out, parallel, sharednothing architecture, capable of running on a large number of nodes without suffering bottlenecks.
- NewSQL systems are approximately 50 times faster than traditional OLTP RDBMS.

NewSQL databases provide an SQL query interface, and clients (users and applications) interact with them the same way they interact with relational databases. They manage read/write conflicts using non-lock concurrency control.

4.2 Selected tool

There are many NewSQL databases being used in big data era, for real time web and big data applications. Every database is having a particular data format for storing its data. Hence a new customer faces the problem of selecting the appropriate NewSQL database that can be used for his business requirements, while migrating from relational database. On the other hand, there is another paradigm of shifting the big data applications from the physical infrastructure into the virtualized data centers in computational clouds. For this reason it has been decided to work with cloud computing as support architecture, more specifically Google Cloud Platform and Spanner as a tool.

Google Cloud Platform (GCP) (Google, n.d.) is a suite of cloud computing services that runs on the same infrastructure that Google contained in Google's data centers. Alongside a set of management tools, it provides a series of modular cloud services including computing, data storage, data analytics and machine learning. In particular, Storage services which provides a variety of storage services, including: Cloud SQL (MySQL or PostgreSQL databases), Cloud Spanner and two options for NoSQL data storage (Cloud Datastore and Cloud Bigtable).

Cloud Spanner (Google, 2017) (Corbett et al., 2013) is a scalable, globally distributed database designed, built, and deployed at Google. At the highest level of abstraction, it is a database that shards data acros many sets of Paxos state machines in datacenters spread all over the world. Replication is used for global availability and geographic locality; clients automatically failover between replicas. Spanner automatically reshards data across machines as the amount of data or the number of servers changes, and it automatically migrates data across machines (even across datacenters) to balance load and in response to failures.

5 Proposed work

In the present work, the performance comparison between a MySQL database and the Spanner database was performed, both working on the Google Cloud Platform.

Due to the need to work with large amounts of data, public data sets are used, in this case the data sets provided by the government of the Autonomous City of Buenos Aires are used. The first data set used contains data from the "Sistema Único de Atención Ciudadana" (SUACI), which is responsible for addressing the needs and claims of the residents of the CABA. This dataset is organized by year. For this work, only the year 2018 has been taken, which has 895,000 records with the following fields: contacto, periodo, categoria, subcategoria, concepto, tipo_prestacion, fecha_ingreso, hora_ingreso,domicilio_cgpc, domicilio_barrio, domicilio_calle, domicilio_altura, domicilio_es.

The second dataset used shows a register of the streets of the CABA, where the name, meaning and code of the streets are included among other data.

To perform the performance evaluation, three queries were defined:

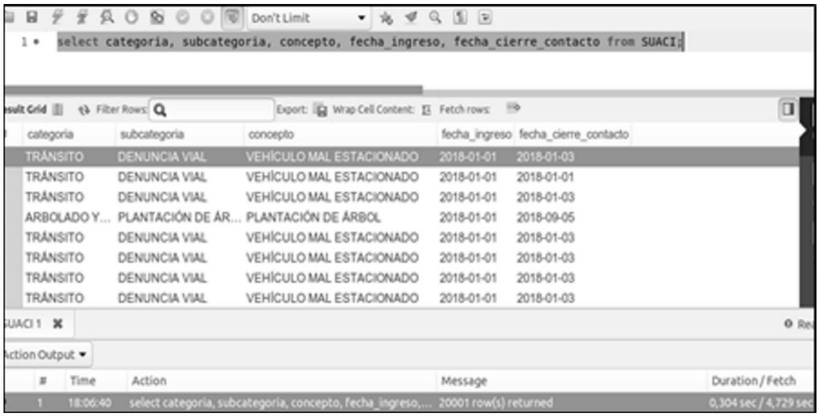
- **Query 1:** `select count(*) from SUACI`
- **Query 2:** `select categoria, subcategoria, concepto, fecha_ingreso, fecha_cierre_contacto from SUACI`
- **Query 3:** `select RECLAMO.categoria, RECLAMO.concepto, RECLAMO.fecha_ingreso, RECLAMO.domicilio_calle as nombre_calle, CALLE.codigo as codigo_calle from SUACI RECLAMO, CALLEJERO CALLE WHERE RECLAMO.domicilio_calle = CALLE.nomoficial limit 11000;`

The three queries were executed with the same syntax in both databases because the implementation of the SQL language in both databases is similar, however it is necessary to emphasize that the data types used by MySQL are different from those uses Spanner, so the migration of data between both databases is not direct and may require adjustments.

The previously defined queries will be executed on three work scenarios: 2000, 10000 and 20000 records. This will allow to evaluate the behavior performance of MySQL and Spanner in front of the increase of workload.

To execute the queries to the MySQL database, the "MySQL Workbench" client was used, while to consult the Google Spanner database the console that provides the tool itself was used in the GCP.

In the case of the MySQL database, the way used to measure time is to connect the proxy "cloud_sql_proxy" provided by the GCP platform with the MySQL database through the console, then connect the MySQL Workbench client and subsequently execute the query and record the time (see Fig.1). In the case of the Google Spanner database, the query is simply executed in the console provided by the tool and the time is recorded (see Fig.2).

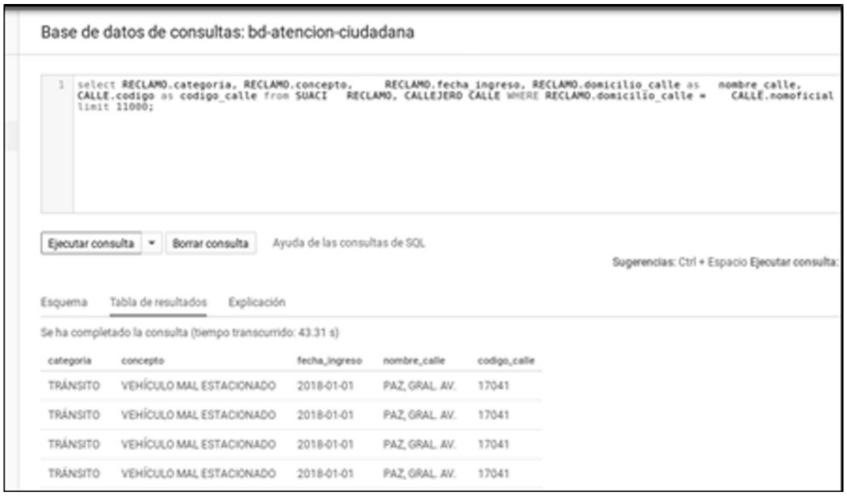


```
select categoria, subcategoria, concepto, fecha_ingreso, fecha_cierre_contacto from SUACI;
```

categoria	subcategoria	concepto	fecha_ingreso	fecha_cierre_contacto
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-03
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-01
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-03
ARBOLADO Y...	PLANTACIÓN DE ÁR...	PLANTACIÓN DE ÁRBOL	2018-01-01	2018-09-05
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-03
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-03
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-03
TRÁNSITO	DENUNCIA VIAL	VEHÍCULO MAL ESTACIONADO	2018-01-01	2018-01-03

#	Time	Action	Message	Duration / Fetch
1	18:06:40	select categoria, subcategoria, concepto, fecha_ingreso,...	20001 row(s) returned	0,304 sec / 4,729 sec

Fig. 1. Query execution in MySQL


Ejecutar consulta
Borrar consulta
Ayuda de las consultas de SQL
Sugerencias: Ctrl + Espacio Ejecutar consulta:
Esquema
Tabla de resultados
Explicación

Base de datos de consultas: bd-atencion-ciudadana

```
select RECLAMO.categoria, RECLAMO.concepto,     RECLAMO.fecha_ingreso, RECLAMO.domicilio_calle as   nombre_calle,
CALLE.codigo as codigo_calle from SUACI   RECLAMO, CALLEJERO CALLE WHERE RECLAMO.domicilio_calle =    CALLE.nomoficial
limit 11000;
```

Se ha completado la consulta (tiempo transcurrido: 43.31 s)

categoria	concepto	fecha_ingreso	nombre_calle	codigo_calle
TRÁNSITO	VEHÍCULO MAL ESTACIONADO	2018-01-01	PAZ, GRAL. AV.	17041
TRÁNSITO	VEHÍCULO MAL ESTACIONADO	2018-01-01	PAZ, GRAL. AV.	17041
TRÁNSITO	VEHÍCULO MAL ESTACIONADO	2018-01-01	PAZ, GRAL. AV.	17041
TRÁNSITO	VEHÍCULO MAL ESTACIONADO	2018-01-01	PAZ, GRAL. AV.	17041

Fig. 2. . Query execution in Google Spanner

For both cases, five measurements are made for each query and the average is calculated, which will be taken as the final result.

5.1 Results analysis

The average times obtained for both databases are analyzed below. These times are expressed in seconds.

Execution with MySQL

Prior to the execution of the consultations it is necessary that the insertion of the records in the database is relied upon. Although these times are not of interest for the evaluation of the performance of the database in front of different workloads, if they are useful when evaluating which is the option that allows a data upload to the cloud faster. Table 1 shows the times incurred in the insertion of the corresponding records.

Table 1. Insertion times for MySQL

	2000 Records	10000 Records	20000 Records
MySQL	1856,77	9318,88	15482,95

In Table 2, the results of the executions of the three queries for 2000, 10000 and 20000 records are shown. Each value is the result of the average of five executions of each query. In Fig. 3, the results shown in Table 2 can be seen graphically.

Table 2. Execution times for MySQL

	2000 Records	10000 Records	20000 Records
Query 1	0,2494	0,2404	0,2666
Query 2	0,25	0,2624	0,3266
Query 3	0,2624	0,2678	0,2714

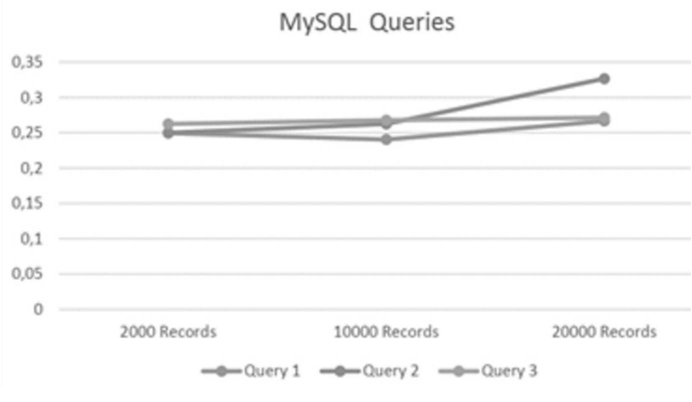


Fig. 3. Execution times for MySQL

Execution with Spanner

As in the case of the MySQL database, previous to the execution of the consultations it is necessary that the insertion of the records in the database is relied upon. Table 3 shows the times incurred in the insertion of the corresponding records.

Table 3. Insertion times for Spanner

	2000 Records	10000 Records	20000 Records
Spanner	34,21	39,72	41,75

In Table 4, the results of the executions of the three queries for 2000, 10000 and 20000 records are shown. Each value is the result of the average of five executions of each query. In Fig. 4, the results shown in Table 4 can be seen graphically.

Table 4. Execution times for Spanner

	2000 Records	10000 Records	20000 Records
Query 1	0,008878	0,03596	0,059554
Query 2	0,028214	0,054388	0,069064
Query 3	0,22888	0,24962	0,26448

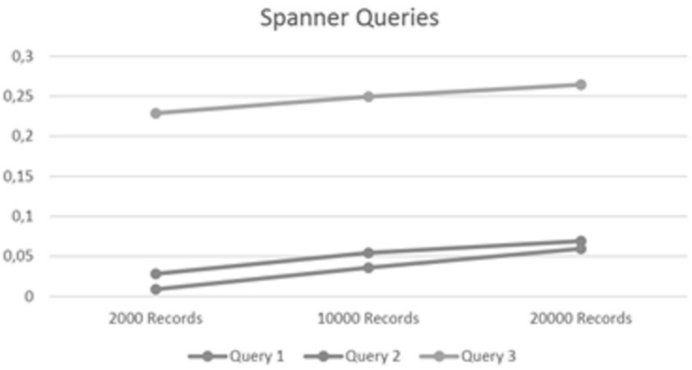


Fig. 4. Execution times for Spanner

Analyzing Tables 1 and 3 you can see that the insertion times of the registers in the database are significantly higher for MySQL. On the other hand, analyzing Tables 2 and 4, the best execution times for queries are obtained with Google Spanner. These results indicate that in the face of an increase in the number of registrations and the increased complexity of the queries, Google Spanner behaves more efficiently.

On the other hand, Table 5 summarizes the average times incurred by each manager in executing the queries. In this table it can be seen that in all cases Spanner has shorter execution times than MySQL, however for query 3 (the query with the greatest complexity because it contains a join) execution times are on average longer than for queries 1 and 2 (see Fig.5). This leads us to the preliminary conclusion that Spanner has better performance than MySQL when the workloads are high, but this efficiency decreases when the complexity of the queries increases.

Table 5. Average execution times

	MySQL	Spanner
Query 1	0,2521	0,0348
Query 2	0,2797	0,0506
Query 3	0,2672	0,2477

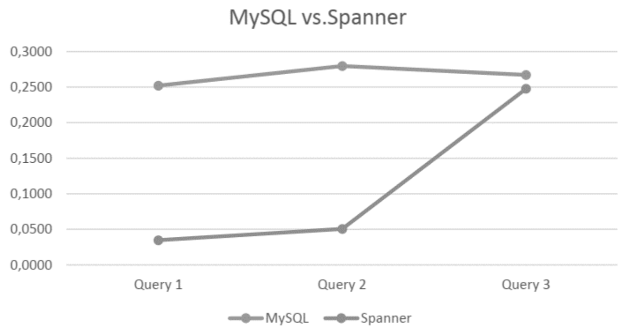


Fig. 5. Average execution times

6 Conclusion and Future work

NewSQL solutions aim to bring the relational data model into the world of NoSQL. Since there are a wide variety of NewSQL databases available now days, a new customer wishing to switch from the traditional physically hosted database server design to using NewSQL on the cloud, faces the problem of selecting the appropriate NewSQL database that can be used for his business needs without drastically changing his existing application. That is why this research offers a performance evaluation between a NewSQL database and a traditional database, MySQL, both hosted and running in the cloud; in order to measure the behavior against the increase of records and complexity of the queries. According to the experiments performed, the use of Spanner offers better results than MySQL in the face of workload and complexity of the queries.

It is proposed as future work, the use of greater workload which will be implemented by increasing records in the database and increasing complexity in queries.

References

1. Barrionuevo, M., Lopresti, M., Lucero, M., Miranda, N. C., Pérez Monte, C., Murazzo, M. A., … Printista, A. M. (2018). Estrategias y análisis orientados al manejo de datos masivos usando computación de alto desempeño. Retrieved from http://sedici.unlp.edu.ar/handle/10915/68233
2. Bernard Marr. (n.d.). How Much Data Do We Create Every Day? The Mind-Blowing Stats Everyone Should Read. Retrieved March 18, 2019, from https://www.forbes.com/sites/bernardmarr/2018/05/21/how-much-data-do-we-create-every-day-the-mind-blowing-stats-everyone-should-read/#61fd023f60ba
3. Corbett, J. C., Dean, J., Epstein, M., Fikes, A., Frost, C., Furman, J. J., … Woodford, D. (2013). Spanner: Google's Globally Distributed Database. *ACM Transactions on Computer Systems (TOCS)*, *31*(3), 8. https://doi.org/10.1145/2491245
4. Google. (n.d.). Google Cloud Platform Overview | Overview | Google Cloud. Retrieved April 1, 2019, from https://cloud.google.com/docs/overview/
5. Google. (2017). Cloud Spanner Documentation.
6. Gubbi, J., Buyya, R., Marusic, S., & Palaniswami, M. (2013). Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, *29*(7), 1645–1660. https://doi.org/10.1016/J.FUTURE.2013.01.010
7. Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., & Ullah Khan, S. (2015). The rise of "big data" on cloud computing: Review and open research issues. *Information Systems*, *47*, 98–115. https://doi.org/10.1016/J.IS.2014.07.006
8. Jing Han, Haihong E, Guan Le, & Jian Du. (2011). Survey on NoSQL database. In *2011 6th International Conference on Pervasive Computing and Applications* (pp. 363–366). IEEE. https://doi.org/10.1109/ICPCA.2011.6106531
9. Kacfah Emani, C., Cullot, N., & Nicolle, C. (2015). Understandable Big Data: A survey. *Computer Science Review*, *17*, 70–81. https://doi.org/10.1016/j.cosrev.2015.05.002
10. Kumar, R., Gupta, N., Maharwal, H., Charu, S., & Yadav, K. (2014). Critical Analysis of Database Management Using NewSQL. *International Journal of Computer Science and Mobile Computing*, *35*(5), 434–438. Retrieved from http://s3.amazonaws.com/academia.edu.documents/33752078/V3I5201499a2.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1495314606&Signature=VhHyc%2BR0An%2FF8Oa6W5EAkCgxO9c%3D&response-content-disposition=inline%3B filename%3DCritical_Analysis_of_Database_Management.pdf
11. Liu, H. (2013). Big Data Drives Cloud Adoption in Enterprise. *IEEE Internet Computing*, *17*(4), 68–71. https://doi.org/10.1109/MIC.2013.63
12. Madden, S. (2012). From Databases to Big Data. *IEEE Internet Computing*, *16*(3), 4–6. https://doi.org/10.1109/MIC.2012.50
13. McFadyen, R., & Kanabar, V. (1991). *An introduction to structured query language*. Wm. C. Brown. Retrieved from https://dl.acm.org/citation.cfm?id=102896
14. Mell, P., & Grance, T. (2011). The NIST Definition of Cloud Computing.
15. Moniruzzaman, A. B. M. (2014). NewSQL: Towards Next-Generation Scalable RDBMS for Online Transaction Processing (OLTP) for Big Data Management. Retrieved from http://arxiv.org/abs/1411.7343
16. Mukherjee, S., Mishra, M. K., & Mishra, B. S. P. (2019). Promises and Challenges of Big Data in a Data-Driven World (pp. 201–211). Springer, Singapore. https://doi.org/10.1007/978-981-13-1498-8_18

17. NIST. (n.d.). NIST Big Data Working Group (NBD-WG). Retrieved March 18, 2019, from https://bigdatawg.nist.gov/home.php
18. Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, *30*(4), 431–448. https://doi.org/10.1016/J.JKSUCI.2017.06.001
19. Plattner, H., & Hasso. (2009). A common database approach for OLTP and OLAP using an in-memory column database. In *Proceedings of the 35th SIGMOD international conference on Management of data - SIGMOD '09* (pp. 1–2). New York, New York, USA: ACM Press. https://doi.org/10.1145/1559845.1559846
20. Pokorný, J., & Jaroslav. (2015). Database technologies in the world of big data. In *Proceedings of the 16th International Conference on Computer Systems and Technologies - CompSysTech '15* (pp. 1–12). New York, New York, USA: ACM Press. https://doi.org/10.1145/2812428.2812429