Jurnal Kejuruteraan SI 4(2) 2021: 65-69
https://doi.org/10.17576/jkukm-2021-si4(2)-10

Performance Comparison between NoSQL (RethinkDB) and MySQL Database Replication from Master to Slave in Big Data

(Perbandingan Prestasi Replikasi dari Master ke Slave dalam Data Raya di antara Pangkalan Data Jenis NoSQL (RethinkDB) dan MySQL)

Dildar Hussain[a], Mohammad Omar[a], Khairol Amali Ahmad[b], Ja'afar Adnan[b] & Khaleel Ahmad[a,*]

[a]*Department of Computer Science & Information Technology, Maulana Azad National Urdu University, India,*
[b]*Department of Electrical & Electronic Engineering, Faculty of Engineering, National Defence University of Malaysia*

**Corresponding author: khaleelahmad@manuu.edu.in*

Received 5 April 2021, Received in revised form 27 May 2021
Accepted 30 September 2021, Available online 31 October 2021

ABSTRACT

When database is stored in one computer or server, certain issues related to disaster (in terms of natural or un-natural) problem may arise with reference to geographical connection and geographical distribution of servers. Here, replication feature plays an important role. There were many techniques and methodologies for geographically distributed servers, but the problem was of one master. When an authorized user updates his database on a client database server then this client database server updates his master database server and at last, the master database server updates all its client's database servers. The described process is called database replication. In this process, there are several main parameters that are important, i.e., bandwidth, memory, and processor time. In this paper, the analysis is focused on comparing which database (NoSQL family-RethinkDB and MySQL) utilize how much memory, processor, and time during replication of database from client to master.

Keywords: MySQL; NoSQL; replication; Master and Slave Database; USN; DRBD; RSYNC

ABSTRAK

Apabila pangkalan data disimpan di dalam sesebuah komputer atau pelayan, masalah mungkin timbul atau isu-isu disebabkan oleh bencana (samada secara semulajadi atau tidak) ataupun masalah berkaitan dengan sambungan dan kedudukan geografi lokasi pelayan-pelayan. Dalam hal ini, ciri-ciri replikasi memainkan peranan yang penting. Terdapat banyak teknik dan kaedah dalam penempatan pelayan-pelayan di lokasi-lokasi geografi yang pelbagai, namun permasalahan wujud disebabkan oleh satu "master". Apabila pengguna yang dibenarkan mengemaskini pangkalan data pada sebuah pelayan pangkalan data pelanggan, pelayan tersebut akan mengemaskini pelayan pangkalan data "master"nya dan seterusnya, pelayan pangkalan data master tersebut akan mengemaskini semua pelayan pangkalan data pelanggannya yang lain. Proses ini dikenali sebagai replkasi pangkalan data. Di dalam proses ini, terdapat beberapa parameter penting seperti lebarjalur, memori, dan masa memproses. Di dalam kertas ini, fokus analisa adalah terhadap perbandingan pangkalan data (diantara keluarga NoSQL-RethinkingDB dan MySQL) dalam penggunaan memori, prosesor, dan jumlah masa yang diambil semasa replikasi pangkalan data dari pelanggan kepada master.

Kata Kunci: MySQL; NoSQL; replikasi; Pangkalan Data Master dan Slave; USN; DBD; RYSYNC

INTRODUCTION

The era of Industrial Revolution 4.0 has the characteristic of utilizing big data. There are many types of data that are more and more becoming relevant and important for various new applications such as for social networks (Lv et al. 2017). There are many more databases developed for storage and each and every database has its special feature. Oracle RDBMS, Microsoft SQL Server, and IBM DB2 are several examples of databases. Lately, a family of NoSQL database have been developed in which varieties of databases are available and their usage is quite popular in big data analytics (Matthew & Kumar 2015). It is important for the database to be updated regularly regardless of any organization, may it be large or small organization. These data should be available for everyone in that organization. For example, when an employee in India updates his "Aadhaar" identification number with his mobile, it is necessary that its number should be updated throughout all databases in that organization. This data update is called replication of data. In a simple way, we can say that a replication is a process by which copies of directory data are created and maintained in several domain controllers (Zawodny & Balling 2004). This data replication is important in increasing data availability to users simply because by having the data copied from one database to another, it will allow all users to use the same data without interfering each other. Hence, fast data transactions and queries are possible. In addition, multiple copies of database also support testing and operational reporting.

DATABASE REPLICATION

Data replication approaches aim to strike a balance between system performance and data consistency. Database replication may be accomplished in several approaches, such as snapshot, merging, and transactional replications. For snapshot approach, data on one server is basically copied to another server (or from one database to another database on the same server); merging replication combines data from two or more databases into a single database; and, in transactional replication, initially, the user systems receive full complete copies of the database but later only receive needed updates as data changes.

Many database management systems (DBMS) usually perform data replication using master-slave concept between the original data and its copies. Any updates will be logged by the master and followed by the slaves. Each slave then output the acknowledgement of successful updates and ready for the upcoming updates.

To replicate the data, there is a single writable Primary Domain Controller (PDC) which is used for writing purposes and a Backup Domain Controller (BDC) for reading read-only (Microsoft Docs 2012). This concept of master and slave replication methodology was available on Windows Server 2000 and Windows NT. Later on, a new methodology was introduced which is called a Multi-master approach in which every Domain Controller is a master and able to replicate the data to other domains. However, this approach is more costly and complex that it may be impractical in certain situations. Transactional conflict prevention or resolution is one of the most common issues faced by multi-master replication approach. While conflict prevention is performed by most synchronous (also known as eager) replication solutions, the asynchronous (also known as lazy) solutions resort to conflict resolution techniques.

Replication works on the basis of three different directories (Oracle 2020) which are schema partition, configuration partition, and the last one is domain partition. The definition of objects and objects' attributes are held by schema container and an update of this schema is replicated by the one-to-many method. The physical layout contains in configuration container and its replication methodology is same as the schema partition.

The replication of active directory is based on the attributes of data. Just take the previous example in which one employee updates his mobile number. The mobile number is an attribute of the employee. If we will modify any attribute of an object, then only the attribute will be modified. It will not modify the object because when an object is created, an update sequence number (USN) (Wiesmann 2000) is assigned to it. Whenever we change any attribute of an object, the USN is incremented. With it, the USN of another domain controller updates their USN. Through the replication, the copy of one data from one database will be available on the other database. If we are unable to find the particular data in a database then we can find this data in another database with the help of management server (Jones & Zarmer 1997). This management server is also called partner. A site may have many partners. The remote site and main site connected to each other with the help of these partners. These partners may either be replication partner or site partner, but there is some difference between the site partner and replication partner. The replication partner can use both an embedded database or a Microsoft SQL server and these partners share a common license key.

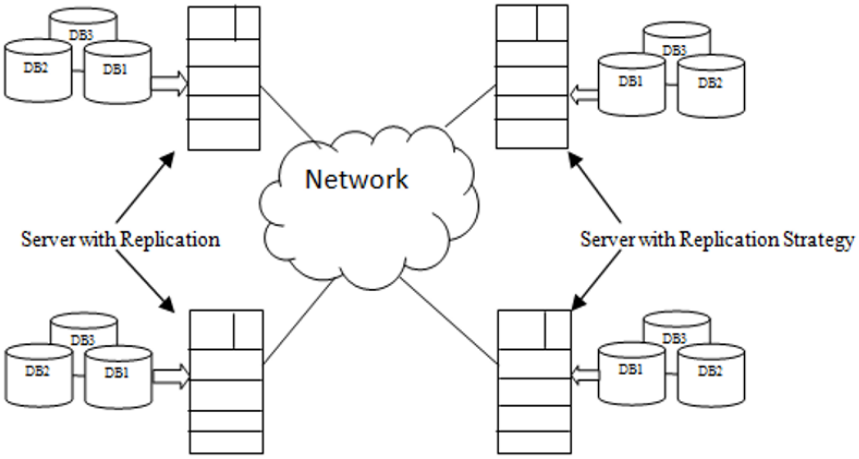

FIGURE 1. Strategy of replication of data

Figure 1 depicts how replication works between the main site and two remote sites. The process of input-output thread connects to the master database for reading the binary log events. Copy the master binary log events to the slave log file that is called a relay log (Hayes 2016). It is a very rare case that copying the replication events is slower. It is possible that it may slow a few hundred milliseconds due to network delay.

Replication strategy ensures that our current data is available at more than one place. As alluded earlier, generally there are two methods used for replication of data that are (a) Synchronous and (b) Asynchronous (Goel & Buyya 2006). In the synchronous method, when we write, delete or update our data in master database, then the slave database is also updated. In asynchronous replication method, if we are writing, updating or deleting our data in master database then with the help of an application, our data is copied or the older data is replaced in slave database. In the Linux based database, there are mainly two techniques used for replication of data, i.e., Distributed Replicated Block Device (DRBD) and RSYNC. (Riasetiawan et al. 2015; Layton 2019). In DRBD, replication works without any interference of human or machine from the master database to slave database. This DRBD technique is used for less than three databases. In RSYNC replication technique, the only directory of files is replicated with the help of human or machine interference and this technique will be best to replicate data in more than three databases.

While replication techniques were originally focused on relational database management systems, they have grown to include non-relational database types due to the growth of virtual machine applications and distributed cloud computing. However, the replication methods vary among these non-relational databases.

In this work, our main objective is to compare briefly the performance between MySQL database and NoSQL (RethinkDB) database on a particular bandwidth when we replicate our data; how much time, processor, and memory are utilized by the server during the replication.

MYSQL DATABASE VS NOSQL (RETHINKDB) DATABASE

MySQL is popularly used together with PHP to store data for websites (Bradley 2018). It has a relational database type. The term “Structured Query Language" (SQL), is the standard language for interacting with databases. Hence, MySQL is an open source database system which was built using the SQL base. As a relational database, MySQL makes use of different tables for the database and allows users to 'relate' data from one table to another.

On the other hand, in a NoSQL ("non SQL" or "non-relational") (NoSQL DEFINITION, 2020) database, the methods for storage and retrieval of data are not based on the tabular relations like those being employed in the relational databases. It is an open-source non-relational database that stores data in JavaScript Object Notation (JSON) format.

It has been developed for real-time Web applications and as such, it was designed to meet the required scalability. In order to provide the real-time push based updates to multiple users in parallel, NoSQL (RethinkDB) make use of a flexible and high-performance query language, ReQL.

METHODOLOGY

NOSQL (RETHINKDB) WITH MASTER AND SLAVE

Figure 2 shows the set-up of the NoSQL (RethinkDB) on Client Machine while Figure 3 depicts NoSQL (RethinkDB) on the server.

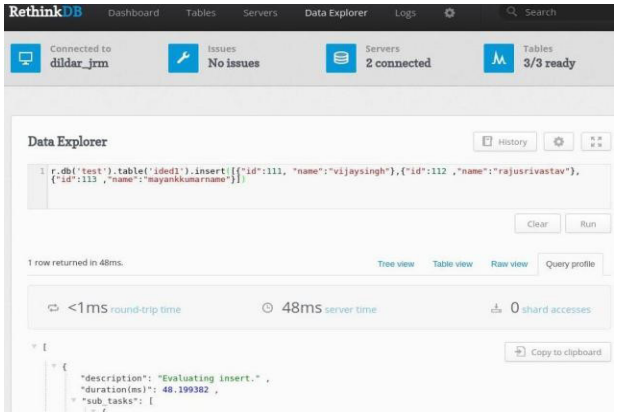

FIGURE 2. NoSQL on client machine

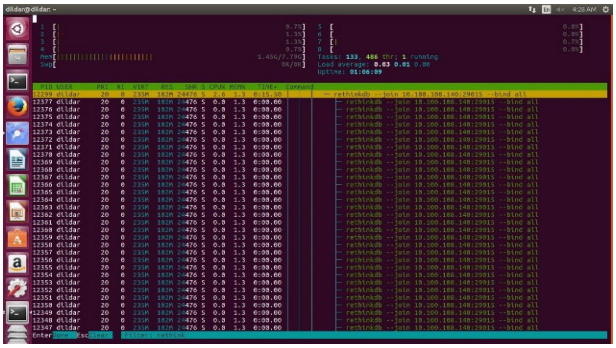

FIGURE 3. NoSQL on server

MYSQL WITH MASTER AND SLAVE

Figure 4 shows the process of MySQL query on Master Machine. Figure 5 captured the MySQL on server.

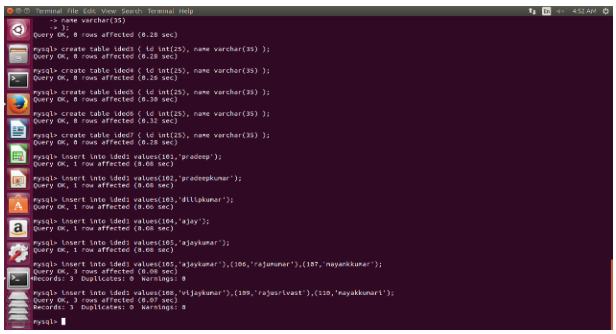

FIGURE 4. MySQL with query on Master Machine

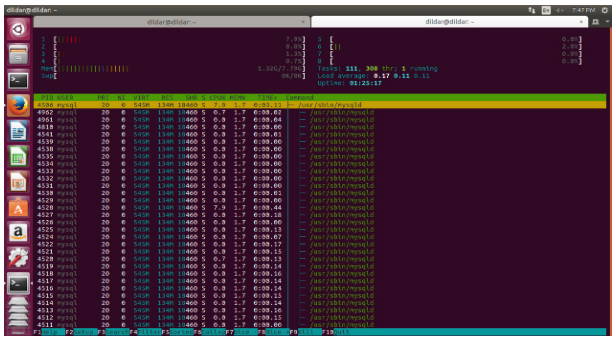

FIGURE 5. MySQL on server

EXPERIMENTAL RESULTS

Figure 6 below shows that in the comparison of CPU utilization, MySQL had a higher percentage of CPU utilization (07.09 %) as opposed to the NoSQL (RethinkDB) (02.06 %). Hence, RethinkDB is more efficient than MySQL in this aspect.

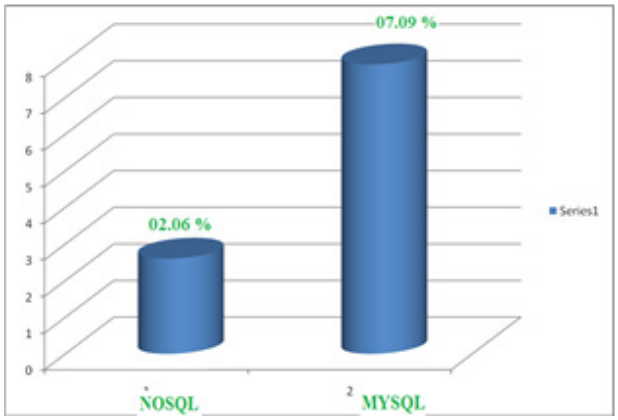

FIGURE 6. CPU (3.40 GHz) utilization

Figure 7 shows that MySQL used more memory (01.07 %) than the RethinkDB (0.103%).

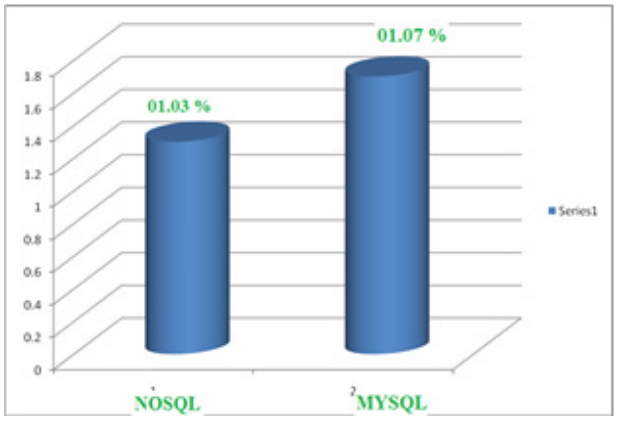

FIGURE 7. Memory (7.89 GB) utilization

In Figure 8, it can be seen that the time taken in the process is longer for the NoSQL (RethinkDB) (which was 15.50 milliseconds) as compared to the MySQL (3.11 milliseconds).

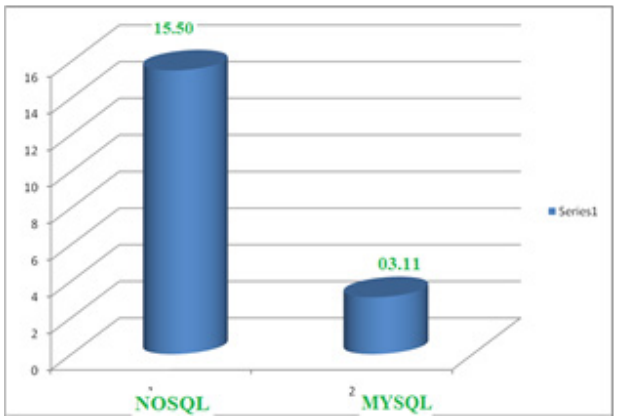

FIGURE 8. Time utilization (milliseconds)

The overall experimental results of CPU utilization, memory utilization, and time utilized by NoSQL and MySQL databases are depicted in Figure 9, where the NoSQL (RethinkDB) performed better than the MySQL in the CPU and memory utilizations but the MySQL fared better in processing time.

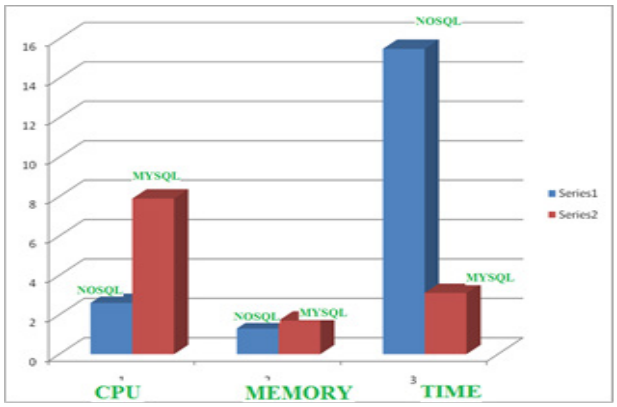

FIGURE 9. CPU, memory and time utilized by MySQL and NoSQL

CONCLUSION

According to this experiment, it shows that by using the same amount of data in NoSQL (RethinkDB) and MySQL on CPU, the CPU utilized by MySQL (07.09%) is more than NoSQL (02.06%) and the memory utilized by MySQL (01.03%) is also more than NoSQL (01.07%). However, the time taken by the NoSQL (15.50 milliseconds) is more than that of MySQL (3.11 milliseconds). Hence, here the NoSQL uses less resource but takes longer processing time during data replications as compared to MySQL.

DECLARATION OF COMPETING INTEREST

None

REFERENCES

Bradley, A. 2018. Understanding how SQL databases work. https://www.thoughtco.com/understanding-how-sql-databases-work-2693878.

Goel, S. & Buyya, R. 2006. Data replication strategies in wide area distributed systems. http://www.cloudbus.org/papers/DataReplicationInDSChapter2006.pdf.

Hayes, A. 2016. What is MySQL replication and how does it work? http://dbadiaries.com/what-is-mysql-replication-and-how-does-it-work.

Jones, A. & Zarmer, C. 1997. Synchronization and replication of object databases in Apple Computer Inc., 1997, United States Patent (No. 5,684,984)

Layton, J.B. 2019. Data replication using rsync. http://www.linuxtoday.com/blog/data-replication-using-rsync.html.

Lv, Z., Song, H., Basanta-Val, P., Steed, A. & Jo, M. 2017. Next-generation big data analytics: State of the art, challenges, and future research topics. *IEEE Transactions on Industrial Informatics.*

Mathew, A.B. & Kumar, S.D.M. 2015. Analysis of data management and query handling in social networks using NoSQL databases. *International Conference on Advances in Computing, Communications and Informatics (ICACCI).*

Microsoft Docs. 2012. How transactional replication works. https://technet.microsoft.com/en-us/library/ms151706(v=sql.105).aspx

NoSQL. 2020. NoSQL definition. http://nosql-database.org.

Oracle. n.d. Fusion Middleware Administrator's guide for Oracle internet directory: How replication works. https://docs.oracle.com/cd/E21043_01/oid.1111/e10029/oid_replic_arch.htm

Riasetiawan, M., A. Ashari, I. Endrayanto. 2015. Distributed Replicated Block Device (DRDB) implementation on cluster storage data migration. *International Conference on Data and Software Engineering.*

Wiesmann, M., Pedone, F. & Schiper, A. 2000. Understanding replication in databases and distributed systems. Proceedings 20th *IEEE International Conference on Distributed Computing Systems.*

Zawodny, J.D. & Balling, D.J. 2004. High performance *MySQL: Optimization, Backups, Replication, Load Balancing.* 1st edition. Kindle Edition.