International Journal of Advanced Computer Technology (IJACT)
ISSN:2319-7900

BENCHMARKING POPULAR OPEN SOURCE RDBMS: A PERFORMANCE EVALUATION FOR IT PROFESSIONALS

THANKGOD SANI ADEYI, Department of Mathematics, Ahmadu bello University,Zaria-Nigeria.; YUSUF ABUBAKAR., Department of Computer Science, Nuhu Bamali Polytechnic,Zaria-Nigeria; ABUDU ABIMBOLA ORIYOMI, Department of Computer Science, Federal College of Education ,Zaria-Nigeria.

Abstract

The main objective of this paper is benchmarking of popular open source relational database management systems (RDBMS) MySQL, PostgreSQL , MariaDB, SQLite, and their comparison using YCSB. The observed results are intended to make it easier for IT professional to determine which database management system is better for certain purposes. We measure the execution time against several record sizes for read, write and update operation. Each measurement was repeated twenty times with the aid of a bash script. The results were then calculated from the average times of each measurement. Comparisons of gained results shows, which database management system is better in certain areas. Comparison of these RDBMS showed, the best choice is usage of PostgreSQL RDBMS. IT professionals who plans to use any particular RDBMS, should first determine, how a selected RDBMS works and justify their selection.

Introduction

The relational database has been the most preferred database model by developers since it was developed by Edgar Codd in 1970 [1]. It is very easy to find IT professionals with Relational Database experience and there is a huge number of projects already in production that where coded using open source databases. Fast performance is key for nearly every data-driven system, and IT professionals work hard to ensure that the database they select is optimized for the success of their application use cases. One of the ways to evaluate data driven systems is to conduct experiment in the environment in which the database will run, under anticipated data and concurrent user workloads. Experiments that include the right benchmarks, such as configurations, parameters and workloads, give developers and architects powerful insight about the system under consideration. This paper examines the performance of four popular open source relational databases using the Yahoo Cloud Serving Benchmark (YCSB).

Background

Relational Database Management Systems

Relational Database Systems implement the relational model to work with the data. Relational model shapes whatever information to be stored by defining them as related entities with attributes across tables (i.e. schemas). This type of database management systems require structures (e.g. a table) to be defined in order to contain and work with the data. With tables, each column (e.g. attribute) holds a different type (e.g. data type) of information. Each record in the database, uniquely identified with keys, translates to a row that belongs to a table, with each row's series of attributes being represented as the columns of a table -- all related together, as defined within the relational model.

SQLite

SQLite is a very powerful, embedded relational database management system. It is an amazing library that gets embedded inside the application that makes use of. As a self-contained, file-based database, SQLite offers an amazing set of tools to handle all sorts of data with much less constraint and ease compared to hosted, process based (server) relational databases. When an application uses SQLite, the integration works with functional and direct calls made to a file holding the data (i.e. SQLite database) instead of communicating through an interface of sorts (i.e. ports, sockets). This makes SQLite extremely fast and efficient, and an also powerful thanks to the library's underlying technology [5].

MySQL:

MySQL is the most popular one of all the large-scale database servers. It is a feature rich, open-source product that powers a lot of web-sites and applications online. Getting started with MySQL is relatively easy and developers have access to a massive array of information regarding the database on the internet. Despite not trying to implement the full SQL standard, MySQL offers a lot of functionality to the users. As a stand-alone database server, applications talk to MySQL daemon process to access the database itself [3].

PostgreSQL

PostgreSQL is the advanced, open-source [object]-relational database management system which has the main goal of being standards-compliant and extensible. PostgreSQL, or Postgres, tries to adopt the ANSI/ISO SQL standards together with the revisions. Compared to other RDBMSs, PostgreSQL differs itself with its support for highly required and integral object-oriented and/or relational database functionality, such as the complete support for reliable transactions, i.e. Atomicity, Consistency, Isolation, and Durability (ACID).

Due to the powerful underlying technology, Postgres is extremely capable of handling many tasks very efficiently. Support for concurrency is achieved without read locks thanks to the implementation of Multiversion Concurrency Control (MVCC), which also ensures the ACID compliance. PostgreSQL is highly programmable, and therefore extendible, with custom procedures that are called "stored procedures". These functions can be created to simplify the execution of repeated, complex and often required database operations. Although this DBMS does not have the popularity of MySQL, there are many amazing third-party tools and libraries that are designed to make working with PostgreSQL simple, despite this database's powerful nature. Nowadays it is possible to get PostgreSQL as an application package through many operating-systems default package manager with ease [4].

MariaDB

MariaDB is a community-developed fork of the MySQL relational database management system, the impetus being the community maintenance of its free status under the GNU GPL. Being a fork of a leading open source software system, it is notable for being led by its original developers and triggered by concerns over direction by an acquiring commercial company Oracle. Contributors are required to share their copyright with Monty Program AB [6].

The intent is also to maintain high compatibility with MySQL, ensuring a "drop-in" replacement capability with library binary equivalency and exact matching with MySQL APIs and commands. It includes the XtraDB storage engine for replacing InnoDB, as well as a new storage engine, Aria, that intends to be both a transactional and non-transactional engine perhaps even included in future versions of MySQL.

Details of the Benchmark Tool

We used an existing tool provided by Yahoo, called the YCSB Client, to execute these benchmarks. A key design goal of this tool is extensibility as it can be used to benchmark new cloud database systems. We have used this tool to measure the performance of four NoSQL systems, as we report in the next section. This tool is available under an open source license. It has ready adapters for different NoSQL Databases. YCSB tool allows benchmarking multiple systems and comparing them by creating "workloads". Using this tool, one can install multiple systems on the same hardware configuration, and run the same workloads against each system. The architecture of YCSB is as shown in Figure 1.

ISSN:2319-7900

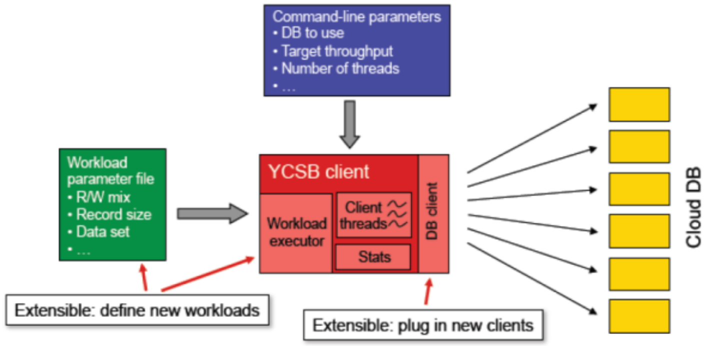

Figure 1: The YCSB Architecture [2]

Workloads

In this section we describe the set of workloads used for the experiment in this paper.

The YCSB framework contains a core set of workloads to evaluate different aspects of a system's performance, called the YCSB Core Package. In YCSB, a package is a collection of related workloads. The workload defines the data that will be loaded into the database during the loading phase, and the operations that will be executed against the data set during the transaction phase, and can be used to evaluate systems at one particular point in the performance space. A package, which includes multiple workloads, examines a broader slice of the performance space. While the core package examines several interesting performance axes, YCSB have not attempted to exhaustively examine the entire performance space. It is developed in such a way that users of can develop their own packages either by defining a new set of workload parameters, or if necessary by writing Java code. The following Workloads were considered in this report;

a) Workload A – 100% Insert

This workload writes 1000, 20000, 40000, 80000 and 100000 thousand one kilobyte (1KB) record into an empty database in each case recording the performance measurements in a text file.

b) Workload B – 100% Read

This is read intensive workload, i.e. it retrieves 1000, 20000, 40000, 80000 and 100000 thousand one kilobyte (1KB) record from a populated database in each case recording the performance measurements in a text file.

c) Workload C – 100% Update

Workload C changes a single field in a record size of 1000, 20000, 40000, 80000 and 100000 thousand one kilobyte (1KB) from a populated database in each case recording the performance measurements in a text file.

Overview of the test

Each workload was tested with 10 client threads combined with overall records of 1000, 20000, 40000, 60000, 80000 and 100000; And each of these combinations repeated 20 times. YCSB by default creates 1k size records.

The Execution time vs number of records using 10 client threads was measured. This utilizes all the cores of the test system and describes how increasing the number of records affects the average response time of a database operation.

Results

In this section, we present benchmarking an experimental evaluation of the four NoSQL Systems using the various workload described before.

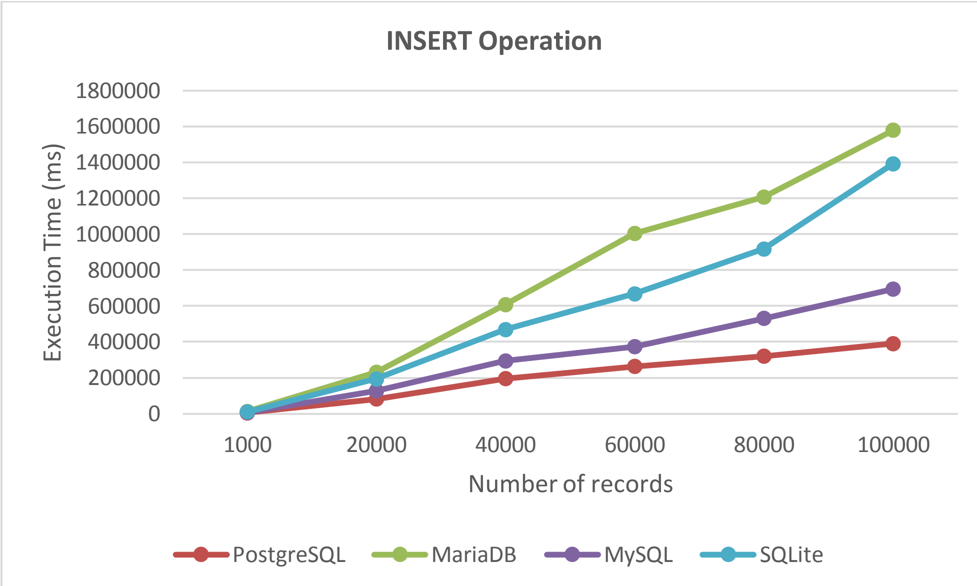

Figure 2: Graph of Execution time Against Number of records for Insert Operation.

Figure 2 shows the graph obtained by inserting 1000, 20000, 40000, 60000, 80000 and 100000 rows of records into the tested databases (the size of each record is 10kb). The result shows that PostgreSQL has the best performance, followed by MySQL, SQLite and MariaDB respectively.

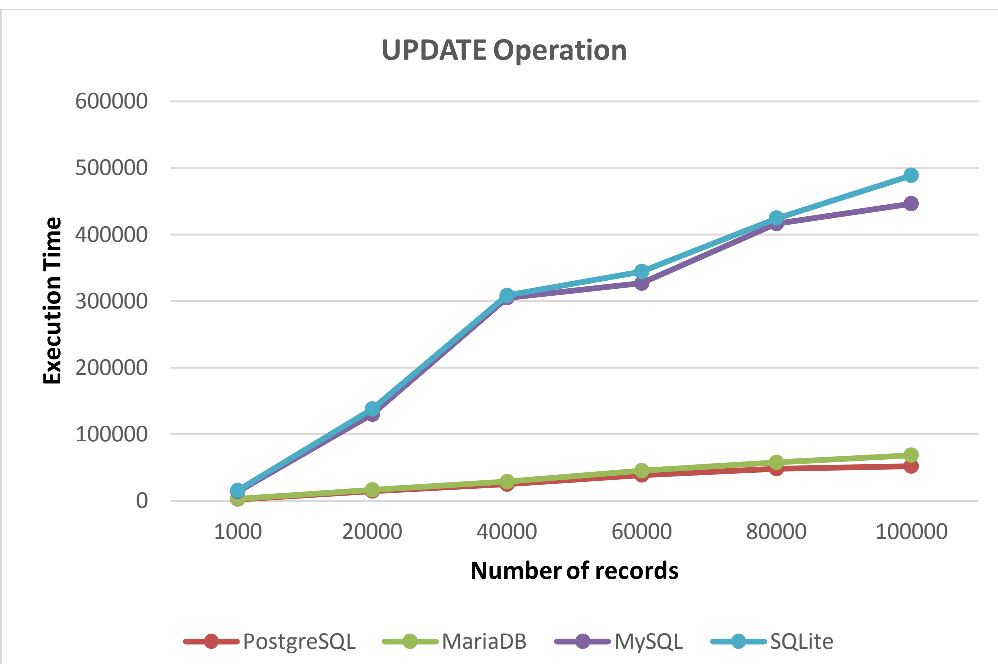

Figure 3: Graph of Execution time Against Number of records for Update Operation.

ISSN:2319-7900

Figure 3 shows the graph obtained by Updating 1000, 20000, 40000, 60000, 80000 and 100000 rows of records on the tested databases (the size of each record is 10kb) using 10 threads. The result shows that PostgreSQL has the best performance, followed by MariaDB, MySQL and SQLite respectively.

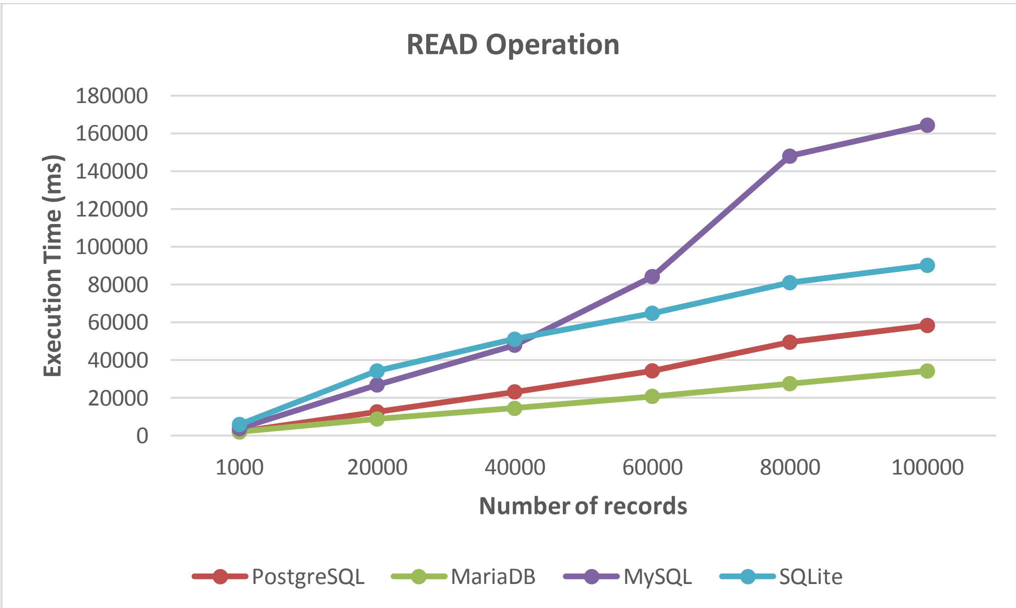

Figure 4: Graph of Execution time Against Number of records for Read Operation.

Figure 4 shows the graph obtained by reading 1000, 20000, 40000, 60000, 80000 and 100000 rows of records from the tested databases (the size of each record is 10kb). The result shows that MariaDB has the best performance, followed by PostgreSQL. MySQL outperforms SQLite for lower record size but poorer as the number of records exceeds 40000.

CONCLUSION

We have benchmarked four popular open source relational databases using yahoo cloud serving benchmark. The result of our benchmark shows, that PostgreSQL have the best performance for Insert operation, followed by Mysql, MariaDB and SQLite respectively. Even though MariaDB perform worst for Insert Operation, It has the best performance for read operation under similar workload. We observed that there are clear tradeoffs between read and write performance that result from each system's architectural decisions. These results highlight the importance of a standard framework for examining system performance so that developers can select the most appropriate system for their needs.

REFERENCES

[1] Shuxin, Y. and Indrakshi, R.. (2005) Relational database operations modeling with UML, Proceedings of the 19th International Conference on Advanced Information etworking and Applications, 927-932.

[2] B. F. Cooper et al. PNUTS: Yahoo!'s hosted data serving platform. In VLDB, 2008.

[3] MySQL. Available at: http://www.mysql.com/?bydis_dis_index=1

ISSN:2319-7900

[4] PostgreSQL. Available at: http://www.postgresql.org/

[5] SQLite. Available at: https://www.digitalocean.com/community/tutorials/sqlite-vs-mysql-vs-postgresql-a-comparison-of-relational-database-management-systems

[6] MariaDB. Available at: http://en.wikipedia.org/wiki/MariaDB

Biographies

THANKGOD SANI ADEYI received the B.S. degree in Computer Science from the Benue State University, Makurdi in 2007, the M.Sc. degree in Computer Science from the prestigious Ahmadu Bello University, Zaria, in 2014. His current research interests include NoSQL Databases, e-payment system strategies and adoptions in Nigeria and cloud computing.. He is currently a staff of Telemagic Communications Limited, Lagos, Nigeria (saniadeyi@gmail.com).

YUSUF ABUBAKAR got diploma certificate in computer science from Nuhu Bamali Polytechnic Zaria and later reeceived the B.Sc. degree in Computer Science from the Ahmadu Bello University, Zaria in 2011, He is currently about to complete his M.Sc. research work at the Ahmadu Bello University, Zaria. His current research interests include Database Performance, Java Programming e.t.c. (yus4haab@yahoo.co.uk).

ABUDU ABIMBOLA ORIYOMI ABUDU ABIMBOLA ORIYOMI received the B.S. degree in Computer Science from the Ogun State University, Ago-Iwoye in 2007. He is currently doing his M.Sc. degree in Computer Science from the prestigious Ahmadu Bello University, Zaria. His current research interests include implementing security solutions in tackling the online social network threats. He is currently working as a lecturer at Federal College of Education, Zaria, Kaduna State.(abuduabimbola@gmail.com)