Available online at www.elixirpublishers.com (Elixir International Journal)

Computer Engineering

Elixir Comp. Engg. 82 (2015) 32094-32097

Awakening to reality

Elixir
ISSN: 2229-712X

Performance and Scaling Comparison Study of RDBMS and NoSQL (MongoDB)

D.R Merlin Shalini and S.Dhamodharan
Department of Computer Applications, Easwari Engineering College, Ramapuram, Chennai-600 089.

ARTICLE INFO

Article history:
Received: 10 November 2014;
Received in revised form:
19 April 2015;
Accepted: 30 April 2015;

Keywords
Massive,
Relational Database Management Systems,
Market.

ABSTRACT

The massive amounts of data collected today by software in fields varying from academia to business and many other fields, is increasingly becoming a huge problem due to storage technologies not advancing fast enough to provide the performance scalability needed. This is even truer for data which are highly organized and require analysis while being stored in databases and being accessed by various applications simultaneously. As database vendors struggle to gain more market share new technologies emerge attempting to overcome the disadvantages of previous designs while providing more features. Two popular database types, the Relational Database Management Systems and NoSQL databases are examined. The aim of this paper was to examine and compare two databases from these two database models and answer the question of whether one performs and scales better than the other.

Introduction

Efficient storage and retrieval of data has always been an issue due to the growing needs in industry, business and academia. Larger amounts of transactions and experimentation result in massive amounts of data which require organized storage solutions. Databases were created in order to satisfy this need of storing and retrieving data in an organized manner. Since their inception in the 1960's different types have emerged, each using its own representation of data and technology for handling transactions. They began with navigational databases which were based on linked-lists, moved on to relational databases, afterwards object-oriented and in the late 2000s. NoSQL emerged and has become a popular trend [1]. Two of the most widely used database types are relational databases and NoSQL databases.

Although the two types differ in many aspects depending on the implementation they could be used for similar applications although it is not recommended as one is not meant as an alternative to the other [2]. One of the main reasons for this recommendation is the problem of NoSQL databases being less reliable compared to relational databases due to less data integrity and reliability.

Comparing these two databases is important as it allows to draw conclusions regarding their ability to process data and how they handle large amounts of transactions and data. It also provides insight to how well suited they are to today's issue of massive amounts of data collected otherwise known as Big Data. Databases play an important role in applications and the wrong choice at the beginning may have disastrous effects as it is difficult to migrate to another database system, more so a completely different type. The performance and scalability of the databases are the most important factors besides reliability when weighing the various options and comparing them for different databases can be difficult due to different designs, configurations and data access methods.

Comparison of the two types is performed in terms of performance and scalability two database systems and compares them by trying to find a middle ground where their implementations are as close as possible and the benchmarks performed do not favour one database system over the other. Amongst the most important issues are finding a data set and representing it in both databases effectively. In addition the correct choice of benchmarks is vital as stressing a database can be a tedious task.

Literature survey

Databases

Databases are defined as organized collections of data. The system which handles the data, transactions, problems or any other aspect of the database is the Database Management System (DBMS).

Relational Databases

Relational databases use the notion of databases separated into tables where each column represents a field and each row represents a record. Tables can be related or linked with each other with the use of foreign keys or common columns.

ACID properties

An important aspect of relational databases which guarantees the reliability of transactions is their adherence to the ACID properties: **A**tomicity, **C**onsistency, **I**solation, and Durability.

Atomicity: Either all parts of a transaction must be completed or none.

Consistency: The integrity of the database is preserved by all transactions. The database is not left in an invalid state after a transaction.

Isolation: A transaction must be run isolated in order to guarantee that any inconsistency in the data involved does not affect other transactions. **Durability**: The changes made by a completed transaction must be preserved or in other words be durable.

NoSQL Databases

NoSQL databases started gaining popularity in the 2000's when companies began investing and researching more into distributed databases. The most common NoSQL database categories are the following:

Document stores: The notion of "documents" is the central concept here with documents being the equivalent of records in relational databases and collections being similar to tables. [8]

Tele:
E-mail addresses: shalini.merlin@gmail.com

Key-value stores: Data is stored as values with a key assigned to each value similarly to hash-tables. Also depending on the database a key can have a collection of values. [8]

Graph databases: Like graph theory the notion of nodes and edges is the primary concept in graph databases. Nodes correspond to entities such as a user or a music record and edges represent the relations between the nodes. An important aspect which differentiates graph from relational databases is the use of index-free adjacency, this means each element contains a pointer to its adjacent element and does not require indexing of every element. [9] An important difference between relational databases and NoSQL databases is they do not fully guarantee ACID properties.

Database replication

Database Replication is the practice of deploying multiple servers which are clones of each other. This practice is used in NoSQL databases often in order to provide higher reliability and performance. In Mongo DB replication is deployed using a primary-secondary server configuration whereby one server is the primary and all others are secondary.

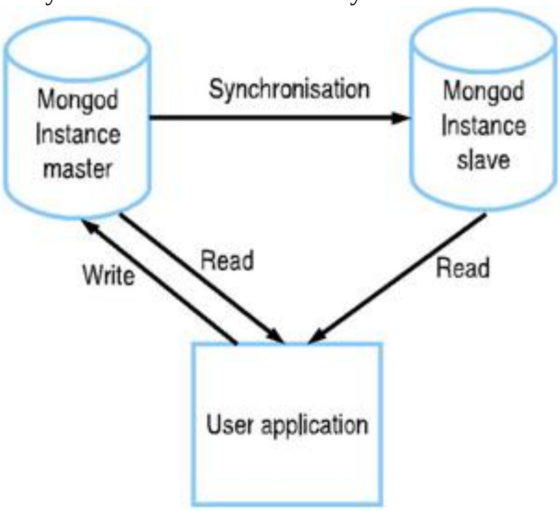


Fig 1. MongoDB Replica deployment and usage

MySQL Cluster also supports replication but is currently limited to a main replica server and a slave replica server at maximum. MySQL Cluster also supports replication but is currently limited to a main replica server and a slave replica server at maximum.

Database Sharding

Sharding is the term used to describe practice of using multiple servers of the same database and configuring them in order for the data stored in the database to be split or separated to different machines.

EDIM1 (EPCC Data Intensive Machine 1)

EDIM1 is an experimental platform designed for data-intensive research at EPCC. It consists of 120 back-end nodes, a login node and data staging server. Each back-end node consists of an Intel Atom CPU, a programmable GPU, 4GB of memory, a 6TB hard disk and 256GB SSD. Nodes are also interconnected with a gigabit Ethernet network and use ROCKS as the operating system which is a GNU/Linux distribution designed for running clusters [12]. EDIM1 was designed as an Amdahl-balanced cluster in order to eliminate the I/O bottleneck which exists in most systems due to the inability of the I/O system to provide data as fast as the CPU can process it. This is an important feature of the machine which affects the project as the databases will be able to make use of all the processing power instead of delaying while the I/O is sending data. The significance of this effect is that observations can be made on how the database works according to the number of connections it handles. For example, having two simultaneous connections requesting the same data could cause a slow down to the transactions.

ROCKS Linux

The operating system running on EDIM1 is the ROCKS Linux is a GNU/Linux distribution which is aimed at setting up clusters easily. It is a 64 bit operating system which allows building, managing and monitoring clusters through a frontend which controls the rest of the nodes. In case of a node failing the frontend automatically reinstalls the base system on that node any preselected packages which in ROCKS are called rolls.

Criteria comparison of RDBMS & NOSQL

Evaluation standards

Methods in C were considered for taking time measurements: The first was to use the **clock()** function provided by C but was quickly ruled out as it only provides the time a process spent using the CPU. The second method was the gettimeofday() function which provided timings with millisecond accuracy. MongoDB records query times in a similar manner but the results are stored in a system specific database and collection, system.profile.

Performance Metrics

To calculate the queries per second the formula below is used.

Queries per second = Total number of queries Total number of threads Average query/time.

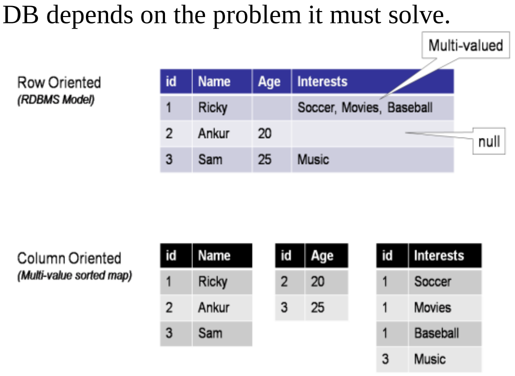
Database Schema- RDBMS vs NoSQL

RDBMS - Schema

A database schema of a database system is its structure described in a formal language supported by the database management system (DBMS) and refers to the organization of data as a blueprint of how a database is constructed. The formal definition of database schema is a set of formulas called integrity constraints imposed on a database. These integrity constraints ensure compatibility between parts of the schema. All constraints are expressible in the same language. A database can be considered a structure in realization of the database language. [16] The states of a created conceptual schema are transformed into an explicit mapping, the database schema. This describes how real world entities are modeled in the database.

NOSQL – Schema

A NoSQL (often interpreted as Not Only SQL [17] [18]) database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases. Motivations for this approach include simplicity of design, horizontal scaling and finer control over availability. The data structure (e.g. key-value, graph, or document) differs from the RDBMS, and therefore some operations are faster in NoSQL and some in RDBMS. There are differences though, and the particular suitability of a given NoSQL DB depends on the problem it must solve.



Row Oriented
(RDBMS Model)

id	Name	Age	Interests
1	Ricky		Soccer, Movies, Baseball
2	Ankur	20	
3	Sam	25	Music

Multi-valued
null

Column Oriented
(Multi-value sorted map)

id	Name
1	Ricky
2	Ankur
3	Sam

id	Age
2	20
3	25

id	Interests
1	Soccer
1	Movies
1	Baseball
3	Music

Fig 2. Schema Comparison

Criteria comparison of RDBMS & NOSQL

Characteristic	RDBMS	NOSQL
ACID compliance (Data, Transaction integrity)	Yes	No
OLAP/OLTP	Yes	No
Data analysis (aggregate, transform, etc.)	Yes	No
Schema rigidity (Strict mapping of model)	Yes	No
Data format flexibility	No	Yes
Distributed computing	Yes	Yes
Scale up (vertical)/Scale out (horizontal)	Yes	Yes
Performance with growing data	Fast	Fast
Performance overhead	Huge	Moderate
Popularity/community Support	Huge	Growing

NOSQL vs. SQL Summary

Category	RDBMS	NOSQL
Types	One type (SQL database) with minor variations	Many different types including key-value stores, document databases, wide-column stores, and graph databases
Development History	Developed in 1970s to deal with first wave of data storage applications	Developed in 2000s to deal with limitations of SQL databases, particularly concerning scale, replication and unstructured data storage
Data Storage Model	Individual records are stored as rows in tables, with each column storing a specific piece of data about that record much like a spreadsheet.	Varies based on database type. Document databases do away with the table-and-row model altogether, storing all relevant data together in single "document" in JSON, XML, or another format, which can nest values hierarchically.
Consistency	Can be configured for strong consistency	Depends on product. Some provide strong consistency (e.g., MongoDB) whereas others offer eventual consistency (e.g., Cassandra)
Data Manipulation	Specific language using Select, Insert, and Update statements,	Through object-oriented APIs
Supports Transactions	updates can be configured to complete entirely or not at all	In certain circumstances and at certain levels (e.g., document level vs. database level)

Result & Analysis

The results which are analyzed here firstly show how the databases respond to different query types, both reads and writes, and total number of queries.

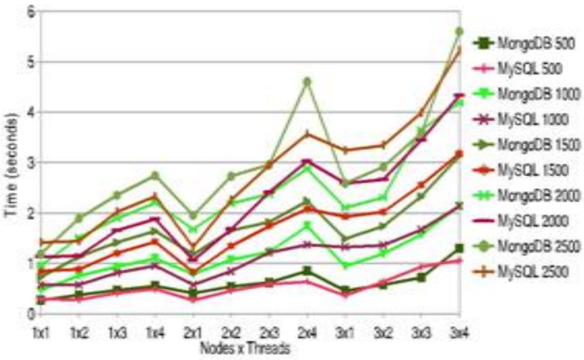


Fig 3. Query with different configurations

Another conclusion can be made by charting the Queries/second metric using the average time.

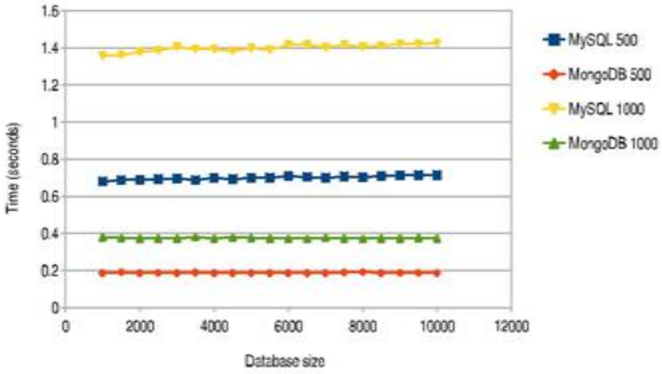


Fig 4. Simple total number of queries per second

Database Size

In order to measure the performance of the database management systems according to the size of the database an iterative approach was considered, whereby a specific process was repeated with only a single variable changing, in this case the variable being the database size

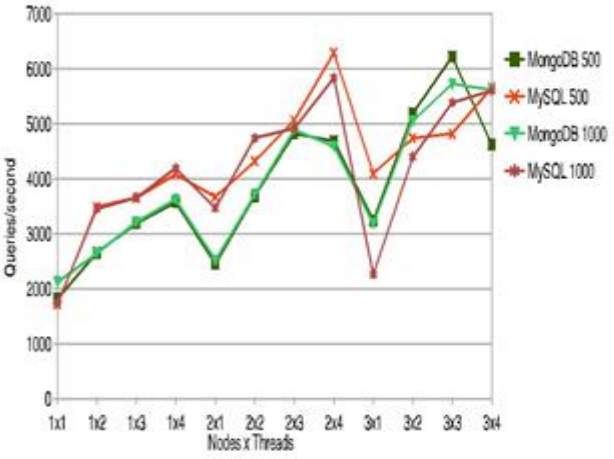


Fig 5. Complex query with different database sizes

Conclusion

Investigation and comparison of the performance and scaling of Relational Database Management Systems and NoSQL databases with the aim of exploring how the different factors affect each database is carried out. The project tested, analyzed and compared the performance and scalability of the two database types. The experiments carried out include running different numbers and types of queries, some more complex than others, in order to analyze how the databases scaled with increased load.

From the comparison of the results it was found that MongoDB can perform much better for complicated queries at the cost of data duplication which in turn results to a larger database. As the two databases behave differently according to the type of queries used the choice of which database to use lies on the type of application the system will be using. In addition it is important to consider the effect that using a database such as MongoDB will have on the hardware storage due to the increased database size.

Future Enhancement

Further work can be done by using larger clusters to test the performance. As it was shown different numbers of connections and configurations affect the behaviour of the databases. By extending this to a larger scale a more broad investigation can be performed with more general conclusions as to this factor.

The reliability of the two database types may not have been part of the project but in the future this aspect can be examined. The reason for doing so is to understand whether increased performance and scalability by one database type over the other affects the reliability. Specifically, since NoSQL do not guarantee the ACID properties a middle ground can be found whereby performance and scaling reach an adequate level but at the same time the reliability of the database is guaranteed as well.

References

[1] Berg K., Seymour T., and Coel R. History of Databases. *International Journal of Management and Information Services*, 17, 2013.

[2] MongoDB. MongoDB Manual. http://docs.mongodb.org/manual/. Accessed: 15-08-2013.

[3] Smith G., Robert T., and Browne C. Tuning Your PostegreSQL Server. http:// wiki.postgresql.org/wiki/Tuning_Your_PostgreSQL_Server. Accessed: 14-08-2013.

[4] Apache. Cassandra Wiki: Getting Started. http://wiki.apache.org/cassandra/ Getting Started. Accessed: 14-08-2013.

[5] Apache. Apache CouchDB Manual. http://docs.couchdb.org/en/latest/. Ac-cessed: 14-08-2013.

[6] Beynon-Davies P. *Database Systems*. Palgrave Macmillan, 3rd edition.

[7] Strozzi C. NoSQL: a Non-SQL RDBMS. http://www.strozzi.it/cgi-bin/CSA/ tw7/I/en_US/nosql/Home%20Page.

[8] Strauch C. NoSQL Databases.

[9] Batra S. and Tyagi C. Comparative Analysis of Relational And Graph Databases. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(2), May 2012. ISSN: 2231 2307.

[10] Kemme B., Jimenez-Peris R., and Patino-Martinez M. Database Repli cation. *Synthesis Lectures on Data Management*, page 17, 2010. doi:10.2200/S00296ED1V01Y201008DTM007.

[11] Oracle Corporation. Differences between the NDB and InnoDB Storage Engines. http: //dev.mysql.com/doc/refman/5.1/en/mysql-cluster-ndb-innodb-engines. html. Accessed: 14-08-2013.

[12] EPCC. EDIM1: Data Intensive Machine. http://www.epcc.ed.ac.uk/facilities/other-facilities/edim1-data-intensive-machine. Accessed: 14-08-2013.

[13] Rocks Group. Rocks: Open Source Toolkit Real and Virtual Clusters. http://www. rocksclusters.org/wordpress/. Accessed: 13-08-2013.

[14] Free Software Foundation. GNU parallel. http://www.gnu.org/software/ parallel/. Accessed: 13-08-2013.

[15] Oracle Corporation. MySQL Manual: The Slow Query Log. http://dev.mysql. com/doc/refman/5.1/en/slow-query-log.html. Accessed: 14-08-2013.

[16] Rybinski, H. (1987). "On First-Order-Logic Databases". *ACM Transactions on Database Systems* **12** (3): 325–349. doi:10.1145/27629.27630

[17] "NoSQL (Not Only SQL)". "NoSQL database, also called Not Only SQL".

[18] Jump up Martin Fowler. "NosqlDefinition". "many advocates of NoSQL say that it does not mean a "no" to SQL, rather it means Not Only SQL"