UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

COS 326: Database Systems
Topic 5: NoSQL MongoDB vs SQL

Grobler, Arno	Harvey, Matthias	Lochner, Amy
14011396	14027021	14038600
Madigoe, Pricilla	Maree, Armand	Obo, Diana
13049128	12017800	13134885

Department of Computer Science, University of Pretoria

Contents

1 Introduction

Though many database systems exists industry to store and query large collections of information, they are often misused in situations where another database could be better suited. To clarify and discuss this problem, an in depth analysis of the research paper, written by Zachary Parker[3], is completed. A discussion of the problem areas faced in databases, a solution to these problem areas, some of the pro and cons we see in databases and the relevance of this discussion in industry today will take place. To provide concrete examples, a large focus will be dedicated to comparing NoSQL MongoDB with SQL databases.

2 The Database Problem Area

In today's world the most common database implementation makes use of the relational model[3]. However, due to the increase in the amount of data collected today and considering the fact that the data is often unstructured[3], an alternative database implementation must be considered, namely NoSQL. The problem is that while unstructured data is becoming more prominent, there is still a need for modest-sized databases that hold structured data[3]. This prompts the question: Will NoSQL databases replace relational databases? As there is a general move towards NoSQL databases and the increasing availability of open-source NoSQL databases[3], it encourages the database designer to question whether he should make use of a NoSQL database. It is well known that NoSQL databases perform very well for large, unstructured collections of data[3] but this begs the question of how a relational database will fair against a NoSQL database for a modest-sized database that holds structured data.

3 Solution to the Problem

The NoSQL document and collection structure works similar to the row and table model of the relational database structure [3] where each row has a key (unique ID) and some data item (value) [4]. MongoDB stores most of these documents in memory which in turn speeds up the querying process significantly compared to the hard-drive querying process that MySQL uses [3]. The reason for this is that since the data item is stored as a single document, the retrieval of these objects is faster than joining the values from separate tables [4]. However, the NoSQL model does not guarantee a specific set of distinct values to be returned from a query, and this is where the SQL model comes in handy [4].

There are a few factors that determine when a database will be more suited for a specific project. Firstly, since NoSQL databases do not conform to some scheme. If the initial requirements are not clear then NoSQL will adopt changes with no complaints [1]. Secondly, since NoSQL prefers the denormalization of data it is significantly faster when it comes to queries, but at the cost of update

speed [1]. And lastly, NoSQL scales a lot better than SQL which will simplify the process of distributed databases - should a medium sized business need to expand [4].

4 Advantages and Disadvantages of the proposed solutions

4.1 Advantages

MongoDB improves scalability [3] by scaling well horizontally i.e. scale across several servers [4]. The data field and the value for that field is stored together as one record [4] which makes data retrieval much faster. MongoDB gains its performance by key value design, ease to scale out and denormalization [2]. It has better run-time performance for inserts, updates and deletes [3]. MongoDB has a flexible schema [2], which allows easy manipulation with the JSON format.

The tight rules that govern a SQL DB structure ensures data integrity and security without having to rely on application rules and logic [4]. It is a simple way of representing data or business logic [4] because of its structured schema. SQL DB has an easy-to-use language (SQL) to retrieve and query data [4]. It performs better when updating and querying non-key attributes [3, 4].

4.2 Disadvantages

MongoDB performs poorly for aggregate functions and querying based on non-key values [3]. It has memory limitations because the size of the DB is limited by virtual memory provided by OS and hardware [2]. MongoDB has no built in way to retrieve an object based on reference [3].

SQL DB requires additional joins in more complex schemas [3, 4] which leads to high transaction loads and decreased performance. It does not scale well across several servers [4] i.e. poor horizontal scalability.

5 Relevance to COS326 and the business industry

COS326, or Database Systems is a third-year Computer Science module that discusses various kinds of SQL and NoSQL databases. It teaches database design in a practical fashion and contrasts the design of SQL-based languages that work well with structured data with that of NoSQL-based databases which work well with unstructured data. Knowing the run-time performance on each

of these databases helps one to learn the various performance and design implications of these two databases as well as their various effective and efficient usage in real-world applications.

Industry has evolved to require vast amounts of data, which is needed to be collected and processed by various (business) organizations daily. This data is usually unstructured since it is not always known in hindsight. Unstructured data can be of any type and may have no form [3]. Understanding the comparison of the workings, design implications and performance of SQL databases with regard to NoSQL databases could help one choose the best database for large volumes of data in terms of scalability and efficiency.

6 Conclusion

It is clear that the choice of NoSQL vs SQL is largely project dependent. Projects with dynamic schemas are generally better off using NoSQL because of its flexibility. NoSQL MongoDB is faster with simple queries largely because of its memory mapped files [3] which is much faster than SQL's reading from disk (provided that there is enough memory). Currently, the use of aggregate functions and querying based on non-key values is better suited to SQL. Finally, when high scalability is required, NoSQL is better suited because of its suitability to distributed computing.

References

[1] Craig Buckler. Sql vs nosql: The differences. `https://www.sitepoint.com/sql-vs-nosql-differences/`, September 2015. [Online; accessed 2-September-2016].

[2] Makble. The advantanges and disadvantages of mongodb. `http://makble.com/the-advantages-and-disadvantages-of-mongodb`, May 2016. [Online; accessed 2-September-2016].

[3] Zachary Parker, Scott Poe, and Susan V. Vrbsky. Comparing nosql mongodb to an sql db. In *Proceedings of the 51st ACM Southeast Conference*, ACMSE '13, pages 5:1–5:6, New York, NY, USA, 2013. ACM.

[4] UpGuard. Mysql vs mongodb. `https://www.upguard.com/articles/mysql-vs-mongodb`, July 2014. [Online; accessed 2-September-2016].