

MySQL and MongoDB as Methods of Storing Data

David Hurley

Computer Systems

Department of Computer Science

Cork institute of Technology

Rossa Ave, Bishopstown, Cork, T12 P928, Ireland

David.hurley2@mycit.ie

The goal of this paper is to compare MySQL and MongoDB as databases, by using the knowledge and experience gained from the previous assignments, lecture theory and personal research. The reason is so that we can gain an understanding of how each system works and apply that knowledge to practical real-world scenario where one implementation might be superior to the other. The comparison will comprise of completing queries specified in the assignments, where questions individually utilize one of the two systems.

The assignment demonstrated where each system had its pros and cons. It guided the user to discover how both systems worked in an organic way that would lead them to knowing which implementation would be most suitable in different situations.

Keywords

SQL – Standard query language

RDMS – relational database management system

JSON – JavaScript object notation

BSON – Binary JSON

API – Application programming interface

I. INTRODUCTION

In the current technological society that we live in, huge amounts of data are created, shared and stored every second. With the internet connecting the world this data can originate from anywhere and be necessary in many applications. The need to store and view this data has given rise to the need for robust databases with good integration with many applications. Video's, text and pictures are continuously being uploaded online where they need to be accurately tracked and updated in the corresponding database. Different databases accomplish this task in different ways, depending on how they are designed fundamentally.

This report will focus on how MySQL and MongoDB store and manipulate data. We will first acclimatize ourselves with both systems by completing the questions given in assignments one and two. Following this we will be able to carry out a comparison between MySQL and MongoDB and give recommendation regarding suitability of each system in certain implementations.

This report will focus on a case study deriving from assignment one and two. Firstly an overview of each system will be given. Following this the advantages and disadvantages of each system will be compared. Next, we will discuss how each system is applicable to scenarios that most suits there

strengths. Finally, a conclusion will be reached based on our usage of each system.

II. MySQL

Its name is composed of “My” and “SQL”. The former referring to the cofounder's daughters name while the latter is an abbreviation of the term Structured query language. MySQL is an open-sourced relational SQL database management system. MySQL was initially released internally in May of 1995 and has since become very common in numerous websites and programs. Large tech companies such as Uber and Netflix with gigantic user bases use MySQL. In the past it was also used by Google.

MySQL is based on a relational model. This means that the database consists of tables and enforced schemas. These tables consist of columns, rows and keys. The two types of key which establish relations are primary keys and foreign keys. In MySQL rows represent records while columns represent attributes. Each row can also have more than one attributes, known as relation keys. These can be used to uniquely identify the row in the table.

MySQL also used database normalization to store data. This normalization ensures that dependencies are enforced and is used in order to minimize data redundancy and make maintenance of the database easier.

One of the reasons MySQL is incredibly popular is because of its support for transactions. A transaction is where multiple SQL statements are treated and executed as a single unit. This transaction feature is especially useful since it will only commit the changes if the entire transaction succeeds. If any part of the transaction fails it will undo all the changes it made and roll back to its previous save point. Since each attempt at changing the database is its own separate instance this ensures that data maintains accuracy and consistency even if multiple users are attempting to view or modify data the same time.

MySQL uses ACID (Atomicity, Consistency, Isolation, Durability) to make sure that all transactions are valid.

Atomicity refers to how the results of transactional statements are stored in a memory buffer before being committed to the database only once the statements in the transaction are all true. Consistency refers to how all of your data is guaranteed to be consistent. Isolation refers to how transactions cannot read data from other transactions that have

yet to finish. Finally, durability refers to how once a transaction is complete, its changes are guaranteed to be record correctly.

The most notable drawback to MySQL is its poor scalability. – For small to medium applications MySQL works great but for large data sets it begins to degrade in terms of query performance. Query's begun to take noticeable longer times to execute or sometimes may time-out before completing. This poor performance scaling in complicated query searches make it impractical for large sized data.

usage of MongoDB make it unsuitable for smaller applications that don't have as much computing power while NoSQL works very well in these situations. Where NoSQL starts to suffer from performance and scalability issues, MongoDB would be better suited.

III. MONGODB

MongoDB is an open source, cross-platform, document orientated database written in C++ that released in February of 2009. It uses JSON and BSON documents and is one of the leading NoSQL databases due to its dynamic schemas, great scalability and high performance. Its used by large company's such as eBay, Google and Sega, and supports API's from a variety of languages such as java and python.

Mongo is a hash-based, schema-less database. It uses collections made up of documents to store data. This schema-less design means that each document in a collection can contain different data giving the user more flexibility with how they want to store and organise their data.

MongoDB was designed for big data storage. It uses a key value design to search for documents which are identified by an id. This key value query is much quicker than MySQL's relational query. Alongside this MongoDB stores working sets in internal memory. This faster access to the data comes at the trade off of cost since more RAM is needed.

MongoDB scales horizontally fantastically well. Its support of auto sharding mean that when the data on one node goes over the threshold, MongoDB will automatically rearrange the data evenly to the nodes. Its support of auto failover means that it's a very robust system. Replica sets are created and stored and can be accessed in the event of failures.

While overall MongoDB seems very attractive it is not without its own flaws. Since it uses memory mapped record and lets the OS handle the storing its very RAM heavy which may be a deal breaker for smaller applications.

As stated, MongoDB is great for large data and is the main reason why Google Search and eBay use it.

IV. Conclusion

Upon completion of the two assignments we had a fairly concrete grasp of each system. Populating MySQL was very different to MongoDB. For MongoDB we used a JSON and once we fixed a syntax error within the file it was very quick and easy to populate the collection. MySQL was much nicer to use and read from compared to MongoDB since it outputted the query results in a table format. It was around the halfway point in both assignments that both systems become much easier to navigate and construct queries for.

In conclusion both databases have many advantages to then, but both are also not without there flaws. The RAM