NAME: Divyang Jain
STUDENT NUMBER: x16110323

Data Storage and Management

Submitted By:

Divyang Jain

16110323

NAME: Divyang Jain
STUDENT NUMBER: x16110323

Introduction

Today's world works on one type of data: big. Data is growing exponentially and will become larger in leaps and bounds. They are also large enough to render traditional database processing software obsolete while processing them. Big data also requires a lot of organisation and processing, for example, capture, storage, analysis, search, sharing, visualization, querying, privacy, and so on. There are around 40 EB of data generated every day, (Sas.com, 2017)

The purpose of this project is to install and implement two instances of MySQL and MongoDB. We create tables in them with ten fields having a VARCHAR type and the standard 255 characters. The normal SQL and MongoDB commands are used to insert values into the tuples and records.

MySQL is an open-source relational database management system (RDBMS) which is used to create, manage, and store databases. MongoDB is a free and open-source cross-platform database program. The difference between the two is that MySQL is a RDBMS and MongoDB is a NoSQL database program.

Both databases were installed on an Ubuntu 17.04 system on VirtualBox. They required some updating of the packages provided by the Ubuntu ISO, using the commands

```
sudo apt-get update

sudo apt-get upgrade

sudo apt-get install mysql-server
```

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 0C49F3730359A14518585931BC711F9BA15703C6
```

```
echo "deb [ arch=amd64,arm64 ] http://repo.mongodb.org/apt/ubuntu xenial/mongodb-org/3.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.4.list
```

```
sudo apt-get install -y mongodb-org
```

The MongoDB server can be started by typing `mongod` at the command prompt. When this is done, we need to open a new terminal window and start with the YCSB test.

NAME: Divyang Jain
STUDENT NUMBER: x16110323

The Yahoo! Cloud Serving Benchmark (YCSB) is a tool to provide a common framework for different systems with different workloads and benchmark their performance. High value and cloud serving stores are the main focus. The project comprises of two tools:

- The YCSB Client, a workload generator with extensions
- The Core workloads, a set of workload scenarios to be executed by the YCSB generator which range from A through F. (GitHub, 2017)

For testing the systems, we will be focusing on two factors: -

(i) Performance
(ii) Scalability, Availability and Reliability

In the performance benchmark, we are to study how the largeness of queries can slow down or otherwise affect a database system. We will change operational counts in the two sets of databases chosen and evaluate them on the time taken for them to complete their queries. They are also evaluated based on update, read, and throughput operations.

The operational counts in the databases are taken as 25000, 30000, 35000, and 40000. The operational counts are taken based on the system load and normalcy of operations in it.

Key Characteristics of Chosen Data Storage Management Systems

MySQL

MySQL is one of the most popular open-source databases used in the world, allowing cost-effective usage of reliable, speedy, and scalable Internet and embedded database applications. It follows standards which are imposed in databases like ACID. It has modern features of a data base like a rollback, crash recovery, etc. full commit or low-level capabilities. It offers ease of use, scalability and high performance, and contains a full set of database drivers and visual tools to help a developer or a DBA manage or build their own appearances. Oracle Ltd is the developer and distributor of MySQL, and they also provide support for it. MySQL has the following features: -

- High performance and scalability to meet the demands of data loads and users which are increasing exponentially.
- Self-Healing Replications Clusters to enhance performance, scalability and reliability.
- Online Schema Change to keep up with business requirements.
- Performance schema for monitoring user and application performance, Plus resource consumption.
- SQL and NoSQL Access to carry out complex queries. Simple, fast key operations can also be done.
- Platform independence gives us the freedom to try other systems.
- Interoperability with Big Data using MySQL as an operation store with Hadoop and Cassandra

(Oracle.com, 2017)

NAME: Divyang Jain
STUDENT NUMBER: x16110323

MongoDB

MongoDB is a database application created by MongoDB, Inc. It helps maximise competitive advantage by leveraging data, reduces the risk of critical usage, increases the value of the deployment with time, and helps reduce cost of ownership to a fraction of what it originally is.

MongoDB has several advantages over traditional relational databases. Some of them are as follows: -

Fast, Iterative Development: Developers will find it very easy to build and deploy applications. The data model of MongoDB is very flexible and has dynamic schemas/idiomatic drivers. Databases can be linked very easily with the automated provisioning and continuous integration offered by the platform.

Flexible Data Model: The data model offered by MongoDB allows you to store and link data of any kind. Validation rules, data access and indexing functionality are not hindered. Schemas can be modified without downtime.

Multi-Data Centre Scalability: MongoDB can be scaled worldwide in different data centres. They provide new levels of scalability and availability. The data can grow or shrink as needed with no problem, since MongoDB will scale up and down with no downtime.

Integrated Feature Set: Analytics and data visualisation, text search, graph processing, in-memory performance, and other characteristics allow us to deliver lots of real-time applications.

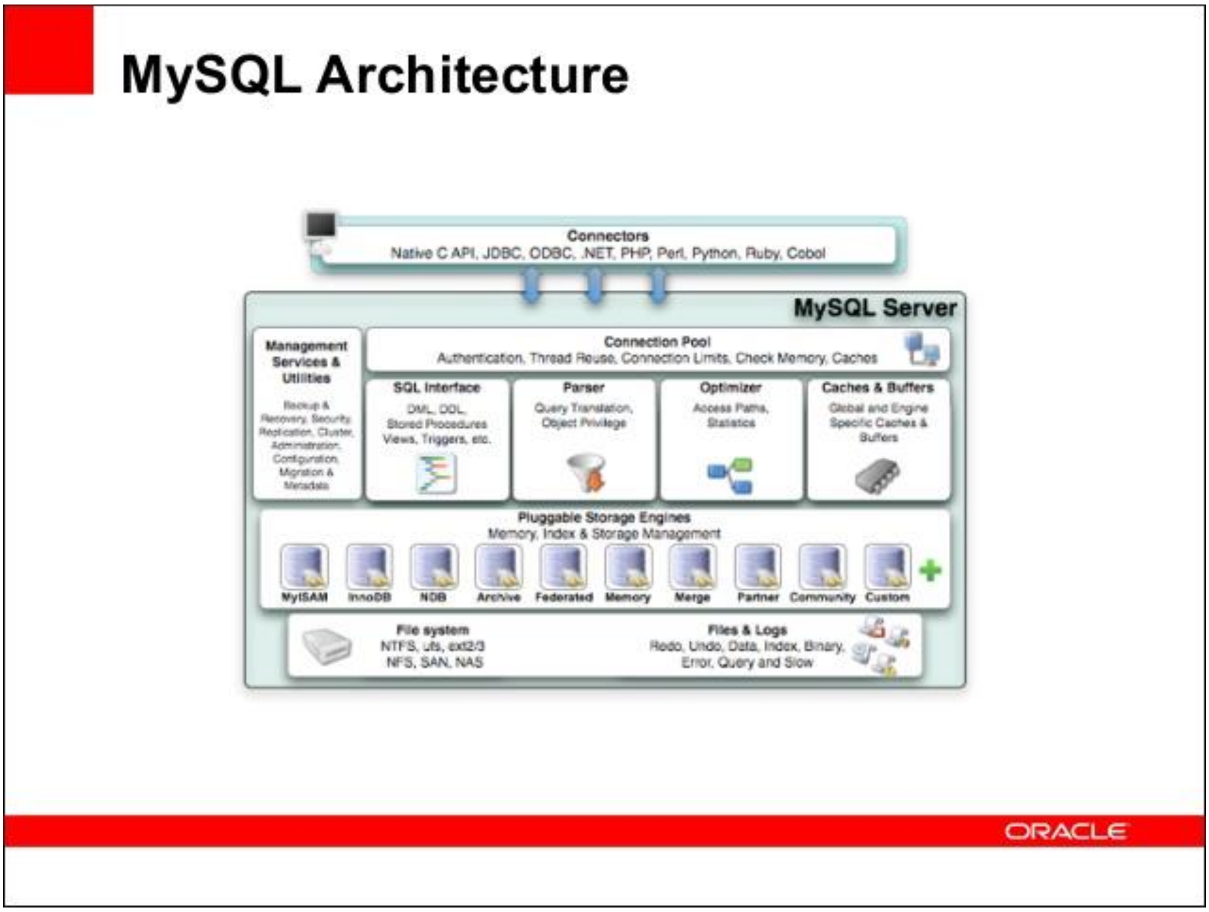
Lower TCO: Productivity increases when using MongoDB. Most of the features are a one-click management method. MongoDB uses community hardware to lower costs, and the subscription fees are very low with 24/7/365 support worldwide.

Long-Term commitment: The ecosystem of MongoDB is very robust. Downloads and customers are in the millions range, and they even include world-famous companies like Fortune500.

(MongoDB, 2017)

NAME: Divyang Jain
STUDENT NUMBER: x16110323

Database Architecture



MySQL Architecture

Connectors
Native C API, JDBC, ODBC, .NET, PHP, Perl, Python, Ruby, Cobol

MySQL Server

Management Services & Utilities
Backup & Recovery, Security, Replication, Cluster, Administration, Configuration, Migration & Metadata

Connection Pool
Authentication, Thread Reuse, Connection Limits, Check Memory, Caches

SQL Interface
DML, DDL, Stored Procedures Views, Triggers, etc.

Parser
Query Translation, Object Privilege

Optimizer
Access Paths, Statistics

Caches & Buffers
Global and Engine Specific Caches & Buffers

Pluggable Storage Engines
Memory, Index & Storage Management

MyISAM, InnoDB, NDB, Archive, Federated, Memory, Merge, Partner, Community, Custom

File system
NTFS, ufs, ext2/3
NFS, SAN, NAS

Files & Logs
Redo, Undo, Data, Index, Binary, Error, Query and Slow

ORACLE

Figure 1: Architecture of MySQL [1]

Query Engine
MySQL is a multithreaded application designed to take advantage of multiple processors and cores. It follows the ANSI standard.

Parser
A query is created and sent to the MySQL parser when clients request one. The MySQL parser uses a large Lex-YACC script that is compiled with Bison.

Query Optimiser
It restructures the query by narrowing down the number of tuples to work with (restrictions), and then removes any attributes in the remaining tuples (projections). It then uses the JOIN instructions to combine the queries.

Query execution
This is handled by library methods which are specific to each query.

(thinkingmonster, 2017)

[1] https://image.slidesharecdn.com/innodb-presentation-140226132613-phpapp01/95/the-innodb-storage-engine-for-mysql-6-638.jpg?cb=1393934495

Query Caching

This caches all the queries and their results for future reference. Future searches are faster. Sometimes, queries have to be purged because the results for the tables have changed with UPDATE operations.

Buffer manager/Cache and buffers

The buffer stores the most commonly used data and structures are available with full efficiency. These caches improve the response time for requests because the data is in memory without extra disk accesses needed.

The Storage Manager

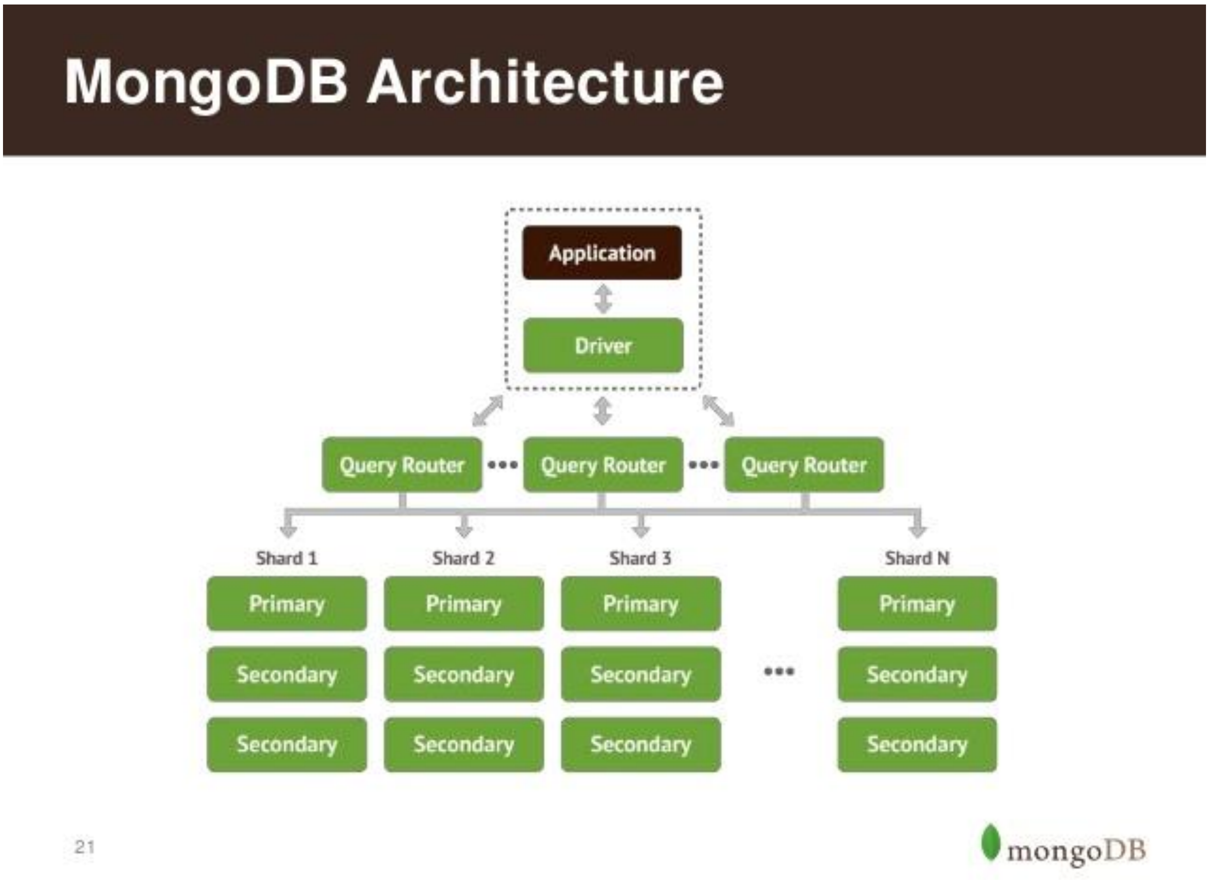
It works with the OS to write data to the disk efficiently. This includes table data, indexes, logs, and internal system data.

The Transaction Manager

This ensures that several users can access the data they need concurrently. There should be no corruption or damage to the data while allowing users to access it while ensuring other operations such as read, write, update, and cleanup take place.

Recovery Manager

It keeps records of the data for later retrieval in case of loss. Logs of modified data and similar events in a database are also recorded.



MongoDB Architecture

Figure 2: MongoDB Architecture [2]

MongoDB follows the Nexus Architecture. MongoDB also doesn’t reinvent the wheel, it builds upon the maturity of former databases like Oracle and continuing their work. It combines key RDBMS concepts with the work of Internet pioneers in designing modern applications.

There are some critical features which relational databases offer.

Expressive query language and secondary indexes: Users can access and manipulate data in ways which make sense. Indexes help provide efficient access to data.

Strong consistency: Applications should instantly read what is written to the database. Building applications which follow this consistency rule is complicated.

Enterprise management and integration: Databases must fit into the IT departments of companies and be suitable for their work. Organisations need a database which can be monitored, secured, integrated, and automated with their existing technology infrastructure, staff, and processes including operations teams, data analysts, and DBAs.

Modern applications have requirements which are not addressed by relational databases, therefore it is necessary to have NoSQL databases which offer: -

Flexible Data Model: NoSQL databases emerged to address the requirements for data with modern applications. Whether document, graph, key-value, or wide-column, they all offer a flexible data model, allowing a user to store and combine data of any structure. The schema can be dynamically modified without downtime or performance impact.

Scalability and Performance: NoSQL databases were built for scalability, so they include partitioning. The community can scale out commodity hardware or cloud servers, allowing unlimited growth with lower latency and higher throughput than relational databases.

Always-On Global Deployments: NoSQL databases were designed for always-available systems which provide a consistent experience for worldwide users. They run across several nodes and replicate across them, automatically syncing over servers, racks, and data centres.

(MongoDB, 2017)

[2] https://image.slidesharecdn.com/enterprisearchitectsview2014-oct-141029130130-conversion-gate01/95/an-enterprise-architects-view-of-mongodb-16-638.jpg?cb=1414587854

NAME: Divyang Jain
STUDENT NUMBER: x16110323

Performance in MySQL and MongoDB

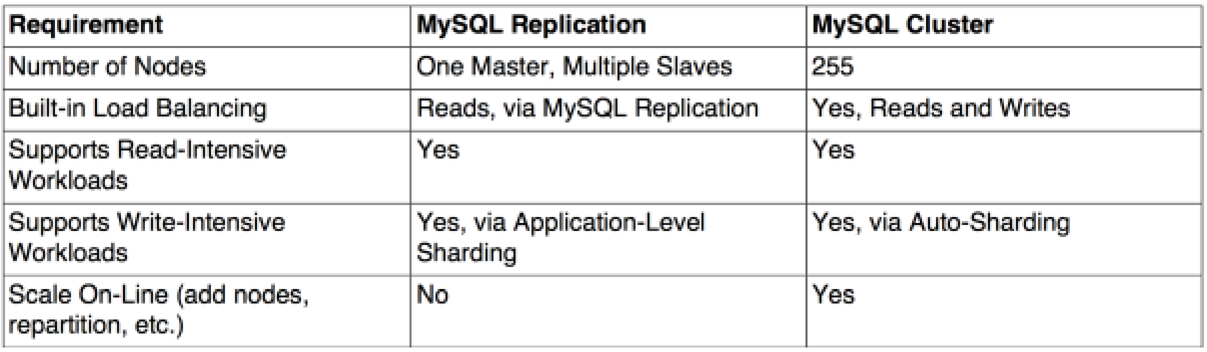
MySQL is a very comprehensive and useful database application, but there are many ways to improve its performance. For example, changing the hardware and processing speed of the system can dramatically improve the response times of the benchmark. Using multiple RAID volumes, SSDs, faster processors, larger amounts of RAM, L3 caches, and similar system resources are other ways of improving performances. These apply to any system and database, however. For specific improvement of MySQL, a database which is relational or has relations between the records and tuples is best. For MongoDB, we are better off with a more modern database which can be accessed online and remain in any location (securely kept). NoSQL database managers like MongoDB are best suited for databases which have little or no structure. They can be documents, graphs, or regarding absolutely any type of data which we can work on.

Scalability, Reliability and Availability in MySQL

Oracle provides a set of database high availability companies which work together in tandem to reduce planned and unplanned downtime.

Oracle uses a set of technologies to provide this maximum availability. At the production site, scalability and server availability are logged and recorded. The continuity of use of applications is also monitored. Global data services offered by MySQL check for server overloading or load balancing. Active Replica helps in protection of data and offloads queries. RMAN, also called Oracle Secure Backup helps backup data to tapes and the cloud.

For scalability, Oracle has a MySQL Replication and Cluster. The table for this is shown below.



Requirement	MySQL Replication	MySQL Cluster
Number of Nodes	One Master, Multiple Slaves	255
Built-in Load Balancing	Reads, via MySQL Replication	Yes, Reads and Writes
Supports Read-Intensive Workloads	Yes	Yes
Supports Write-Intensive Workloads	Yes, via Application-Level Sharding	Yes, via Auto-Sharding
Scale On-Line (add nodes, repartition, etc.)	No	Yes

(MySQL Downloads, 2017)

Scalability, Reliability and Availability in MongoDB

MongoDB is a NoSQL database, so it was already designed from the ground up with scalability in mind. It is very easy to scale MongoDB databases up or down since the storage is done on community servers or the cloud. Cloud services are very simple to increase when you need more storage since you simply have to increase your budget by a few dollars to pay the increased subscription fees.

NAME: Divyang Jain
STUDENT NUMBER: x16110323

(MongoDB, 2017)

Test Evaluation

In YSCB Bench mark package tool includes the default workloads are as follows: -

- Workload A has a 50/50 ratio for read and update
- Workload B has a 95/5 ratio for read and update
- Workload C has work 100 percent read
- Workload D has 95/5 ratio of read or insert
- Workload E has scan and insert ratio of 95/5
- Workload F consists 100/0 read, modify and write operations.

The operation counts used in workloads are 25000, 30000, 35000, and 40000.

These tests are taken from Ubuntu 17.04 Desktop ISO on VirtualBox.

Machine used for execution: MacBook Pro 13” Retina

CPU: 2.7 GHz Intel Core i5

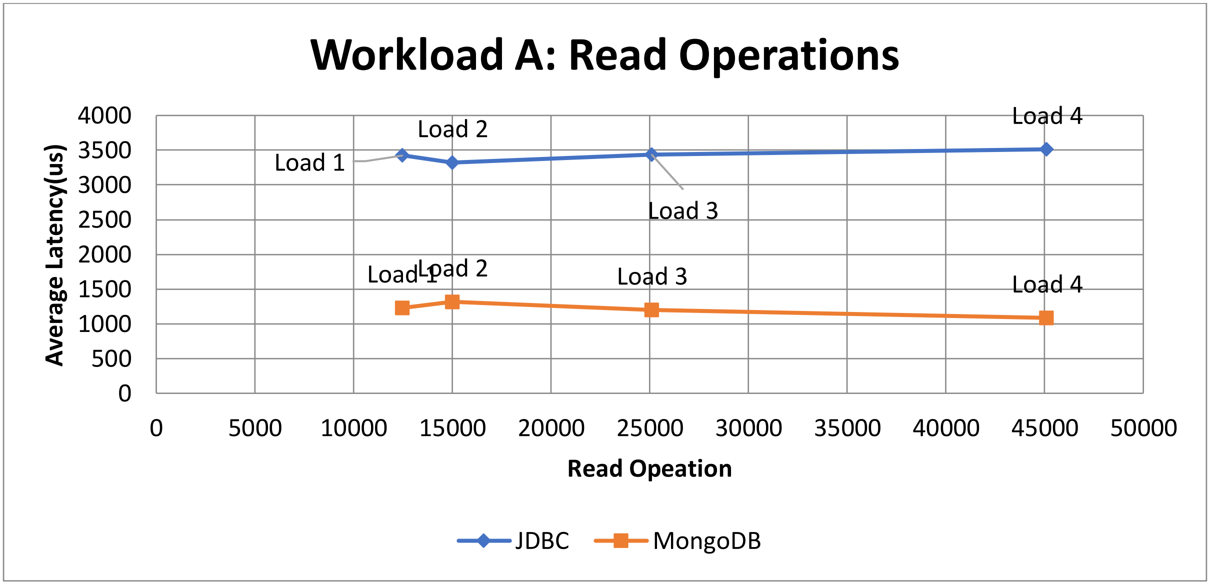
Virtual RAM is given as 2 GB 1867 MHz DDR3

MySQL version: 2.3

MongoDB: 3.4.6

YCSB version: 0.12.0

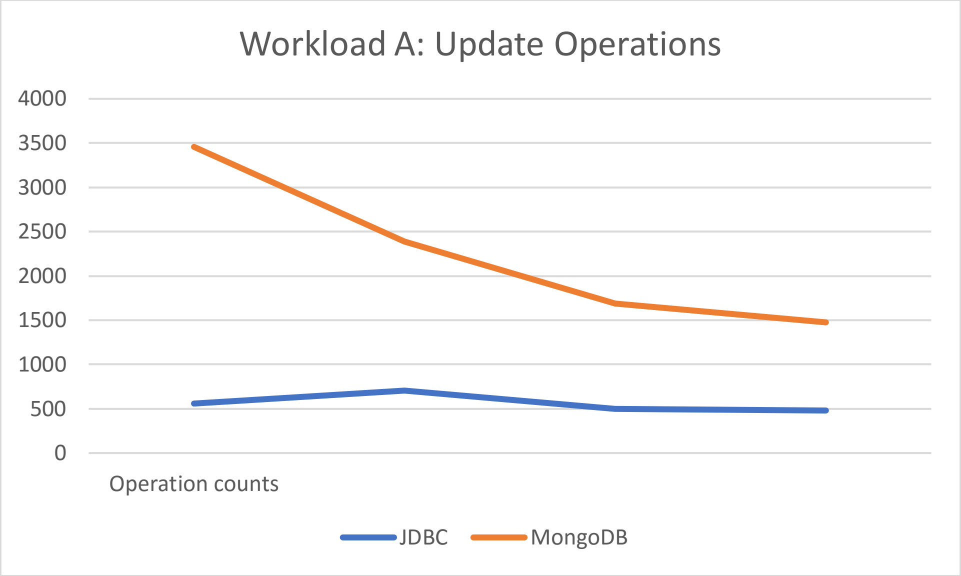
Workload A: 50% Read, 50% Write, Update Heavy



MongoDB is taking a much lower time to complete its update operations than JDBC/MySQL. They are proceeding at a similar rate. Load 1 has the highest time taken, and it decreases in

Load 2 for JDBC. Load 2 time increases very slightly in MongoDB. Load 3 takes a longer time than Load 2 in JDBC, but less in MongoDB. Load 4 time increases in MongoDB, but reduces in MongoDB. We can see that Loads 3 and 4 increases in JDBC but go down in MongoDB.

MongoDB is the better database here when doing read operations for Workload A.



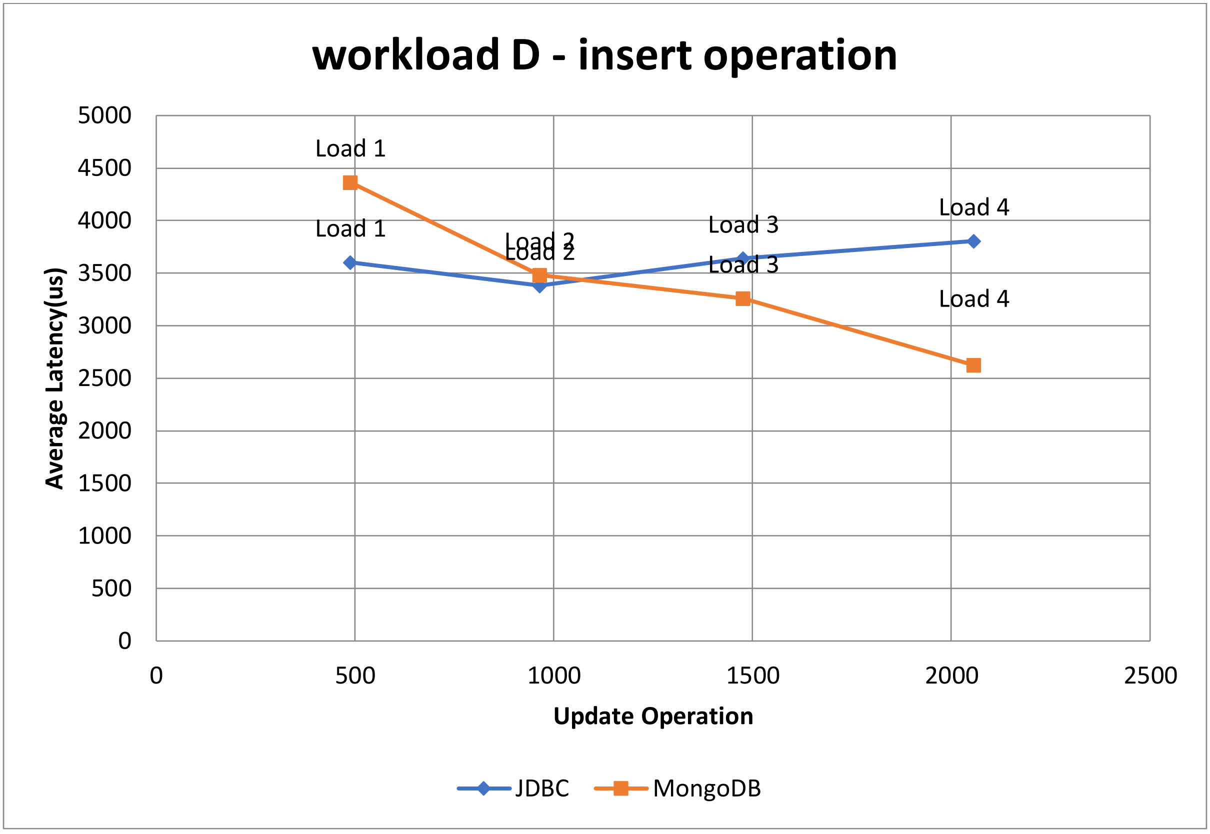
MySQL starts and ends at a lower rate than MongoDB here. Load 2 is higher than Load 1, but the graph decreases from Load 2 to Loads 3 and 4.

On the other hand, MongoDB has a steadily decreasing graph for all four loads, although it still remains above JDBC even at Load 4.

MySQL is faring better than MongoDB in the update operations field.

NAME: Divyang Jain
STUDENT NUMBER: x16110323

Workload D: Read 95%, Update 5%, Read Latest

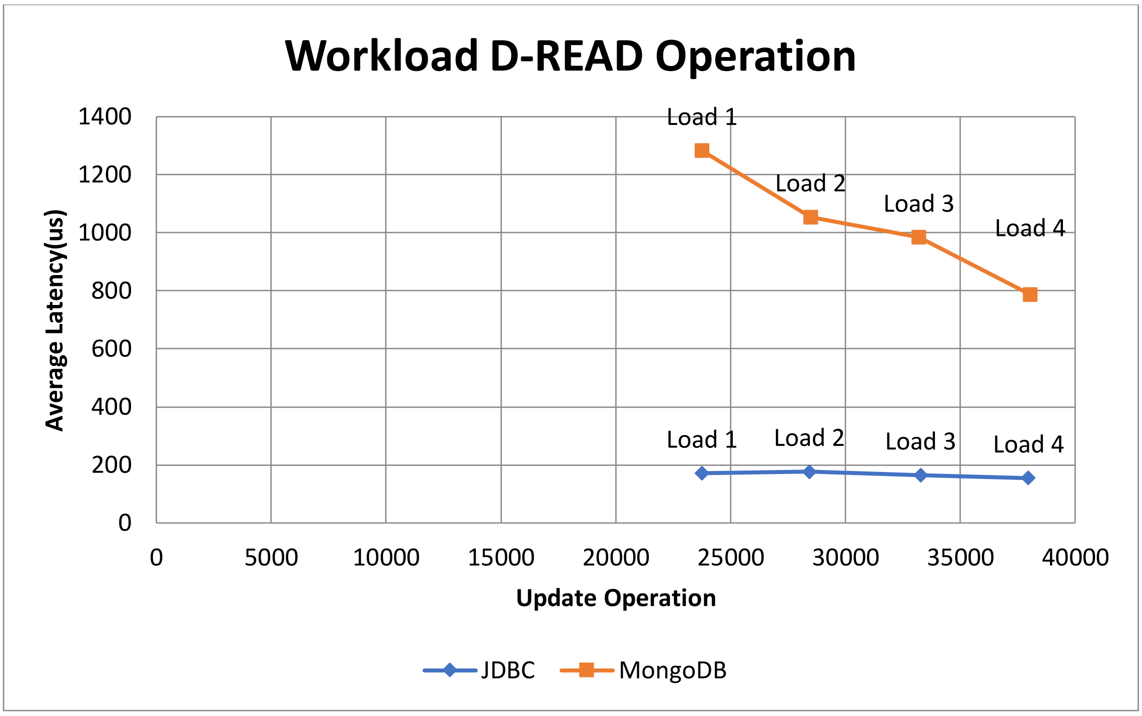


MySQL has a lower time in Load 1 while MongoDB's is higher. Load 2 timings are lower, but MongoDB's time has decreased while MySQL has increased from Load 1. MySQL's latency timings rises again for Loads 3 and 4, while MongoDB's keeps decreasing steadily for both loads.

MySQL starts off well with lower latency than MongoDB, but it decreases its time only for Load 2 and rises steadily for the other two.

MongoDB has a higher Load 1 time and similar Load 2 timing, but it decreases the time taken for all its loads steadily until Load 4.

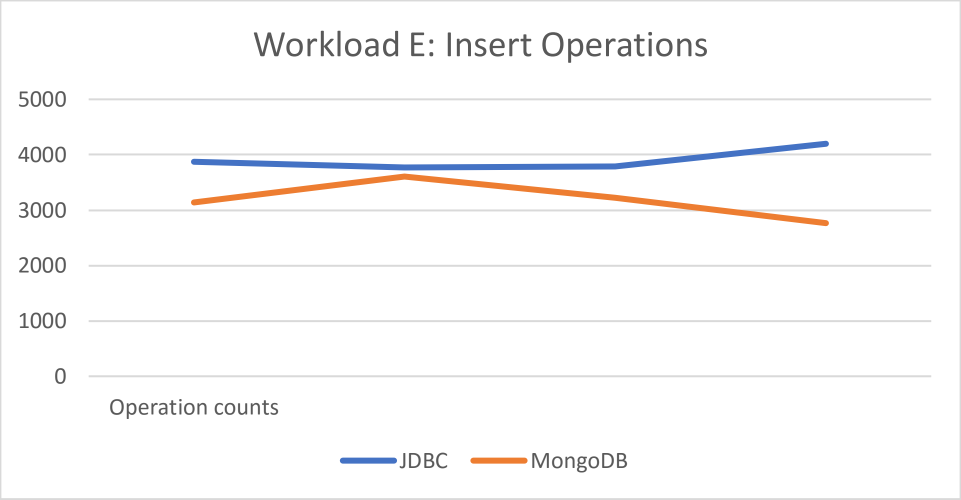
MongoDB is taking lesser time overall for its insert operations than MySQL.



Here, MongoDB is taking longer in its average latency to complete the READ operations as compared to JDBC. JDBC starts its latency timings lower than MongoDB and keeps decreasing steadily at a low rate. MongoDB starts Load 1 at a higher timing, but it decreases quickly in all the other loads. MySQL has a very low latency in all four loads with a very low decrease, but it remains lower than all four loads of MongoDB output.

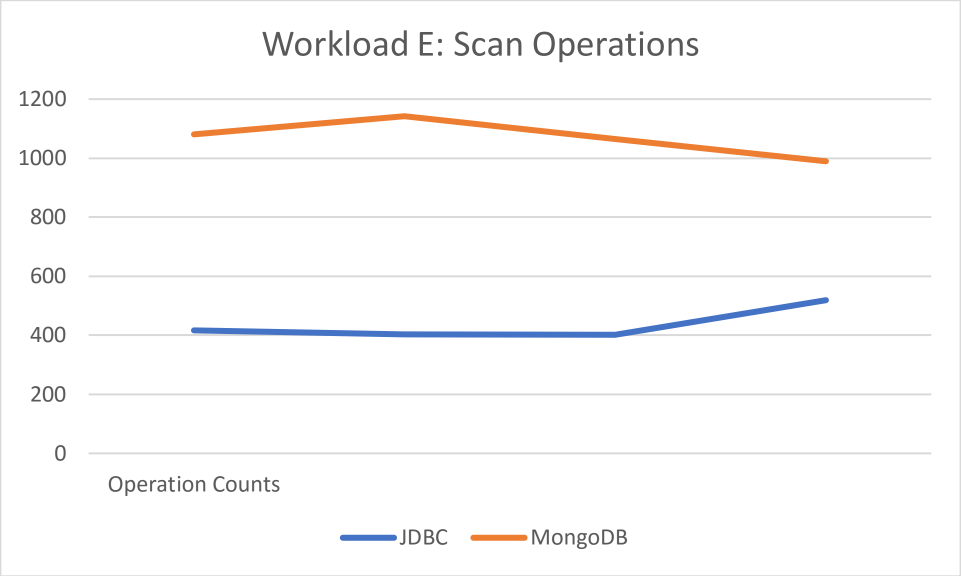
MySQL fares better in Workload D's read operations.

Workload E: Scan 95%, Insert 5%, Scan short ranges



Here, MySQL has a slight decrease in the graph from Load 1 to Load 2, but the increases steadily to Loads 3 and 4. MongoDB increases in latency from Load 1 to Load 2, but then decreases in Loads 3 and 4.

MongoDB has a steady decrease and lower latency in Insert operations for Workload E.



MongoDB is taking a longer time for SCAN operations than MySQL. In MySQL, we have a similar latency for Loads 1, 2, and 3, but a slight increase for Load 4. MongoDB, on the other hand, has a much higher starting latency value in Load 1 and increases a little more for Load 2. However, it decreases for Loads 3 and 4, but remains more than MySQL.

MySQL is outperforming MongoDB in the SCAN operation using average latency values.

Conclusion

MySQL is an RDBMS and MongoDB is a hybrid database manager which is NoSQL, but does combine useful features of RDBMS and NoSQL databases. Both have their advantages and disadvantages, as well as different performance benchmarks depending on the operation running on them. After comparing them, it is hard to say which database is overall better for general-purpose database work, since in Workload A, MongoDB outperforms MySQL in READ operations, but not the UPDATE ones. In Workload D, the INSERT operations take lesser time for MongoDB and more for MySQL, but the READ operations take MongoDB much longer than MySQL. Lastly, Workload E shows that MongoDB is taking somewhat lesser time for READ operations than MySQL, but the SCAN operations are completed by MySQL much faster than MongoDB. This is due to the size of the table and the relations present in it, as well as the simplicity of the given records and tuples. They also may have been caused by fluctuations and interruptions in the virtual machine itself while a certain database manager was running. It may also be that MongoDB is better at scanning data for most READ operations, but doesn’t perform as well in UPDATE and SCAN ones because of the database size and complexity of design compared to MySQL. Another explanation is that MongoDB needs more time to make the database available globally and indexing the content for unrelated databases which can be of any type takes a while, but subsequent reading and insertion operations will be much faster.

NAME: Divyang Jain
STUDENT NUMBER: x16110323

MySQL is best suited for relational databases which have lots of relations between the tuples and records, but MongoDB will outperform it in databases which are general-purpose and suited for random access, as well as not related since the databases are documents, graphs, or any generic type. Indexing and later access to data will be much faster in MongoDB as it is suited for this purpose.

REFERENCES

1. GitHub. (2017). brianfrankcooper/YCSB. [online] Available at: https://github.com/brianfrankcooper/YCSB/wiki [Accessed 24 Jul. 2017].

2. MySQL Downloads. (2017). MySQL HA/Scalability Guide. [online] Available at: https://downloads.mysql.com/docs/mysql-ha-scalability-en.pdf [Accessed 24 Jul. 2017].

3. Oracle.com. (2017). MySQL. [online] Available at: http://www.oracle.com/technetwork/database/mysql/index.html [Accessed 24 Jul. 2017].

4. Sas.com. (2017). Big Data in Big Companies. [online] Available at: https://www.sas.com/en_us/whitepapers/bigdata-bigcompanies-106461.html [Accessed 24 Jul. 2017].

5. thinkingmonster. (2017). MySQL Architecture. [online] Available at: https://thinkingmonster.wordpress.com/database/mysql/mysql-architecture/ [Accessed 24 Jul. 2017].

6. MongoDB. (2017). MongoDB Architecture. [online] Available at: https://www.mongodb.com/mongodb-architecture [Accessed 24 Jul. 2017].