OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

A SURVEY ON DATABASE FOR STORAGE OF VOTING DATA

Sushant Nanaware[1], Dr. Ayesha Butalia[2], Vaibhav Thakare[3]'Swapnil Randhave[4]' Sagar Bhor[5]

Student,Department of Computer Engineering PGMCOE, Wagholi Pune University.[1 3 4 5]

Phd.DR, Department of Computer Science and Engineering, PGMCOE Engineering College, Pune, India[2]

sushant2328@gmail.com[1], ayeshabutalia@yahoo.co.in[2], [4]vaibhav.vat@gmail.com[3],swpnlrandhave@gmail.com

Sagarbhor70@gmail.com[5]

Abstract: **Decision of right and suitable database is always important for any data framework. Because database is an integral and important part. In our SMART Voting System Project there is need to manage and store voting data. In our SMART voting project government voting database consists of authentication data means Aadhar card details of peoples and also consists of individual votes casted by voters at the time of voting and finally voting result. All these data needs to be store, so in this paper I did a survey on various databases that are available today. Databases were initially "flat files" displaying simple columns and rows to store data, but today, databases are relational, allowing complex queries across multiple database tables and database sets.[2] Relational databases provide users with greater use and flexibility to manage the data in the repository, using database programs such as Microsoft SQL and MySQL. We choose to write the performance analysis of different type of databases in context to government voting data. In this paper i compare and analyze the performance of different databases over each others. In this paper i have compared the performance of MySQL (relational) and Mongodb (non-relational) databases. This comparison yields to help to choose one database among these databases which is best suitable for our SMART voting project because different databases are having different characteristics and features that they are providing. And also in certain environments there are some databases that work best in that environment.**

Keywords *– Database Management System; database, MySQL; NoSQL; Mongodb; Performance evaluation*

I INTRODUCTION

Storage design of e-voting system plays important role in a real world implementation. Because how we store the voting data is the key to protect privacy and data integrity, including total number of votes for each candidate. In a democracy, a voting system is like in which voters make a choice between well-defined options, often as part of an election. Such systems are having aim of finding out the winner or winners of an election, by determining how individuals express their vote and how votes are tallied. Electronic voting make the voting, tallying and counting processes much faster and easier when compared with a manual process, because we are storing voting data in database which will located on a server, and tallying and counting will be done automatically by the server machine using DBMS. A voting database is a database containing information on voters for the purpose of assisting a political party or an individual politician.

II DATABASE OVERVIEW

A collection of data designed to be used by different peoples is called a database. And this data is stored in a systematic way so that computer program can check it to access the information from the database. Database is also defined as an structured collection of data. It is a collection of interrelated data stored together with controlled redundancy to serve one or more applications in an optimal way. A database is organized in such a way that a computer program can quickly select desired piece of data. DBMS a database management system is nothing but a software that manages all the data in the database and performs queries to database.

The database is a very important aspect in information systems that the database is a repository of data to be processed further. The database is important because it can organize the data; avoid duplication of data, relationships between data that is not clear and also updates complicated. A database management system is a collection of interrelated data and a set of program to access those data. The system which handles the data, transactions, problems or any other aspect of the database is the Database Management System (DBMS). The primary goal of DBMS is to provide a way to store and retrieve database information that is both convenient and efficient. Database system is designed to manage large bodies of information. Management of data involves both defining structure for storage of information and providing mechanism for the manipulation of information. In addition the database system must ensure the safety of the information stored, despite system crashes or attempts at unauthorized access. If data are to be shared among several users, the system must avoid possible anomalous result.[1]

Relational DBMS which use structured query language (SQL) was traditional database system. The latest trend in market is the non-relational database known as NoSQL. Both these databases are equal potential competitors of each other. This study is to find out the suitable database for storage of voting data.

Before going to the comparative study of databases for storing of voting data, let's see basic characteristics of database. These characteristics of database are nothing but CAP theorem that means Consistency, Availability and Partitioning. [1]

Consistency:

Means that once an update operation is finished, everyone can read that latest version of the data from the database and such system is consistent system. While, a system in which, updated data cannot be seen by all users at once is known as eventually consistent. Availability: is achieved if the system always provides continuous operation. Availability is achieved by deploying the database as a cluster of nodes, using replication or partitioning data across multiple nodes. In such case, if one node crashes, the other nodes can still continue to work.

Partition Tolerance:

A database system which can operate even if one of the nodes fail or is inaccessible. This is done by redirecting all queries to the failed node to some other active node of that system. The traditional database systems like SQL, focused on consistency and supports following ACID properties.

Atomicity: Partially accomplished transactions are discarded. Transaction can only be successful or unsuccessful. **Consistency:** In the event of transaction failure, system reverts the transaction and goes back to the previous stable state. And hence, system always remains stable.

Isolation: Transactions are completed without any interference and are processed independently.

Durability: All the committed transactions are saved in logs and will not be lost. This helps to recover the system in case of abnormal terminations.

On the other hand, NoSQL database systems focus more on Availability and Partitioning and give eventual consistency. These systems follow BASE properties.

Basic Availability:

NoSQL database focuses on availability of data as per CAP theorem requirements. Soft State: State of the database system is dynamic and may change over time due to eventual consistency. All the replicas of database do not have to be consistent all the time. Eventual Consistency: After any Write or update or delete operation, system may not immediately reflect the modifications done. But, eventually it will become consistent showing the modified data in all replicas.

III TYPES OF DATABASE

A. Relational DBMS

Relational Database Administration Frameworks (RDBMS) have became most popular to store the information in databases and mostly used for financial records, manufacturing information, staff and salary data, fabricating data, and so on. RDBMSs are based on the relational model, and make use of predefined schema. This model uses two concepts: table and relationship.[3] A table represents a well arranged set of columns and rows, data is stored in these rows and columns and the relationship is nothing but the relation established between the rows of the tables. SQL (Structured Query Language) is used for the purpose of querying and managing the relational data.[8].RDBMS supports all the user requirements along with simplicity, robustness, flexibility, scalability, performance. In RDBMS Related tables can be interlinked together. Various relational databases available are MySQL, Oracle, SQL Server, etc.

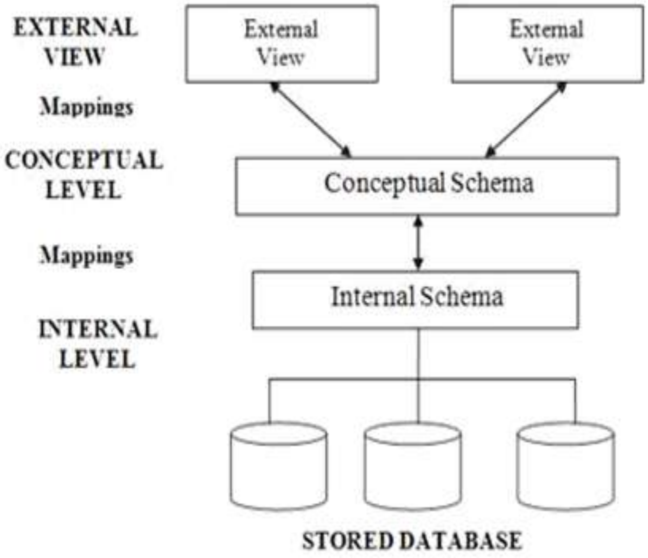


Figure 1 Relational Database

B. Non-relational DBMS:

The NoSQL databases are non-relational, schema free, no joins, easy replication support, horizontally scalable, support for unstructured data, etc. NoSQL databases store data in a different manner as compared to the traditional methods of relational database systems. They are meant for data of schema-less structure, it means NoSQL supports schema-free storage of data in various forms such as document, graph. NoSQL databases can be easily distributed and hence they provide high scalability and availability. NoSQL, as Not Only SQL, is nothing but the non relational data storage, which is developed in 1991, at that time first edition of Berkeley DB was published. It is a key/value database for storage of data at the time of the embedded occasion which requires a high speed of inserting, reading and writing with a relatively simple data type [3]. The idea to design a database like a NoSQL is by reducing the constraints like data consistency and integrity, in exchange for high availability and partition tolerance (as horizontal scalability) because these two constraints are the most demanding, or these are important requirements of Internet application Because now a day's internet applications are producing huge amount of data, and this data requires huge amount of data storage with high performance and low-cost scalability. With the increasing use of the Internet and the availability of cheap storage, massive amounts of structured, semi structured and unstructured data are created and stored by a variety of applications.

Usually such data in large-scale is known as big data [2]. Processing large amounts of data requires fast machines, flexible database schemas and distributed architectures that do not fit into relational databases. NoSQL databases claim that they provide easy access, high speed and development capabilities for working with large data. NoSQL databases has become very famous because of the impressive features that they are providing like high scalability, easy access, and distributed architecture, etc. After 2007 till now, a lot of the popular products has emerged, such as Google's BigTable, Apache's HBase, FaceBook's Cassandra, 10gen's MongoDB, Yahoo!'s PNUTS, Redis, CouchDB, etc. These databases generally provide the high performance of read and write and great horizontal scalability, which perform well in many practical Internet applications.

IV MONGODB

A. Overview

MongoDB is one of the NoSQL document-oriented database. MongoDB is an open-source database, means you can use it at free of cost. It is scalable in nature. MongoDB is written by using C++ language libraries. This instance made the NoSQL databases to follow high scale systems with faster access [8]. MongoDB is a NoSQL database means it is non-relational database. MongoDB which is a NoSQL database and Relational-DBMSs are different from each other in terms of their implementation and operational concepts.

B. Data Storage Mechanism:

Document oriented Mechanism:

MongoDB is one of the most famous NoSQL databases, whose important intent is to fill the gap between key-value stores providing high performance and scalability and traditional Relational DBMS having rich management, and take the advantages of both in one. MongoDB is developed by the 10gen company, and now many SNS (Social Networking Services) applications like shutter fly, foursquare, bit.ly etc. is using it. the most major production and service of MongoDB, like file storage, session server, and user tracking, that MongoDB is currently powering them. MongoDB is a database that supports document-oriented storage [3] that means document is the minimum independent storage unit. So that field design can be different in different document, even the documents are in the same collection. This is providing the schema-free feature to MongoDB. There is no need schema to specify the schema at the time of creating a document or collection. Each document in the same collection can have the different schema. we can dynamically add or delete the fields In document at any time when the database is running. All of these bring the convenience to the design of database storage model. As same using the social site as the example, the amount of fields in each state document are different depending on the response number. When a response is submit, we add a field to store it. The user's response is changed dynamically. If we want to pre-determine the schema of the database, that would be difficult. The schema-free feature of MongoDB solves the problem completely.

C. Architecture of MongoDB

MongoDB architecture is built with many components; Configuration servers, Shared nodes and Routing services or mongos. The architectural instance is shown in Fig. 2. Shard Nodes consist of one or more nodes which are responsible for storing the data. Corresponding read and write queries are routed to the appropriate node that holds the data. A replicated node comprises of one or more servers and acts as a primary replica to give the backup for failure instances [3]. Configuration servers store metadata and routing related information of MongoDB which is acting as a current primary replica [3]. Routing services are also called as mongos and perform tasks related to different clients. Routing server also handles different queries issued by various clients. The result of a query will be send to the related node before it sends back to the client. MongoDB uses memory-mapped files to increase the performance. It also supports auto-sharding instance which increases the scaling behavior across multiple nodes.

A MongoDB system holds a set of databases, each database holds a set of collections, a collection is a set of documents, each document is a set of fields, and a field is a key-value pair, key is the name of the field with the string type, value is the data of the field and can be any type. Fig. 3 shows the data model.

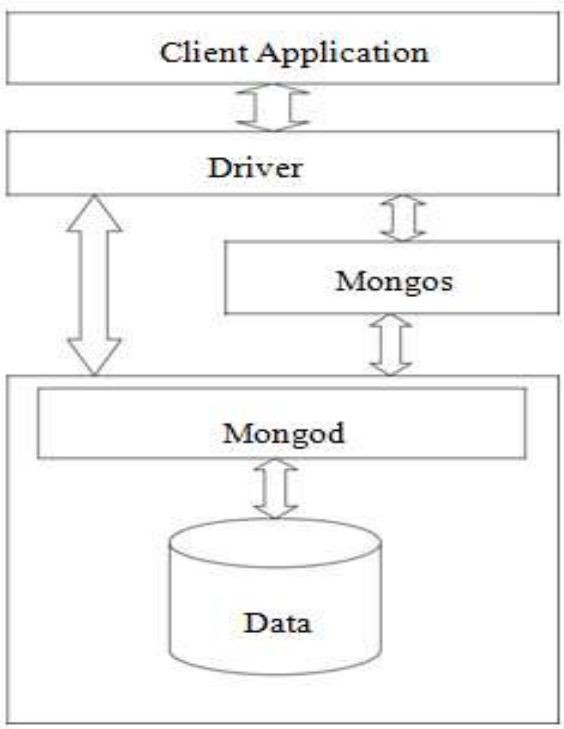


Figure .2 Architecture of MongoDB

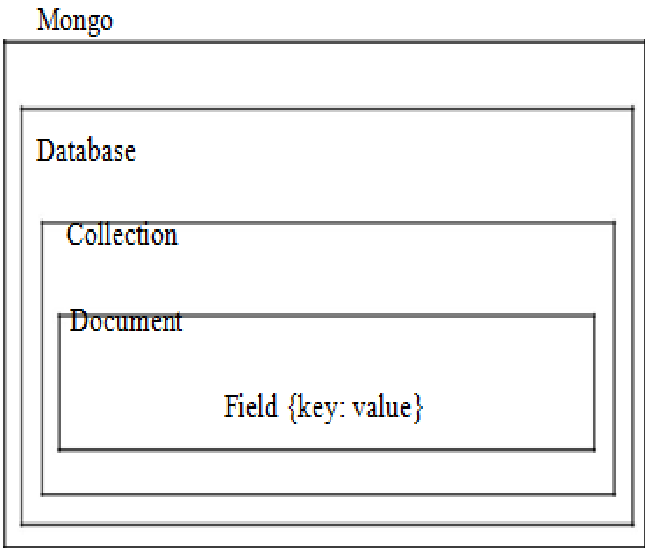


Figure.3 Data Model of MongoDB

D. Features of MongoDB

BSON

MongoDB makes use of BSON for the purpose of data storage and network transfer format documents. BSON is a binary-encoded serialization of JSON-like documents BSON stores key-value pair in binary byte with the key to store a string type data and the value to store data of any type such as document and array. BSON provides an impressive feature to MongoDB– nested document and array that means the type of value that is stored in a document's field that value is not changed is still an array or document, for which we can unite number of collections consider such as joining with foreign key to each other. It improves the performance of database by minimizing the joins between numbers of collections.[3] BSON (Binary encoding of JSON) are used to store documents. In MongoDB specific schema are not designed, so that we can insert JSON (Java Script Object Notation) documents.

GridFS

MongoDB also supports a special feature called GridFS; which is nothing but a complex query language. GridFS is used to store and retrieve files having size more than 16MB[3] such files are like audio, video etc. Because of these added features, MongoDB projects have been considered to process large amount of data over Relational DBMS.

(CIA) Confidentiality, integrity and availability related Issues in MongoDB

Instead of the various facts there are many security suggestions given in the current MongoDB manual [6], number of security weaknesses have been uncovered by the hackers group. Arranging security alternatives and leaving trusted correspondence to the application developer isn't a strong way to provide security. There are some well known security loopholes in mongodb, and these are as follows :

a) Database can be installed without any authentication means without password credentials.

b) Any created user has read-only access to the whole database.

c) Port 28017 is susceptible to internet based attacks.

d) Database does not inherently provide encryption to the data. External protection mechanism has to be used.[7]

Recent attacks on MongoDB

Authors in [6] clearly clarified that a French media transmission organization's information was traded off in an assault on the default MongoDB port 27017. Shodan search engine was utilized to remove the points of interest of the IPs engaged with MongoDB. Information breach that occurred in a firm that was supported by MongoHQ, database as an administration, claimed that the attackers may have accessed a supported application that contained user credentials of one of its users [7], thereby impacting confidentiality and integrity.

A. Security Extension

MongoDB coordinates with outside security improving items like Kerberos authentication software and IBM's Guardium encryption software[6]. Compared with Hadoop and MySQL MongoDB's security upgrading outside items and potential outcomes are lesser.

B. Cloud Security

MongoDB can be conveyed on various cloud stages like Rack space cloud [7], Amazon EC2. In cloud computing, MongoDB's CIA is improved utilizing condition's capacities as clarified here. Whenever MongoDB is sent on the cloud, it should be ensured that the cases are available just by approved clients, which is done utilizing "Ip tables" command line device. Once the instances are secured, firewall setups are utilized to empower port22 to acknowledge any TCP association. As given in, when MongoDB is sent on Amazon EC2, the access is limited utilizing the security gathering of the amazon web services (AWS). Security Group is set of firewall arrangement rules that scans and limits incoming packets. MongoDB appear to depend more on the cloud's local security approach to improve its own access control in this manner guaranteeing confidentiality and integrity.

F. Concurrency Control

MongoDB additionally implements concurrency control. It gives multi-granularity locking and because of that it is allowing locking at worldwide, database or record collection level. There is shared access considered readers for reading an information and selective access for authors who writes or alter a resource, i.e. different clients can get an access to resources for reading reason however just a single client can keep in touch with an source at any given moment. Better granularity locking is finished utilizing intent locks.[6]

V MYSQL

A. Overview

MySQL is the most trusted and depended-on open source database platform in use today. Many of the most popular and highly-trafficked websites in the world are built on MySQL because of its ubiquity across heterogeneous platforms and application stacks and for its well-known performance, reliability and ease of use. MySQL is a relational database system and has support for all the user requirements along with simplicity, robustness, flexibility, scalability, performance.

B. MySQL's Storage Engines

MySQL has various storage engines, and each storage engine has a mechanism to store data. MySQL stores each database (also called a schema) as a subdirectory of its data directory in the underlying file system. When you create a table, MySQL stores the table definition in a .frm file with the same name as the table. Thus, when you create a table named MyTable, MySQL stores the table definition inMyTable.frm.[13] Because MySQL uses the file system to store database names and table definitions, case sensitivity depends on the platform. On a Windows MySQL instance, table and database names are case insensitive; on Unix-like systems, they are case sensitive. Each storage engine stores the table's data and indexes differently, but the server itself handles the table definition.

1. The InnoDB

Engine InnoDB

Overview

InnoDB is the default value-based storage engine for MySQL and the most imperative and comprehensively helpful engine generally. It was intended for handling some short lived transactions that typically entire as opposed to being moved back. Its execution and automatic crash reference make it prominent for non-transactional capacity needs, too. You should utilize InnoDB for your tables unless you have a convincing need to utilize an alternate engine.

InnoDB Storage Mechanism

InnoDB stores its data in a sequence of one or more data files that are together known as a tablespace. Essentially tablespace is a black box that InnoDB manages all by itself. In MySQL 4.1 and newer versions, InnoDB can stores data from each table and make indexes in separate files. For building its tablespace, InnoDB utilizes raw disk partitions but for modern file systems this is unnecessary. InnoDB utilizes MVCC to accomplish high simultaneousness, and it executes every one of the four SQL standard isolation levels. It defaults to the REPEATABLE READ isolation level, and it has a next-key locking strategy that averts phantom reads in this isolation level: as opposed to locking just the lines you've touched in a question, InnoDB secures holes in the list structure too, keeping phantoms from being embedded. InnoDB tables are based on a clustered list. InnoDB's list structures are altogether different from those of most other MySQL storage engines. Therefore, it gives extremely fast primary key queries. However, secondary indexes (indexes that aren't the essential key) contain the essential key sections, so if your essential key is vast, different records will likewise be extensive. You ought to make progress toward a little essential key on the off chance that you'll have numerous files on a table. The capacity organize is stage unbiased, which means you can duplicate the information and list documents from an Intel-based server to a PowerPC or Sun SPARC with no inconvenience.[12]

InnoDB has various interior enhancements. These incorporates predictive read-ahead for prefetching information from disk, a versatile hash list that consequently assembles hash files in memory for quick queries, and an embed buffers to speed inserts.[13] As a value-based storage engine, InnoDB supports truly "hot" online reinforcements through variety of ways, including Oracle's restrictive MySQL Enterprise Backup and the open source Percona XtraBackup. MySQL's other stockpiling motors can't take hot reinforcements—to get a steady reinforcement, you need to end all keeps in touch with the table, which in a blended read/compose workload more often than not winds up stopping peruses as well.

2. The MyISAM

Engine Overview:

As MySQL's default storage engine in version 5.1 and older, MyISAM gives large number of features, for example, full-content ordering, compression, and spatial (GIS) functions. MyISAM doesn't support transactions or row level locks. Its greatest shortcoming without a doubt is the way that it isn't even remotely crash-safe. MyISAM is the reason MySQL still has the notoriety of being a non-transactional database management system, over 10 years after it picked up exchanges! In any case, MyISAM isn't too terrible for a non-transactional, non-crash-safe storage engine.[13] In the event that you require read-only data, or if your tables aren't extensive and won't be difficult to repair, it isn't not feasible to utilize it. (Be that as it may, kindly, don't utilize it as a matter of course. Utilize InnoDB.)

Storage Mechanism:

MyISAM regularly stores each table in two files, a data file and an index file.The two records bear .MYD and .MYI extensions, separately. MyISAM tables can contain either dynamic or static (settled length) lines. MySQL chooses which format to utilize in view of the table definition. The quantity of rows a MyISAM table can hold is constrained fundamentally by the available disc space on your database server and the biggest

file your working framework will give you a chance to make. MyISAM tables made in MySQL 5.0 with variable-length rows are arranged as a default of course to deal with 256 TB of data, utilizing 6-byte pointers to the information records. Prior MySQL forms defaulted to 4-byte pointers, for up to 4 GB of information. All MySQL adaptations can deal with a pointer size of up to 8 bytes. To change the pointer measure on a MyISAM table (either up or down), you should alter the table with new values for the MAX_ROWS and AVG_ROW_LENGTH options that speak to rough approximations for the measure of room you require. This will cause the whole table and the greater part of its files to be modified, which may take quite a while.[12]

MyISAM features

As one of the oldest storage engines included in MySQL, MyISAM has many features that have been developed over years of use to fill niche needs:

1. Locking and concurrency:

MyISAM locks whole tables, not rows. Readers get shared (read) locks on all tables they have to read. Writers acquire exclusive (compose) locks. Be that as it may, you can embed new rows into the table while select queries are running against it (concurrent embeds).

2. Repair:

MySQL supports manual and automatic checking and repairing of MyISAM tables, however don't mistake this for exchanges or crash recovery. In the wake of repairing a table, you'll likely locate that a few information is basically gone. Repairing is moderate, as well. You can utilize the CHECK TABLE mytable and REPAIR TABLE mytable commands to check a table for blunders and repair them. You can likewise utilize the myisamchk command-line apparatus to check and repair tables when the server is offline.

3. Index features:

You can create indexes on the first 500 characters of BLOB and TEXT columns in MyISAM tables. MyISAM supports full-text indexes, which index individual words for complex search operations.

4. Delayed key writes:

MyISAM tables marked with the DELAY_KEY_WRITE create option don’t write changed index data to disk at the end of a query. Instead, MyISAM buffers the changes in the in-memory key buffer. It flushes index blocks to disk when it prunes the buffer or closes the table. This can boost performance, but after a server or system crash, the indexes will definitely be corrupted and will need repair. You can configure delayed key writes globally, as well as for individual tables.[13]

VI ARCHITECTURE OF MYSQL

A good picture of how MySQL’s components work together will help you understand the server. Following fig. shows a logical view of MySQL’s architecture.[14]

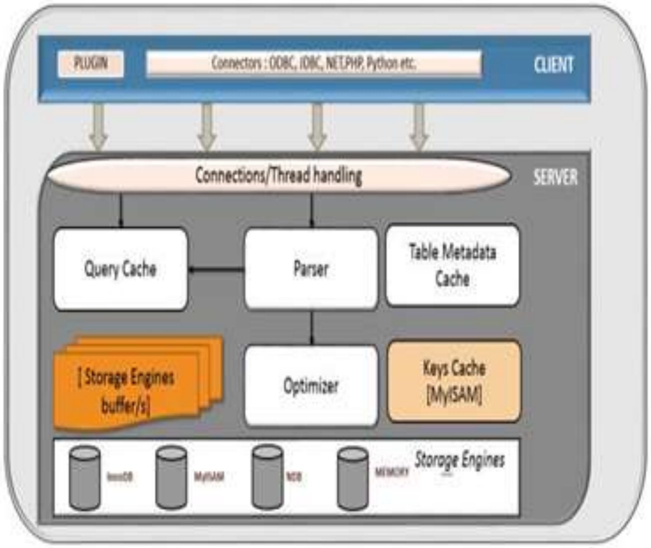


Figure 4. MySQL Architecture [14]

Components:

Client:

This is the first and topmost layer in MySQL architecture, which is nothing but a utility to connect MySQL server. This layer contains the services that are not unique to

MySQL, means MySQL is known about the services that are submitted by the clients. They're services most network-based client/server tools or servers need: connection handling, authentication, security, and so forth.

Server:

This is the second layer in MySQL architecture, which is nothing but a MySQL instance where actual data getting stored and data processing is happening. The second layer is where things get interesting. Much of MySQL's brains are here, including the code for query parsing, analysis, optimization, caching, and all the built-in functions (e.g., dates, times, math, and encryption). Any functionality provided across storage engines lives at this level: stored procedures, triggers, and views, for example.

mysqld:

MySQL Server daemon program which runs in the background and manages database related incoming and outgoing requests from clients. mysqld is a multi-threaded process which allows connection to multiple sessions listen for all connections and manages MySQL instance.

MySQL Memory Allocation:

Main MySQL memory is dynamic, examples innodb_buffer_pool_size key_buffer_size etc. Working on shared nothing principal which means, every session has unique execution plan and we can share data sets only for the same session.[14]

The functioning modules in MySQL architecture are as follows:

Connection/ThreadHandling:

This module Manages client connections or sessions which are nothing but mysql threads.

Parser:

Parser Check for SQL syntax by checking every character in SQL query and generate SQL_ID for each SQL query.

Optimizer:

Optimizer creates efficient query execution plan as per the storage engine. It will rewrite a query. Example: InnoDB has shared buffer so optimizer will get pre-cached data from it. Using table statistics optimizer will generate an execution plan for a SQL query.

Metadatacache:

This module Cache for object metadata information and statistics.

Querycache:

This module is responsible to Share identical queries from memory. If an identical query from client found in query cache then, the MySQL server retrieves the results from the query cache rather than parsing and executing that query again. It's a shared cache for sessions, so a result set generated by one client can be sent in response to the same query issued by another client. Query cache based on SQL_ID.SELECT data into view is the best example of pre-cache data using query cache.

Keycache:

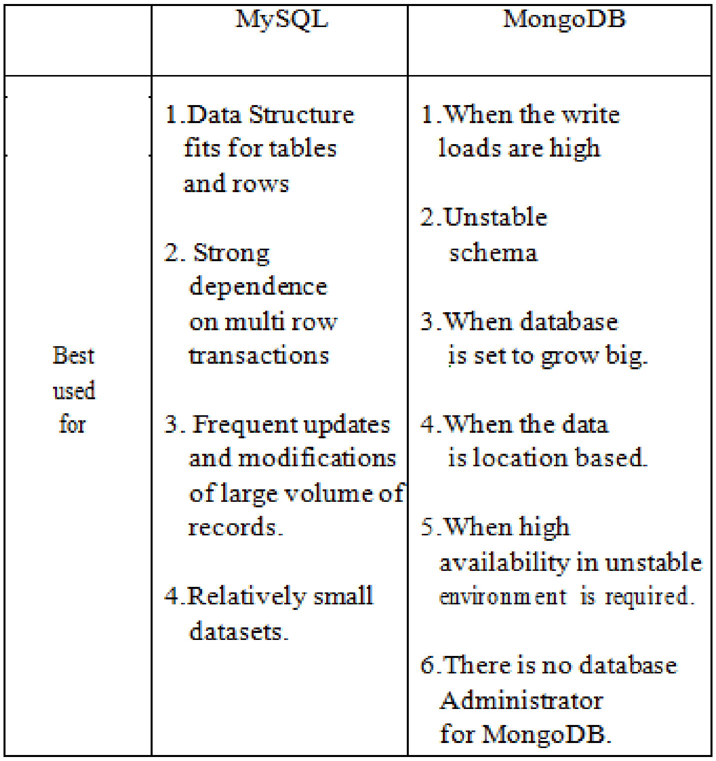
Cache table indexes. In MySQL keys are indexes (In oracle keys are constraints) if index size is small then it will cache index structure and data leaf. If an index is large then it will only cache index structure. Used by MyISAM storage engine.

Storage engine:

This is the last and important module in MySQL architecture which is nothing but the storage engines. They are responsible for storing and retrieving all data stored "in" MySQL. Like the various file systems available for GNU/Linux, each storage engine has its own benefits and drawbacks. The server communicates with them through the storage engine API. This interface hides differences between storage engines and makes them largely transparent at the query layer. The API contains a couple of dozen low-level functions that perform operations such as "begin a transaction" or "fetch the row that has this primary key." The storage engines don't parse SQL or communicate with each other; they simply respond to requests from the server. MySQL component that manages physical data (file management) and locations. Storage engine responsible for SQL statement execution and fetching data from data files. Use as a plugin and can load/unload from running MySQL server.[14]

VII RESULT ANALYSIS

Table 1 .Comparative analysis.



	MySQL	MongoDB
Best used for	1.Data Structure fits for tables and rows 2. Strong dependence on multi row transactions 3. Frequent updates and modifications of large volume of records. 4.Relatively small datasets.	1.When the write loads are high 2.Unstable schema 3.When database is set to grow big. 4.When the data is location based. 5.When high availability in unstable environment is required. 6.There is no database Administrator for MongoDB.

There are some environments nothing but the applications in which a particular database is well suitable. Because every application's requirements is different, and the database which will fulfill the requirements will be best suitable for that particular application

VIII CONCLUSION

From the survey of the databases: MySQL and MongoDB, we surveyed various functionalities, their working, their storage mechanisms, different performance impacting features of these databases and assessing maturity of these database. it is much clear that NoSQL databases are evolving at a great speed in terms of handling big data and its storage. Relational databases are primarily meant to be transactional databases, But relational databases seem to have more in-built security features while NoSQL databases, being open source, is less secure. Being open source non-relational database, MongoDB does not fulfill the security requirements at full extent. As compared to MongoDB, MySQL database provides security in much more extent. Because of the security provided by the MySQL many big organizations like NASA, Facebook, Twitter, US Navy, others are using MySQL database to store their sensitive data.[12] That's why we choose to use MySQL database for our SMART Voting project, because votes are very sensitive data.

ACKNOWLEDGEMENT

We would like to express thanks of gratitude towards our guide Dr. Ayesha Butalia, HOD Shreekant Dhamdhere and our close supporters.

REFERENCES

[1] Sharvari Rautmare, Dr. D. M. Bhalerao “MySQL and NoSQL database comparison for IoT application”, in proceeding of 2016 IEEE International Conference on Advances in Computer Applications (ICACA), 24-24 Oct. 2016

[2]Sudhanshu Kulshrestha, Shelly Sachdeva, “Performance Comparison for Data Storage - Db4o and MySQL Databases”, in proceeding of Contemporary Computing (IC3), 2014 Seventh International Conference, 7-9 Aug. 2014

[3] Yunhua Gu, Xing Wang, Shu Shen, Jin Wang, Jeong-Uk Kim, “Analysis of Data Storage Mechanism in NoSQL Database MongoDB”, in proceeding of Consumer Electronics - Taiwan (ICCE-TW), 2015 IEEE International Conference, 6-8 June 2015

[4] Yunhua Gu1, Shu Shen1, Jin Wang1, Jeong-Uk Kim2, “Application of NoSQL Database MongoDB”, in proceeding of Consumer Electronics - Taiwan (ICCE-TW), 2015 IEEE International Conference, 6-8 June 2015

[5] Liana Stanescu, Marius Brezovan, Dumitru Dan Burdescu, “Automatic Mapping of MySQL Databases to NoSQL MongoDB”, in proceeding of Computer Science and Information Systems (FedCSIS), 2016 Federated Conference,11-14 Sept. 2016

[6] Lior Okman, Nurit Gal-Oz, Yaron Gonen, Ehud Gudes, Jenny Abramov, “Security Issues in NoSQL Databases”, in proceeding of Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference,16-18 Nov. 2011

[7] Sethuraman Srinivas, Archana Nair, “Security Maturity in NoSQL Databases – Are they Secure Enough to Haul the Modern IT Applications?”, in proceeding of Advances in Computing, Communications and Informatics (ICACCI), 2015 International Conference,10-13 Aug. 2015

[8] Dharavath Ramesh, Shankar Nayak Bhukya2, “Inclusion of e-Commerce Workflow with NoSQL DBMS: MongoDB Document Store”, in proceeding of Computational Intelligence and Computing Research (ICCIC), 2016 IEEE International Conference,15-17 Dec.