International Journal of Electrical, Electronics and Computer Engineering **13**(1&2): 01-04(2024)

ISSN No. (Online): 2277-2626

Performance Evaluation of IoT Database Management using Mongo DB versus MYSQL Databases

Shivani Sharma*
Department of School of Computer Science and Engineering
Govt. P.G College Dharamshala, Himachal Pradesh Technical University (HPTU) (Himachal Pradesh), India.

(Corresponding author: Shivani Sharma)*

(Received: 03 January 2024, Accepted: 15 March 2024)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Internet of Things (IoT) is the latest enhancing technology in which all the physical devices are interconnected through internet. Physical devices can be refrigerator, door, watches, geyser and many more. These devices use Sensors to percept the environment and send the information to connected device. There is large amount of data that an IoT device stores. So, it's challenging to store that huge amount of data. In this research paper we will study and evaluate the performance of SQL and NOSQL DBMS further we shall conclude why IoT prefers NOSQL DBMS to store its Data.

Keywords: Sensors, percept, database management, SQL, NOSQL.

INTRODUCTION

Internet of things is the most emerging and enhanced technology in today's World. By using Internet of Things, we can provide internet facilities to all the Physical Things that already exist in this era of technology. The main motive of using IoT is to make the things SMARTER. The word 'Things' in Internet of Things can be any physical entity that is existing in this world. These things can be AC, Alarm watches, curtains and many more. It is a Giant Network with connected devices. These devices gathers and store data about how they are used and environment in which they are operated Eyada *et al.* (2020). IoT devices uses Sensors to detect the environment and send the detected data to cloud and further analysis is performed on the received data and useful data is fetched from it and then it is sent to the user for further action. Let us understand IoT with an example of Smart Home.

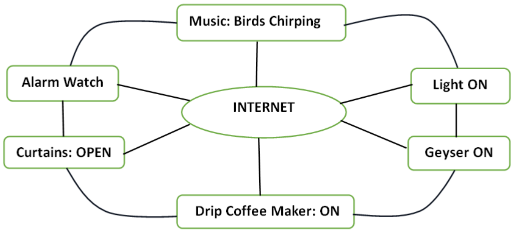


Fig. 1. Smart Homes.

In this example: All the physical things in the home are interconnected and also connected to the internet. If the alarm watch rings in the morning, then curtain will open automatically then following the order drip coffee maker then geyser then light then music will automatically play without the human need. In this way we can make our homes smarter by using IoT.
These IoT devices percepts the data and send it continuously and every time this data is varying. For Eg: If an IoT device has to keep track on the temperature then it will continuously keep checking it and will also send the data constantly. This will be huge amount of varying data. So, it becomes challenging to manage this data. So, we need a database management system by which we can manage our continuously varying huge data. There are two types of DBMS, SQL and NOSQL.

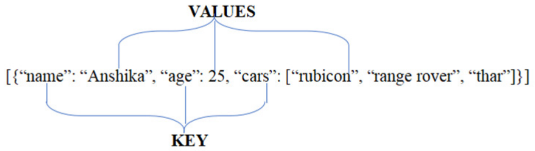
SQL DATABASE MANAGEMENT SYSTEM

SQL databases are those databases which stores the data in a structural format. MYSQL is a DBMS which uses SQL (Structured Query Language) to store the data in tabular form. It is a RDBMS (Relational Database Management System) in which we store our data in tabular form. In tables there are rows and columns which contains the actual data. Rows are called Tuples and columns are called Attributes.
MYSQL is a Rigid way of storing the data in which manipulation in the data is little complex. SQL databases are the Schema based databases. Schema is a structural representation of the Database. In a Schema based database it is difficult to modify or add new applications. MYSQL databases uses Binary Tree format to store huge amount of data. Data is stored in Data Pages and huge amount of data can contain large number of Data Pages as the size of one Data Page is 8Kb.

NOSQL DATABASE MANAGEMENT SYSTEM

NOSQL databases are those databases in which we can store our data in non-structural or semi-structural format. MONGODB is a NOSQL DBMS in which we store the data in BSON (Binary JavaScript Object Notation) format. It is a binary representation of JSON (JavaScript Object Notation) format. We import the data to database in JSON format but internally it is converted to BSON format by MONGODB Drivers. MONGODB stores the data in the Documents. These documents contain the fields which are formed by key-value pairs. Documents represents the rows and fields are the columns. MONGODB is a Schema-less database in which we can add new applications without any difficulty Rautmare & Bhalerao (2017). Two documents can have same fields or different which makes it more flexible to adopt the changes more efficiently.

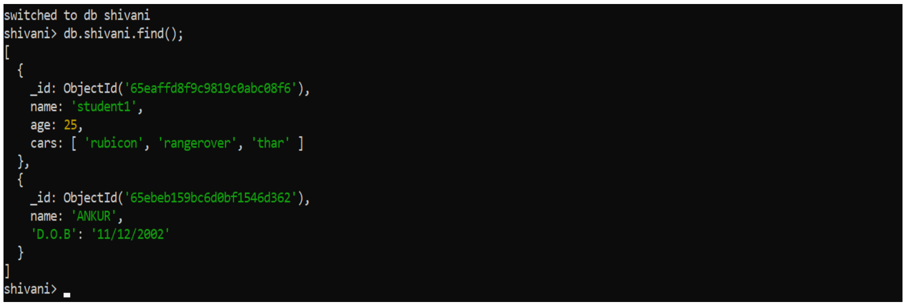
JSON Format: It is a text-based notation to exchange the data between server and client. It is a user-friendly format in which data is store in the form of Key-Value pairs and arrays. The data types supported by JSON are String, Number, Object, Array, Boolean (True or false) and null. EG:



```
[{“name”: “Anshika”, “age”: 25, “cars”: [“rubicon”, “range rover”, “thar”]}]
```

Fig. 2. JSON Representation.

BSON Format: It is a binary representation of the JSON format. It is a machine-friendly format in which data is internally stored in the database. JSON format is basically used for transportation while BSON format is used for Storage. BSON supports some extra datatypes such as it adds an object id automatically with each document which is uniquely identified to all the documents and cannot be same for two documents. The data types supported by BSON are String, Number (Integer, Float, Decimal), Array, null, date, BinData.



```
switched to db shivani
shivani> db.shivani.find();
[
  {
    _id: ObjectId('65eaffd8f9c9819c0abc08f6'),
    name: 'student1',
    age: 25,
    cars: [ 'rubicon', 'rangerover', 'thar' ]
  },
  {
    _id: ObjectId('65ebeb159bc6d0bf1546d362'),
    name: 'ANKUR',
    'D.O.B': '11/12/2002'
  }
]
shivani> _
```

Fig. 3. BSON Representation.

In this picture we can see how the id is generated with each document automatically as data is converted from JSON format to BSON by MONGODB drivers.

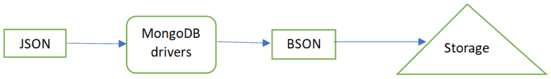


Fig. 4. Mongo DB Drivers.

IoT ARCHITECTURE

IoT architecture is the standard way of representing the working of an IoT device. It tells us how an IoT device works. The working of IoT device is divided into several layers where each layer performs several tasks Mohammad Ali Jabraeil Jamali (2020). There are mainly four layers in IoT architecture and they are as follows:

1. Sensing Layer 2. Transport layer
3. Data Processing Layer 4. Application layer

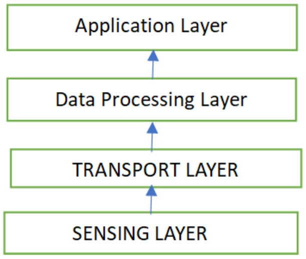


Fig. 5. IoT Layers.

1. Sensing layer: This is the first layer in IoT architecture and it is responsible for collecting the information from the environment using sensors and actuators. For Eg: In a temperature tracking system, it is mandatory that temperature in the room should not go down 6 deg Celsius, and if it goes down then an alert is sent to the user mobile phone. So, here sensors will keep track on the temperature and keep checking it again and again in every second.

2. Transport layer: This is the second layer in IoT architecture and it is responsible for transmitting the data from sensors to other connected devices mainly to the cloud. It includes various protocols like TCP (Transmission Control Protocol) and technologies to connect and transfer the data between connected devices. It uses technologies like Internet (3G, 4G, 5G), Wi-Fi, Bluetooth and can also use gateways and routers to connect to other devices in the network.

3. Data Processing Layer: This is the main layer in IoT architecture in which we analyze the data received from sensing layer. In this layer all the calculations are performed and all the algorithms work in this layer. For Eg: The temperature data is received from the sensing layer and here all the conditions (if else) are interpreted and checked whether the received temperature is below 6 deg Celsius or not.

4. Application Layer: It is the topmost layer in IoT architecture which interacts with the user directly. By Application layer user can access or control Io T devices. This can include mobile apps and other interfaces by which user can monitor IoT devices. For Eg: After processing the data, it is analyzed that temperature has gone below 6 deg Celsius, now an alert is sent to the user mobile and then user can monitor the temperature remotely just by using the mobile phone.

HETEROGENITY IN IOT DATA

IoT data is heterogenous in nature. Heterogenous means data is coming from various sources and in various formats. This data can include Structured, Semi-Structured, Non-Structured, Sensor data, Device status, Unique Identifiers, Location data and many more. The various types of data that an IoT device can have Cooper & James (2009).

1. Unique Identifiers: Unique Identifiers are the addresses which are given to each device connected to the internet. Each device is associated which unique IP(Internet Protocol) address. Due to increasing IOT devices there is shortage in IP addresses. So, the IPV4 (Version 4) is now replaced with IPV6 (Version 6) as the Version 4 was suffering from exhaustion in number of IP addresses generation. IPV4 is 32-bit IP address and it can generate 2^{32} IP address. But number of IOT devices are increasing and IPV6 is 128-bit IP address and it can generate 2^{128} unique IP addresses.

2. Location Data: Location data provides the position of the IoT object within global positioning system (GPS). GPS real time tracking software are integrated with IoT device which tells the exact location of the device on the map [2].

3. Sensor Data: Sensor data is data which an IoT device collects from the environment using sensors and actuators. This data is continuously changing data and it can by anything about the environment. For Eg: Temperature detection, Humidity detection, Rainfall detection, Smoke detection and many more. It can be of any form like text format, degree, percentage etc.

4. Metadata: Metadata is the data about data. It is in depth information about the data. It can be of various types like, the time when data was generated or the system who generated it and many more. It is used to describe and explain each device in the network.

5.Past/ Historical Data: It is the data collected in past and we need to store this data as Historical Data is use for comparing the performance of the devices in the internet and for many purposes. Historical data is always helpful when we need to analyze the changes in the environment. So, it becomes challenging to store this huge volume of past data.

PROBLEM IN MANAGING IOT DATA

IoT data is heterogeneous, rapidly changing, huge amount, un-structured, of various types, So, it becomes challenging to store this vast volume of data. We need a proper database management system to manage this data. As the IoT technology is enhancing day by day so, we cannot neglect its database management as it will decrease the overall performance of IT technology. Also, we need to shard our vast volume of IOT data.

Sharding: Sharding is a technique to distribute vast single dataset across multiple machines so that we can handle huge data on various machines. It is just the process of managing the huge data by dividing it. IoT data is huge and it is necessary to shard this data. Now we will evaluate SQL and NOSQL DBMS to store and manage IoT Data.

MONGODB vs MYSQL FLEXIBILTY

1. Flexibility in database is the term which defines how easily the database can adopt the changes or how easily we can modify it.

2. As we discussed MONGODB is schema-less, we do not need to follow any format to add new applications. We can add new fields to any document in MONGODB without any restriction. So, MONGODB is very Flexible.

3. While on the other hand MYSQL databases are the schema-based databases. It becomes challenging to add new applications in existing table. So, MYSQL is not very Flexible.

4. As the data in IoT is of various types, if any IoT device wants to add extra functionality to data then it will be easier if DBMS is flexible. So, for flexibility we will choose MONGODB Mongo (2024).

MONGODB vs MYSQL USER-FRIENDLINESS

1. In MONG0DB we add the data in documents which are in JSON format which are easy to understand.

2. While in MYSQL we add the data into tables, proper queries need to be executed to add new data into tables. Understanding the Queries is little complex. So, in terms of easiness we will choose MONGODB.

MONGODB vs MYSQL SCALABILTY:

1. Scalability is the ability of the DBMS to handle or manage huge volume of data. This can also be defined as the ability to handle multiple requests at one time. Scalability in DBMS is achieved by sharding.

2. By Sharding we divide the big dataset into smaller datasets by dividing it horizontally that will run on different machines and then we can handle multiple queries at one time.

3. MYSQL Sharding is a complex process because in MYSQL there are many relationships. This will make it difficult to manage sharded databases. It does not allow cross joins across multiple sharded databases.

4. MYSQL supports joins where two tables are dependent on each other and can share their data but by sharding the MYSQL datasets, there will be loss of shared data as we divide the data in sharding.

5. In MONGODB there are no relationships and also it supports less joins. So, Sharding in MONGODB is a great task to handle the huge volume of data without any loss. For Scalability we will choose MONGODB.

MONGODB vs MYSQL SPEED

We can define Speed as how fast Database is responding to the query.

1. In MONGODB there is no schema and one document contains related data. But in MYSQL when it comes to related data, there are joins by which two tables are connected and they share the data.

2. So, it will be faster to fetch the data from one collection rather than fetching it from two join tables. So, MONGODB is faster than MYSQL.

CONCLUSIONS

After studying and comparing both the DBMS (MONGODB AND MYSQL) we conclude that MONGODB (NOSQL) is the best choice to store and manage huge, rapidly changing, of various types data.

MONGODB leads in Flexibility, Scalability, easiness and in many properties from MYSQL. MYSQL is a popular choice when we are dealing with structured data. But now a days we are dealing mostly with unstructured data, that is it can be in form of Images, Audios, Videos and many more. Even when we are uploading any pictures they are in the form of Unstructured data. So, we can clearly say it is the generation of Unstructured data. That's why MYSQL is not that much popular now. In IoT, data is heterogeneous, Unstructured (mostly), rapidly changing, huge volume. So, we evaluated MONGODB is the best choice to store and manage IoT Data.

Acknowledgement. The author would like to express her sincere gratitude to all the faculty members of department for their invaluable guidance, unwavering support, and continuous encouragement throughout the course of this research endeavor.

REFERENCES

Cooper, J., & James, A. (2009). Challenges for Database Managemnt in the Internt of Things. IETE Technical Review.

Datarade (nS.d.). IoT Data: Best IoT Datasets & Databases. Retrieved from https://datarade.ai/data-categories/iot-data.

Eyada, M., Saber, W., Genidy, M. M., & Amer, F. (2020). Performance Evaluation of Iot Data Management Using Mongo DB Versus MYSQL Databases in Different Cloud Environments. IEEE Access.

Mohammad Ali Jabraeil Jamali, B. B. (2020). Towards the Internet of Things.

Mongo, D. B. (2024). MongoDB vs. MySQL Differences. Retrieved from https://www.mongodb.com/compare/mongodb-mysql.

Rautmare, S., & Bhalerao, D. M. (2017). MySQL and NoSQL database comparison for IoT application. 2016 IEEE International Conference on Advances in Computer Applications (ICACA). Coimbatore, India: IEEE.