MONGO DB FOR DATA AS A SERVICE

Nipunika. A[1], Sai Rohita Desai[2]
Sreenidhi Institute of Technology and Science, Medchal , Telangana.

Abstract : Relational databases are used store financial records, keep track of inventory and to keep records on employees. In a relational database, information is stored in tables. The expensive of setting up and maintaining the database system Advances in the complexity of information databases have limits on field lengths and type. Mongo DB is a cross-platform document-oriented database program, Classified as a NoSQL database program and Mongo DB uses JSON-like documents with schema. It is non relational database technologies which arose under the NoSQL banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model Document databases store semi-structured data and descriptions of that data in document format. They allow developers to create and update programs without needing to reference master schema. Document databases are used for content management and mobile application data handling. Mongo DB has developed a tried and tested approach to constructing an Operational Data Layer. This process for constructing an Operational Data Layer has been successfully implemented with many customers. A successfully implemented ODL is a springboard for agile implementation of new business requirements.

***IndexTerms* - Document-based , NoSQL, Time stamp, scalability , Sharding.**

I. INTRODUCTION

Relational databases are two dimensional storage tables which are used to store financial records, keep track of inventory and to keep records on employees. In a relational database, information stored is inflexible and static. The cost of setting up and maintaining the database system is considerable. Advances in the complexity of information have further increased the difficulty in maintaining the databases consistent and properly functioning. Also, databases have limits on field lengths, type, and negligible flexibility for increasing varieties of data.

At present, where the data is largely being produced and consumed, traditional databases are falling short to satisfy the needs of the organizations. The data is not only growing in terms of volume but also variety and variability. Organizations are now in a need of fast, flexible and affordable databases.

With the advent of concepts such as Big Data, Data Analytics and Data Science relational databases are now replaced with various other databases which have various performance extensions such as scalability, speed, etc. Also, presently, people are not just behind storage facility given by the databases. Efficiently managing and performing various strategies has resulted in many of the organizations.

Mongo db is one of these databases with various additional features available which makes it easier for the user to use and access data in a better and efficient way.

Mongo db was created by Eliot Horowitz, who had encountered development and scalability issues with traditional relational database approaches. Meaning –huMongo us-represents the idea of supporting large amounts of data. The Mongo db Inc. Company began to develop a document-oriented database system it called Mongo DB.

II. LITERATURE REVIEW

This paper emphasizes on two main concepts – Firstly, the need and features of Mongo db. The drawbacks of the existing relational databases, how are they falling short in the present day era of Big data and machine learning fields. Mongo dbhas additional and advanced features different from the normal relational database which helps a user to effectively store, retrieve and manage the data. This has brought a lot of difference in the working and development of the companies. Switching from a relational database to a non-relational database can be a challenge in many ways, among which the need to study carefully all possible types of non-relational databases and finding the optimal solution for the specific application, and the creation of a non-relational database which can provide exactly the same features and querying operations as the database that will replace (Cornelia Győrödi, june 2015).

III. THE ADVENT OF MONGO DB

Mongo db is an open source database management system (DBMS) that uses a document-oriented database model which supports various forms of data. It is one of numerous non relational database technologies which arose in the mid-2000s under the NoSQL banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model. Instead of using tables and rows as in relational databases, the Mongo db architecture is made up of collections and documents (Margaret Rouse, 2018).

Mongo db is a cross-platform document-oriented database program, Classified as a NoSQL database program and Mongo db uses JSON-like documents with schema. It is nonrelational database technologies which arose under the banner for use in big data applications and other processing jobs involving data that doesn't fit well in a rigid relational model (wikipedia.org).

Mongo db Inc. (formerly **10gen**) is an American software company that develops and provides commercial support for the open source database Mongo DB, a NoSQL database that stores data in JSON-like documents with flexible schemas (wikipedia.org).

MONGO DBCONSISTS OF THE FOLLOWING :

3.1.1. NOSQL (Not Only SQL)

- The NoSQL systems were not required to follow an established relational schema.
- Large-scale web organizations such as Google and Amazon used NoSQL databases to focus on narrow operational goals

- It uses a single master architecture for data consistency, with secondary databases which maintain copies of the primary database. Operations are automatically replicated to those secondary databases for automatic failover

3.1.2. Document databases

- Document databases store semi-structured data and descriptions of that data in document format. They allow developers to create and update programs without needing to reference master schema.
- Document databases are used for content management and mobile application data handling
- JSON A record in Mongo DB is a document, which is a data structure composed of field and value pairs. Mongo DB documents are similar to JSON objects but use a variant called Binary JSON (BSON) that accommodates more data types.

3.1.3. JSON format

- JSON (Javascript Object Notation) is a text-based document used for representing simple data structures and objects in Web browser-based code. JSON was originally based on the Javascript programming language. Like XML, JSON is language-independent and may be combined with C++, Java, Python, Lisp and many other languages.
- Creating tables may seem to be easier, but with fast moving data especially in present era of Big Data , tables fall short as many fields cannot be condensed into a 2-D table's row and column relation, many fields may require multiple sets of data or many such cases where in data cannot be restricted to a particular schema.
- In Mongo DB, data is stored as documents. These are stored in Mongo DB in JSON (JavaScript Object Notation) format.
- JSON is formatted as name/value pairs. In JSON documents, fieldnames and values are separated by a colon, If you wanted to begin to model one of the rows above, for example this one:

name	quantity	size	status	tags	rating
notebook	50	8.5x11,in	A	college-ruled,perforated	8

In JSON these fields would look like:
{"name": "notebook", "qty": 50}
This is JSON format (Team).

3.2. FEATURES OF MONGO DB

- Ad hoc queries

Mongo DB supports field, range query, and regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions.

- Indexing

Fields in a Mongo DB document can be indexed with primary and secondary indices.

- Replication

Mongo DB provides high availability with replica sets.[1] A replica set consists of two or more copies of the data. Each replica set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

- Load balancing

Mongo db scales horizontally using sharding the user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more replicas.). Alternatively, the shard key can be hashed to map to a shard – enabling an even data distribution. Mongo db can run over multiple servers, balancing the load or duplicating data to keep the system up and running in case of hardware failure.

- File storage

Mongo db can be used as a file system, called Grid FS, with load balancing and data replication features over multiple machines for storing files. This function, called grid file system, is included with Mongo db drivers.

- Aggregation

Mongo db provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single-purpose aggregation methods.

Map-reduce can be used for batch processing of data and aggregation operations. But according to Mongo DB's documentation, the Aggregation Pipeline provides better performance for most aggregation operations.

The aggregation framework enables users to obtain the kind of results for which the SQL GROUP BY clause is used. Aggregation operators can be strung together to form a pipeline – analogous to Unix pipes. The aggregation framework includes the $lookup operator which can join documents from multiple documents, as well as statistical operators such as standard deviation.

- Server-side JavaScript execution

JavaScript can be used in queries, aggregation functions (such as Map Reduce), and sent directly to the database to be executed.

- Transactions

Support for multi-document ACID transactions was added to Mongo db with the General Availability of the 4.0 release in June 2018.

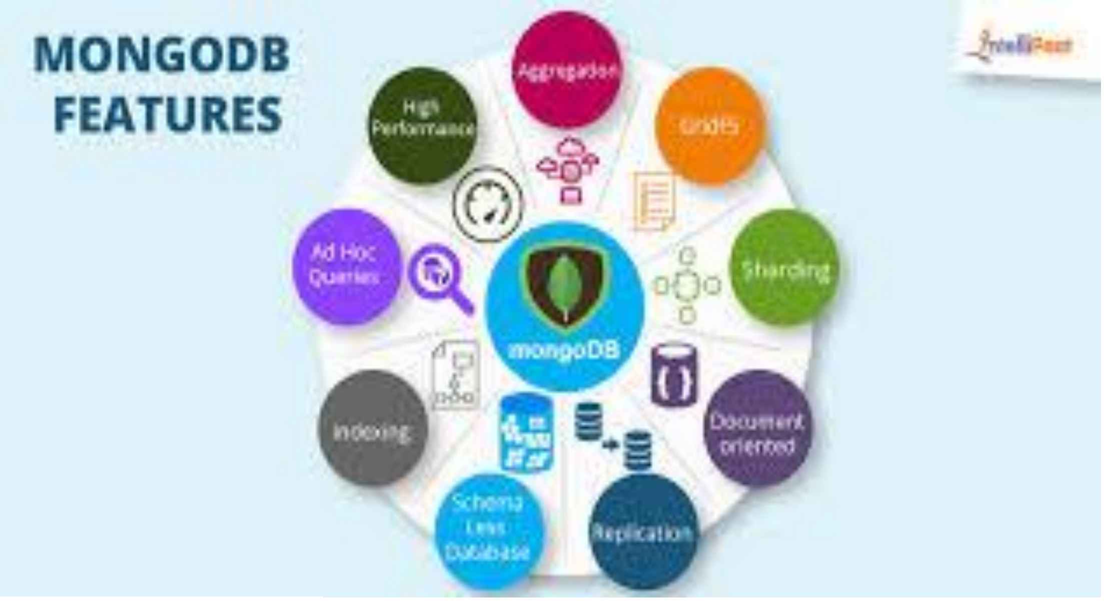

Fig 1 : features of Mongo db

IV. PROBLEM STATEMENT

3.1.4. DATA AS A SERVICE

Organizations are turning to a new approach: Data as a Service. This strategic initiative is an investment in consolidating and organizing your enterprise data in one place, then making it available to serve new and existing digital initiatives. Data as a Service becomes a system of innovation, exposing data as a cross-enterprise asset. It unlocks data from legacy systems to drive new applications and digital systems, without the need to disrupt existing backends.

3.1.5. CHALLENGE

The keys to success in the digital age are how quickly you can build innovative applications, scale them, and gain insights from the data they generate – but legacy systems hold you back.

- Data Locked in Silos: No complete view of your data? That means poor customer experience, missing insights, and slower app development
- Poor Data Accessibility: Existing systems aren't built for the modern access patterns of 24/7 customer experiences on web, mobile, and social – and they're single points of failure.
- Limited Data Support: New classes of web, mobile, social, IoT, and AI applications produce data in a volume and variety that legacy systems just can't handle.
- Cloud Blockers: Brittle legacy systems prevent the shift to cloud computing, holding developers back from on-demand access to elastically scalable compute and storage infrastructure.
- High Cost: Expensive hardware, huge jumps in costs as workloads scale, and punitive licensing impose barriers to innovation**.**

3.1.6. SOLUTION

Deliver Data as a Service within your organization to speed development, integrate data, and improve accessibility and performance. Mongo db is not only faster in most of the queries a Oracle database faces, it is also more flexible and can store large data with ease. It's flexibility and speed is the main up-votes to switch to Mongo DB. Also, as mentioned earlier one can migrate over to Mongo db from SQL based Oracle, with little difficulties.

The path to Data as a Service is to implement an Operational Data Layer (ODL). This data layer sits in front of legacy systems, enabling you to meet challenges that the existing architecture can't handle – without the difficulty and risk of a full rip and replace.

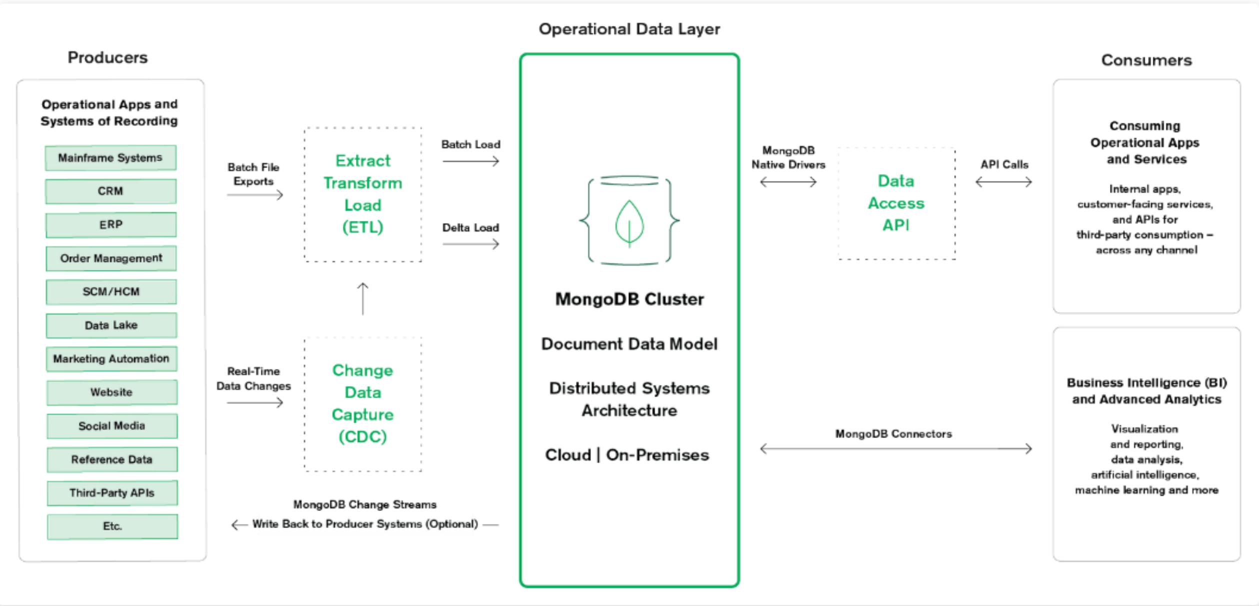
Operational Data Layer
Producers
Operational Apps and Systems of Recording
Mainframe Systems
CRM
ERP
Order Management
SCM/HCM
Data Lake
Marketing Automation
Website
Social Media
Reference Data
Third-Party APIs
Etc.
Batch File Exports
Extract Transform Load (ETL)
Batch Load
Delta Load
Real-Time Data Changes
Change Data Capture (CDC)
MongoDB Change Streams
Write Back to Producer Systems (Optional)
MongoDB Cluster
Document Data Model
Distributed Systems Architecture
Cloud | On-Premises
MongoDB Native Drivers
Data Access API
API Calls
MongoDB Connectors
Consumers
Consuming Operational Apps and Services
Internal apps, customer-facing services, and APIs for third-party consumption – across any channel
Business Intelligence (BI) and Advanced Analytics
Visualization and reporting, data analysis, artificial intelligence, machine learning and more

Fig 2 : introducing ODL layer

- Here's how Mongo db can help:

Data Layer Realization

Mongo db has developed a tried and tested approach to constructing an Operational Data Layer. The Data Layer Realization methodology helps you unlock the value of data stored in silos and legacy systems, driving rapid, iterative integration of data sources for new and consuming applications. Data Layer Realization offers the expert skills of Mongo DB's consulting engineers, but also helps develop your own in-house capabilities, building deep technical expertise and best practices.

This process for constructing an Operational Data Layer has been successfully implemented with many customers. Starting with clear definitions of project scope and identifying required producing and consuming systems is the first step to ensure success. Based on these findings, we assign data stewards for clear chains of responsibility, then begin the process of developing and deploying the Operational Data Layer with loading and merging, data access API creation, validation, and optimization. This process is iterative, repeating in order to add new access patterns and consuming apps or enrich the ODL with new data sources.

A successfully implemented ODL is a springboard for agile implementation of new business requirements.

I. Why the Mongo db Intelligent Operational Data Platform?

5.1. Mongo db is the best way to work with data

- Ease :
- Mongo DB's document model makes it simple to model – or remodel – data in a way that fits the needs of your applications
- Speed : Unifying data in Mongo db means you can write less code and get better performance when accessing data
- Flexibility : A flexible data model is essential to integrate multiple source systems to offer a unified DaaS: adapt your schema at any time, without disruption
- Versatility : Query data in any way your applications require, meeting the demands of different workloads while providing ACID guarantees to ensure data integrity

5.2. intelligently put data where you need it

- Availability: Built-in redundancy and self-healing recovery ensure service continuity of DaaS.
- Scalability : Mongo db ensures that you can scale your DaaS to store all your enterprise data and serve the most intensive workloads and demanding users
- Workload Isolation : Run operational apps while also serving analytics and BI to unlock critical insights in real time – all on a single data platform
- Data Locality : Distribute your Mongo db cluster globally for worldwide DaaS coverage and regulatory compliance

5.3. Mongo db gives you the freedom to run anywhere

- Portability : Mongo db runs the same everywhere – on-prem, on your developers' laptops, in the cloud, or as an on-demand, fully managed Database as a Service
- Global Coverage : Deploy a Mongo db cluster across the globe – or turn to Mongo db Atlas, our Database as a Service, for coverage in 50+ regions of all the major cloud providers

5.4. Mongo db enables data access and APIs

Consuming systems require powerful and secure access methods to the data in the ODL. Mongo DB's drivers provide access to a Mongo DB-based ODL from the language of your choice. Data as a Service reaches its fullest potential when you present a common Data Access API for applications; this layer can be custom built, or Mongo db Stitch can be used to expose access methods with a built-in rules engine for fine-grained security policies.

Data as a Service should also be available for analytics. The Connector for Business Intelligence allows analysts to connect to a Mongo db

ODL with their BI and visualization tools of choice, or Mongo db Charts can connect directly to the ODL for native visualization.

II. BENEFITS

i. Reduce Risk

- Avoid exposing source systems directly to new consuming applications
- Implement a system of innovation without the danger of a full “rip and replace” of legacy systems

ii. Improve Innovation

- Build new applications and digital experiences that weren’t possible before
- Make full use of your data to build unique differentiators vs. the competition
- Improve customer experience

iii. Move Faster

- Develop new applications 3-5x faster
- Iterate quickly on existing services, adding new features that would have been impossible with legacy systems
- Deliver insights that improve your competitiveness and efficiency

iv. Lower Costs

- Reduce capacity on source systems, cutting costs for licensing, MIPS, and expensive hardware (anonymous).

V. CONCLUSION

EVEN THOUGH THE RELATIONAL DATABASES CANNOT BE ENTIRELY WIPED OUT OF THE PICTURE, REPLACING IT WITH MONGO DB IN CERTAIN DATA SPECIFIC REGIONS SUCH AS ANALYSIS, QUERIES AND ANY OTHER SUCH PROCESS WOULD HELP THE ORGANIZATIONS TO FIND MANY BENEFICIAL INFORMATION AND HIDDEN PATTERNS OUT OF THEIR DATA. A SUCCESSFULLY IMPLEMENTED ODL IS A SPRINGBOARD FOR AGILE IMPLEMENTATION OF NEW BUSINESS REQUIREMENTS. THE ADDITIONAL FEATURES AND ADVANTAGES AVAILABLE IN MONGO DB HAS ALREADY PROVED ITSELF TO BE VERY USEFUL IN TERMS OF THE ORGANIZATION’S GROWTH. MANY COMPANIES HAVE ALREADY STARTED USING THIS DATABASE TO REACH THEIR ORGANIZATIONAL NEEDS.

REFERENCES

[1] (n.d.). Retrieved from wikipedia.org: https://en.wikipedia.org/wiki/MongoDB

[2] (n.d.). Retrieved from wikipedia.org : https://en.wikipedia.org/wiki/MongoDB_Inc.#History

[3] Margaret Rouse. (2018, august). Retrieved from TechTarget.com: https://searchdatamanagement.techtarget.com/definition/MongoDB

[4] anonymous. (n.d.). Retrieved from mongodb.com: https://www.mongodb.com/initiatives/data-as-a-service

[5] Cornelia Győrödi, R. G. (june 2015). A Comparative Study: MongoDB vs. MySQL.

[6] Team, M. D. (n.d.). Retrieved from www.mongodb.com: https://docs.mongodb.com/guides/server/introduction/.