

MongoDB or no MongoDB?

When do you know if this modern NoSQL database is the right database for your needs?



nagarro

Table of Contents

Abstract		3
1.	Purpose	4
2.	Background	4
3.	Use cases	5
	3.1 Do you need transactions?	
	3.2 Intuitive customer interactions	
	3.3 Processing real-time information	
	3.4 Is your company growing?	
	3.5 Security	
	3.6 Time dependent data points	
Со	nclusion	10
Re	ferences	11



ABSTRACT

MongoDB was introduced in 2007 by its founders Dwight Merriman, Eliot Horowitz, and Kevin Ryan as part of a planned Platform as a Service (PaaS) service model. It came about after the founders realized how the traditional databases struggled with scalability and agility. Over the years, MongoDB has evolved a lot and is ready for commercial production as an easy-to-use, scalable open-source NoSQL database.

Often in the limelight thanks to its massive horizontal scaling, MongoDB is seen as a much-needed flexible database structure, which is ideal for many use cases. Here's a non-exhaustive list of scenarios where it can be deemed fit for use.

1. PURPOSE

The good old days when a single database would fit all our needs, are history now. With emerging technologies, the database landscape has evolved in different ways through NoSQL and Timeseries, instead of relying on the traditional Relational Database Management Systems (RDBMS) databases. Today, selecting the right database is way more important than finding one single database that can fit every product requirement. This selection is a challenging task, as every database has its own strengths and weaknesses. A database that seems perfect for one scenario can be unfit for another scenario.

The purpose of this whitepaper is to provide some insights into one such NoSQL database used for modern applications: MongoDB. This white paper will assess MongoDB with the help of relatable use cases wherever they would be relevant to the study.

2. BACKGROUND

MongoDB was introduced in 2007 by its founders Dwight Merriman, Eliot Horowitz, and Kevin Ryan as part of a planned Platform as a Service (PaaS) service model. It came about after the founders realized how the traditional databases struggled with scalability and agility. Over the years, MongoDB has evolved a lot and is ready for commercial production as an easy-to-use, scalable open-source NoSQL database.

MongoDB Inc now markets this mature, document-based, fastest-growing, high

performing NoSQL database. The founding trio also decided to maintain MongoDB in both enterprise and open-source versions. MongoDB has also made its debut on Amazon Web Services (AWS) and Azure cloud platforms.

Often in the limelight thanks to its massive horizontal scaling, MongoDB is seen as a much-needed flexible database structure, which is ideal for many use cases. Here's a non-exhaustive list of the use cases where it can be deemed fit for use.



3. USE CASES

Choosing the correct database is an important step when developing a product. Here is a helpful breakdown that can assist you in determining whether MongoDB is the right tool for the job.

3.1. Do you need transactions?

The Transaction consistency is one of the basic challenges in NoSQL databases. Traditional databases do not guarantee consistency of stored information if concurrent transactions are performed on them. MongoDB Inc. addresses this challenge by re-focusing on WiredTiger - its storage engine - for data management. Hence, MongoDB is completely Atomicity-Concurrency-Isolation-Durability (ACID) complaint, that is, all-or-nothing, by using snapshot transaction isolation along with the storage engine.

With the above in mind, we can say that:

- MongoDB behaves like an RDBMS with the scalable and flexible features of a NoSQL database.
- With MongoDB, the ACID feature can be controlled even at the transaction level. This means that while one transaction can be ACID compliant, another can be non-ACID compliant.
- An added advantage of this feature is to protect multi-document transactions from having a performance impact for the workloads that don't require transactional consistency

This implies that such a database is a great fit at many places, like the stock market, where updating positions by aggregating hundreds of thousands of trades requires both NoSQL database and transactional consistency.

3.2. Intuitive customer interaction

Gone are the days when a customer's name and address were enough to keep them engaged with a business. The more you know your customers, the more choices you will have to keep them engaged. Personalized interactions with customers such as "Welcome back, John" is one such example. When you return to a website and see a personalized greeting or feedback based on your browsing history, it gives you an interactive experience. While browsing through any e-commerce website, it is most likely that you will see recommended products as per your browsing record and preferences.

This interaction is known as Browsing Behavior. The analysis of such behavior entails collecting metrics, which leads to a massive volume of structured, semi-structured, and unstructured data. These metrics need a database with rich analytical features and the capability to provide real-time results by interpreting such a huge volume of data. This capability is a vital cog in dealing with different business scenarios. MongoDB, with its wide array of analytical functions (like aggregation framework, MongoDB Map-Reduce, etc.) provides the perfect solution for analytical query capabilities and performance. It can easily extract information from a massive database in almost real-time, even with limited hardware.

3.3. Processing real-time information

How data can be accessed in a NoSQL database is an important consideration. Being a premier NoSQL database, MongoDB offers a very elegant way to do so. It uses a lightweight format called Binary JSON (BSON), which is a binary extension over JavaScript Object Notations (JSON), to represent data internally.

This empowers MongoDB with the following advantages while processing information:

- Ease and flexibility over JSON documents
- Speed and beneficial properties of

binary storage in the form of BSON

Besides elegant data storage, MongoDB Inc. has also designed memory utilizations optimally by:

- Incorporating data locality and in-memory caching, to enhance performance
- Capping each JSON document size at 16 MB, for efficient memory management. Hence, a document which is required by a query gets well-placed in the memory, enabling faster access.

All these factors ensure excellent read-write performances within realistic amounts of hardware.

MongoDB is a good choice for businesses in the areas of stock market, tweet analysis, satellite imagery, etc. that demand real-time processing of information and analyzing from huge and varied data streaming sources.



This database offers the following advantages to them:

- Connection managers and wellsupported out-of-the-box connectors/ drivers to connect Business Intelligence (BI) tools along with programing platforms. This ensures the best performance from the network.
- Better locking(through multi-granularity locking)allows operations to be locked at the global, database, or collection level. It also allows individual storage engines to implement their own concurrency control below the collection level.
- Profiling and baselining properties enhance MongoDB's performance even more.

3.4. Is your company growing

Scaling a database to accommodate business growth is one of the most significant challenges, which MongoDB addresses brilliantly. The database provides location-aware sharding with two sharding strategies:

• Range

• Hash

Such advantages make a database administrator's life easy in distributing data across shards, with the ease of data locality or writing query-based on clusters. MongoDB's query router exploits this location-aware sharding strategy and makes scaling completely transparent for applications. It also allows developers to write the same code for querying its appropriate shards, without compromising with the query's performance, making the developer's tasks relatively easier.

MongoDB can also deliver an up-time of 99.99% without human intervention because it is equipped with horizontal scaling along with a built-in, self-healing, and configurable replicaset. The secondary node of the replica-set automatically becomes a primary node, using a well-defined election process. When the secondary node goes down beyond a defined period, it recovers completely by using primary node logs for data consistency.

Due to all the advantages listed above, DBAs and architects often prefer MongoDB for a fastgrowing business because of its flexible scaling-out and high availability features.

3.5. Security

NoSQL databases are vulnerable to threats because of their design & implementation challenges. Hence, security remains a concern for such databases. However, MongoDB addresses this aspect too as it has been built while considering the importance of database security.

Despite a few leftover security issues such as Denial of Service (DoS), MongoDB has come a long way in its security roadmap, with countless security feature implementations. Some of these features are listed below:

- Encryption at rest that is encrypting data file using Transparent Data Encryption(TDE)
- Storage engine encryption by WiredTiger

- User authentication with Lightweight Directory Access Protocol(LDAP)
- Kerberos client authentication using access control
- Client-side field-level encryption configurations
- Transport Layer Security (TLS)/Secure Socket Layer (SSL) encryption
- Query assembling in BSON, to eliminate SQL injection threats

When it comes to security, MongoDB may not be the best solution amongst databases, but it takes the lead over many other non-relational databases.



3.6. Time-dependent data points

Some of the best characteristics like agility, scalability, and flexibility of time-dependent data points, such as the Internet of Things (IoT), also become its key challenges. IoT devices and their respective data storage require the same agility, scalability, and flexibility. Such a requirement becomes more demanding with the exponentially-increasing sensors, devices, and applications being added as data sources. The amount of data generated from these sources is voluminous and comes at relatively quickly. Such data also has diversified formats like graphs, key-value, documents, etc. RDBMS is certainly not a suitable candidate for storing such voluminous, speedy, and diversified data.

NoSQL databases are born from such scenarios and as a premier NoSQL database, MongoDB can be a good fit here with its document-based and flexible JSON data model.

The flexible data model of MongoDB ensures it can accommodate any change in the format of data. It can also store diversified data, whether it is structured, semi-structured, or unstructured. Examples: High-frequency financial services market data, sensors temperature measurements, product data, and data on e-commerce websites. Hence, MongoDB becomes the preferred pick for DBAs and architects.

CONCLUSION

To summarize, multiple factors should be considered before choosing the right database, such as:

- Analytical query performances
- Consistency, availability, and partition tolerant (CAP)
- Schema-less design
- Seamless scaling
- Security

MongoDB complies with all these factors and also has strong community support available, which helps with easier development.

However, if your requirements comprise any one or more from the following scenarios, then MongoDB may not be the most suitable solution:

- When the data needs to be relational.
- When referential integrity cannot be established between the data.
- When the data needs to be obtained from multiple documents by combining them, MongoDB can turn our to be an expensive alternative.
- When there is a lack of procedure, functions, and packages like programming construct for the business logic to be written in the database.
- When cascade Data Manipulation Language (DML) cannot be achieved, as there is no provision for triggers and trigger-based features.
- When there is a Data Warehouse (DWH) environment or large graph representations.
- When there is strict ACID transactional behavior in financial and science-related domains.

Therefore, it is recommended that while considering MongoDB, you must make an informed decision by evaluating your specific business requirements with your technical experts first.

REFERENCES

- 1. www.mongodb.com/
- 2. https://university.mongodb.com/
- 3. https://www.mongodb.com/blog/post/multi-document-transactions-in-mongodb
- 4. https://docs.mongodb.com/manual/faq/
- 5. https://docs.mongodb.com/manual/sharding/
- 6. https://docs.mongodb.com/manual/data-center-awareness/
- 7. https://www.mongodb.com/blog/post/mongodb-multi-document-acid-transactionsgeneral-availability
- 8. mongodb-architecture-guide-july-2018.pdf



AUTHOR



Sanjeeva Kumar has more than 15 years of experience and has been working with Nagarro as Architect (Database). He has expertise in many areas of database management like database security, performance tuning, capacity planning, backup strategy, database migration, and database solutioning.

MARKETING TEAM



Shailesh Dhaundiyal Editor



Deeksha Mamtani Content Marketing



Nitish Singh Designer



ABOUT NAGARRO

Nagarro drives technology-led business breakthroughs for industry leaders and challengers.When our clients want to move fast and make things, they turn to us. Working with some of our SAP clients, we continuously push the boundaries of what is possible related to innovations through process optimization and technology progress. Today, we are more than 6500 experts across 21 countries. Together, we form Nagarro, the global services division of Munich-based Allgeier SE.

/nagarro

f /nagarroinc

in /nagarro

► /nagarroinc

(O) /lifeatnagarro

©2020 Nagarro – All rights reserved

www.nagarro.com

CONNECT WITH US

Minfo@nagarro.com