Data Engineering

Cassandra Vs MongoDB In 2018

By Matan Sarig | May 1, 2018

When it comes to NoSQL databases, MongoDB and Cassandra may seem similar but are quite different. Both have similar use cases, but are not intended for transactional data (i.e., accounting systems).

Who Uses These Databases?

In the case of Cassandra vs MongoDB, both have a strong following with big names using each one.

Cassandra: Cassandra, released in 2008, has been used by many organizations including AppScale, Constant Contact, Digg, Facebook, IBM, Instagram, Spotify, Netflix, and Reddit.

MongoDB: MongoDB, released in 2009, has been used by many organizations including. Google, UPS, Facebook, Cisco, eBay, BOSH, Adobe, SAP, Forbes, and really many more. You can check the full list here: https://www.mongodb.com/who-uses-mongodb.

What About Database Structure?

Cassandra: One of Cassandra's biggest strengths is being able to handle massive amounts of unstructured data. In cases where your database needs to rapidly scale with minimal increase of administrative work, Cassandra may be a good choice.

How big can it scale? Cassandra can handle the load of applications like Instagram that have roughly 80 million photos uploaded to the database every day.

Cassandra uses wide column stores which utilize rows and columns but allows the name and format of those columns to change. It uses a blend of a tabular and key-value. Unlike a typical relational database management system (RDBMS), tables can be created, altered, and dropped while the database is running and processing queries.

Column families are similar to tables in RDBMS and contain rows and columns, with each row having a unique key. Unlike a traditional RDBMS, all rows in a table are not forced to have the same columns. These columns can also be added on the fly and are accessed using the Cassandra Query Language (CQL). While CQL is similar to SQL in syntax, Cassandra is non-relational, so it has different ways of storing and retrieving data.

MongoDB: MongoDB uses JSON-like documents that can have varied structures. It uses the MongoDB query language to allow access to the stored data. Since it is schema-free, you can create documents without having to create the structure for the document first.

Database hierarchy:

A useful comparison with Relational database management systems (RDBMS) in which you have: `Table | Column | Value | Records`. In comparison, in MongoDB, you have: `Collection | Key | Value | Document`. This means that collections in MongoDB are like tables in RDBMS.

Documents are like records in an RDBMS. Documents can easily be modified by adding or deleting fields without having to restructure the entire document.

Are Indexes Needed?

Cassandra: In Cassandra, multiple secondary indexes are not fully supported; you can only query using the primary key.

MongoDB: Indexes are preferred in MongoDB. If an index is missing, every document within the collection must be searched to select the documents that were requested in the query. This can slow down read times.

How Are Their Queries Different?

Selecting records from the customer table:

Cassandra: `‘SELECT * FROM customer;’`

MongoDB: `‘db.customer.find()’`

Inserting records into the customer table:

Cassandra: `‘INSERT INTO customer (custid, branch, status) VALUES('appl01', 'main', 'A');’`

MongoDB: `‘db.customer.insert({ cust_id: 'appl01', branch: 'main', status: 'A' })’`

Updating records in the customer table:

Cassandra: `‘UPDATE Customer SET branch = ‘main' WHERE custage > 2;’`

MongoDB: `‘db.customer.update( { custage: { $gt: 2 } }, { $set: { branch: 'main' } }, { multi: true } )’`

Where (And How) Are These Databases Deployed?

Cassandra: Cassandra was written in Java. It can be deployed on BSD, Linux, OS X, and Windows.
MongoDB: MongoDB was written in C++, and has support for the following programming languages: Actionscript, C, C#, C++, Clojure, ColdFusion, D, Dart, Delphi, Erlang, Go, Groovy, Haskell, Java, JavaScript, Lisp, Lua, MatLab, Perl, PHP, PowerShell, Prolog, Python, R, Ruby, Scala, Smalltalk.

What Types Of Replication / Clustering Are Available?

Cassandra: Cassandra does replication out-of-the-box. You tell it the number of nodes it should copy your data to and it takes care of the rest of the process.

Cassandra allows for multiple masters where losing a single node still lets you write to the cluster. This can allow for better fault tolerance without the 10 to 40 second downtime required with MongoDB.

MongoDB: MongoDB has built-in replication with auto-elections. This allows you to set up a secondary database that can be auto-elected if the primary database becomes unavailable. However, MongoDB requires some setup (and maybe some help from support) to do replication. MongoDB has replica sets where one member is the primary and all others have a secondary role. The reads and writes are committed to the primary replica first and then replicated to the secondary replicas.

MongoDB has a single master. While the auto-elect process happens automatically, it can take 10 to 40 seconds for it to occur. While this is happening, you can not write to the replica set.

Who's Currently Behind The Databases?

Cassandra: Avinash Lakshman and Prashant Malik developed Cassandra at Facebook for the Facebook inbox search feature. Facebook released Cassandra in July 2008 as an open source project. The original developers got the name for the project from Cassandra, a Trojan mythological prophet. The Apache Software Foundation is currently behind the database.

MongoDB: MongoDB was started in 2007 by 10gen, which created the product based on the word "humongous". In 2009, it was released, and 10gen later changed their company name to MongoDB, Inc. MongoDB, Inc. provides development of the software and sells their enterprise solution.

Who Maintains The Project?

Cassandra: Apache Software Foundation maintains the project.

MongoDB: MongoDB, Inc. maintains the project.

Who Provides Support?

Cassandra: Support for Cassandra comes from third-party companies like Datastax, URImagination, Impetus, and more. A complete list of Cassandra DB third-party support providers can be found at https://wiki.apache.org/cassandra/ThirdPartySupport.

MongoDB: MongoDB has an enterprise-grade support that provides 24 x 7 support along with the option for an extended lifecycle support. Extended lifecycle support allows you to continue using older versions and upgrade when you want. Getting support from MongoDB gives you unlimited access to security fixes and updates.

Who Maintains The Documentation?

Cassandra: The Apache Software Foundation maintains the Cassandra documentation and can be found at http://cassandra.apache.org/doc/latest/. While there, you can learn how to get started with Cassandra, the Cassandra Query Language, Tools, FAQS, and more. Datastax also maintains documentation at http://docs.datastax.com/en/landing_page/doc/landing_page/current.html.

MongoDB: MongoDB, Inc. maintains the MongoDB documentation, and it can be found at https://docs.MongoDB.com/. From there, you can find information about the MongoDB Server, Atlas (database as a service), cloud manager for hosted MongoDB, and Ops Manager.

Very useful community sites are the omnipresent StackOverflow and a bit more database-specific StackExchange for Databases.

Is There An Active Community?

Cassandra: Apache Software Foundation offers a community site with a mailing list, IRC, along with links to books and publications. This information can be found at http://cassandra.apache.org/community/.

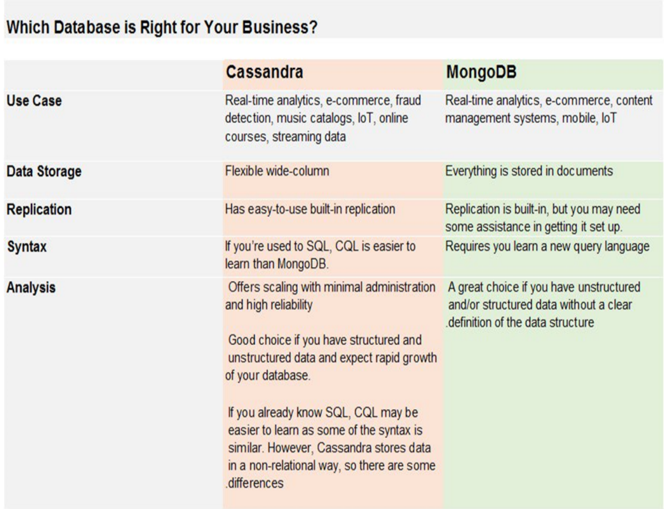
MongoDB: The MongoDB community offers information about webinars, events, user groups, and the MongoDB University.

Which Database Is Right For Your Business?

Cassandra: One of Cassandra's greatest strengths is its ability to scale while still being reliable. It is possible to deploy Cassandra across multiple servers built-in without a lot of extra work. Part of this is because Cassandra handles replication with minimal configuration, making it easy to set up.

If you need a database that is easy to setup and maintain regardless of how much your database grows, Cassandra can be a good option. If you work in an industry where you need rapid growth of your database, Cassandra offers easier rapid growth than MongoDB.

MongoDB: MongoDB can be a great choice if you need scalability and caching for real-time analytics; however, it is not built for transactional data (accounting systems, etc.). MongoDB is frequently used for mobile apps, content management, real-time analytics, and applications involving the Internet of Things. If you have a situation where you have no clear schema definition, MongoDB can be a good choice.

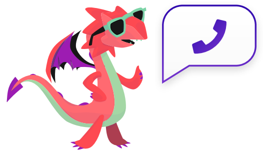


Which Database is Right for Your Business?

	Cassandra	MongoDB
Use Case	Real-time analytics, e-commerce, fraud detection, music catalogs, IoT, online courses, streaming data	Real-time analytics, e-commerce, content management systems, mobile, IoT
Data Storage	Flexible wide-column	Everything is stored in documents
Replication	Has easy-to-use built-in replication	Replication is built-in, but you may need some assistance in getting it set up.
Syntax	If you're used to SQL, CQL is easier to learn than MongoDB.	Requires you learn a new query language
Analysis	Offers scaling with minimal administration and high reliability Good choice if you have structured and unstructured data and expect rapid growth of your database. If you already know SQL, CQL may be easier to learn as some of the syntax is similar. However, Cassandra stores data in a non-relational way, so there are some .differences	A great choice if you have unstructured and/or structured data without a clear .definition of the data structure

If you have a situation where you are de-normalizing your database schema, MongoDB documents can be used to store the unstructured data in a way that is easier to update. In a situation where the write load is high, MongoDB can be a good choice. It offers a high insert rate.

Cassandra or MongoDB: whichever you pick for your organization, Panoply, smart data warehouse, offers a single data management solution that can connect Cassandra, MongoDB, cloud, and more without coding.



Get a free consultation with a data architect to see how to build a data warehouse in minutes.

Request Demo