MongoDB 5.0 and Rapid Releases: Guide to What's New

March 2022

MongoDB.

Table of Contents

Introduction

MongoDB 5.0 and the 5.x Rapid Releases make it even easier for you to support a broader range of workloads, introducing new ways of future-proofing your applications, and further enhancing privacy and security. Key highlights include:

- **Native time series** collections, clustered indexing, and window functions make it easier, faster, and lower cost to build and run applications like IoT and financial analytics, and to enrich your enterprise data with time series measurements.
- **Live Resharding** allows you to change the shard key for your collections on-demand as your workloads grow and evolve — with no database downtime or complex migrations.
- The **Stable API** future-proofs your applications. Starting with MongoDB 5.0, you can consume the latest features at your own pace and won't need to change your application to accommodate any new behaviors introduced in the later versions.
- MongoDB's unique **Client-Side Field Level Encryption** now extends some of the industry's strongest data privacy controls to multi-cloud databases.

Highlights from the 5.1, 5.2, and 5.3 Rapid Releases include:

- **Time series enhancements** including support for sharding, data densification, gap filling, columnar compression, & delete operations. Time series collections deployed in Atlas can also automatically archive data to Atlas Online Archive.
- **Richer, more flexible analytics** with cross-shard support for $lookup (joins) and $graphlookup (graph traversal). Aggregations that write the results of the pipeline to a collection (e.g., updating a view) can now execute on secondaries and then direct the output to the primary, enabling more comprehensive workload isolation for analytics queries.
- **New operators** including accumulators to compute items based on their place in a dataset and $sortArray to sort the items in an array.
- **Improved C# experience** with a redesigned LINQ

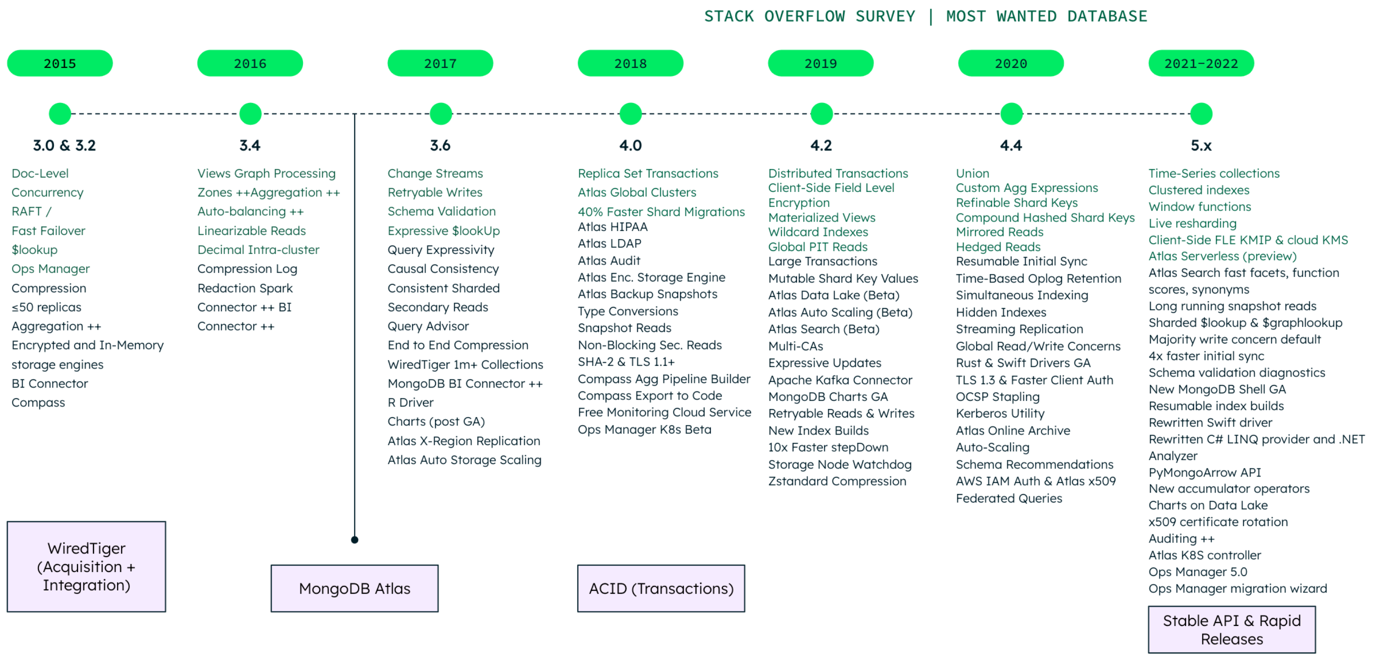

Figure 1: MongoDB Version History: Expanding the feature set with every release

interface and a MongoDB Analyzer for .NET.

- Improved resilience with **faster initial sync via file copy**, speeding up the process of adding new replica set members by as much as 4x.
- MongoDB's unique **Client-Side Field Level Encryption** will now work with any KMIP-compliant key management solution.

We will be covering these new capabilities and more in this guide.

MongoDB 5.0 is Generally Available (GA) now, and ready for production deployments. There are two ways to get started:

1. Run 5.0 as a fully-managed service in the cloud with MongoDB Atlas, including in our new Atlas Serverless tier. The 5.3 Rapid Release is also available in MongoDB Atlas.
2. Download and run MongoDB 5.0 on your own infrastructure, either with the community edition of MongoDB or with MongoDB Enterprise Advanced, which offers sophisticated operational tooling via Ops Manager, advanced security controls, proactive 24x7 support, and more. 5.3 and other Rapid Releases are available to download for evaluation purposes.

"As developers continually look to improve their velocity, they are increasingly looking for versatile platforms that allow them to offload operational responsibilities. Whether it's adding native time series capability or introducing live resharding, MongoDB is striving to offer developers what they want."

Stephen O'Grady, Principal Analyst with RedMonk

New Release Cadence for the MongoDB Database

Delivering on our announcement from October 2020, MongoDB 5.0 marks a new era in our release cadence – designed to get new features and improvements into your hands faster.

What **Doesn't Change**: Major Releases Annually

Major Releases of the MongoDB database will still be published once per annum — exactly as they have been prior to this new cadence. MongoDB 5.0 ships in 2021, followed by 6.0 scheduled in 2022, 7.0 in 2023, and so on.

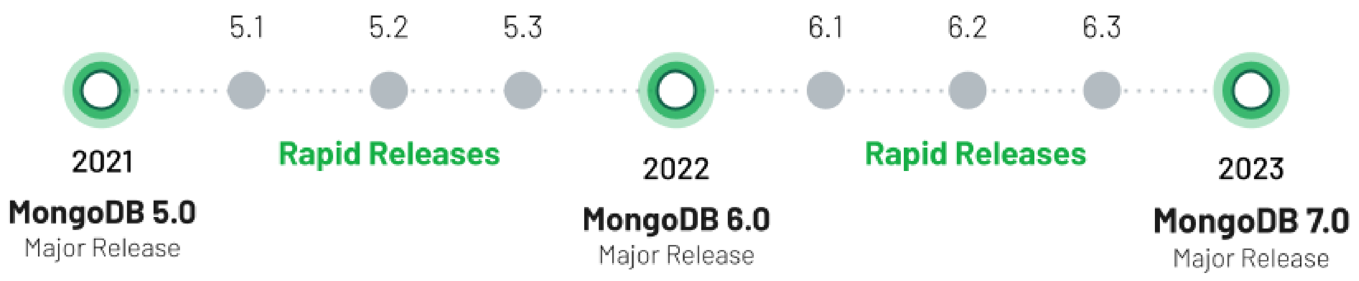

Figure 2: New MongoDB Release Cadence: annual Major Releases and quarterly Rapid Releases

These Major Releases will be available to the community and supported in MongoDB Atlas and with MongoDB Enterprise Advanced, where they will be certified with Ops Manager. Consistent with our existing support policy, patch releases will be published for each Major Release throughout its product lifecycle.

Maintaining our established annual release cadence with Major Releases will ensure there is no impact to MongoDB users and customers who don't wish to (or who aren't able to) upgrade often.

What **Does Change**: Rapid Releases Every Quarter

We will no longer be holding back all of our new features and enhancements for a single Major Release once per year. Starting from MongoDB 5.0, we will be publishing new Rapid Releases every quarter (and even faster in the future).

As shown in Figure 2:

- One quarter after the GA of the MongoDB 5.0 Major Release, we shipped our first MongoDB 5.1 Rapid Release.
- The MongoDB 5.2 Rapid Release shipped one quarter later, followed by 5.3 one quarter after that.
- All of the features and enhancements from those Rapid Releases — as well as the three months of additional development work following 5.3 — will roll up into the MongoDB 6.0 Major Release in 2022.

Unlike Major Releases, Rapid Releases will be supported **only** in MongoDB Atlas. Rapid Releases **will be available** as development builds from the MongoDB Download Center for evaluation but will not be supported outside of Atlas or certified with Ops Manager.

Focusing the qualification and support of new Rapid Releases to the MongoDB Atlas builds enables our engineering organization to ship even higher quality software faster. Otherwise we would have to qualify each release on more than 20 different supported MongoDB platforms every quarter, which would significantly reduce the rate of feature delivery to you. It would also mean users incurring the operational overhead of upgrading each quarter to stay on the current release, which is something we can simply automate for them in MongoDB Atlas.

Review the MongoDB Support Policy for more details on the support lifecycle and coverage for each MongoDB version.

Stable API: Consuming New Releases While Future-Proofing your Applications

While a shorter release cycle gets new features and improvements into your hands faster, we also recognize that upgrading to any new version of a database can sometimes be a long and complex process.

With MongoDB 5.0, we introduced the Stable API which defines a set of commonly used commands and parameters that **will not change** between database releases — whether those are annual Major Releases or quarterly Rapid Releases. You can simply pin your driver to a specific release of the Stable API with the confidence that your application code will not need to change to accommodate new database behavior, even as the database is upgraded and improved beneath it. This allows you to better **future-proof your apps**, making your application lifecycle less tied to the database lifecycle and providing you with a level of investment protection that is ahead of other databases.

The Stable API also gives MongoDB engineers the flexibility to add new features and improvements to the database with each release, but only in ways that are backwards compatible with earlier versions. When changes do need to be made in the future, then we will cut a new version of the API that will run concurrently with the existing Stable API on the same server. This will enable you to upgrade to the newer API in the future, but only when it makes sense for your application.

With the acceleration of MongoDB releases, the Stable API enables you to easily take advantage of new database features faster. You can learn more from our Stable API documentation.

Native Time Series Platform

With the 5.0 release, MongoDB expands its general purpose application data platform to enable developers to work more easily with time series data, harnessing the innovation within IoT, financial analytics, logistics, and more. MongoDB 5.0 makes it faster and lower cost to build and run time series applications by natively supporting the entire time series data lifecycle – from ingestion, storage, querying, real-time analysis, and visualization through to online archiving* or automated expiration as data ages.

MongoDB's new time series collections automatically store time series data in a highly optimized and compressed format, reducing your storage footprint as well as I/O to achieve greater performance and scale. These optimizations eliminate lengthy development cycles, enabling you to quickly build a schema tuned for the unique performance and analytical demands of time-series applications. All you need do to create a collection that will optimize time series data storage is run the following command:

```
db.createCollection("collection_name",
{ timeseries: { timeField: "timestamp" } } )
```

Even with an optimized schema, you retain the flexibility to model metadata and the time-based level of granularity based on your application's needs and how frequently data is arriving. As your application requirements evolve, you can adapt your schema without friction, just as you have always been able to with MongoDB's flexible document data model.

You can ingest vast volumes of streaming time series data, while MongoDB seamlessly adjusts ingestion rates and automatically handles out-of-order measurements based on dynamically generated time partitions for you. To give you a choice of options for data ingestion, the latest release of the MongoDB Connector for Apache Kafka adds native support for time series. You can automatically create time series collections directly from Kafka topic messages. This enables you to collect data at the edge, process and aggregate it as needed before persisting it into time series collections in MongoDB.

Time series collections can take advantage of MongoDB's native sharding to horizontally distribute massive data sets and co-locate nodes with data producers. This allows users to easily support local write operations and adhere to data sovereignty regulations.

Time series collections automatically create a clustered index against data ordered by time, so that you can query data with low latency. We have also expanded the MongoDB Query API with Window Functions, allowing you to run analytical queries like moving averages and cumulative sums to uncover hidden patterns.

In relational database systems, these are often referred to as SQL analytical functions and support windows defined in terms of rows — i.e., a three-row moving average. MongoDB takes this a step further by also adding powerful time series functions like exponential moving average, derivative and integral, allowing you to define windows in terms of time, for example a 15-minute moving average. Window functions can be used to query both time series and regular collections in MongoDB, opening up new classes of analytics use cases across multiple application types.

New temporal operators, including $dateAdd, $dateSubstract, $dateDiff, and $dateTrunc allow you to aggregate and query data across custom time intervals.

Review the New Aggregation Operators section of the 5.0 release notes to learn more.

With time series data, it's not uncommon to have uneven data or missing readings (e.g. a sensor going offline), which can make working with the data difficult. Time series collections will support data densification and gap filling, allowing teams to better handle missing data so that they can more easily build time series applications and run analytics on time series data.

Time series collections also support delete operations. While most time series applications are append-only, in some cases end users may need to be able to invoke their right to erasure. MongoDB's time series collections give developers an easy way to comply with modern data privacy regulations.

When running MongoDB 5.0+ in Atlas, you can use MongoDB Charts to visualize your time series data in real time. You can economically retain your time series data for historical or compliance purposes by automatically tiering aged data out to low cost object storage while preserving the ability to query it at any time.

Of course time series data does not exist in isolation, so we make it easy for you to combine it with the full complement of your enterprise data. Time series collections can sit right alongside regular MongoDB collections within a single database, making it straightforward to extract valuable insights and create intelligent applications. For example:

- A manufacturer can identify issues in the production line by capturing and analyzing sensor data, then combine it with their orders collections to identify customers that will be affected by a downstream inventory shortage.
- A retailer analyzing weather data can combine it with sales data to understand seasonality patterns.

You don't have to choose a specialized time series database that can't serve any other application types and that requires complex integrations to blend time series with enterprise data. MongoDB eliminates the costs and complexities of integrating and running multiple disparate databases by providing a single platform that lets you build performant and efficient time series applications while also powering any modern use case or workload.

Review the MongoDB time series web page for all of the latest resources for working with time series data in the MongoDB application data platform.

Live Resharding: Scale without Fear or Friction

Live Resharding provides you with friction-free scaling, allowing you to change the shard key for your collection on-demand as your application evolves with no database downtime.

Sharding enables you to scale out your database across multiple nodes: you do that to accommodate growth in data or load, or for data locality — for example pinning data onto a set of shards in a specific region for low latency local access and for data sovereignty.

MongoDB sharding has always been highly flexible — you can select any field or combination of fields in your documents to shard on. The choice of shard key defines the distribution of data across the available shards. Ideally you want to select a shard key that gives you low latency and high throughput reads and writes, while also evenly distributing your data across the cluster so you get linear scalability as you add more shards.

Choosing the shard key is an important decision that used to be difficult to change. This was because resharding was a manual and complex process that could only be achieved through one of two approaches:

1. Dumping the entire collection and then reloading it into a new collection with the new shard key. This is an offline process, and so your application is down until reloading is complete — for example, it could take several days to dump and reload a 10 TB+ collection on a three-shard cluster.
2. Undergoing a custom migration that involved writing all the data from the old cluster to a new cluster with the resharded collection. You had to write the query routing and migration logic, and then constantly check the migration progress to ensure all data had been

successfully migrated. Custom migrations were highly complex, labor-intensive, risky, and expensive. It took one MongoDB user three months to complete the live migration of 10 billion documents.

The complexity, time, and effort of resharding applies to almost every distributed database. With live resharding in MongoDB, this all goes away.

Now you just run the reshard command from the shell, pointing at the database and collection you want to reshard, specify the new shard key and let MongoDB take care of the rest.

```
reshardCollection: "<database>.<collection>",
key: <shardkey>
```

When you invoke the reshard command, MongoDB clones your existing collection, then starts applying all new oplog updates from the existing collection to the new collection. This enables the database to keep pace with incoming application writes. When all oplog updates have been applied, MongoDB will automatically cut-over to the new collection and remove the old collection in the background.

With Live Resharding now you can:

- **Evolve with your apps with simplicity**: As your applications evolve or as you need to improve on the original choice of shard key, a single command kicks off resharding. It is also efficient. Rather than simply rebalancing data, it copies and rewrites all of the current collection data to the new collection in the background, while also keeping pace with incoming application writes.
- **Compress weeks/months to minutes/hours**: Live resharding is fully automated, so you eliminate disruptive and lengthy manual data migrations. To make scaling-out even easier, now you can evaluate the effectiveness of different shard keys in dev/test environments before committing your choice to production. Even then, you can change your shard key when you want to.
- **Extend flexibility and agility across every layer of your application stack**: You have seen how MongoDB's flexible document data model instantly adapts as you add new features to your app. Now you get that same flexibility when you shard. New features or new requirements? Simply reshard as and when you need to.

Richer and More Flexible Analytics

Many developers start out with MongoDB for their operational use cases, and then expand to leverage the platform's versatility in powering analytics. The 5.1 Rapid Release includes new features and enhancements that make it easier to unlock insights from your data.

Cross-shard joins and graph traversals

For most transactional and operational workloads, the document data model largely eliminates the need to join data from different collections. This is because related data can be embedded in sub-documents and arrays within a single, richly structured document – following the principle that what is accessed together is often best stored together.

However analytical applications can sometimes require joins to be executed – for example bringing together customers and orders from separate collections. Through the $lookup aggregation pipeline stage, you can have the database join collections for you. The $graphlookup stage gives you the ability to traverse related data, performing “friend-of-friend” type queries to uncover patterns and surface previously unidentified connections in your data.

MongoDB 5.1 allows you to use $lookup and $graphlookup to combine and analyze data that is distributed across shards which was not previously possible. Our design gives you even more precision in your code by enabling you to target individual shards as needed. However you don't need to understand sharding or even know your collection is sharded to run these queries as there is no new syntax for developers to learn.

Materializing results for operational analytics

The $merge and $out aggregation stages can be used to write the results of an aggregation pipeline in order to create a new collection or create/update an on-demand materialized view. These stages enable users to reduce processing overhead by reading pre-computed results instead of re-running the aggregation each time, and by writing only incremental results when the aggregation results change.

Users often want to run resource-intensive analytical queries on secondary nodes in order to avoid performance impacts on the primary — but since only primaries can serve writes, aggregations including $out or $merge could not previously run on a secondary node.

Many 5.1 compatible drivers now will be able to run such pipelines successfully, performing their query execution work on a secondary node, then automatically directing any writes to the primary. This allows you to offload computationally expensive analytics work to secondary nodes while still being able to materialize the results of that work. This will be accessible via drivers in their respective releases.

Next Generation Privacy and Security

MongoDB's unique Client-Side Field Level Encryption (FLE) now brings some of the strongest data privacy controls to multi-cloud clusters. Backed by online audit configuration and certification rotation, MongoDB 5.0 helps you maintain a strict security posture with no interruption to your applications, and with the freedom to run your applications anywhere.

Client-Side FLE was initially released with integration to Amazon's Key Management Service (KMS). With MongoDB 5.0 and the 5.1 Rapid Release, native support is now Generally Available for Azure Key Vault and Google Cloud KMS. Client-Side FLE will also work with any KMIP-compliant key management solution. Providing additional flexibility and ease-of-use, MongoDB drivers can now authenticate to Amazon's KMS using temporary credentials.

These enhancements not only provide first class support for the key management services available in each major cloud provider, they also mean you can take advantage of multiple layers of encryption that use keys from different clouds. For example, you can have PII data encrypted client-side with AWS KMS keys, then stored in both an AWS and Google Cloud region on MongoDB Atlas and further encrypted at-rest with a key managed via Azure Key Vault.

Client-Side FLE was introduced in 2019. It uses the MongoDB drivers to encrypt the most sensitive fields in your documents **before** they leave the application. This enables you to do three things that you can't do with in-flight or at-rest encryption alone:

- Protect data while it is in-use, in the memory of your active database instance. The database never sees plaintext, but data is still queryable.
- Make data unreadable to anyone running the database for you, or who has access to the underlying database infrastructure — this includes MongoDB SREs running the Atlas services as well as cloud provider personnel.
- Simplify the process of enforcing right to erasure (sometimes called right to be forgotten) mandates in modern privacy regulations such as the GDPR or the CCPA. This is because you simply destroy the key encrypting a user's PII, and their data is rendered unreadable and unrecoverable — in-memory, at-rest, in backups, and in logs.

Certificate Rotation and Auditing

Also new in MongoDB 5.0 is the ability to rotate x509 certificates without restarting the database cluster. The audit log has been enhanced to capture more system event audit messages, providing a more holistic view of activities against your database. You can now also reconfigure audit log filters as an online operation. These filters allow you to control the specific events and user activity you are capturing in the audit log, allowing you to quickly respond to new business or regulatory demands.

Cloud Services and On-Premise Operational Automation

MongoDB 5.0 includes new options and tooling to support you running the database wherever you need it for your business.

MongoDB Atlas Serverless Instance (Preview)

We want developers to be able to build MongoDB applications without having to think about database infrastructure or capacity management. With serverless instances on MongoDB Atlas, now available in preview, you can automatically get the database resources you need based on your workload demand, and pay for only what you use.

It's really simple: the only decision you need to make is the cloud region hosting your data. After that, you'll get an on-demand database endpoint that dynamically adapts to your application traffic. Serverless instances will support the latest MongoDB 5.0 GA release, Stable API, and upcoming Rapid Releases so you never have to worry about backwards compatibility or upgrades. You pay only for reads and writes your application performs and the storage resources you use (up to 1TB of storage in preview) and leave capacity management to MongoDB Atlas's best-in-class automation. The preview release is just the beginning – we will be working with partners such as Vercel and Netlify to deliver an integrated serverless development experience in the coming months.

Ops Manager 5.0

Ops Manager is the management platform that makes it easy to deploy, monitor, back up, and scale MongoDB on your own infrastructure. Ops Manager has been enhanced for MongoDB 5.0, with highlights including:

- Support for the automation, monitoring, and backup/restore of MongoDB 5.0 deployments.
- Improved load performance with parallelized client-side restores.
- A quick start experience for deploying MongoDB in Kubernetes with Ops Manager.
- And lastly, a guided Atlas migration experience that walks users through provisioning a migration host to push data from their existing environment into the fully managed cloud service.

Other MongoDB 5.x Highlights: Developer Productivity

New MongoDB Shell: GA

The new MongoDB Shell has been redesigned from the ground up to provide a modern command-line experience with enhanced usability features and a powerful scripting environment. It makes it even easier for users to interact and manage their MongoDB data platform, from running simple queries to scripting admin operations.

A great user experience, even in a command-line tool, should always be a major consideration. With the new MongoDB Shell we have introduced syntax highlighting, intelligent auto-complete, contextual help, and useful error messages creating an intuitive, interactive experience for MongoDB users.

Data Science with PyMongoArrow

Run complex analytics and machine learning on your data in MongoDB using Python with the release of our new PyMongoArrow API, available in beta. PyMongoArrow offers fast and easy conversion of MongoDB query results to popular data formats such as Pandas data frames and numPy arrays, helping you streamline your data science workflows.

Improved Schema Validation

Schema validation is a powerful way of applying data governance controls to your MongoDB schema. In MongoDB 5.0 schema validation is now easier and more user-friendly with descriptive error messages generated whenever an operation fails validation.

You gain a deeper insight into which parts of a document failed to validate against which parts of a collection's validator, and why. Then you can quickly identify and remediate code errors that are throwing off your validation rules without having to slice documents into pieces to isolate the problem.

Read the Improved Error Messages in Schema Validation blog post, which showcases the addition of new detailed error messages.

Long-Running Snapshot Queries

Long-running snapshot queries increase the versatility and resilience of your applications. With this feature — now available in preview — you can run queries that take up to five minutes by default (or tune them for a custom duration*) while maintaining consistent snapshot isolation against a live, transactional database. You can route snapshot queries to secondary nodes to separate different workloads within a single physical cluster, and scale them across shards for performant reads over large data sets

Long-running snapshot queries simplify your data architecture. They eliminate the need to create a static copy of your data and move it out into a separate silo to power reconciliations and other complex algorithms, and to isolate these queries from interfering with your live, operational workloads.

MongoDB enables long-running snapshot queries through a project in our underlying storage engine called durable history, implemented back in MongoDB 4.4. The durable history will store snapshots for all field values that have changed since the query started. By using this durable history, the query can maintain snapshot isolation, even as data is changing in the database. Durable history also helps lower storage engine cache pressure so your applications can achieve higher query throughput for heavy write workloads.

- Note that tuning for a custom duration is initially available in Atlas through a cloud support specialist only; self-serve tuning will be available after MongoDB 5.0 GA.

C# Enhancements

The LINQ provider for the MongoDB C# driver has been rewritten from the ground up to provide higher performance, unlocking additional aggregation pipeline features, and expanding GSSAPI authentication support to Linux. It also offers backwards compatibility, so you can continue using the driver version your application is running on until you are ready to consume the latest features.

Additionally, we've released the MongoDB Analyzer for .NET, which enables C# developers to more easily troubleshoot queries and aggregations, and prevent errors from cropping up at runtime. The MongoDB Analyzer builds on earlier releases of the MongoDB .NET driver and makes it easier and faster for developers to use MongoDB with C#.

Swift Vapor + async/await API

Vapor, the leading framework for Swift web applications, is now officially integrated with the MongoDB Swift driver. The Swift Vapor integration provides a smooth user experience for developers with full access to MongoDB features and functions. Additionally, the driver and Vapor integration now support using Swift's brand new async/await feature to easily maintain and build more robust apps.

New accumulators & expression to sort arrays

MongoDB 5.2 brings new operators that streamline your queries. The $top and $bottom operators allow you to compute the top and bottom elements of a data set and return related fields within the same query without complex logic. We are also introducing $maxN, $minN, and accumulators such as $firstN, $lastN, which return elements while taking into account the current order of documents in a dataset.

A highly requested feature, the new $sortArray expression allows you to sort the elements in an array directly in your aggregation pipeline in an intuitive, optimized way. The input array can be as simple as an array of scalars or as complex as an array of documents with embedded subdocuments.

Atlas Search

Atlas Search, which delivers powerful full-text search functionality without the need for a separate search engine, has several new capabilities for building rich end user experiences:

- Function Scoring allows you to apply mathematical formulas on fields within documents to influence their relevance, such as popularity or distance — e.g. closer restaurants with more or better reviews will show up higher in a list of search results.
- You can now define collections of synonyms for a particular search index. By associating semantically equivalent terms with each other, you can respond to a

wider range of user-initiated queries in your applications.

- Faceted search allows users to filter and quickly navigate search results by categories and see the total number of results per category for at-a-glance statistics. Atlas Search will now enable lightning fast facets and counts over large data sets. With the new facet operator, facet and count operations are pushed down into Atlas Search's embedded Lucene index and processed locally – taking advantage of 20+ years of Lucene optimizations – before returning the faceted result set back to the application. Facet-heavy workloads such as ecommerce product catalogs, content libraries, and counts run up to 100x faster.

Both function scoring and synonyms are supported against any version of the MongoDB database, from 4.2 and up. Facets in Atlas Search are supported against any version of the MongoDB database, from 4.4 and up.

Other MongoDB 5.x Highlights: Platform Resilience

Default Majority Write Concern

Starting with MongoDB 5.0, the database's default write concern has been elevated to the majority concern (w:majority). Operations will commit only when they have been applied to the primary node and persisted to the journals of a majority of replicas, providing stronger durability guarantees "out of the box".

With majority writes, data can survive both elections and complete node failures. Through replication performance enhancements delivered in recent MongoDB releases, including the introduction of streaming replication and replicate-before-journaling, the latency impact of majority writes is low.

Write concerns are fully tunable, so you can maintain the earlier primary-acknowledged default or configure custom write concerns that balance the performance with the durability requirements of your application. Read the default majority write concern blog post to learn more about upgrade and performance considerations.

Resumable Index Builds

Reducing the impact of planned maintenance to your users, in-flight index builds will now automatically resume from where they left off after a node restarts. Now you can decouple long-running index builds over large collections from scheduled operational processes such as patching or upgrading nodes in your database cluster.

Connection Management

MongoDB 5.0 improves the resilience of your deployments by helping prevent the occurrence of connection storms at the driver level.

While there is no single root case of connection storms, they tend to be self-propelling and often occur once the deployment is already compromised. So we have taken a multi-faceted approach to implementing mitigations to reduce situations where runaway connection creation can occur:

- The drivers will limit the number of connections they attempt at any given time, preventing overloading the database server in a simple and effective way.
- The drivers also decrease the frequency of checks while monitoring the connection pool, giving unresponsive or overloaded server nodes a chance to breathe and resume.
- Instead of choosing randomly from available servers, the drivers route workloads to faster servers with the healthiest connection pools.

These features, in addition to past improvements at the mongos query router layer, help create more robust and scalable server architectures that can withstand extremely high application loads. They also provide a solid foundation for additional resilience initiatives that will further enhance connection management in upcoming releases.

Performance & Operational Overhead Improvements

A number of changes have been made to the WiredTiger internals that improve backups, including minimizing the checkpoints pinned while a backup cursor is open and improving handling of backup cursors that are open for long periods. These improvements will reduce both the operational overhead and storage consumption on the replica node from which the backup is taken. This improvement is available for backups taken from MongoDB Atlas and from self-hosted deployments controlled by Ops Manager or Cloud Manager, and has been backported to MongoDB 4.2 and above.

In addition to enhancements affecting backups, WiredTiger checkpointing and locking have been improved to enhance performance when MongoDB is managing many concurrently active collections in a single instance. This is especially useful to multi-tenant applications built on MongoDB.

Improving Resilience with Faster Initial Sync via File Copy

Initial sync is how a replica set member in MongoDB loads a full copy of data from an existing member. This process occurs when users are adding new nodes to replica sets. Initial sync is also occasionally used to recover replica set members that have fallen too far behind the other members in a cluster. Prior to 5.2, this process was done with a logical initial sync, in which every collection in the source node is scanned and all documents are then inserted into matching collections in the target node (with indexes being built at the time of document insertion). However, users and customers leveraging logical initial sync, especially those trying to synchronize large data sizes, have reported frustratingly long initial sync times.

Starting with 5.2, we have added the option of initial sync via file copy to significantly improve the performance of initial syncs. With this method, MongoDB will copy files from the file system of the source node to the file system of the target node. This process can be faster than a logical initial sync, especially at larger data sizes. In our testing with a 630 GB dataset, initial sync via file copy was nearly four times (4X) faster than a logical initial sync on the same dataset. This new capability builds upon the continuous enhancements we've made to improve resilience and scalability, including the ability for initial sync to automatically resume after a network failure, and allowing users to specify their preferred initial sync source – both introduced with MongoDB 4.4.

For more information, see the documentation on initial sync.

Summary

MongoDB 5.0 marks the arrival of a new release cadence that delivers new features to you faster than ever. The Stable API coupled with Live Resharding helps future-proof your applications, while Native Time Series support makes it even easier to address a broad range of workloads.

These along with the other features and improvements discussed in this guide make MongoDB 5.0 the best choice for your new workloads and upgrade target for existing MongoDB deployments.

> *"Clearly, MongoDB is massively popular with developers due to its flexibility, ease of use, and agile capability, all of which are resultant from its document model foundation. These latest improvements also add breadth of integrated data management and deployment with enhanced security and privacy capabilities. MongoDB's apparent aim is to provide support for a broader range of workloads and enable its customers to achieve better competitive advantage moving forward in a world that is increasingly data-unified and cloud-based."*

Carl Olofson, Research Vice President, Data Management Software at IDC

Safe Harbor

The development, release, and timing of any features or functionality described for our products remains at our sole discretion. This information is merely intended to outline our general product direction and it should not be relied on in making a purchasing decision nor is this a commitment, promise or legal obligation to deliver any material, code, or functionality.

We Can Help

We are the company that builds and runs MongoDB. Over 29,000 organizations rely on our commercial products. We offer cloud services and software to make your life easier:

MongoDB Atlas is the global cloud database service for modern applications. Deploy fully managed MongoDB across AWS, Azure, or Google Cloud with best-in-class automation and proven practices that guarantee availability, scalability, and compliance with security standards.

MongoDB Enterprise Advanced is the best way to run MongoDB on your own infrastructure. It's a finely-tuned package of advanced software, support, certifications, and other services designed for the way you do business.

MongoDB Atlas Data Lake allows you to quickly and easily query data in any format on Amazon S3 using the MongoDB Query Language and tools. You don't have to move data anywhere, you can work with complex data immediately in its native form, and with its fully-managed, serverless architecture, you control costs and remove the operational burden.

MongoDB Charts is the best way to create, share and embed visualizations of MongoDB data. Build visualizations quickly and easily to analyze complex, nested data. Embed individual charts into any web application or assemble them into live dashboards for sharing.

Realm Mobile Database allows developers to store data locally on iOS and Android devices using a rich data model that's intuitive to them. Combined with the MongoDB Realm sync-to-Atlas, Realm makes it simple to build reactive, reliable apps that work even when users are offline.

MongoDB Realm allows developers to validate and build key features quickly. Application development services like Realm Sync for mobile and Realm's GraphQL service, can be used with Realm Functions, Triggers, and Data Access Rules – simplifying the code required to build secure and performant apps.

MongoDB Cloud Manager is a cloud-based tool that helps you manage MongoDB on your own infrastructure. With automated provisioning, fine-grained monitoring, and continuous backups, you get a full management suite that reduces operational overhead, while maintaining full control over your databases.

MongoDB Consulting packages get you to production faster, help you tune performance in production, help you scale, and free you up to focus on your next release.

MongoDB Training helps you become a MongoDB expert, from design to operating mission-critical systems at scale. Whether you're a developer, DBA, or architect, we can make you better at MongoDB.

Resources

For more information, please visit mongodb.com or contact us at sales@mongodb.com.

Case Studies (mongodb.com/customers)
Presentations (mongodb.com/presentations)
Free Online Training (university.mongodb.com)
Webinars and Events (mongodb.com/events)
Documentation (docs.mongodb.com)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Enterprise Download (mongodb.com/download)
MongoDB Realm (mongodb.com/realm)

MongoDB.

US 866-237-8815 • INTL +1-650-440-4474 • info@mongodb.com