MongoDB: Delivering Real-Time Insight with Business Intelligence & Analytics

November 2017

mongoDB

Table of Contents

Introduction

Business Intelligence (BI) and analytics provides an essential set of technologies and processes that organizations have relied upon over many years to inform operational insight and guide strategic business decisions.

With the emergence of new data sources such as social media, mobile apps, machine learning, and sensor-equipped “Internet of Things” networks, organizations can extend BI to deliver real-time insight and discovery into such areas as operational performance, customer satisfaction, and competitor behavior.

However, these new data assets, often referred to as “big data”, challenge many of the previous assumptions around data storage and processing for BI and analytics applications. Not only do organizations have to manage much higher volumes of information, the data itself arrives at much faster rates and is more complex, multi-structured, and dynamic than existing transactional data sources. To be useful, the data must be analyzed and visualized in shorter intervals than traditional reporting cycles.

While many BI systems are adapting to these new requirements, the underlying databases powering BI and analytics processes may not afford the same flexibility. Organizations need to explore alternative technologies that augment their systems to fully integrate and benefit from modern application data.

With its rich document model, powerful analytical capabilities over high volumes of multi-structured data sets, and native Connector for BI providing integration with the leading BI and analytics tools, MongoDB provides a foundation to evolve BI to support real-time analytics for modern applications. Combining the leading BI and visualization platforms with the fastest-growing database enables organizations to realize real time analytics against live, operational data without the time and expense of ETL processes. This approach offers many benefits to business analysts tasked with extracting insight from modern, data driven applications:

- Developers can build more functional applications faster, using a single database technology.
- Operations teams eliminate the requirement for shuttling data between separate operational and analytics infrastructure, each with its own unique

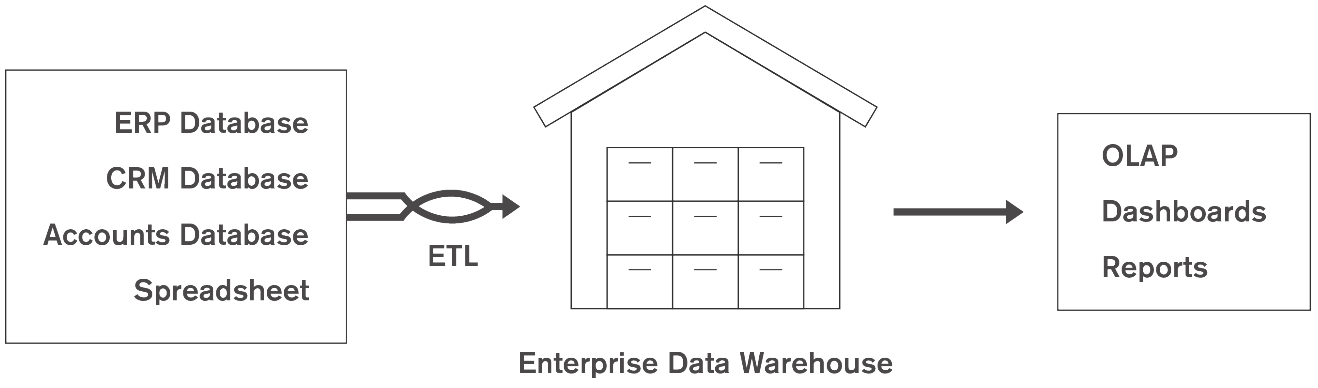

Figure 1: Traditional BI Process

configuration, maintenance and management requirements.

- CIOs deliver faster time-to-insight for the business, with lower cost and risk.

The Modern "Big Data Challenge" for Business Intelligence & Analytics

In traditional BI platforms, the flow of data – starting with its acquisition from source systems through to transformation, consolidation, analysis, and reporting – follows a well-defined sequential process, as illustrated in Figure 1.

Operational data from multiple source systems is integrated into a centralized Enterprise Data Warehouse (EDW) and local data-marts via Extract Transform Load (ETL) processes. Reports and visualizations of the data are then generated by BI tools. This workflow is optimized for users, enabling the deep historical analysis used to inform strategic decision making at senior levels within the organization. Databases and reporting queries are predicated on a number of assumptions:

1. **Predictable Frequency.** Data is extracted from source systems at regular intervals – typically measured in days, months and quarters.
2. **Static Sources.** Data is sourced from controlled, internal systems supporting established and well-defined back-office processes.
3. **Fixed Models.** Data structures are known and modeled in advance of analysis. This enables the development of a single schema to accommodate data from all of the source systems, but adds significant time to the upfront design.
4. **Defined Queries.** Questions to be asked of the data (i.e., the reporting queries) are pre-defined. If not all of the query requirements are known upfront, or requirements change, then the schema has to be modified to accommodate changes.
5. **Slow-changing requirements.** Rigorous change control is enforced before the introduction of new data sources or reporting requirements.
6. **Limited users.** The consumers of BI reports are typically business managers and senior executives.

Evolving BI and Analytics for Modern Data

Businesses want to harness new data sources and fast time-to-insight in new and compelling ways. Examples include:

- Retailers tracking user preferences, web clicks, and social sentiment to identify and automatically target geo-aware and device-aware personalized content and promotions
- Utilities capturing household energy usage levels to predict outages and to incent more efficient energy consumption
- Governments detecting and tracking the emergence of disease outbreaks via social media signals

- Oil and gas companies taking the sensor output from their drilling equipment to make more efficient and safer exploration decisions

The availability of new data sources generating “big data” is challenging the previous assumptions of data management and reporting within the BI platform.

The Need for Speed & Scale

Time to value is everything. For example, having access to real-time customer sentiment or logistics tracking is of little benefit unless the data can be analyzed and reported in real-time. As a consequence, the frequency of data acquisition, integration and analysis must increase from days to minutes or less, placing significant operational overhead on BI systems. In a growing number of cases, source data needs to be analyzed in-place, directly within operational systems avoiding lengthy ETL cycles, in order to provide the responsiveness and low latency insight demanded by the business.

The availability of new data sources drives an explosion in the amount of data organizations must manage, with analysts estimating a doubling in volumes every 12 to 18 months. Not only do BI databases have to handle much higher ingestion rates (often referred to as “data velocity”), there is also the challenge of how data is moved through the BI pipeline, from source systems to the EDW, data-marts and into analytical and reporting processes.

Agile Analytics and Reporting

With such a diversity of new data sources, business analysts can not know all of the questions they need to ask in advance. Therefore an essential requirement is that the data can be stored before knowing how it will be processed and queried.

The Changing Face of Data

Data generated by such workloads as social, mobile, sensor, machine learning, and logging is much more complex and variably structured than traditional transaction data from back-office systems such as ERP, CRM, PoS (Point of Sale), and Accounts Receivable.

The existing relational databases used by these back-office systems are designed to model cleansed and neatly structured data into tabular row and column formats with defined values, enforced by rigid schemas. They were never designed for the polymorphic, semistructured or unstructured data that is now typical in many of today’s modern, data-driven applications.

Higher Uptime Requirements

The immediacy of real-time analytics accessed from multiple fixed and mobile devices places additional demands on the continuous availability of BI systems. Batch-based systems can often tolerate a certain level of downtime – for example, for scheduled maintenance. Online operational systems, on the other hand, need to maintain service during both failures and planned upgrades.

Taking BI to the Cloud

The drive to embrace cloud computing to reduce costs and improve agility means BI components that have traditionally relied on databases deployed on monolithic, scale-up systems have to be re-designed for the elastic scale-out, microservices architecture of the cloud.

Impacts to Traditional BI Databases

The relational databases underpinning many of today’s traditional BI platforms are not well suited to the requirements of big data:

- **Semi-structured and unstructured data** typical in mobile, social, and sensor-driven applications cannot be efficiently represented as rows and columns in a relational database table.
- **Rapid evolution of database schema** to support new data sources and rapidly changing data structures is not possible in relational databases, which rely on costly ALTER TABLE operations to add or modify table attributes.
- **Performance overhead** of JOINs and multi-table transaction semantics prevents relational databases

from keeping pace with the ingestion and processing of high-velocity data sources.

- **Quickly growing data volumes** require scaling databases out across commodity hardware, rather than the scale-up approach typical of most relational databases.

Relational databases' inability to handle the speed, size, and diversity of rapidly changing data generated by modern applications is already driving the enterprise adoption of so called "NoSQL" and "Big Data" technologies in both operational and analytical roles.

Integrating Operational Data with BI and Analytics

Modern data workflows can take both online and offline forms. It is important to differentiate these forms in terms of performance, availability and data usage requirements to better understand where technologies such as MongoDB and Hadoop can be used within a modern BI system.

Differentiating Between Online and Offline Data

Online Big Data refers to data that is created, ingested, transformed, managed, and/or analyzed in real time to support operational applications and their users. Modern data is born online. Latency for these operational applications must be very low and availability must be high in order to meet SLAs and user expectations for modern application performance. This includes a vast array of applications, from social networking news feeds to analytics, from real-time IOT data streams and ad servers to complex CRM applications. Examples of databases powering online modern data applications include MongoDB and other non-relational databases with the capability to handle the real-time, richly structured, rapidly changing, high volume data sets now available to the business.

Offline Big Data encompasses applications that ingest, transform, manage and/or analyze data in a batch context. They typically do not create new data. For these applications, response time can be slow (up to hours or days), which is often acceptable for this type of use case. Since they usually produce a static (vs. operational) output, such as historical reports or dashboards, they can even go offline temporarily without impacting the overall goal or end product. Examples of offline "Big Data" applications include Hadoop-based workloads and enterprise data warehouses.

Organizations evaluating which data management technologies to adopt should consider how they intend to use their data. Those looking to build applications that support real-time, operational use cases and analytics will need an operational data platform like MongoDB. For those that need a place to conduct long-running analysis offline, technologies like Hadoop can be an effective tool.

Organizations pursuing both use cases can do so in tandem, and they will often find integrations between online and offline technologies. For instance, MongoDB provides integration with Hadoop to create an operational data layer on top of a batch-based Hadoop data lake. This is discussed in more detail later in the paper.

MongoDB for Integrating Online Big Data

With its rich document model, powerful query functionality, scalable architecture, and integration with leading BI and analytics tools, MongoDB can be deployed as a key database component both within, and as an extension of, a BI platform, including:

- A conventional data source for regular ETL processes integrating data into the EDW or Hadoop.
- A single view data platform aggregating data from operational and EDW sources, allowing for cross-function, complete 360-degree view reporting and visualization. This is illustrated in Figure 3.
- An Operational Data Store (ODS) enabling real-time analytics and dashboards to be generated against live, operational data. This is shown in Figure 4, alongside the traditional EDW data flow.

As the most widely deployed non-relational database MongoDB is at the forefront of brining modern application

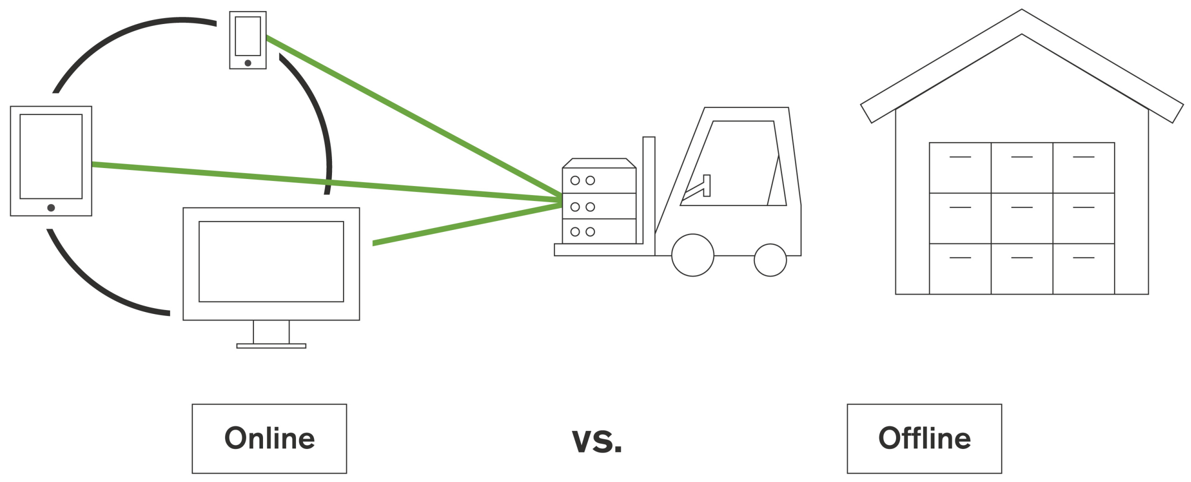

Figure 2: Data Management with MongoDB and Hadoop

data to BI platforms. The following sections explore MongoDB's capabilities for real time BI and analytics.

Modeling Complex Data with MongoDB's BSON Documents

Rich data structures with dynamic attributes comprising text, geospatial coordinates, media, arrays, embedded elements, and other complex types are common in today's web, mobile, social, and sensor-driven applications.

Unlike relational databases that flatten data into 2-dimensional tabular structures of rows and columns, MongoDB efficiently models and stores this rich data as documents (analogous to rows in a relational database) in a binary representation called BSON (Binary JSON). The BSON encoding extends the popular JSON (JavaScript Object Notation) representation to include additional types such as int, long, date, floating point, and decimal128. BSON documents contain one or more fields comprising a value with a specific data type. Each field can be indexed and queried, giving additional flexibility to BI reporting.

With sub-documents and arrays, BSON documents also align with the structure of objects at the application level. This makes it easy for developers to map the data used in the application to its associated document in the database, reducing complexity and making it easier to bring new applications to market faster.

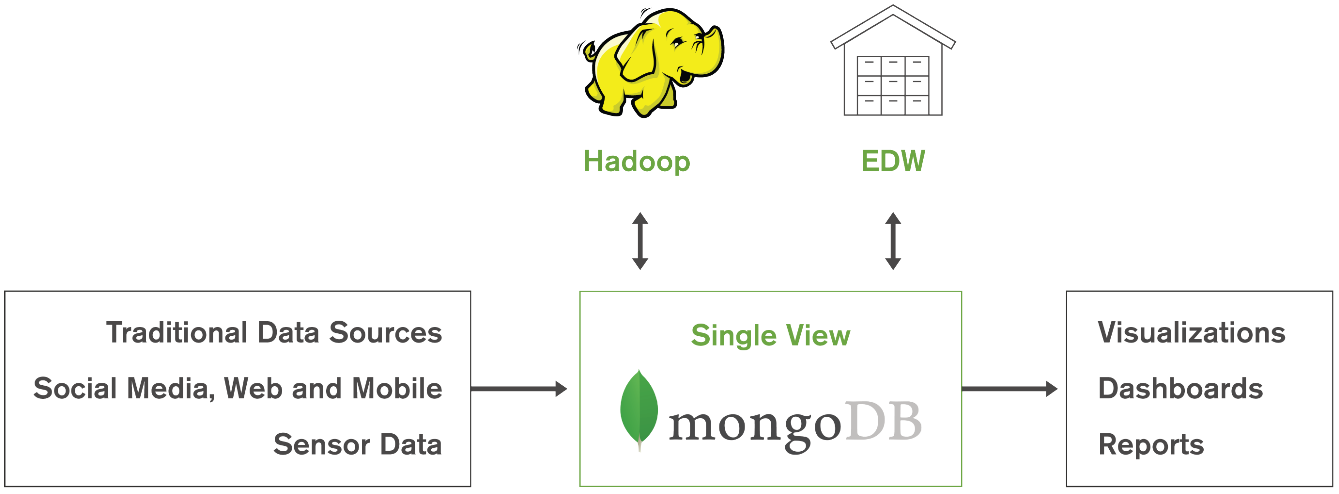

Figure 3: Single View Powered by MongoDB

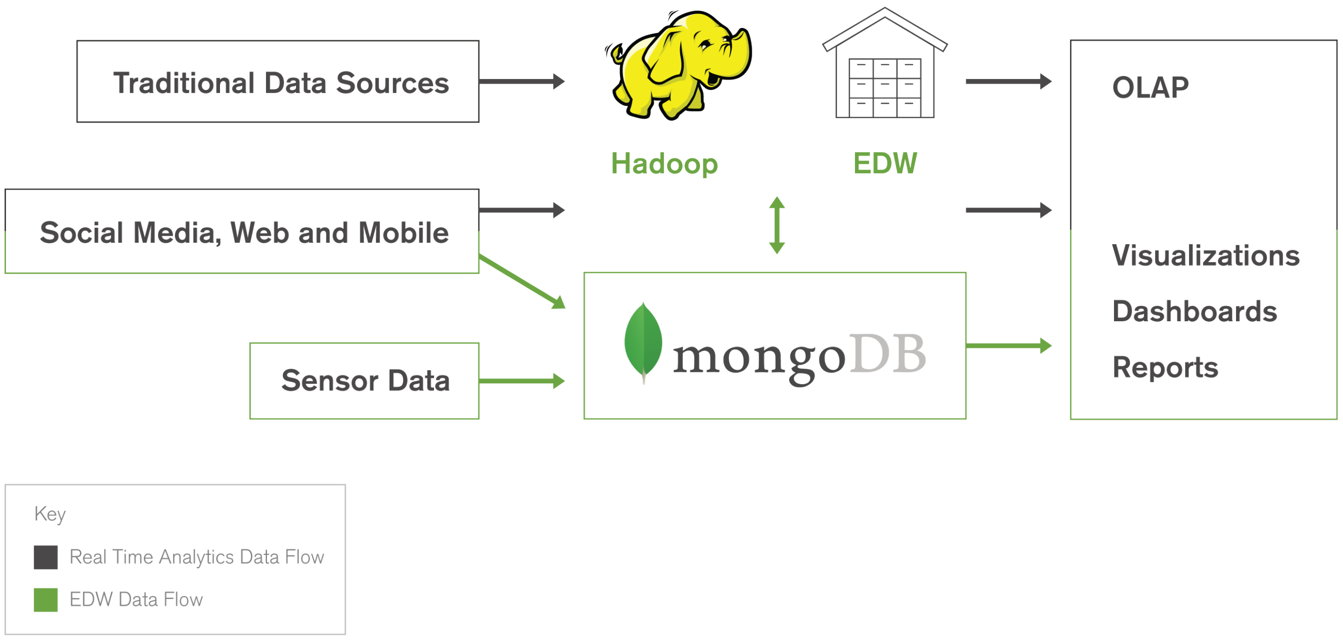

Figure 4: Real-Time Analytics Powered by MongoDB

Rapidly Evolving the Data Model with Dynamic Schemas

MongoDB's dynamic schema provides a major advantage for BI applications that need to ingest, store, and process rapidly evolving data streams from new sources.

Collections (analogous to tables in a relational database) can be created without first defining their structure. Documents in a given collection need not all have the same set of fields. Users can adapt the structure of documents just by adding new fields or deleting existing ones, making it very simple to extend BI applications by adding new attributes for analysis and reporting.

Dynamic schemas bring great agility, but it is also important that controls can be implemented to maintain data quality, especially for reporting and analytics applications. Unlike NoSQL databases that push enforcement of these controls back into application code, MongoDB provides schema validation within the database. Users can enforce checks on document structure, data types, data ranges and the presence of mandatory fields. As a result, business analysts can apply data governance standards, while developers maintain the benefits of a flexible document model.

Real-Time Analytics for Operational Data

Through powerful query functionality and indexing, MongoDB enables users to run analytics in real-time directly against their data.

MongoDB users have access to a broad array of query, projection and update operators supporting analytical queries against live operational data, supporting tunable consistency to balance performance against data freshness:

- Aggregation and MapReduce queries, discussed in more detail below
- Range queries returning results based on values defined as inequalities (e.g., greater than, less than or equal to, between)
- Geospatial queries returning results based on proximity criteria, intersection and inclusion as specified by a point, line, circle or polygon
- Search queries return results in relevance order and in faceted groups, based on text arguments using Boolean operators (e.g., `AND`, `OR`, `NOT`), and through bucketing, grouping and counting of query results
- JOINs allow documents from separate collections to be conveniently combined in a single operation.

- Graph queries bring native graph processing within MongoDB, enabling efficient traversals across trees, graphs and hierarchical data to uncover patterns and surface previously unidentified connections
- Key-value queries returning results based on any field in the document, often the primary key
- Native BI connectivity for rich data visualisations, connectivity with Apache Spark for complex analytics – such as machine learning – and Hadoop integration to expose analytics generated in the data lake to operational applications.

With the combination of MongoDB's dynamic document model and comprehensive query framework, users are able to store data before knowing all of the questions they will need to ask of it.

Data Aggregation

The MongoDB Aggregation Pipeline is similar in concept to the SQL GROUP BY statement, enabling users to generate aggregations of values returned by the query (e.g., count, min, max, average, standard deviation) that can be used to power analytics dashboards and visualizations.

Using the Aggregation Framework, documents in a MongoDB collection (analogous to a table in a relational database) pass through an aggregation pipeline, where they are processed by operators. Expressions produce output documents based on calculations performed on the input documents. The accumulator expressions used in the $group operator maintain state (e.g., totals, mins, maxs, averages) as documents progress through the pipeline.

Beyond Group_By type queries, the aggregation pipeline allows developers and data engineers to declaratively create sophisticated processing pipelines for data analytics and transformations, powering faceted search, JOINs and graph traversals natively within MongoDB.

Result sets from the aggregation pipeline can be written to a named collection with no limit to the output size (subject to the underlying storage system). Existing collections can be replaced with new results while maintaining previously defined indexes to ensure queries can always be returned efficiently over rapidly changing data.

In-Database MapReduce

MongoDB provides native support for MapReduce, enabling complex data processing that is expressed in JavaScript and executed across data in the database. Multiple MapReduce jobs can be executed concurrently across both single servers and sharded collections.

The incremental MapReduce operation uses a temporary collection during processing so it can be run periodically over the same target collection without affecting intermediate states. This mode is useful when continuously updating statistical output collections used by reporting dashboards.

Optimizing Analytics Queries with Indexes

MongoDB provides a number of different index types to optimize performance of real-time and ad-hoc analytics queries across highly variable, fast moving data sets. These same indexes can be used by the MongoDB Connector for BI, and by Apache Spark and Hadoop to filter and extract only relevant subsets of data against which analytics can be run.

Indexes can be created on any field within a document. In addition to supporting single field and compound indexes, MongoDB also supports indexes of arrays with multiple values, short-lived data (i.e., Time To Live), partial, sparse, geospatial and text data, and can enforce constraints with unique indexes. Index intersection allows indexes to be dynamically combined at runtime to efficiently answer explorative questions of the data. Refer to the documentation for the full list of index types.

The MongoDB query optimizer selects the best index to use by running alternate query plans and selecting the index with the best response time for each query type. The results of this empirical test are stored as a cached query plan and are updated periodically.

MongoDB also supports covered queries – where returned results contain only indexed fields, without having to access and read from source documents. With the appropriate indexes, analytics workloads can be optimized to use predominantly covered queries.

Scaling MongoDB for New Data Sources

As data volumes grow, it is essential for users to select a database that will grow with them. MongoDB provides horizontal scale-out of data sets using a technique called sharding. Transparent to the underlying applications, sharding distributes data across multiple physical partitions and nodes. Sharding allows MongoDB deployments to address the hardware limitations of a single server, such as storage and RAM or bottlenecks in disk I/O, without adding complexity to the application. Sharding enables MongoDB to quickly scale both capacity and performance as new data sources are added to the BI platform.

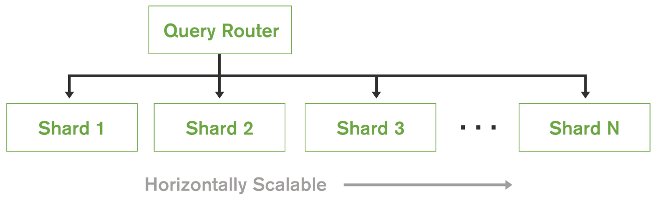

Figure 5: Automatic Scaling Across Commodity Compute Clusters

MongoDB automatically balances the data in the cluster as the data grows or the number of nodes in the cluster increases or decreases, making it ideal for cloud-based deployments.

MongoDB Integration with BI and Analytics Tools

To make modern data actionable through dashboards, reports, visualizations and integration with other data sources, it must be accessible to established BI and analytics platforms.

The MongoDB Connector for BI

MongoDB's flexible data model and dynamic schema allow you to store data in rich, multi-dimensional documents to quickly build and evolve your apps. But your BI platform expects fixed schemas and tabular data, and to query it via SQL.

The MongoDB Connector for BI lets you use MongoDB as a data source for your SQL-based BI and analytics platforms. The MongoDB Connector for BI acts as a layer that passes queries and data between a MongoDB instance and your BI tool. The Connector stores no data, and purely serves to bridge your MongoDB server with business intelligence platforms.

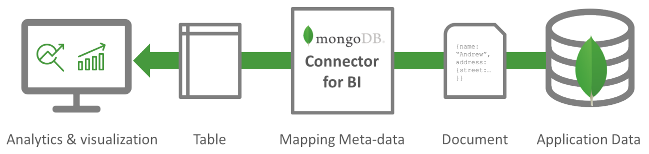

Figure 6: MongoDB Connector for BI

In order for BI tools to query MongoDB as a data source, the Connector for BI does the following: * Provides the BI tool with the schema of the MongoDB collection to be visualized. Users can review the schema output to ensure data types, sub-documents and arrays are correctly represented as tabular structures. * Translates SQL statements issued by the BI tool into equivalent MongoDB queries that are then sent to MongoDB for processing within the database. * Converts the returned results into the tabular format expected by the BI tool, which can then visualize the data based on user requirements.

The MongoDB Connector for BI can be used with any BI or analytics platform based on SQL. The platforms below are just a selection:

Figure 7: Sample BI Platforms for Use with the MongoDB Connector

The connector takes advantage MongoDB's secondary indexes and aggregation pipeline – including equi and nonequi JOINs, pushing query predicates and operations down to the database. This approach reduces the amount of data that needs to be moved and computed in the BI layer, providing faster time to insight. In addition, performance metrics are observable via the Show Status function, enabling deep performance insights and optimizations.

The MongoDB Connector for BI Version is compatible with SQL-99 SELECT statements, connects to BI tools via the MySQL wire protocol. It can be installed manually or deployed with MongoDB Ops Manager.

The MongoDB Connector for BI is available with MongoDB Enterprise Advanced, and can be evaluated as part of a free trial. Review the documentation to learn more.

Beyond BI connectivity, MongoDB also offers native analytics integration with Apache Spark and Hadoop.

Integrating MongoDB and Spark

While MongoDB natively offers rich real-time analytics capabilities, there are use cases where integrating Apache Spark can extend the processing of operational data managed by MongoDB. By using the MongoDB Connector for Apache Spark, many organisations are building new classes of sophisticated, real-time analytics applications.

The MongoDB Connector for Apache Spark exposes all of Spark's libraries, including Scala, Java, Python and R. MongoDB data is materialized as DataFrames and Datasets for analysis with machine learning, graph, streaming, and SQL APIs.

The Spark Connector can take advantage of MongoDB's aggregation pipeline and rich secondary indexes to extract, filter, and process only the range of data it needs – for example, analyzing all customers located in a specific geography. This is very different from simple NoSQL datastores that do not offer secondary indexes or in-database aggregations. In these cases, Spark would need to extract all data based on a simple primary key, even if only a subset of that data is required for the Spark process. This means more processing overhead, more hardware, and longer time-to-insight for data scientists and engineers.

To maximize performance across large, distributed data sets, the MongoDB Connector for Apache Spark can co-locate Resilient Distributed Datasets (RDDs) with the source MongoDB node, thereby minimizing data movement across the cluster and reducing latency.

Integrating MongoDB and Hadoop

Many organizations are harnessing the power of Hadoop and MongoDB together to create complete data analytics pipelines:

- MongoDB powers the online, real time operational application, serving business processes and end-users, exposing analytics models created by Hadoop to operational processes
- Hadoop consumes data from MongoDB, blending it with data from other sources to generate sophisticated analytics and machine learning models. Results are loaded back to MongoDB to serve smarter and contextually-aware operational processes – i.e., delivering more relevant offers, faster identification of fraud, better prediction of failure rates from manufacturing processes.

Organizations such as eBay, Orbitz, Pearson, Square Enix and SFR are using MongoDB to operationalize their Hadoop-based data lakes.

The MongoDB Connector for Hadoop presents MongoDB as a Hadoop data source allowing Hadoop jobs to read data from MongoDB directly without first copying it to HDFS, thereby eliminating the need to move TBs of data between systems. Hadoop jobs can pass queries as filters, thereby avoiding the need to scan entire collections and speeding up processing; they can also take advantage of MongoDB's indexing capabilities, including text and geospatial indexes.

As well as reading from MongoDB, the Hadoop connector also allows results of Hadoop jobs to be written back out to MongoDB, to support real-time operational processes and ad-hoc querying by both MongoDB and Spark processes.

The Hadoop connector supports MapReduce, Spark on HDFS, Pig, Hadoop Streaming (with Node.js, Python or Ruby) and Flume jobs. Support is also available for SQL queries from Apache Hive to be run across MongoDB data sets. The connector is certified on the leading Hadoop distributions from Cloudera, Hortonworks, and MapR.

To learn more about creating an operational data lake with Hadoop and MongoDB, download the whitepaper.

Creating Reactive Data Pipelines with Change Streams

MongoDB change streams enable developers and data engineers to build reactive, real-time, web, mobile, and IoT apps that can view, filter, and act on data changes as they occur in the database. Change streams enable seamless data movement across distributed database and analytics engines, making it simple to stream data changes and trigger actions wherever they are needed, using a fully reactive programming style. In addition to updating dashboards, analytics systems, and search engines as operational data changes, other use cases enabled by MongoDB change streams include:

- Powering trading applications that need to be updated in real time as stock prices rise and fall.
- Refreshing scoreboards in multiplayer games.
- Creating powerful IoT data pipelines that can react whenever the state of physical objects change.
- Synchronizing updates across serverless and microservices architectures by triggering an API call when a document is inserted or modified.

Conclusion

Data from modern apps is an essential extension of BI and analytics platforms, presenting new sources of operational insight and discovery to the business. However, the rate of data ingestion coupled with its complexity and volume are beyond the design constraints of many traditional databases used in BI systems today.

With its rich document model, powerful analytical capabilities over large, multi-structured data sets and integration with leading BI and analytics platforms, MongoDB provides a foundation to integrate modern application data with existing BI and analytics platforms, providing real-time insight and analytics.

We Can Help

We are the MongoDB experts. Over 4,300 organizations rely on our commercial products, including startups and more than half of the Fortune 100. We offer software and services to make your life easier:

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

MongoDB Atlas is a database as a service for MongoDB, letting you focus on apps instead of ops. With MongoDB Atlas, you only pay for what you use with a convenient hourly billing model. With the click of a button, you can scale up and down when you need to, with no downtime, full security, and high performance.

MongoDB Stitch is a backend as a service (BaaS), giving developers full access to MongoDB, declarative read/write controls, and integration with their choice of services.

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 Professional helps you manage your deployment and keep it running smoothly. It includes support from MongoDB engineers, as well as access to MongoDB Cloud Manager.

Development Support helps you get up and running quickly. It gives you a complete package of software and services for the early stages of your project.

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 Enterprise Download (mongodb.com/download)
MongoDB Atlas database as a service for MongoDB (mongodb.com/cloud)
MongoDB Stitch backend as a service (mongodb.com/cloud/stitch)