Couchbase

Couchbase 4.0

What's New in Couchbase 4.0

Couchbase 4.0 is a major release that includes significant advances in both architecture and features. Major innovations in Couchbase 4.0 include Multi-Dimensional Scaling (MDS); a comprehensive, SQL-compatible query language we've named N1QL (pronounced "nickel"); global secondary indexes; further optimizations for cross–data center replication; and significant enhancements to security. These capabilities dramatically extend the scalability and performance advantages of Couchbase, and enable support for a much broader set of use cases.

With MDS and N1QL, Couchbase 4.0 becomes the first and only database to combine the powerful query capabilities of a relational database with the performance, scalability, and flexibility of a NoSQL database. Couchbase 4.0 now enables you to query billions of documents and meet the performance and scalability requirements of enterprise web, mobile, and Internet of Things (IoT) applications with a NoSQL database.

N1QL = JSON + SQL

The Bottom Line: N1QL is a comprehensive and declarative query language that brings the power and familiar syntax of SQL together with the flexibility of a JSON data model — making it faster and easier for developers to build web, mobile, and IoT enterprise applications on top of a scalable database. Couchbase 4.0 is the first NoSQL database to combine high performance, scale, and comprehensive query with ease of development.

Couchbase N1QL is a comprehensive, declarative query language that leverages the flexibility of JSON and the power of SQL. It dramatically increases developer agility, as it enables you to query and transform semi-structured JSON data in any manner your application requires. This includes the ability to build a flexible JSON data model consisting of related documents that can be queried via **JOINS**, and to **NEST** or **UNNEST** data to query or transform complex documents. Additionally, N1QL is accessible via your preferred development framework — whether LINQ, Spring, Ottoman (our object-document mapping [ODM] framework for Node.js), or anything else. There's no learning curve. Since the application processes query results directly as streams of JSON documents, there's no longer an impedance mismatch, and no need for a complex translation layer.

Why was N1QL necessary?

Quite simply, there was a need for a more powerful NoSQL query solution. Relational databases, with SQL, have been the historical standard, but their rigid, tabular structure is ill suited to support the scale and semi-structured data used by most web, mobile, and IoT applications. Document databases such as Couchbase, which support the JSON data format, are more flexible and scalable — but until now, their adoption has been limited by the lack of a powerful query language. N1QL changes that by extending SQL — recognized by virtually every developer in the world — to JSON, the industry-standard data model for web, mobile, and IoT applications.

Why SQL for NoSQL?

SQL is proven and powerful. It's been the database industry's standard query language for more than 40 years; millions of developers around the world are building scalable, enterprise applications today using SQL either directly or indirectly through application development frameworks. By extending SQL to JSON, Couchbase applies all the SQL experience and capability to JSON, and by leveraging existing SQL constructs, N1QL is familiar and easy for developers to adopt.

N1QL connects the SQL ecosystem to NoSQL.

N1QL further benefits the enterprise by making access to data stored in Couchbase easy and efficient. N1QL has full compatibility with the SQL ecosystem via connectors and standard JDBC/ODBC drivers. This allows enterprises for the first time to connect popular ETL, reporting, and BI tools to Couchbase. Companies like Databricks, Looker, Simba Technologies, Informatica, Tableau, and Metanautix are all partnering with Couchbase to provide supported integrations.

Multi-Dimensional Scaling

The Bottom Line: With Multi-Dimensional Scaling, enterprises can isolate different workloads and independently scale the index, query, and data services. As a result, adding database capacity is easier, faster, and cheaper. Rebalancing and redistributing data also becomes more efficient because it doesn't require re-creating the local indexes. MDS improves performance, reduces hardware costs, and enables enterprises to support a much broader set of applications with a single Couchbase database.

With the introduction of Multi-Dimensional Scaling, Couchbase 4.0 has redefined the way enterprises can scale distributed databases. MDS enables you to separate, isolate, and scale individual services — query, index, and data — to improve application performance and resource utilization. With MDS, Couchbase 4.0 is the first and only distributed database capable of scaling with the speed and precision required by web, mobile, and IoT applications.

More specifically, MDS enables enterprises to optimize hardware performance by (1) allocating resources based on the workload requirements of each specific service, and (2) isolating workloads of a specific type to avoid resource contention. Enterprises can now perform queries, maintain indexes, and write data on separate nodes. This configuration greatly improves performance because it removes inefficiencies from the traditional scale-out model, where every node has to participate in performing a query or maintaining an index.

The diagrams below illustrate the difference between traditional scale-out and Multi-Dimensional Scaling models.

Traditional scale-out architecture spreads all workloads across all nodes.

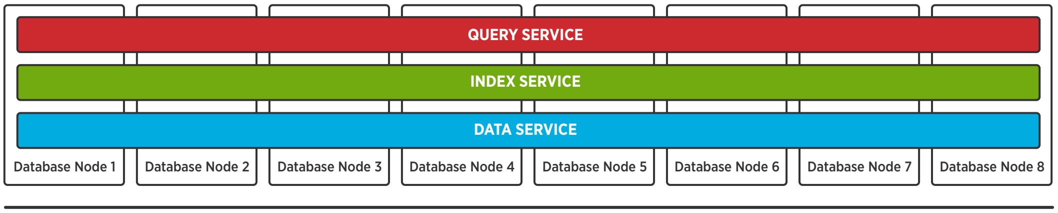

Figure: Homogeneous Scaling

Couchbase Multi-Dimensional Scaling isolates and optimizes database workloads.

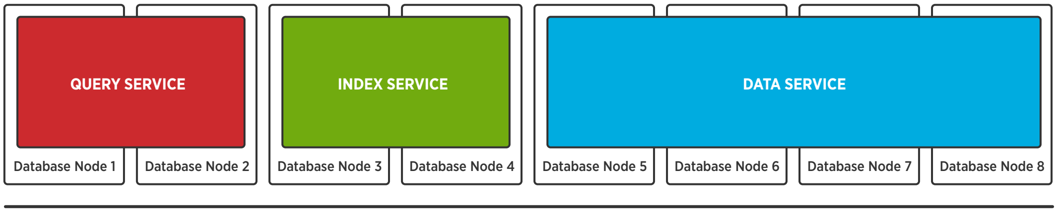

Figure: Independent Scaling with Multi-Dimensional Scaling

Global Secondary Indexes

The Bottom Line: Global Secondary Indexes provide a new access path to your data for faster lookup and higher throughput. It's a global index, so index lookup is much easier — there's no need to query a local index on each data node and aggregate the results. A Global Secondary Index can be independently scaled to maximize index performance, and can be deployed separately from query and data services to isolate workloads. Global Secondary Indexes improves query consistency and reliability for distributed databases.

At a high level, indexes speed access to data by enabling you to quickly look up objects that meet user-specified criteria without having to search every object in the database. Indexing is especially important in a distributed database such as Couchbase, because data in Couchbase clusters are often distributed across many nodes, and any item in any node may need to be accessed in order to satisfy a query.

Traditional local secondary indexes reside with data on every node of a Couchbase cluster. They're great when you need to query all of the data, but because they query every node every time — even to return data that's stored on a subset of the nodes — performance gets worse as the number of nodes increases.

Now, with Couchbase 4.0, we introduce Global Secondary Indexes, a unique new way to create indexes that greatly speeds up and scales queries. Global Secondary Indexes enable you to create indexes that are isolated from data processing and are aggregated on dedicated nodes. The benefit is that your queries only touch nodes with data that's being queried, resulting in faster query processing. By keeping the entire index on fewer nodes — perhaps just one node — the operational overhead remains constant even as the cluster grows. This ensures that requests to the index can be satisfied using local data without incurring network latency. In sum, the new Global Secondary Indexes capability lets you build new types of applications that weren't possible before — i.e., applications that can query big data with near real-time latencies.

Table: Comparison between Local and Global Secondary Indexes

Local Secondary Indexes	Global Secondary Indexes
Co-located with data	Isolated from Key-Value operations
High write performance	Async writes to a large number of global indexes
Lower read performance: scatter-gather	Higher query performance
Scaling bottleneck, as the number of indexes or data nodes gets larger	Independently scaled and partitioned

Enhanced Security

The Bottom Line: Couchbase 4.0 facilitates secure deployment and administration by adding simplified compliance with security standards, LDAP integration, and auditing capabilities.

Data security is a top concern for all businesses, as enterprises impose internal controls and must comply with external rules and regulations around data management. With Couchbase 4.0, we've introduced a number of important security controls, including simplified compliance with security standards, from PCI, to HIPAA, to FISMA, and more. Couchbase 4.0 also has native LDAP integration for admin account management and configurable audit trails that capture who does what, when they do it, and how they do it. This is all in addition to Couchbase's existing security capabilities, such as encryption of data at rest and on the wire.

In sum, Couchbase 4.0 takes data security to a higher level. It facilitates secure deployments by providing security controls to access the entire stack, from physical protection of the network infrastructure to Couchbase 4.0 and the deployed applications.

ForestDB Storage Engine

The Bottom Line: ForestDB is a high-performance and scalable storage engine designed to support compact and efficient index structures for Global Secondary Indexes. It is optimized to take advantage of new SSD technology for extreme performance and throughput. Its compact index structure and efficient use of storage make it blazingly fast across a mix of read and write workloads.

ForestDB is a fast and space-efficient Key-Value storage engine that's based on a Hierarchical B+-Tree based Trie, or HB+-Trie. We created ForestDB to provide a scalable and high-performance storage engine for the Couchbase NoSQL database. ForestDB is designed to efficiently manage variable-length keys and to perform well in read and write workloads, which are common in modern web, mobile, and IoT applications. ForestDB is optimized to take advantage of emerging SSD storage technology, and is designed as a unified storage engine that scales from small devices to large servers. It's included in Couchbase 4.0 to support indexes accessed via N1QL and it also powers the local database of Couchbase Mobile.

Cross–Data Center Replication (XDCR) Filtering

The Bottom Line: We've added filtering capabilities to Cross–Data Center Replication (XDCR) in order to significantly reduce the amount of data replicated across data centers. XDCR Filtering achieves this reduction by replicating only data relevant to the destination. Couchbase customers no longer need to create many different buckets just to segment data for Cross–Data Center Replication.

Couchbase 3.0 introduced Cross–Data Center Replication to provide an easy yet powerful way to replicate data from one cluster to another for increased high availability, disaster recovery, and geographic load balancing. Until Couchbase 4.0, however, you had to replicate all your data to each data center managed by XDCR, which could be inefficient for those whose primary objective is to make data available to geographically distributed locations at low latency. Now, with XDCR Filtering, you can select specific data to be replicated to a specific geography — a much more efficient process.

Geospatial Views

The Bottom Line: With Geospatial Views, developers can now develop rich applications that query and visualize data easily and efficiently in multiple dimensions, including but not limited to spatial dimensions.

Geospatial Views enable location-aware applications to query data based on geographic coordinates. Applications can incorporate queries that identify containment of points and shapes in a bounding box. Geospatial Views have been available as an experimental capability in earlier versions of Couchbase. With Couchbase 4.0, Geospatial Views become a supported part of the Couchbase experience. In addition, Geospatial Views are multi-dimensional. You can query not only on location but on location *and* attribute — for example, "restaurants in a city that are (a) open after midnight, and (b) deliver."

SQL and Big Data Ecosystem Support

The Bottom Line: Couchbase 4.0 enables you to integrate your data with your enterprise SQL and big data ecosystems, allowing your users to extract insights from data using the tools and applications they're familiar with.

Standard JDBC/ODBC drivers and N1QL's support for SQL make Couchbase the easiest NoSQL database to integrate with the broad SQL tool and application ecosystems. Customers can use tools like Excel and Tableau to visualize and analyze data, and they can easily migrate data from their legacy RDBMS into Couchbase.

Connectors for Spark and Kafka allow for near real-time integration of operational data in Couchbase with the Hadoop ecosystem. Enterprises can now perform real-time analysis of Couchbase data, and use those insights to drive personalized interactions with their customers.

SDK Support

The Bottom Line: Couchbase 4.0 SDKs improve developer productivity by allowing developers to query and manipulate data with their favorite language and framework. With the release of Couchbase 4.0, we're further empowering developers through additional language support, a rich query API, improved flexibility of programming models, improved support for durability, stronger and more specific logging information, and integrated support for popular platforms like Spark and Kafka.

All supported SDKs provide native support for the new Couchbase 4.0 query API.

- In Java, this means a DSL with a fluid API for autocomplete and improved developer productivity.
- In Spring, support is also first class, including rapid bootstrapping and prototyping of applications.
- In .NET, this means native support for our LINQ controller.
- In node.js, N1QL support is native and we've extended functionality by including our own ODM (Ottoman) that allows the familiar paradigm of working in objects.
- In libcouchbase (the C SDK), support for the new query API is native, including functionality normally reserved for higher-level languages.

We've also made several across-the-board developer enhancements, such as improved durability control, faster performance, improved connection management, and improved error handling through advanced logging (first in the Java SDK, with others to follow). In addition, we've introduced a new Golang SDK that was in development for over a year, which includes native query support as standard. And we now included native Spark and Kafka support within the JVM for the Couchbase SDK.

Full-Text Search — Developer Preview

The Bottom Line: Couchbase 4.0 supports integrated full-text search with simplified administration, enabling developers to easily add full-text search capabilities to any application, without having to deploy and manage additional components. This feature is included in Couchbase 4.0 as a developer preview.

Couchbase Full-Text Search (FTS) is an integrated full-text search engine, available in Couchbase 4.0 as a developer preview. It's a distributed, clusterable data indexing server that includes the ability to manage full-text and other kinds of indexes for JSON documents that you've created and stored in a Couchbase bucket and other data sources. The indexes that Couchbase FTS manages can be automatically distributed across multiple, clustered FTS server processes on different machines to support larger indexes, higher performance, and higher availability. With FTS, developers can easily add full-text search capabilities to any application, without deploying additional components — significantly reducing operational complexity. Alternatively, Couchbase customers who use third-party full-text search engines such as Elasticsearch or LucidWorks can leverage the available connectors to continuously replicate data from the Couchbase cluster to those search engines.

Bloom Filters

The Bottom Line: Couchbase 4.0 now includes bloom filters to significantly improve latency for accessing data in which the cached working set is much smaller than the data on disk (DGM).

Couchbase 4.0 leverages a bloom filter to avoid unnecessary storage I/O when trying to read data based on a key and that data may or may not exist. If the bloom filter returns a negative, Couchbase will not try to find the data on disk because there's no data for the key. As a result, Couchbase won't perform unnecessary storage I/O, thereby significantly improving latency for accessing data when the cached working set is much smaller than the data on disk.

Views and Indexes

The Bottom Line: In Couchbase 4.0, queries using views are two times faster for some scenarios compared to the 3.0 release.

In addition to using N1QL to query data, Couchbase also enables indexing and querying of data via views. At a base level, a view creates an index on the data according to the defined format and structure. The view consists of specific fields and information extracted from the objects in Couchbase. Views are used for a number of reasons, including producing interactive reports that pre-aggregate and summarize data, or interactive reports that require complex reshaping of data through programmable map and reduce functions. With Couchbase 4.0, we've made improvements to views, such that queries using views can run up to two times faster compared to the 3.0 release.

Improved Availability and Reliability

The Bottom Line: Couchbase 4.0 continues to improve availability of the system through enhancements to online operations and memory management.

All supported SDKs provide native support for the new Couchbase 4.0 query API.

- ***Scale memcached connections without restart:*** With Couchbase 4.0, the number of memcached connections can be increased on the fly without restarting the cluster. This means more concurrent client connections, and no downtime for connection scaling.
- ***Reduced memory fragmentation:*** In Couchbase 4.0, JEMalloc is the default memory allocator. This benefits applications with a heavy update workload by providing fast heap allocations and a compact memory footprint with low fragmentation. With JEMalloc, Couchbase has improved stability, eliminating unnecessary server restarts and application rewrites.
- ***Mini-core dumps with Google Breakpad integration:*** In Couchbase 4.0, memcached crashes are tracked with Google Breakpad, a widely used open source crash reporting toolkit. Thanks to this integration, a full core dump is no longer needed, and a crash report can reproduce the call stack all the way down to the line of code that crashed, without any user intervention. This vastly simplifies troubleshooting.
- ***Reduced memory pressure during backfill:*** In Couchbase 3.0, a rebalance or backup operation of clusters with a low resident memory ratio would trigger a backfill, where data was read from disk and loaded into memory before getting streamed out. Now, with Couchbase 4.0, we've enhanced the Database Change Protocol (DCP) that supports this process to reduce memory requirements for the backfill operation, thereby keeping the resident ratio more stable.

About Couchbase

Couchbase delivers the world's highest performing NoSQL distributed database platform. Developers around the world use the Couchbase platform to build enterprise web, mobile, and IoT applications that support massive data volumes in real time. The Couchbase platform includes Couchbase Server, Couchbase Lite - the first mobile NoSQL database, and Couchbase Sync Gateway. Couchbase is designed for global deployments, with configurable cross data center replication to increase data locality and availability. All Couchbase products are open source projects. Couchbase customers include industry leaders like AOL, AT&T, Bally's, Beats Music, BSkyB, Cisco, Comcast, Concur, Disney, eBay, KDDI, Nordstorm, Neiman Marcus, Orbitz, PayPal, Rakuten / Viber, Tencent, Verizon, Wells Fargo, Willis Group, as well as hundreds of other household names. Couchbase investors include Accel Partners, Adams Street Partners, Ignition Partners, Mayfield Fund, North Bridge Venture Partners, and West Summit.

2440 West El Camino Real | Ste 600
Mountain View, California 94040

1-650-417-7500

www.couchbase.com