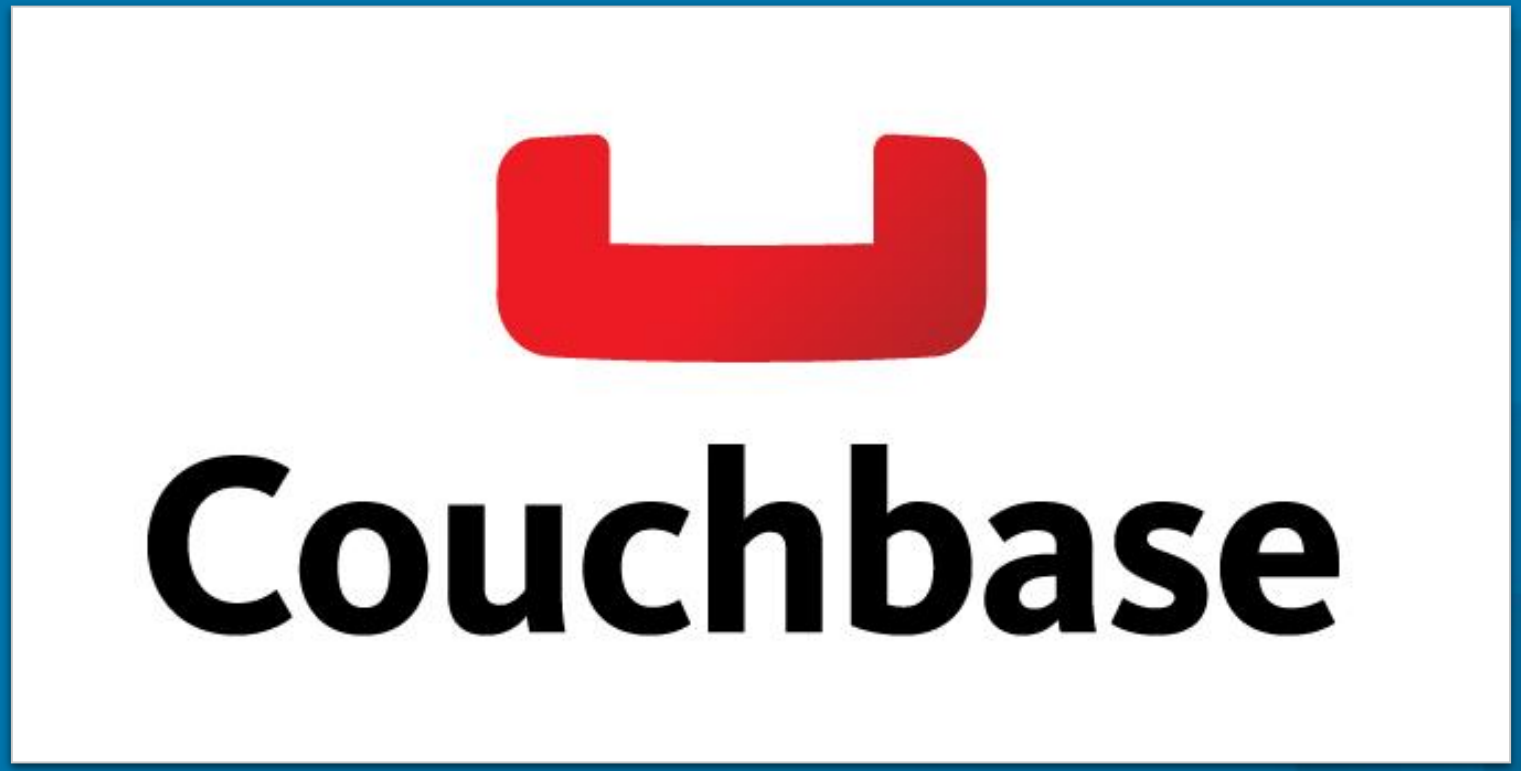
THE GEORGE WASHINGTON UNIVERSITY

WASHINGTON, DC



DATS 6102 Group Project: Lydia Teinfalt, Joe Valle, Natasha Vij

Fall 2020

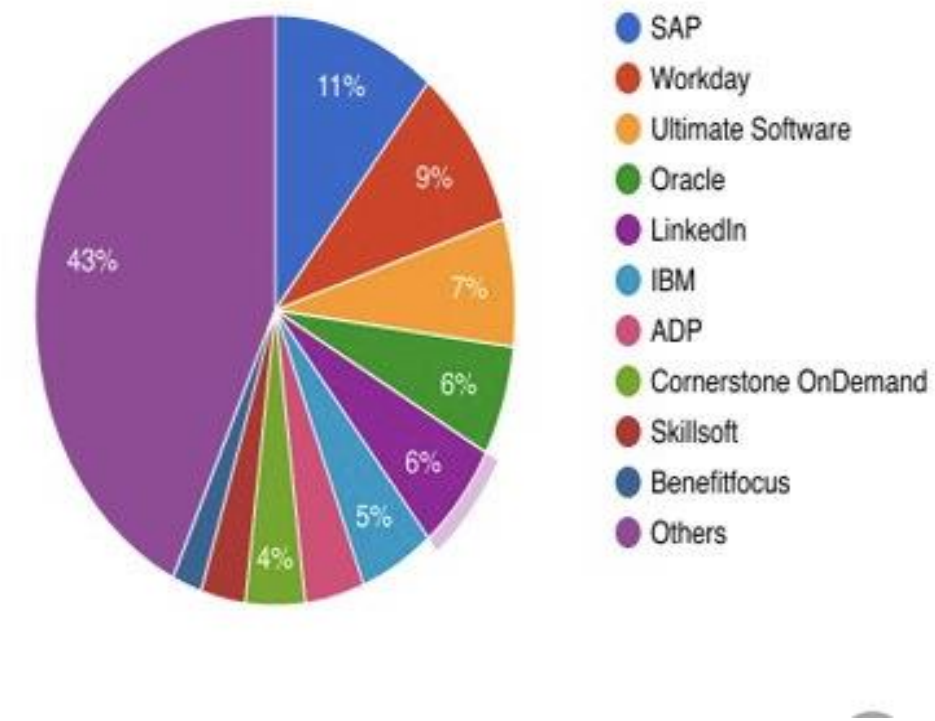
Table of Contents

What is Couchbase?

- Open-source, distributed multi-model NoSQL document-oriented database
- Developed by Couchbase, Inc. Headquartered in Santa Clara, CA
- Created in 2009 from the merger of CouchOne and Membase
  - Compared to MongoDB – created in 2007
- Written in C++, Erlang, C, Go
- NoSQL database technology

Sources
- https://www.linkedin.com/company/couchbase/
- https://en.wikipedia.org/wiki/Couchbase_Server
- https://www.couchbase.com/about

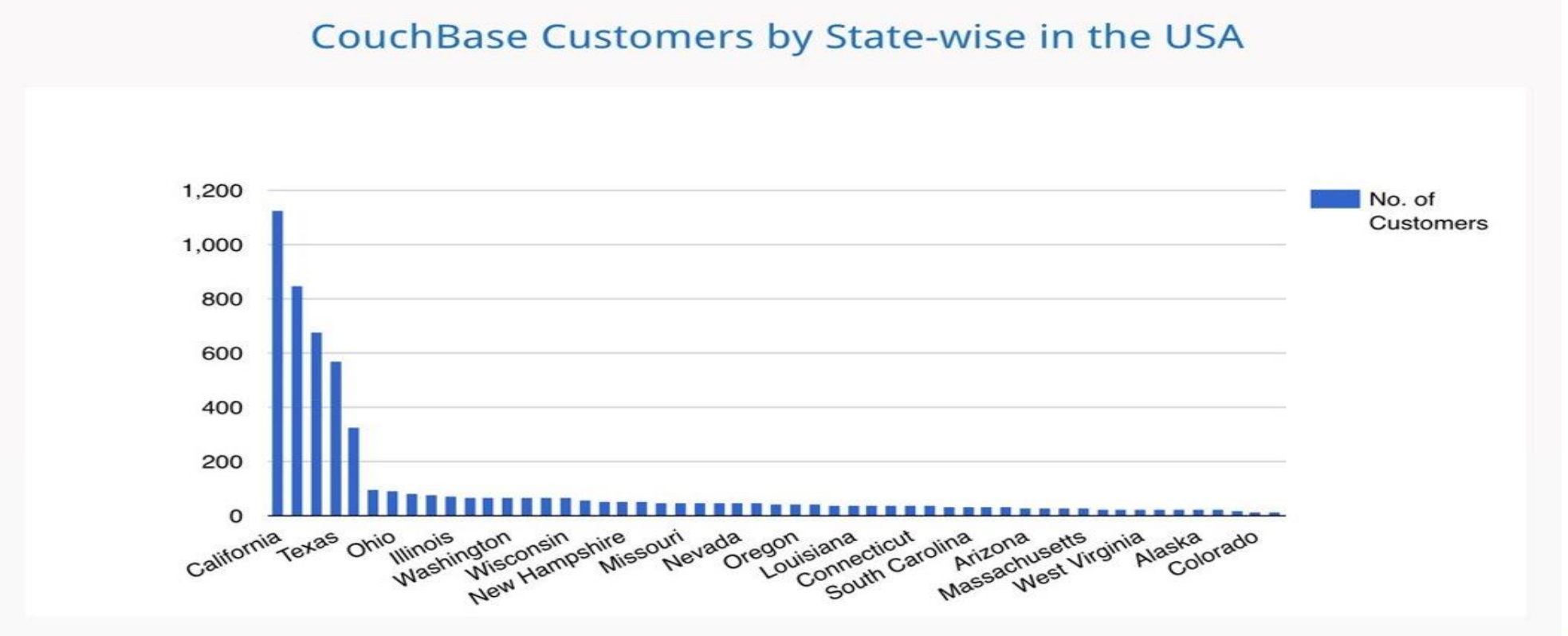
Customers



Company Name	Market Share
SAP	11%
Workday	9%
Ultimate Software	7%
Oracle	6%
LinkedIn	6%
IBM	5%
ADP	4%
Cornerstone OnDemand	4%
Skillsoft	3%
Benefitfocus	2%

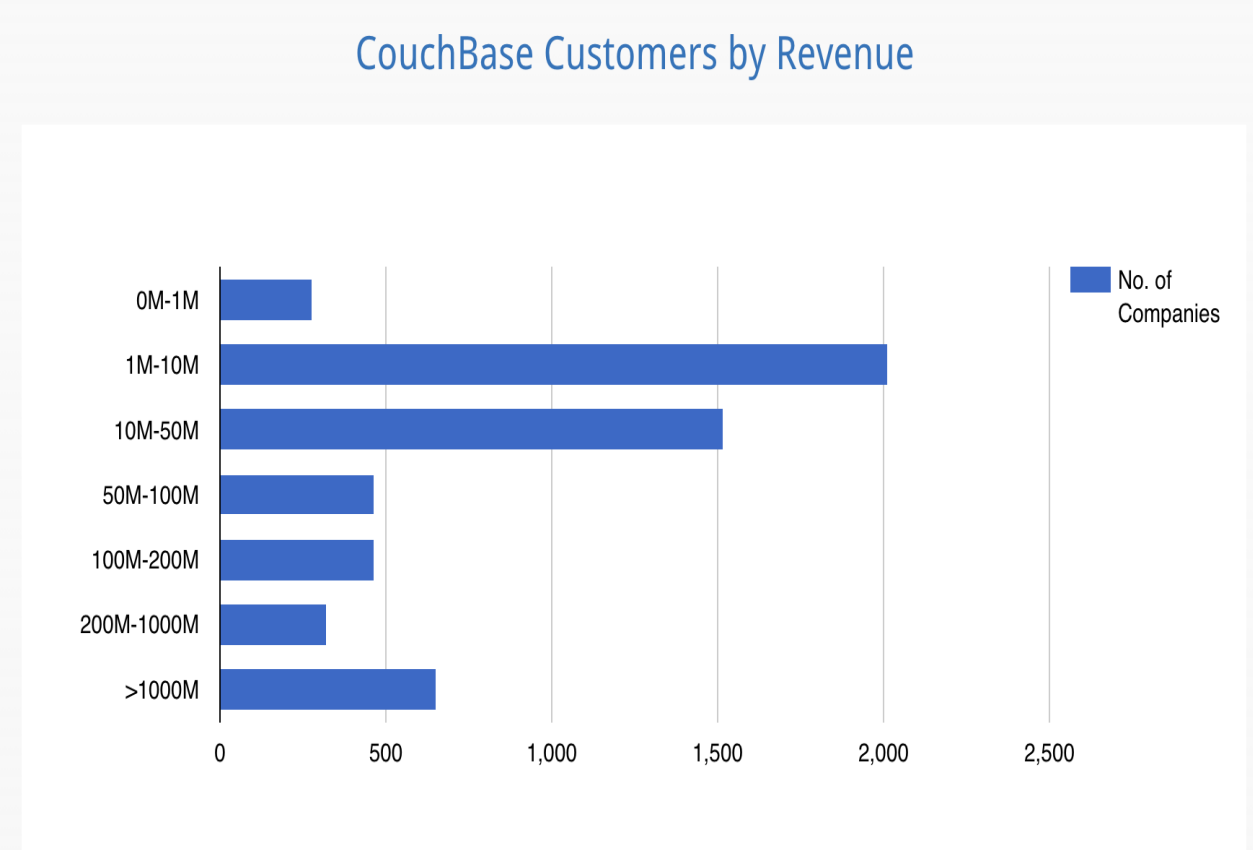
Source:
- https://www.infoclutch.com/installed-base/hrms/couchbase/#counts-by-revenue
- https://www.couchbase.com/customers

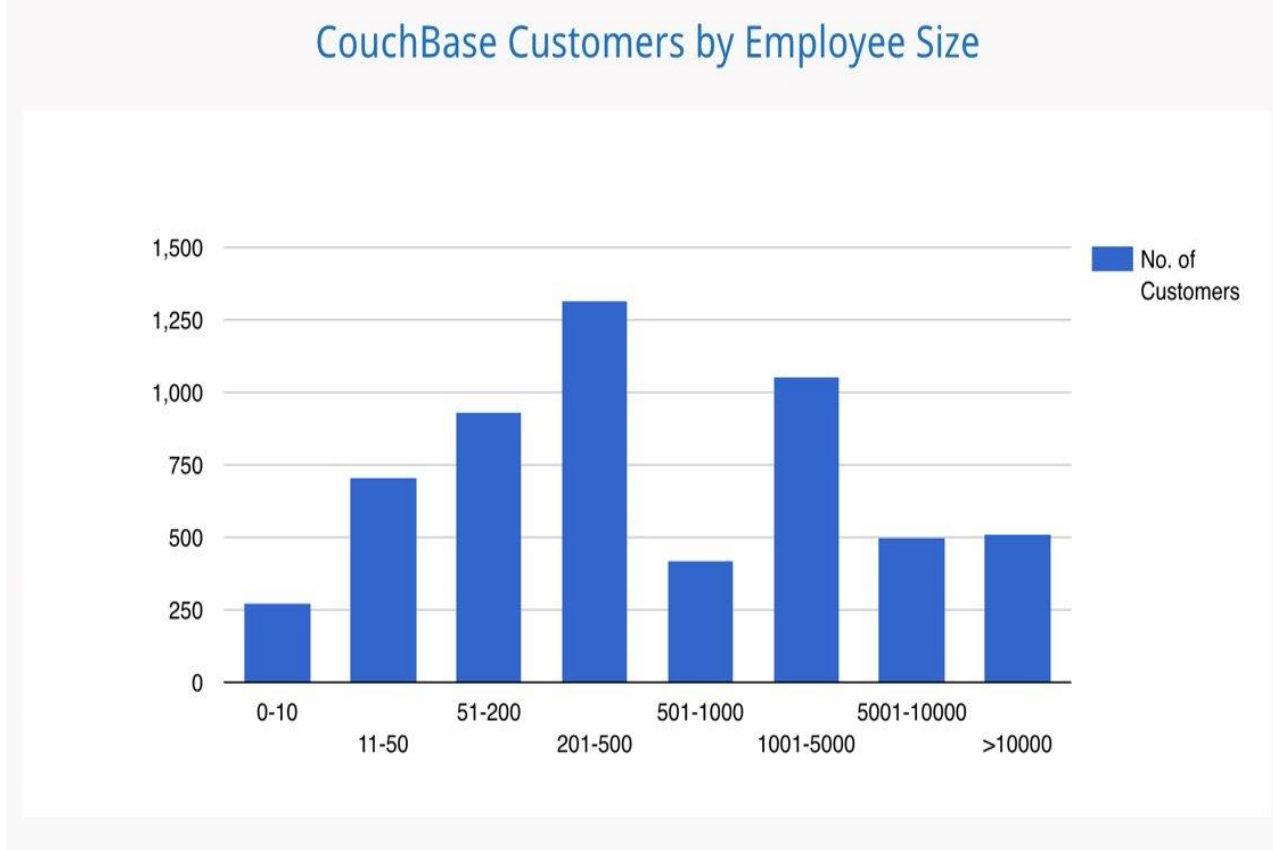
State-Wide Distribution



Source:
- https://www.infoclutch.com/installed-base/hrms/couchbase/#counts-by-revenue

Customers by Revenue and Company Size





Source:
- https://www.infoclutch.com/installed-base/hrms/couchbase/#counts-by-revenue

Use Case and Solutions

- Aggregate data collected from different sources
- Publish new product and inventory content in real time
- Ensure that data is available online and offline.
- Manage, support, and drive real-time data insights

Source:
- https://www.couchbase.com/solutions
- https://www.couchbase.com/customers

Couchbase Server

Community Version

- The open-source foundation of NoSQL data platform
- Best suited for non-enterprise developers
- Available free of charge for both development and production

Enterprise Version

- Full-featured, multi-service NoSQL database
- Available free of charge
- Common uses
  - Evaluation
  - Development
  - Unlimited preproduction testing

Sources:
- https://www.couchbase.com/products/cloud/faq
- https://www.couchbase.com/products/server
- https://www.couchbase.com/downloads

Couchbase Cloud

FEATURES

- Powerful NoSQL database
- In-house Virtual Private Cloud deployment that secures and isolates data under customers' control
- Can be powered through AWS, Microsoft Azure, and Google Cloud Platform

BENEFITS

- Control data and security
- Control clusters anywhere
- Control costs and maintain predictability

Sources:
- https://www.couchbase.com/products/cloud/faq
- https://www.couchbase.com/products/cloud
- https://www.couchbase.com/downloads

Cloud and Server Pricing

Support level	Cloud Developer Pro	Cloud Enterprise	Server and Mobile Silver	Server and Mobile Gold
Initial response time	8 hours	1 hour	5 hours	2 hours
Hours	10x5	24x7	10x5	24x7
Hours of operation	9 a.m. - 5 p.m.	24x7	9 a.m. - 5 p.m.	24x7
Entry price	$0.31/hr per node*	$0.47/hr per node*	Contact	Contact

Node Type

		CS[2]		CM[2]		S		L		T		U		X	
	Product	Couchbase Server	Couchbase Mobile	Couchbase Server	Couchbase Mobile	Couchbase Server	Couchbase Mobile	Couchbase Server	Couchbase Mobile	Couchbase Server	Couchbase Mobile	Couchbase Server	Couchbase Mobile	Couchbase Server	Couchbase Mobile
	Max Cores	4	2	8	4	16	8	24	12	32	16	48	24	64	32
	Max RAM	32GB	2GB	64GB	4GB	256GB	8GB	384GB	12GB	512GB	16GB	768GB	24GB	1,024GB	32GB
USD List Price	Silver	$3,600		$5,700		$8,900		$11,500		$14,000		$18,100		$22,100	
	Gold	$4,800		$7,600		$11,900		$15,300		$18,700		$24,100		$29,400	
	Platinum	$6,800		$10,800		$17,000		$21,800		$26,600		$34,300		$41,900	

Source:
- https://www.couchbase.com/pricing

Couchbase Licensing

- Need to purchase as many Enterprise Edition subscriptions as you have physical commodity servers
- All Servers in cluster must be at the same subscription level
- Standard duration of subscription is one year

Source:
- https://www.couchbase.com/licensing-and-support-faq

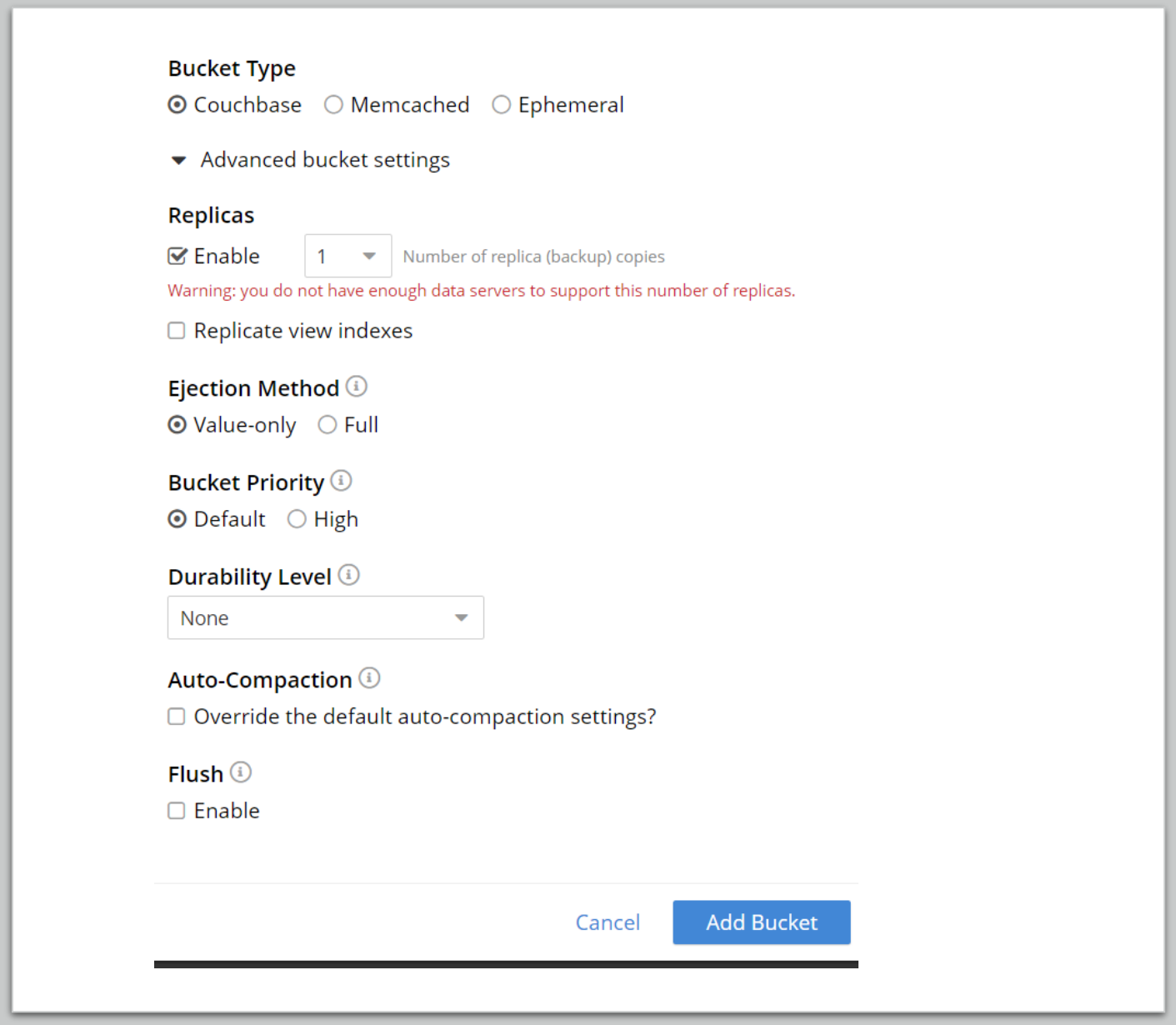
Couchbase Key Concepts

- SQL like query based on N1QL standards
- Replication based on master-master model
- Full text searching. This supports language-aware searching; allowing users to search for, say, the word *beauties*, and additionally obtain results for *beauty* and *beautiful*.
- Couchbase server 6.6 has web-based UI that supports
  - Couchbase Web Console
  - Importing/exporting data
  - Query Workbench
- Couchbase is classified as a CP server under the CAP Theorem
  - Consistency and partition tolerance (protection)
  - Data by default is eventually consistent
- ACID transactions can be performed on data and are very similar to those conducted by RDBMS
- Data is separated into virtual buckets that function as collections
- Horizontal and vertical scaling are jointly supported
  - Depends on the scaling model applied when working with data

Sources:
- https://en.wikipedia.org/wiki/Couchbase_Server
- https://blog.couchbase.com/transactions-n1ql-couchbase-distributed-nosql/#:~:text=N1QL%20Transactions%20%3A%20An%20Elastic%20and%20Scalable%20Distributed,START%20TRANSACTION%3B%20UPDATE%20customer%20SET%20b%20...%20
- https://resources.couchbase.com/nosql_comparison_web

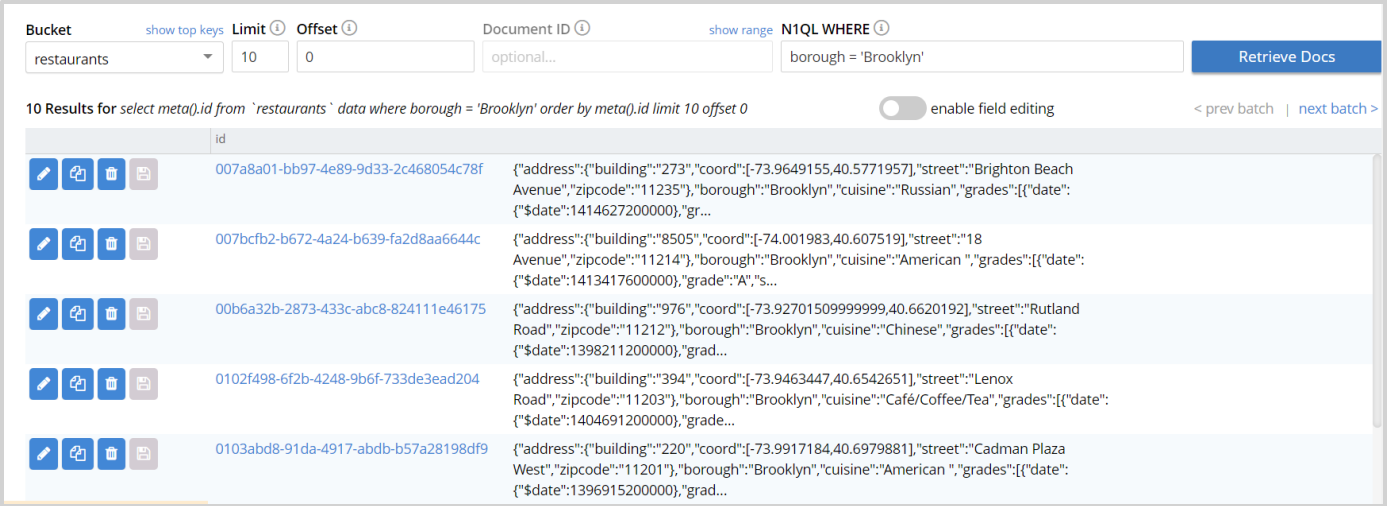
Buckets

- Bucket is a collection of documents
- Bucket types
  - Couchbase
  - Ephemeral
  - Memcached
- Specify memory quota
- Enable up to 3 replicas
- Create index to search buckets
- Import documents into bucket using JSON list, CSV



Document Editor

- Document editor built into web client
- Index must be created first to use "WHERE" clause
- Enable field editing to provide interface to update data without using JSON
- Document operations
  - Edit JSON format
  - Copy (edit new doc ID) into a new document
  - Delete



CAP Theorem and Transactions

- Outline of the CAP Theorem
  - Consistency
  - Availability
  - Protection (partition tolerance)
- Couchbase is designated as a CP server
  - May be set up as an AP system with multiple clusters
- Provides support for replication between servers
- Transaction format is very similar to RDBMS
  - N1QL applies SQL-like queries to JSON files under Couchbase

- Couchbase supports ACID transactions
  - Atomicity: All or nothing (automatic rollback after any failure)
  - Consistency: Data is valid under all constraints
    - N1QL checks for any pre-existing keys and rolls back transactions that violate any constraint
  - Isolation: Unaffected by other transactions
    - Applies optimistic concurrency control (assumes that multiple transactions can be completed without interfering with one another)
  - Durability: Data remains the same in the system

Sources:
- http://info.couchbase.com/rs/302-GJY-034/images/MongoDB_vs_Couchbase_Architecture_WP.pdf
- https://en.wikipedia.org/wiki/Couchbase_Server
- https://blog.couchbase.com/transactions-n1ql-couchbase-distributed-nosql/#:~:text=N1QL%20Transactions%20%3A%20An%20Elastic%20and%20Scalable%20Distributed,START%20TRANSACTION%3B%20UPDATE%20customer%20SET%20b%20...%20

Inserting Data and Transactions

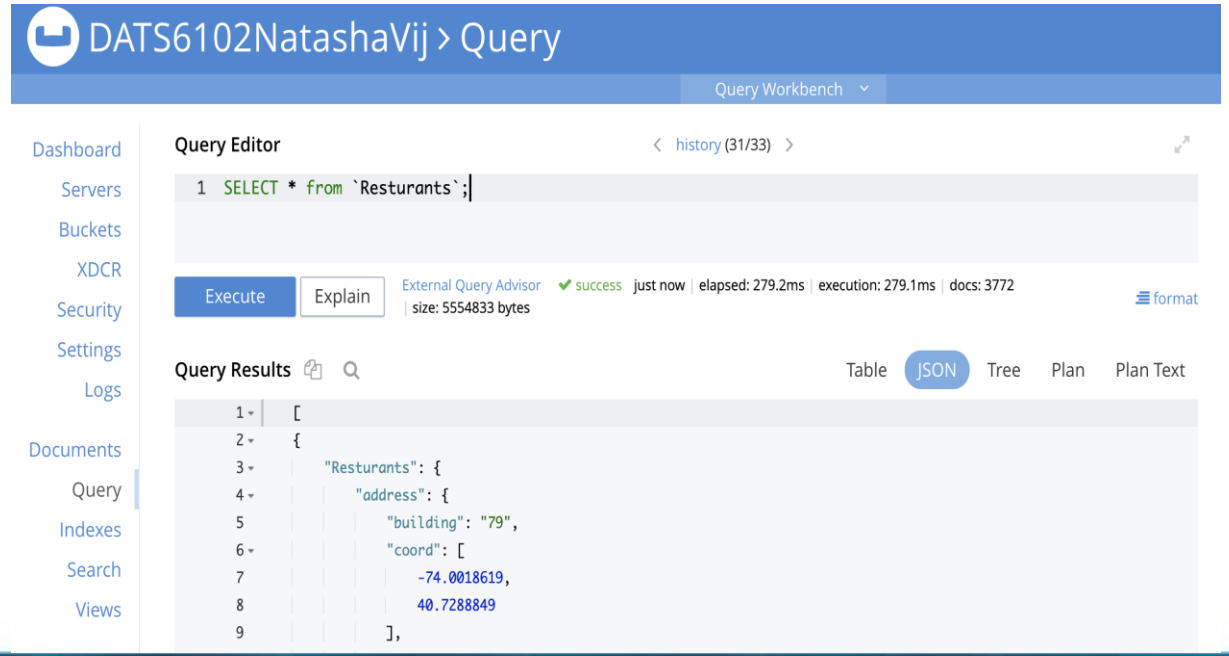
Transactions	MySQL MySQL Database (Statements are same/similar in Oracle, SQL Server, Informix & DB2)	Couchbase Couchbase Database (Cheshire-Cat)
Insert data. Tuples in MySQL, JSON documents in Couchbase	INSERT INTO customer(cid, name, balance) VALUES(4872, "John Doe", 724.23); INSERT INTO customer(cid, name, balance) VALUES(1924, "Bob Stanton", 2735.48);	INSERT INTO customer VALUES("cx4872", {"cid": 4872, "name":"John Doe", "balance":724.23}); INSERT INTO customer VALUES("cx1924", {"cid": 1924, "name":"Bob Stanton", "balance":2735.48});
Simple transaction, debit, and credit. Intermediate selects have to be read their own updates (RYOW)	START TRANSACTION; UPDATE customer SET balance = balance + 100 WHERE cid = 4872; SELECT cid, name, balance from customer; UPDATE customer SET balance = balance – 100 WHERE cid = 1924; SELECT cid, name, balance from customer; COMMIT;	START TRANSACTION; UPDATE customer SET balance = balance + 100 WHERE cid = 4872; SELECT cid, name, balance from customer; UPDATE customer SET balance = balance – 100 WHERE cid = 1924; SELECT cid, name, balance from customer; COMMIT;

	MySQL	Couchbase
The second transaction with partial rollback.	START TRANSACTION; UPDATE customer SET balance = balance + 100 WHERE cid = 4872; SELECT cid, name, balance from customer; SAVEPOINT s1; UPDATE customer SET balance = balance – 100 WHERE cid = 1924; SELECT cid, name, balance from customer; ROLLBACK WORK TO SAVEPOINT s1; SELECT cid, name, balance from customer; COMMIT;	START TRANSACTION; UPDATE customer SET balance = balance + 100 WHERE cid = 4872; SELECT cid, name, balance from customer; SAVEPOINT s1; UPDATE customer SET balance = balance – 100 WHERE cid = 1924; SELECT cid, name, balance from customer; ROLLBACK WORK TO SAVEPOINT s1; SELECT cid, name, balance from customer; COMMIT;

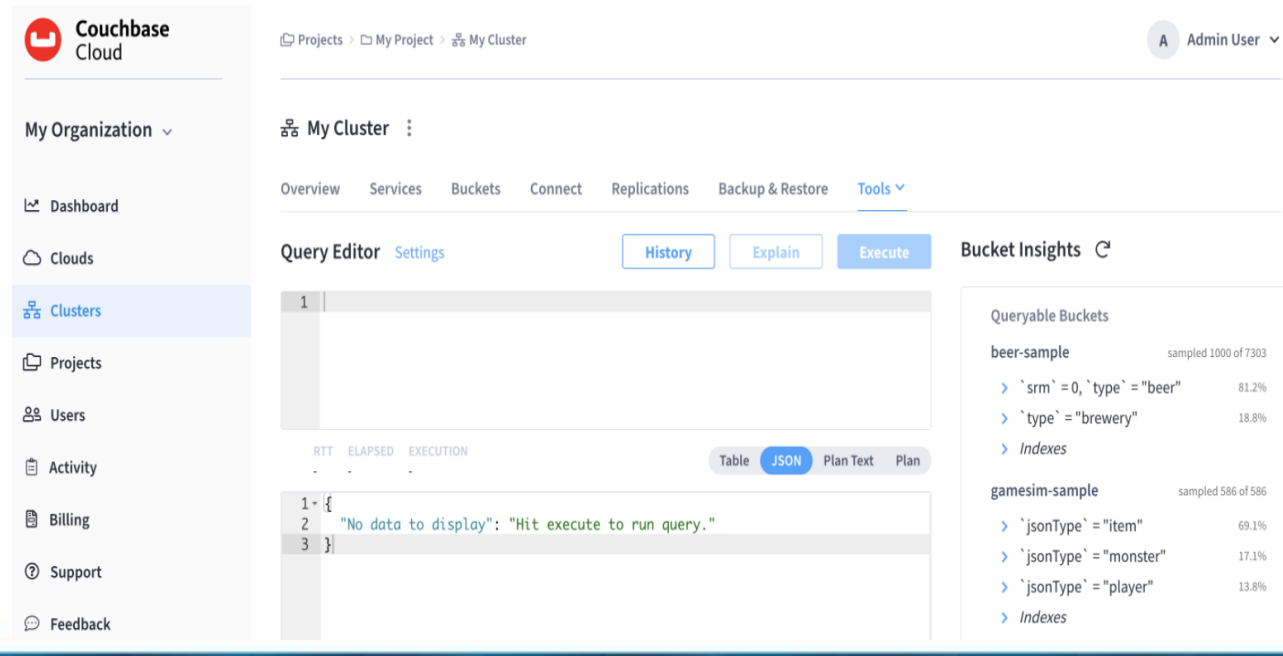
Source:
- https://blog.couchbase.com/transactions-n1ql-couchbase-distributed-nosql/#:~:text=N1QL%20Transactions%20%3A%20An%20Elastic%20and%20Scalable%20Distributed,START%20TRANSACTION%3B%20UPDATE%20customer%20SET%20b%20...%20

Couchbase Server vs. Cloud Query Workbench

Server-Based Query Workbench



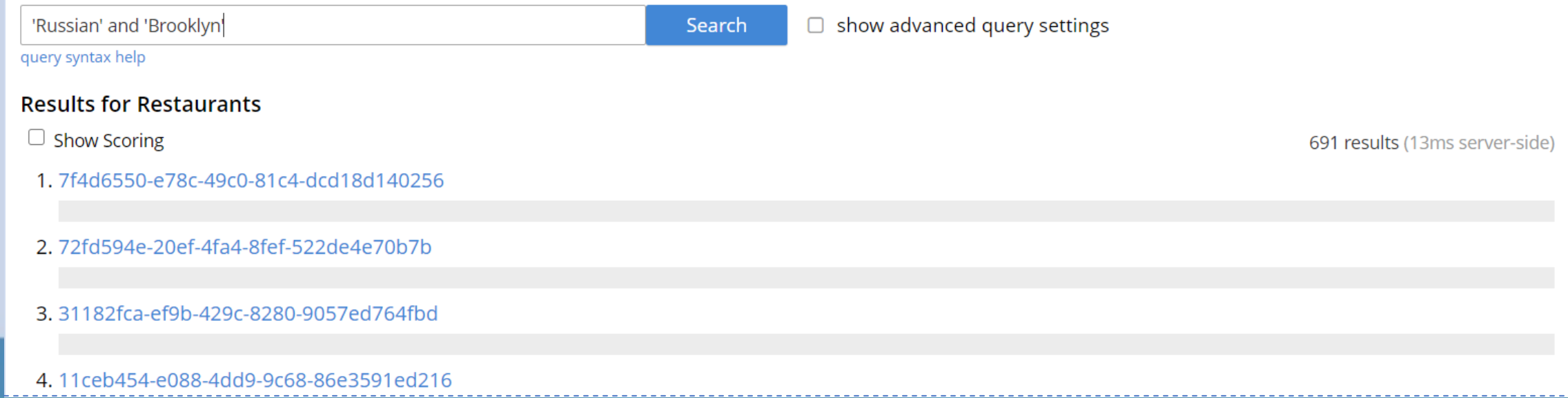
Cloud-Based Query Work Editor



Source
- https://docs.couchbase.com/cloud/clusters/query-service/query-workbench.html

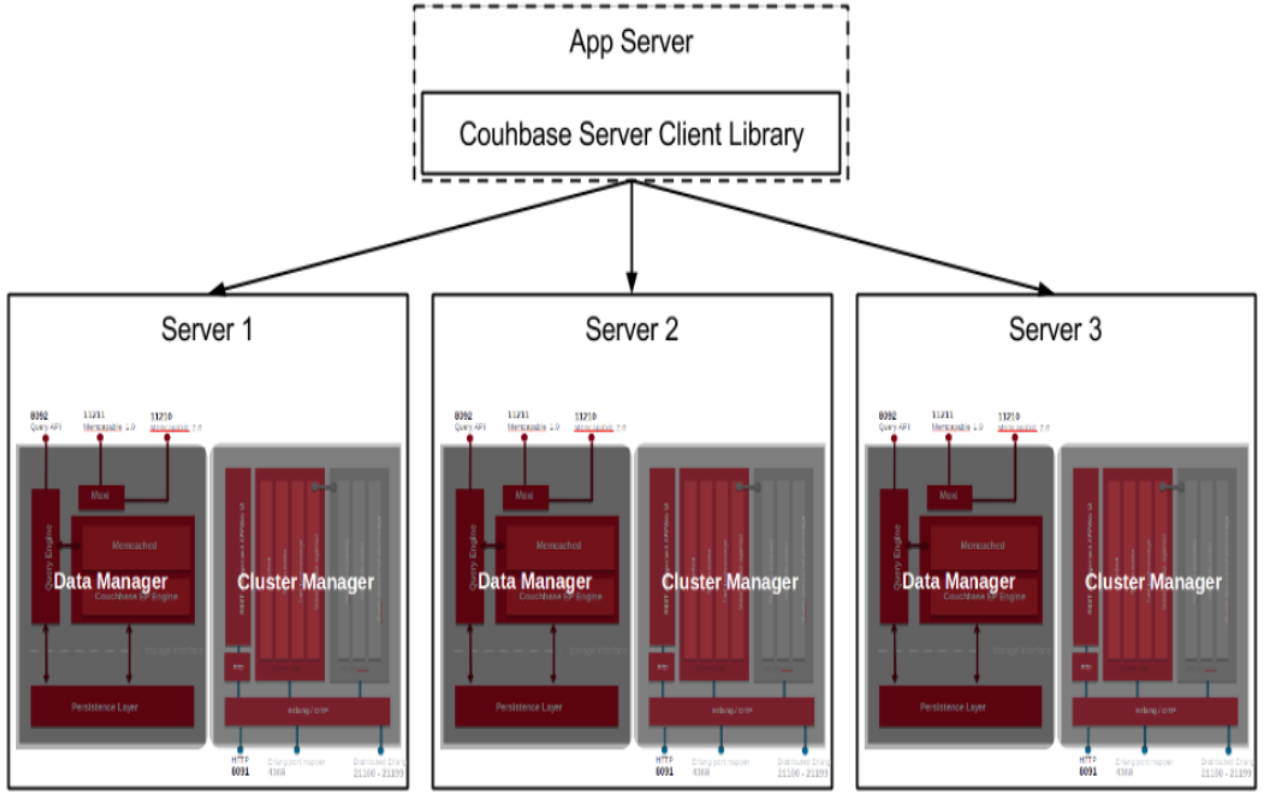
Full Text Search

- Document collection must be indexed
- Query language Google- like
- No need to know specific field names but field scoping is available.
  - +borough: Brooklyn
- Search items can be required, optional or excluded
- You can specify numeric and date ranges with >, >=, <, and <= operators



Topology and Partitioning

- Couchbase has a flat topology with a single node type
  - All nodes are equal and communicate to each other on demand
  - Nodes run processes assigned with a Data Manager and Cluster Manager
- Couchbase partitions data in 1,024 virtual buckets
  - Assigns buckets to nodes once opened
  - Buckets store all data within a given range
  - Cluster map that outlines buckets to nodes makes routers or configuration servers unnecessary



Sources:
- https://blog.couchbase.com/transactions-n1ql-couchbase-distributed-nosql/#:~:text=N1QL%20Transactions%20%3A%20An%20Elastic%20and%20Scalable%20Distributed,START%20TRANSACTION%3B%20UPDATE%20customer%20SET%20b%20...%20
- http://info.couchbase.com/rs/302-GJY-034/images/MongoDB_vs_Couchbase_Architecture_WP.pdf

Comparing Couchbase with Other Databases

Couchbase vs RDBMS

Couchbase	RDBMS
Collection stored in buckets	Tables
Key value	Rows
Query	Column
Index	Joins
Supports ACID transactions	Primary and foreign keys
No schema	Schema

Similarities with MongoDB

- Both are document-oriented databases
  - Schema-less and classified under NoSQL
- Under CAP Theorem consistency and partition tolerance are prioritized in both
- Both may apply secondary indexes
- By default both are set to be eventually consistent
  - Option is available to switch to immediate consistency
- Both can perform maintenance using CLI
- Both offer node-to-node-encryption for security purposes
- Both support ACID and multi-document transactions
- Both can scale horizontally
- Neither are solely cloud-based systems
- Foreign keys are not required in either

Sources:
- https://db-engines.com/en/system/Couchbase%3BMongoDB
- https://pediaa.com/what-is-the-difference-between-couchbase-and-mongodb/#:~:text=%20Difference%20Between%20Couchbase%20and%20MongoDB%20%201,was%20written%20in%20C%2C%20Go%2C%20JavaScript...%20More%20
- https://www.couchbase.com/nosql-database-cloud-comparison
- https://resources.couchbase.com/nosql_comparison_web
- https://blog.couchbase.com/distributed-multi-document-acid-transactions-in-couchbase/#:~:text=Couchbase%20ACID%20transactions%20are%20multi-document%20and%20multi-node%20%E2%80%93,with%202%20replicas%20%28total%203%20copies%20of%20data%29.

Differences from MongoDB

- Couchbase writes data in multiple data files in an append-only manner
- MongoDB stores data into collections using BSON
- Secondary database models
  - Couchbase: Key-value store
  - MongoDB: Search engine
- Scalability
  - Couchbase
    - Homogenous scaling model: Distributes a workload evenly for each node across a cluster
    - Multidimensional scaling model: Allows services to independently expand in order to limit competition and interference between each other
  - MongoDB
    - Master-slave replication
    - Horizontal data partitioning (sharding)

Sources:
- https://db-engines.com/en/system/Couchbase%3BMongoDB
- https://pediaa.com/what-is-the-difference-between-couchbase-and-mongodb/#:~:text=%20Difference%20Between%20Couchbase%20and%20MongoDB%20%201,was%20written%20in%20C%2C%20Go%2C%20JavaScript...%20More%20
- https://resources.couchbase.com/nosql_comparison_web
- https://www.couchbase.com/nosql-database-cloud-comparison

Homogenous Scaling versus Multidimensional Scaling

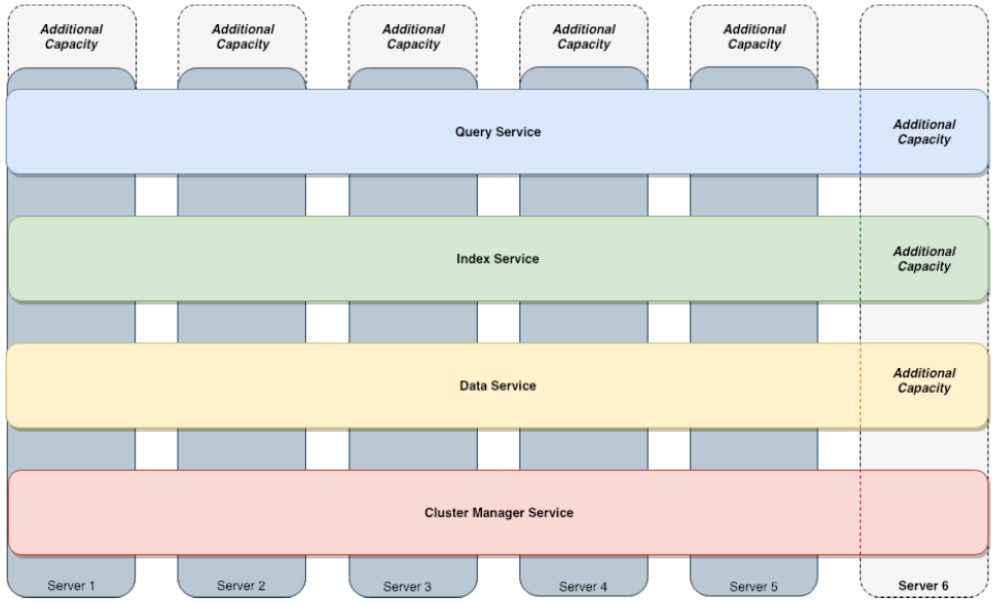


Figure 3.1.2.1 The homogeneous scaling model of Couchbase Server

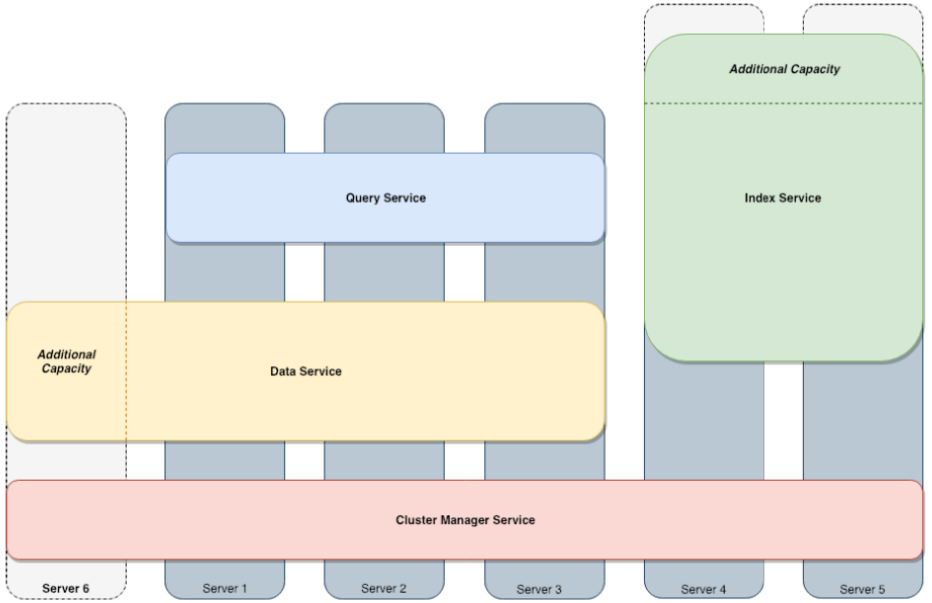


Figure 3.1.2.2 The independent scaling model of Couchbase Server

Source:
- https://resources.couchbase.com/nosql_comparison_web

Advantages

Couchbase

- Fully integrated caching layer that doesn't require other caching products to improve performance
- Multi-dimensional scaling (MDS) can be applied to isolate workloads, and can be done vertically
- Extends ANSI SQL to support JSON's flexible schema
- Accesses all functionality more easily without using a unique query language
- Adaptive index partitioning to isolate data
- Applies SQL-like querying that goes beyond read-only queries
- Nodes can read and write
- Does not require master-slave replication between servers
- Can also perform maintenance with Web UI or REST API

MongoDB

- Supports more programming languages
- Compatible with more operating systems
- Performs somewhat better with small-sized clusters and data sets

Sources:
- https://www.couchbase.com/comparing-couchbase-vs-mongodb
- https://db-engines.com/en/system/Couchbase%3BMongoDB
- https://resources.couchbase.com/nosql_comparison_web
- https://www.couchbase.com/nosql-database-cloud-comparison
- https://pediaa.com/what-is-the-difference-between-couchbase-and-mongodb/#:~:text=%20Difference%20Between%20Couchbase%20and%20MongoDB%20%201,was%20written%20in%20C%2C%20Go%2C%20JavaScript...%20More%20
- https://resources.couchbase.com/c/altoros-nosql-performance-benchmark?x=N-I_ik

References

- https://nordicapis.com/what-is-the-difference-between-an-api-and-an-sdk/
- https://www.couchbase.com/comparing-couchbase-vs-mongodb
- https://db-engines.com/en/system/Couchbase%3BMongoDB
- https://db-engines.com/en/system/CouchDB%3bCouchbase%3bMongoDB
- https://pediaa.com/what-is-the-difference-between-couchbase-and-mongodb/#:~:text=%20Difference%20Between%20Couchbase%20and%20MongoDB%20%201,was%20written%20in%20C%2C%20Go%2C%20JavaScript...%20More%20
- https://www.couchbase.com/nosql-database-cloud-comparison
- https://docs.couchbase.com/server/current/n1ql/n1ql-language-reference/join.html
- https://docs.couchbase.com/server/6.5/fts/query-string-queries.html
- https://docs.couchbase.com/server/current/learn/services-and-indexes/indexes/indexing-and-query-perf.html
- https://docs.couchbase.com/server/current/manage/manage-buckets/create-bucket.html
- https://www.linkedin.com/company/couchbase/

References Continued

- https://en.wikipedia.org/wiki/Couchbase_Server
- https://resources.couchbase.com/nosql_comparison_web
- https://resources.couchbase.com/c/altoros-nosql-performance-benchmark?x=N-I_ik
- https://dzone.com/articles/how-acid-mongodb
- https://www.infoclutch.com/installed-base/hrms/couchbase/#counts-by-revenue
- https://www.couchbase.com/products/cloud
- https://www.couchbase.com/downloads
- https://www.couchbase.com/products/cloud/faq
- https://www.couchbase.com/licensing-and-support-faq
- https://blog.couchbase.com/transactions-n1ql-couchbase-distributed-nosql/#:~:text=N1QL%20Transactions%20%3A%20An%20Elastic%20and%20Scalable%20Distributed,START%20TRANSACTION%3B%20UPDATE%20customer%20SET%20b%20...%20
- http://info.couchbase.com/rs/302-GJY-034/images/MongoDB_vs_Couchbase_Architecture_WP.pdf
- https://www.couchbase.com/customers