

Technical Overview: Anatomy of the Cloudbant DBaaS

Guaranteed Data Layer Performance, Scalability, and Availability

The End of Scale-It-Yourself Databases?

Today's applications are expected to manage a variety of structured and unstructured data, accessed by massive networks of users, devices, and business locations, or even sensors, vehicles and Internet-enabled goods.

Deciding which data management platform will provide the best performance, up-time, and economics takes a lot of research and planning. Developers ask themselves:

- What's the right DBMS?
- What's the right hardware?
- What's the right database design?
- Should we hire new DB administrators?

Many teams lack the time and expertise to research the plethora of database software, hardware, and design possibilities that are available on the market today. So they often choose to live with the status quo. As the database workload continues to grow, they lose precious development time in order to handle database crises, and eventually run the risk of losing up time, losing customers, losing data, or losing revenue and profit.

Next Wave: Guaranteed Performance from DBaaS Providers

Database as a Service (DBaaS) solutions from Cloudant and Amazon AWS enable you to buy into a guaranteed data management "Service Level Agreement" (SLA) instead of merely buying into a database technology.

This greatly simplifies application development and delivery:

Step 1: Know what SLA you require

- Storage? Throughput? Latency? Up-time? Data access methods? Support? etc.

Step 2: Partner with a DBaaS that guarantees delivery of it

- At the best price, with no lock in

This paper describes what the Cloudant DBaaS is, what makes it unique, how it's used, and the types of SLAs that Cloudant guarantees for its customers.

**Rosetta
Stone**



"Fast development cycles keep our users happy. Wrangling our database makes us unhappy. With Cloudant, we each do what we do best. Our developers can iterate quickly, while Cloudant engineers keep our database humming along."

Matthew Von Bargaen, Software Development Manager, RosettaStone, Inc.

Cloudant Performance Profile

Companies of all sizes use Cloudant to manage data for many types of large or fast-growing web and mobile apps in ecommerce, on-line education, gaming, financial services, networking, and other industries. Cloudant is best suited for applications that require an operational data store to handle a massively concurrent mix of low-latency reads and writes. Its data replication & synchronization technology also enables continuous data availability, as well as off-line app usage for mobile or remote users.

As a JSON document store, Cloudant is ideal for managing multi- or un-structured data. Advanced indexing makes it easy to enrich applications with location-based (geo-spatial) services, full-text search, and real-time analytics.

As explained later in the paper, a Cloudant account can be hosted within a multi-tenant Cloudant cluster, or on a single-tenant cluster running on dedicated hardware hosted within a top-tier cloud provider like Rackspace or IBM SoftLayer.

The following tables provide real-life performance data characteristics for actual customers:

Large Single-Tenant Customer:

Read:Write Ratio	20 : 1
Data Volume	130 TB
Transactional Throughput	Over 2 billion requests per day
Latency	Under 10 milliseconds
User base	Global (mobile app)
Cluster growth	From 6 servers to over 200 in 12 months

Note: read:write ratios vary from customer to customer. Cloudant handles mixed workloads that are read heavy or write heavy or evenly balanced.

Large Multi-Tenant Customer:

Read:Write Ratio	3 : 1
Data Volume	215 GB
Transactional Throughput	1 million requests per day (42,000 per hour)

These can be used as guidelines to help determine whether Cloudant can fulfill your data layer performance requirements. We can keep growing Single-tenant clusters to scale up performance.

What we don't do:

Cloudant is an operational data store. It is not meant to handle ad-hoc query intensive workloads like those seen in data warehousing applications. It is also accessed via a web-based RESTful interface, therefore its data access latency can be in the very low milliseconds. If you need micro-second response time, consider using an in-memory database.

Cloudant – The Industry’s First Global Data Delivery Network

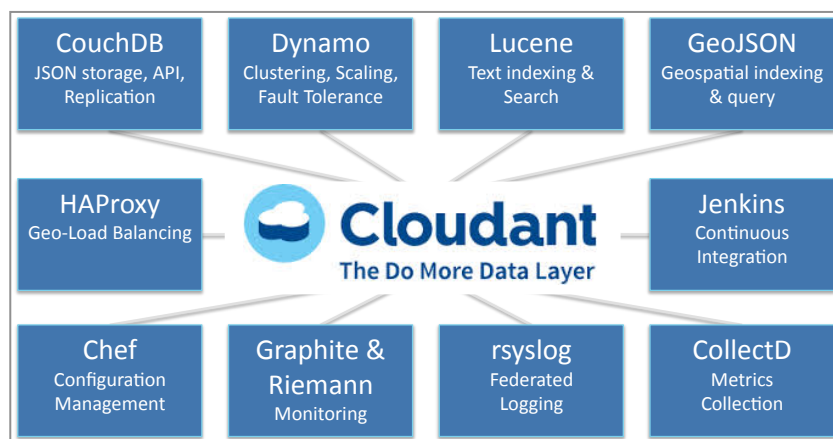
Cloudant is the first data management platform to leverage the availability, elasticity, **and reach** of the cloud to create a global data delivery network (DDN) that enables applications to scale larger and remain available to users wherever they are.

- Stores data of any structure as self-describing JSON documents.
- Distributes readable, writable copies of data to multiple locations or devices.
- Users read and write the closest available data source.
- Synchronizes data continuously via filtered, multi-master replication.
- Integrates via a RESTful API.
- Provides full-text search, geo-location services, and flexible, real-time indexing.
- Is monitored and managed for you 24x7 by the big data experts at Cloudant.



Cloudant’s DNA

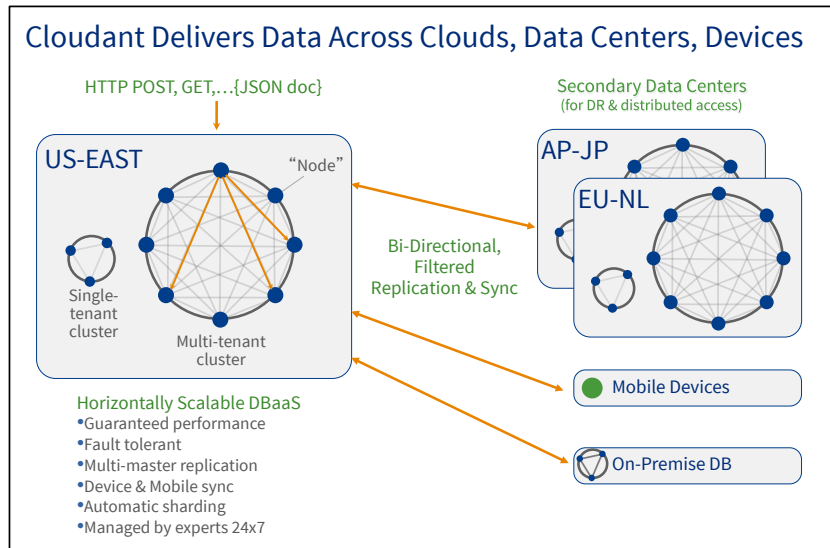
Cloudant was born and raised in the cloud era. It was built to be elastic, highly available, and to manage popular web and mobile data types like JSON, full-text, and geospatial data. Cloudant was created by combining the best open source code and thought leadership into an innovative DBaaS that powers some of the biggest web and mobile apps in the world. Cloudant regularly contributes code back to these open source projects as well.



How the Cloudbant Data Layer Works

The Cloudbant Data Layer is massive; your databases grow within it, not out of it, and we continually monitor and grow Cloudbant to meet our customers' collective storage and performance demands.

Application data resides in logical databases that you create. Cloudbant physically stores the data on server **Nodes**, which are joined into multi-tenant or single-tenant **Clusters**. Our clusters are spread across a **Global Network** of data centers.



Cloudbant automatically handles load balancing, clustering, backup, growing/shrinking the clusters, distributed query execution, and high availability. As your database and user load grows, Cloudbant scales the data layer for you to keep up with the workload.

Getting started and using Cloudbant is very straightforward:

1. Create a Cloudbant DBaaS Account

Cloudbant saves you the trouble of choosing and provisioning hardware, installing and configuring DBMS software and load balancers. You simply sign up for a free account at Cloudbant.com, and your data layer is ready. You're provisioned a custom URL against which you can immediately begin creating databases and creating, reading, updating, deleting, and indexing data.

When you activate your Cloudbant account, you choose a physical platform (such as Rackspace, IBM SoftLayer, AWS, Azure, Joyent, etc.) and geographic zone (e.g., US, Europe, Asia, etc.) as the primary location for your data layer. Typically, the choice is based on where your application code is hosted in order to get the benefits of app and data layer co-location. You can move your data layer from one hosting provider or location to another at any time.

Creating Multi-Tenant versus Single-Tenant Accounts

By default, your data layer will reside on a multi-tenant cluster that is also used by other multi-tenant account holders. Databases within multi-tenant clusters are secure and not accessible by other Cloudant users unless you explicitly set sharing permissions with other users to enable team development, etc. Cloudant includes special IO elevators that help ensure your application performance isn't impacted by "noisy neighbors."

For those that prefer dedicated data layer resources, Cloudant can set up single-tenant clusters on dedicated hardware that can span multiple data centers or even cloud providers to guarantee that your unique performance and availability requirements are met.

2. Store JSON Data

Cloudant is designed to store, index and report on flat collections of self-describing JSON documents. If you are familiar with relational SQL databases, then JSON documents are analogous to rows, and the fields within them are analogous to columns.

Database: product-catalog

<pre>{ "_id": "Thunder-Muscle-Case", "productName": "Thunder Muscle", "productPrice": "28.37", "productUnit": "Case of 24 cans", "manufacturer": { "name": "Global National" } }</pre>	<pre>{ "_id": "Thunder-Muscle-Can", "productName": "Thunder Muscle", "productPrice": "1.50", "productUnit": "One 20 ounce can", "productCategory": "Energy Drink" "manufacturer": { "name": "Global National", "contact": "Todd Margaret" } }</pre>
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Cloudant assigns each JSON document a unique identifier (`_id`). Document fields can be numbers, strings, Booleans, objects (e.g., "manufacturer"), arrays (e.g., "tags"), dates, attachments (video, images, or any other file type), etc. There is no limit to the number of elements or to the size of a document.

The fields in the JSON docs define the schema for your Cloudant databases. You can safely add new document types to a database or change the content structure of existing documents as needed without having to migrate data as you would with a SQL database. This schema flexibility makes Cloudant an excellent fit for data such as product catalogs, experiment results, content management information and others that don't fit particularly well within tabular, relational schemas. You can index and query on any field in any JSON document in your database.

Documents are accessed via HTTP using a RESTful web services API. Database access will be explained in more detail later in this white paper.

3. Index Your Data

Chainable, Incremental MapReduce

Cloudant includes several ways to index and query your data. The primary way is to create special “design documents” in the database that contain MapReduce functions written in Javascript. The Map function defines which JSON documents belong in the index and which fields to include from each member JSON doc. The optional Reduce function defines an aggregate operation to perform against the data in the index such as sums, averages, counts, etc. Every design document has a unique URL, and accessing it from the client app returns the data in the index (or some subset you define using an optional query parameter in the URL) to the client as JSON.

MapReduce indexes are updated incrementally. Cloudant distributes the MapReduce functions to all the nodes in your database. Changes to JSON documents trigger incremental index updates rather than having to rebuild the whole index. This reduces IO overhead for the Cloudant Data Layer, and also enables you to perform faster, more real-time analyses of your data.

To further improve performance and make development easier, Cloudant removes the restriction that there can be only one map phase and one reduce phase. “Chainable” MapReduce queries allow the output of one MapReduce job to be persisted and fed to a chain of subsequent MapReduce jobs. This makes it easier and faster to create more advanced analytics.

Lucene Indexing and Full-text Search

In order to handle full-text search and provide more flexible ad-hoc querying of JSON data, Cloudant integrated Apache Lucene text indexing and search into Cloudant’s storage and scalability framework. This enables you to enrich indexing and querying of your JSON data with:

- Ranked searching – Search results can be ordered by relevance or by custom sort fields
- Powerful query types – Including phrase queries, wildcard queries, proximity queries, fuzzy searches, range queries and more
- Language-specific analyzers
- Faceted search and filtering
- Bookmarking – Paginate results in the style of Web search engines

GeoJSON-based 2D & 3D Geospatial Indexing and Querying

GeoJSON support was also integrated into Cloudant to enable the development of high-precision location-based services, which are a common requirement for mobile and sensor network applications. Cloudant allows you to:

- Build queries using bounding polygons, rectangles, circles, or ellipses,
- Perform nearest neighbor and predictive path analyses.
- Store a wide range of data types, including complex geometries and metadata.
- Use proven CRS libraries since Cloudant supports any coordinate reference system (CRS)

4. Access Your Data via a RESTful JSON API

Cloudant has a RESTful API that makes every document in your Cloudant database accessible as JSON via a URL; this is one of the features that makes Cloudant so powerful for web and mobile applications.

JSON documents can be retrieved, stored, or deleted individually or in bulk using GET, PUT, POST, DELETE statements over HTTP(s). Cloudant always encrypts data in flight. In Cloudant, JSON documents can have files of any type attached to them, which makes it easy to manage media, or even code.

Indexes stored in Cloudant as “design” documents are also accessible via HTTP GET calls and return JSON data. Here is an example of a RESTful Cloudant query that accesses a design document called “product-catalog.” The optional [Lucene] query string limits the returned JSON docs to the first 10 JSON docs that have a category field containing the value “electronics” and a brand field containing “XYZCorp,” and a price field with a value between 100 and 200, inclusive:

```
https://[username].cloudant.com/productsdb/_design/views101/_search/product-catalog?q=category:electronics AND brand:XYZCorp AND price:[100 TO 500]&limit=10
```

The Cloudant API is compatible with Apache CouchDB. Therefore you can access Cloudant from CouchDB-compatible client libraries, tools, and frameworks available for all major development platforms.

5. Replicate Your Data

Multi-master replication enables Cloudant to distribute readable—and writeable—replicas of your data across multiple data centers, devices, and cloud providers and keep changes to them synchronized. This increases up time and reduces access latency by connecting users to the closest copy of the data. It does for your database data what a content delivery network (CDN) does for your static content. Compare this to the “master-slave” replication architecture in MongoDB and most relational databases, where only one replica can be updated (bottleneck); all others are read-only.

Cloudant customers use replication to:

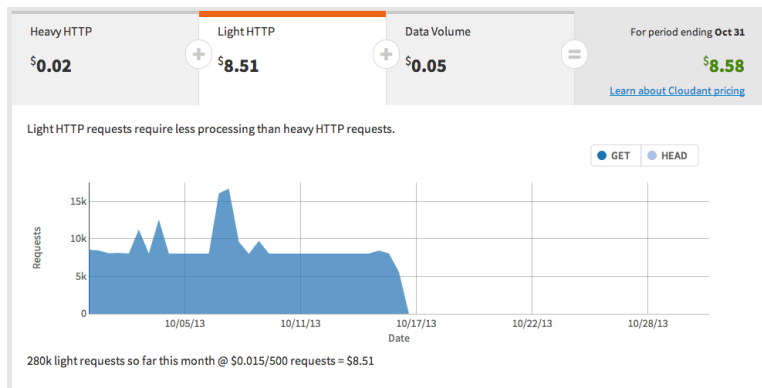
- Enable mobile data synchronization
- Push data to “Edge” databases – such as data marts or spreadsheets; making it great for independent analytic projects
- Enable off-line computing via hub-and-spoke databases
- Keep copies of their Cloudant databases on-premise; your data is never locked in to Cloudant.

6. Monitor and Visualize

In order for Cloudant and its customers to monitor data layer performance, we have defined APIs and dashboards for collecting and reporting system metrics.

Cloudant generates 50,000 metrics per second. You don't need to be an expert at interpreting this information because we do it for you to help keep your database running and growing smoothly.

If you want to view these metrics yourself, you can access your account metrics via the Cloudant dashboard and through APIs.



Scaling Cloudant

Dynamo-Style Horizontal Clustering Framework

Cloudant found its inspiration for clustering in the Amazon white paper about Dynamo database clustering. Cloudant implemented its own version of Dynamo's master-less, quorum-based clustering in Erlang (a language designed specifically for creating massively parallel applications), and it handles:

- Cluster membership
- Routing and coordination of database interactions
- Coordination of distributed queries
- Remote procedure calls to get better performance and better resilience to node failures running jobs on remote nodes.

Tune-able Eventual Consistency:

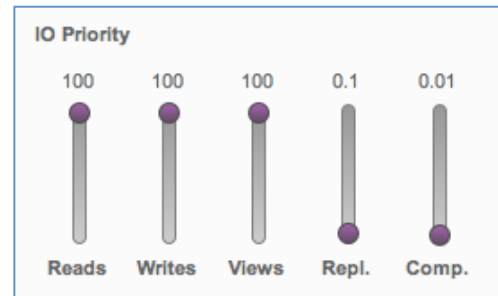
Quorum-based clustering enables you to specify the number of copies of data that you want to store (for high availability), and how many of them must be written to disk or how many must match in order for Cloudant to consider the data safely written or safely consistent. These quorum values are known as N, W, and R.

For example, on a 3-node cluster, you could decide to store N=3 copies of the data, once copy per node. W=2 would mean Cloudant would tell you the data is safely committed after 2 copies have been written. R=2 would mean that Cloudant would consider a copy of read data to be consistent with others if at least 2 copies match.

The quorum values can be changed in order to tune performance and consistency in a partitioned environment (i.e., CAP theorem). By default, Cloudant optimizes for availability.

The IO Queue (IOQ)

Cloudant handles billions of database interactions every day for thousands of databases. The IOQ is a sophisticated IO prioritization layer that analyzes the priority of each database IO request to ensure that all clients get their fair share of the IO layer. It makes sure low-latency requests get higher priority, while also ensuring that performance is good for lower priority IOs such as database compaction.



Getting Started with Cloudant

You can visit <http://cloudant.com> to create a free Cloudant Data Layer account, and the Cloudant dashboards provide tools that make it easy to create and load databases with JSON documents.

Get help with Cloudant via support@cloudant.com or via IRC: #Cloudant