Cloud CMS and MongoDB

Enterprise Content Management in the Cloud

Cloud CMS makes it easy to publish fresh content and engaging digital experiences to mobile and web applications.

Founded on the belief that successful customer experiences need to span an increasing number of devices and form factors, over long periods of time, Cloud CMS brings your brand to life by enabling you to manage, deliver and source meaningful content to your audience at just the right time. Cloud CMS provides your business with an intelligent content backbone that supplies the infrastructure and authoring tools to source, curate, collaborate and respond with crisp content for an always-connected world.

Cloud CMS uses MongoDB as the foundation for its real-time, multi-tenant content services, cross-cluster distributed caching and elastic job queue. MongoDB stores all customer-defined content types and instances and is further utilized to store binary files within GridFS for complete sharding of the backend. MongoDB provides the Cloud CMS platform with the high performance and reliability the product requires.

The Problem

As a cloud platform, Cloud CMS handles a large amount of real-time requests for content creation, search, query and retrieval. The platform must provide a consistent, transactional view of content at all times while remaining flexible and dynamic for end user content needs. It works with a wide range of content – from arbitrary JSON to images, media files, desktop and office documents, PDFs, and custom content types.

The system manages role-based permissions, validation logic, auditing records and a content services including mimetype transformation, workflow for content lifecycle management and data warehousing for analytics collection and reporting. Users are able to contribute data via a free and public API. As a result, data is constantly being created, updated, refreshed and delivered to mobile and web applications worldwide.

Cloud CMS was initially designed with CouchDB in mind. But the team quickly hit design and performance roadblocks. Fundamentally, with a growing number of social systems and an exploding number of possible content schemas for all of the different kinds of front-end applications you might imagine, Couch DB proved to be far too rigid. The fixed-schema nature of the product would have required us to essentially start over with a new indexing and view design for each new customer that signed up for a tenant subscription. Cloud CMS began an extensive evaluation of other non-relational, JSON document-oriented database options. MongoDB offered the high performance, flexible schema design and indexing control that the company required, coupled with a long runway for scale-out and a fast, easy solution for storing, locating and retrieving data.

Why MongoDB?

Compared to other non-relational databases, MongoDB proved to be faster, more reliable and easier to implement. It also provided a greater number of higher quality software drivers. These drivers enabled our developers to lower the learning curve and focus more quickly on introducing MongoDB into the Cloud CMS product. The team also appreciated that MongoDB had an active and invigorated developer community that contributed driver features, bug fixes, code samples and best practices.

MongoDB now provides storage, query and retrieval for all customer content including structured JSON, images, audio and video files. Customers are free to upload any kind of content they like. They are also at liberty to define their own custom schemas on the fly. This includes custom types, properties, validation callback logic and server-side aspects and behaviors. The behavior of the server can be adjusted on a tenant-by-tenant basis by simply uploading server-side script files (JS). No coding, no deployment and no server restarts.

All of the software that touches MongoDB is written in Java, and runs on top of Amazon EC2. It utilizes EC2 Auto Discovery for dynamic network configuration as new servers come online and old servers go offline. This allows Cloud CMS customers to only ever pay for the server capacity they use. Cloud CMS integrates Hazelcast with MongoDB to provide a fully cross-cluster, persisted grid of content objects, OAuth2 authentication tokens, and more. Cloud CMS also stores and indexes JSON properties into ElasticSearch to provide real-time full-text search for all customer content.

High Performance

MongoDB is the only document-oriented database that focuses explicitly on high performance data storage and retrieval. The Cloud CMS team evaluated other products and felt that many were focused on other, non-data problems. CouchDB, for example, offers nice document replication, but the Cloud CMS product has no need for this as it already offers distributed changeset versioning (similar to Git) in the application tier.

In addition, Cloud CMS provides transactional commits and replication of data sets between the cloud and on-premise installations. The team feels that MongoDB strikes the right balance and by focusing on the data, gives us the guarantee that we're getting the purely high performance data operations.

Adaptable

MongoDB is a schema-less database by design in that it doesn't impose any mandatory structure to your JSON. As such, Cloud CMS customers can upload their existing content with ease. In addition, Cloud CMS nicely layers JSON Schema support on top so that customers can define types, properties, validation logic, integrity checking and server-side behaviors as they wish.

Cloud Ready

MongoDB provides a well thought-out, forward thinking architecture for scaling out and delivering high performance. The developers are able to work locally with a single MongoDB instance and deploy the very same code without any changes to multi-instance test and production cloud environments. These environments utilize sharding and replica sets for scale-out and failover respectively.

Developer Friendly

"The most impressionable thing for me was the engineering focus at MongoDB and their desire to produce a high performance product," said Cloud CMS CTO and Founder Michael Uzquiano. "Everyone at MongoDB is focused on making sure the tires are on the road. Even Eliot [Horowitz, MongoDB CTO and co-founder] contributes code to the MongoDB Java Driver on GitHub and posts to the community forums. That's very authentic and very inspiring to us as a business."

Results

When Cloud CMS moved to MongoDB, they saw performance increase by a factor of 10 and were able to offer a customer-friendly content management platform that imposed no migration barriers for customers with existing content. MongoDB reduced code by 50% and sped up fully authenticated and audited data retrieval from approximately 20ms to 2ms on average.

Said Michael: "It's been a pleasure to work with MongoDB. We're able to offer the feature set that we want and find ourselves operating today much leaner. We have less code to maintain and don't have to spend our time worrying out database architecture and scale out. We can spend more of our time focused on our own customers and their success."

Highlights

⇒ Migrating from CouchDB to MongoDB reduced code by 50% and improved performance by a factor of 10x.

⇒ Ported all customer data in a few hours with zero downtime.

⇒ Shard and replica-set architecture minimizes # of servers, reduces cost and keeps costs low for multitenant customers