Couchbase

Building Content-Driven Apps

Couchbase and Elasticsearch for content and metadata stores

Interactive content-driven apps need to store and share different kinds of content and metadata. Whether you're designing an online publication, a product catalog, a website landing page, or other digital content, it's likely you want to support rich, personalized content for millions of users worldwide.

To handle the needs of these types of applications, you'll want to consider a NoSQL database. In this paper, you'll learn what makes Couchbase Server a good fit for apps that store and serve content and metadata.

What content-driven apps need

Your database is critical for your app because it is a central store for data and metadata. That means your app needs a database that lets you:

- **Store unstructured content and metadata**. The vast majority of data generated today is unstructured, with a variety of new and changing data types. So, if you're building a content catalog, you may store tens of millions of different objects in a variety of forms – unstructured content and metadata that may be hierarchical, sparse, free-form text, or varying length. Keeping your app fresh means being able to add new types of data quickly, without dealing with the complexity and time it takes to change a schema. Storing content in a RDBMS requires knowing ahead of time the exact structure of the data your app needs to store, a requirement that is hard to meet for content and metadata apps that must support new data types as they emerge.

- **Easily scale to millions of users.** Content and metadata stores have workloads that are often spiky and unpredictable. If some of your content suddenly goes viral, your database needs to scale out rapidly without app changes or downtime. Plus, your app data and workload should be evenly distributed across the database cluster to avoid hotspots. Most content consumers today are mobile, and web apps can be consumed from practically anywhere in the world. You want to scale quickly across geographies, devices, and platforms.

- **Deliver high-performance apps.** It's no longer enough to serve static content on the web. A typical content-driven app has a mix of static and dynamic content. Static content includes images and text while dynamic content includes elements that are app generated, transformed and personalized for each viewer. At the database level, this means you need consistent low-latency for metadata access and high-throughput for content and stat updates.

- **Data availability, all the time.** With smart devices, people are consuming content everywhere, all the time. Relational databases often struggle with round-the-clock operations, needing offline maintenance windows. You need a database that keeps data available (and your app running) 24x365 with no downtime – not for hardware, software, or anything.

Architecture for content store apps

Typically, as shown in Figure 1, the architecture for content-driven apps calls for:

- a database like Couchbase Server, for storing frequently accessed content and metadata
- a full-text search engine like Elasticsearch for full-text search capabilities across JSON documents and
- an external storage systems like Amazon S3 or content delivery network (CDN) for storing larger content like audio and video files.

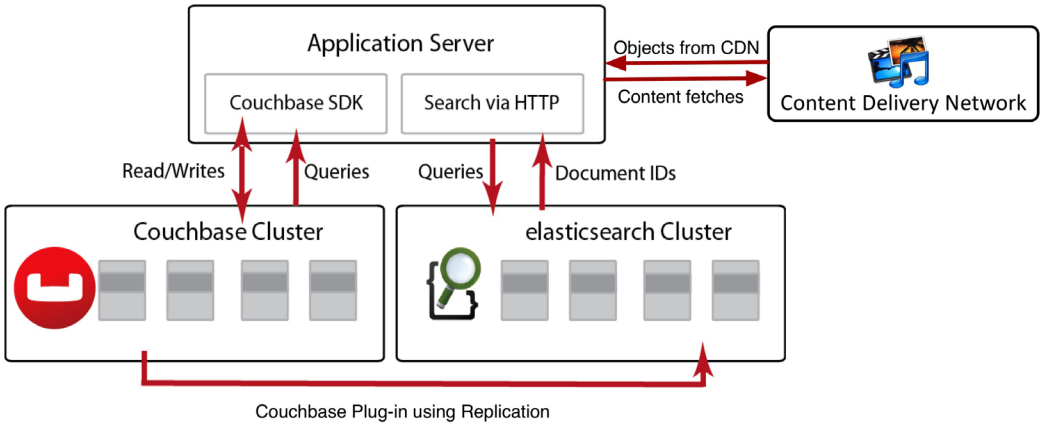

Figure 1: Content store reference architecture

Couchbase makes an ideal home for content and metadata by providing the following: easy scalability, consistent high performance, flexible data model, and always-on 24x365 characteristics. And here's how -

Easy scalability

An internally-used application may have hundreds or thousands of users, but a content store app with social integration may grow to attract millions of users in a matter of days. To manage that unpredictable growth, your database needs to scale quickly, on demand, and without any app changes.

Scaling with Couchbase Server is easy, both within a cluster and across clusters in multiple datacenters. As shown in Figure 2, with one click of a button and zero changes to your app, you can grow your clusters from 1 to 25 to 100 servers. Every server in the cluster is architecturally identical. This design scales out linearly without any single point of failure.

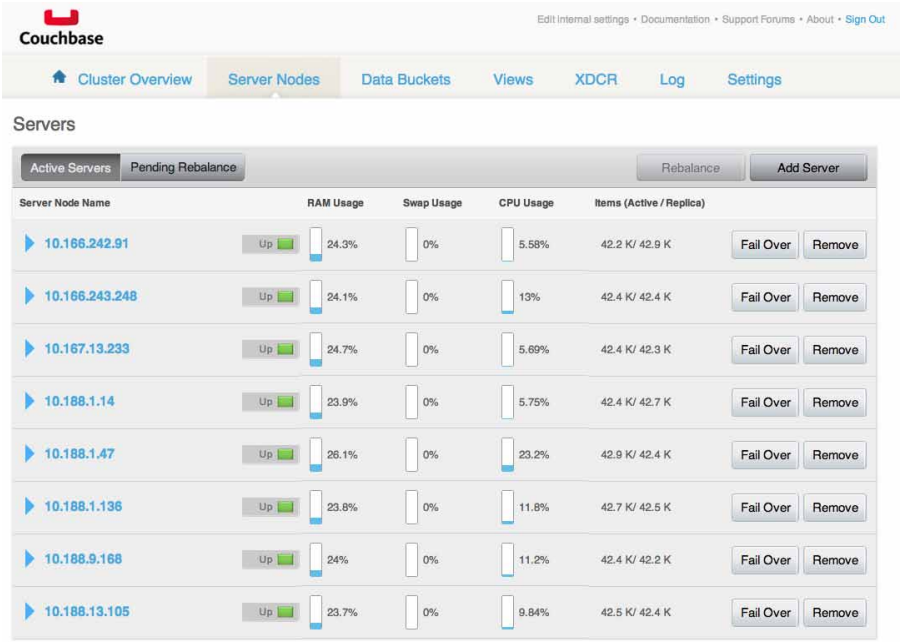

Figure 2: Couchbase Server is easy to scale or shrink with a single button click

Data is auto-sharded and uniformly distributed across servers in a Couchbase cluster. Client requests are directly routed to the appropriate server, which is completely transparent to the app and the developer. When servers are added or removed from the cluster, online rebalancing maintains even data distribution throughout the cluster and ensures that your app's data access is evenly load balanced across servers.

Scaling is also important on a global level. To provide a good experience to global users, such as students and other digital content readers, you need to bring data closer to them to reduce latency. With cross datacenter replication in Couchbase Server, data is replicated across datacenters to improve data locality (getting data closer to your users).

Consistent high performance

Content and metadata stores are highly interactive apps where content consumers discover, share and leverage information at record speed. They need to be rich and powerful, supporting hundreds of thousands of document operations every second. To have this high performance, your database needs consistent low latency and high throughput. Consistent low latency means sub-millisecond response time 99% of the time regardless of the mix of reads and writes. The built-in object cache in Couchbase Server holds the working set for your app in memory for fast access.

High throughput, an essential ingredient for high app performance, means making the best use of your hardware resources so that you can serve more users with fewer servers. It is needed when you have concurrent updates to documents that store user or content statistics. Couchbase Server provides high throughput that scales linearly with the number of server nodes. Fine-grain locking in Couchbase Server minimizes lock contention and enables highly concurrent read and write operations.

Couchbase's low latency and high throughput has been demonstrated in a benchmark result published by Cisco, based on a mixed workload.

Flexible data model

Content-driven apps have highly unstructured data. With different schemas that constantly change, as well as content that is used by millions of users across different devices and platforms, it is hard to neatly fit data into the rows and columns that are managed by a RDBMS. What's needed is a system for all your content and metadata.

With NoSQL databases such as Couchbase, each document can have a completely different structure from other documents. JSON documents can model all kinds of data, even ones that are sparse, hierarchical, semi- or unstructured. Because of this flexibility, Couchbase provides a great fit for customized content applications or apps that need to integrate data from different sources. For example, as shown in Figure 3, a JSON document in Couchbase can be used to store something as unstructured as a reddit post.

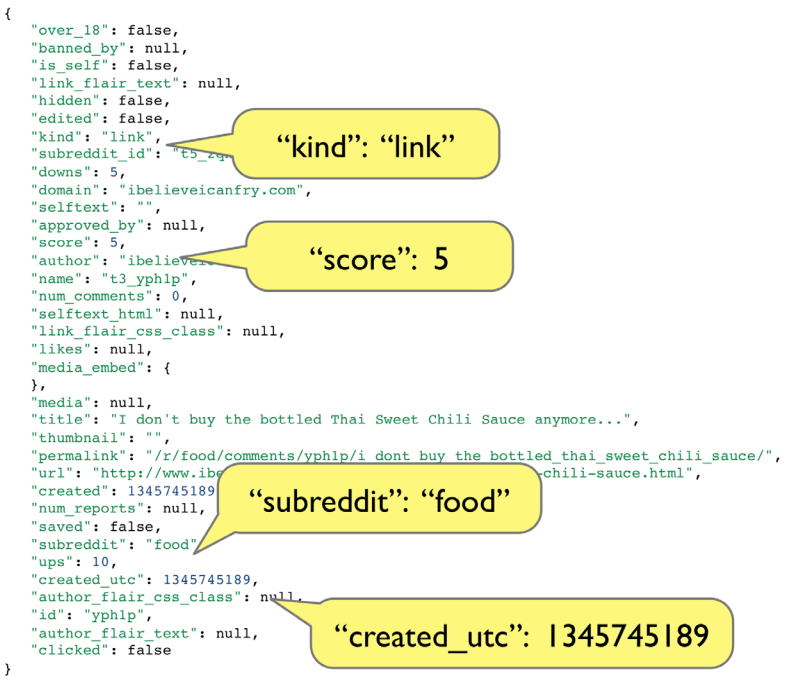

Figure 3: A Reddit post modeled as JSON in Couchbase

In a content and metadata store app, you can use JSON documents in Couchbase Server to store:

- **Content metadata** for media objects for articles, including categories, contributors, and type information. The metadata can have pointers to the media file stored in external storage systems like Amazon S3 or a CDN.
- **User profiles** for content viewing history. This data is typically updated every time a user views a document and can be used for customizing search results based on user preferences.
- **Content stats** for when a piece of content gets viewed. This data is typically updated every time a document is viewed and can be used for boosting search results based on popularity.

With Couchbase's indexing and querying you can create primary and secondary indexes on JSON documents, query documents for specific key values and ranges as well as do aggregation. This is useful for getting simple real-time analytics on content in your application. For example, you can use a view to generate a feed of recently published articles and top viewed content in your application. As shown in Figure 4, a view can be written to order sub-reddit's by time and score.

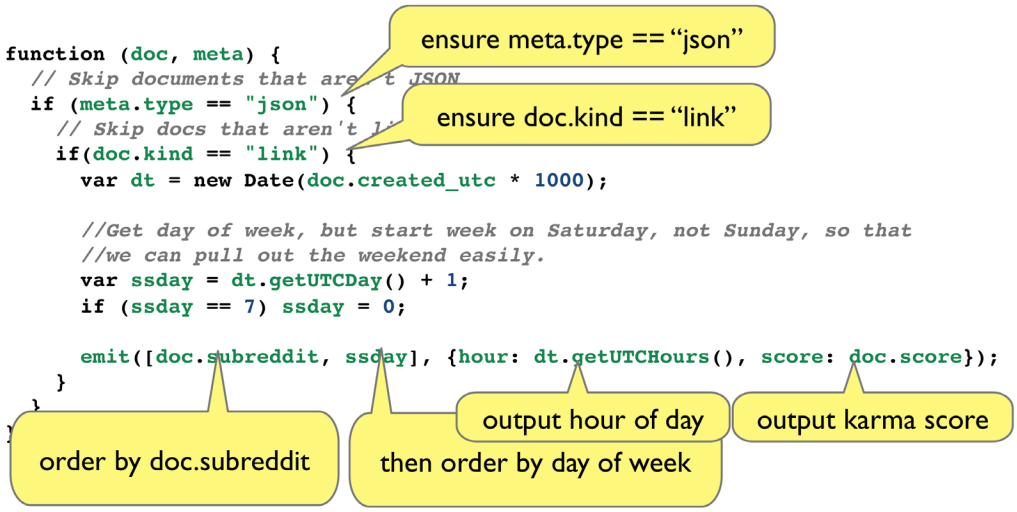

Figure 4: Secondary index to order subreddit's by time and karma score

Always-on 24x365

To make sure your users have a great experience, your platform needs to be up and running 24x365 without any downtime. With Couchbase, you can keep your platform running even when you do admin tasks or when your database fails – a server or your datacenter.

All of this is made possible in Couchbase Server by replicating documents across different servers in the cluster. So, whenever a server fails in the cluster, Couchbase Server detects this failure and promotes corresponding replica documents on available nodes to active status. The client detects the topology change and requests information about the cluster topology from any available node in the cluster. With the updated topology information, clients can route requests to the appropriate node. This means that your app can continue to send requests to the database without any changes.

Completing administrative tasks online like database upgrades, operating system or hardware upgrades, cluster resizing, data compaction, backup and index building are all possible without taking your app down. Should a datacenter crash, data is still available and your app can read and write to any geo-location using active-active cross data center replication (see Figure 5).

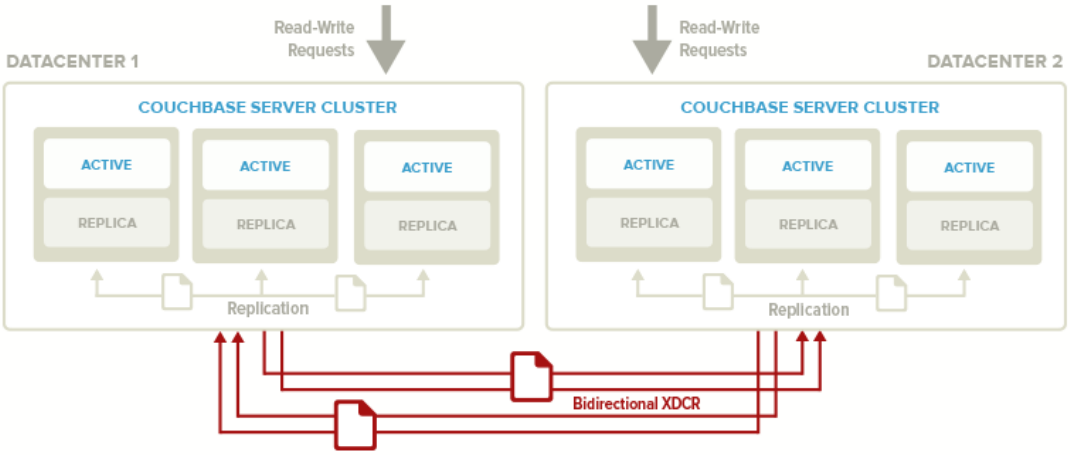

Figure 5: Intra-cluster and cross datacenter replication between two datacenters

Full-text search

Content and metadata store apps are gateways to digital content. Critical content assets are usually made up of many different smaller content elements, which might not be easily located. Additionally, with large amounts of rich content, it is also hard to find what exactly you are looking for quickly. Using a RDBMS to search data is largely time consuming since you need to pre-define how your data is stored and indexed to make it searchable.

If your app needs search capabilities, different techniques can be used to index your database for full-text search. The preferred approach is to use a real-time stream of data between Couchbase and Elasticsearch, a full-text search engine.

Elasticsearch is a distributed full-text search engine for documents. Users can write simple and complex queries using the Elasticsearch query DSL. In addition to just searching for phrases, Elasticsearch also provides relevance-based queries with features like boosting and faceted filtering that enable you to build rich and powerful apps.

By using the Couchbase Plug-in for Elasticsearch, you can continuously stream data in real time to Elasticsearch after it is written to disk on Couchbase. This ensures that the search index in Elasticsearch is always kept up-to-date with the data in Couchbase. The plug-in is installed on the Elasticsearch cluster and uses Couchbase's cross datacenter replication to propagate data.

The plug-in is cluster-topology aware so if a Couchbase node or Elasticsearch server goes down or if new servers are added, the updated cluster topology information is obtained and replication continues to the available servers in the destination cluster. Additionally, the plug-in intelligently tracks what data is replicated from Couchbase to Elasticsearch. If data transfer is interrupted due to network connection failures, it can pick up from where it left off when the connection is back up.

Sample application – online learning portal

Now, let's say you want to build a content app. Using the online learning portal app shown in Figure 6 as an example, let's look at the different portions of the app you can build using Couchbase Server and Elasticsearch.

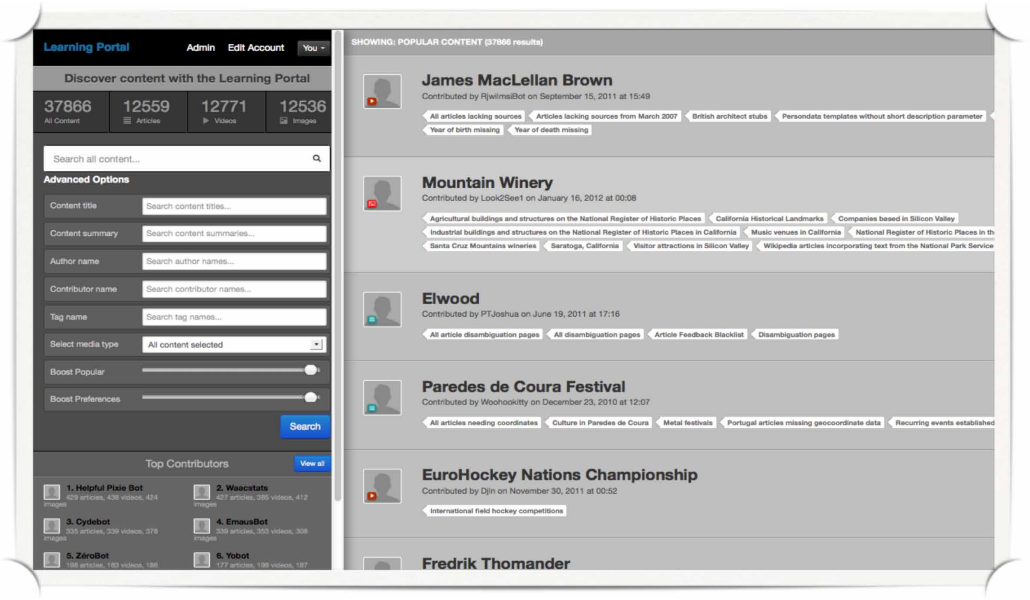

Figure 6: Learning portal application using Couchbase and Elasticsearch

Couchbase Server is ideal for storing content and metadata. It can be used for:

- Storing content that can be created, modified, updated, and tagged in real-time as well as usage details related to them.
- Storing per user preferences.
- Browsing content in your app with Couchbase's map reduce views.
- Viewing and editing content using the Couchbase admin portal.
- Analyzing content popularity using simple real-time analytics (see Figure 7)

Figure 7: Calculating statistics via Couchbase Server Views

As you can see in Figure 8, by using Couchbase and Elasticsearch, you can:

- Get ad-hoc full-text search queries of content summaries.
- Update the full-text index in Elasticsearch as new content is added or content tags are updated in Couchbase.
- Boost the relevancy of a field in an ad-hoc user search query based on popularity score.
- Boost the relevancy score based on user preferences.

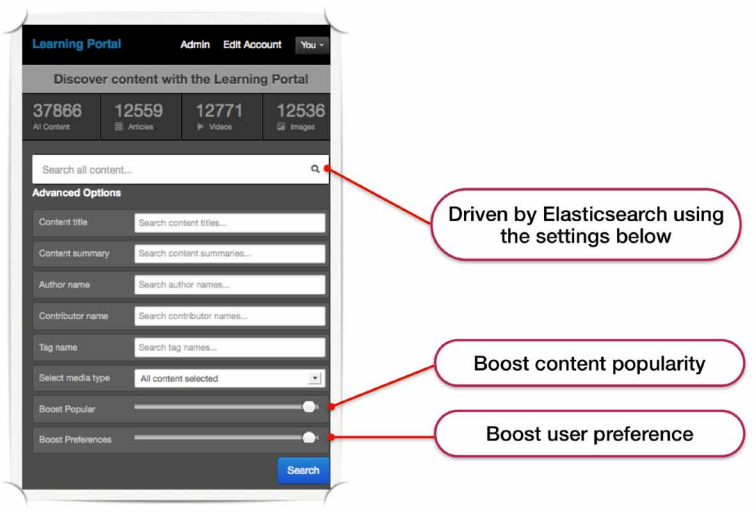

Figure 8: Tuning content ranking via Elasticsearch

Additional references

Couchbase Plug-in For Elasticsearch

Couchbase at McGraw-Hill Education

Create a Content Store with Couchbase – The Learning Portal

Couchbase Learning Portal – Source Code

Cross Datacenter Replication in Couchbase Server

Couchbase Performance Benchmark – Cisco and Solarflare

About Couchbase

We're the company behind the Couchbase open source project, a vibrant community of developers and users of Couchbase document-oriented database technology. Our flagship product, Couchbase Server, is a packaged version of Couchbase technology that's available in Community and Enterprise Editions. We're known for our easy scalability, consistent high performance, 24x365 availability, and a flexible data model. Companies like AOL, Cisco, Concur, LinkedIn, Orbitz, Salesforce.com, Shuffle Master, Zynga and hundreds of others around the world use Couchbase Server for their interactive web and mobile applications. www.couchbase.com