> Bases de Datos 2, Resumen 6
> Miguel Ku Liang - 2019061913

Introduction

Couchbase's core architecture supports a flexible JSON data model and uses relational and multimodel data access services to supply data to operational and analytic applications.

- Essential NoSQL requirements and features: Couchbase's purpose is to ensure high performance, provide data model and data access flexibility, support distributed cluster networks and mobility and provide incredible value and low TCO.
- The original multi-model NoSQL database: Couchbase was founded through the merge of CouchOne and Membase.
    - Couchbase is an open source database company: Speed, flexibility, familiarity and affordability are built in the core of the database engine to ensure low latency and reliable replication.
- Core performance design principles: memory and network-centric architecture for speed and low latency, multimodel data access blending JSON flexibility with Key Value speed, workload isolation and asynchronous approach to everything.

JSON Data Model and Access Methods

- JSON Document data model: Couchbase stores data as individual documents that consists of a key and value.
    - JSON Document Flexibility: Couchbase created the Bucket-Scope-Collection-Document organizational hierarchy to allow maximum flexibility in defining application data meta models.
- Document access methods: Key-value, query and Analytics, full-text search, eventing and Couchbase Mobile.
- Key, values, and sub-documents
    - Keys: each value is identified by a unique key that cannot be changed
    - Values: attributes have their own value that can be a basic or complex type
    - Sub-documents: inner component of a JSON document
- Key Organizing Concepts for Documents
    - Flexible, dynamic data containment model
        - Buckets: logical equivalent of a database in relational systems
        - vBuckets: shards of data automatically distributed across nodes
        - Ephereral Buckets: allows the database to support applications where data is processed but not persisted
        - Scope: intermediate data organization structure
        - Collections: categorical or logically organized groups of documents
- Cluster Design Concepts

    - Nodes: physical or virtual machines that host instances of Couchbase Server
    - Clusters: consists of one or more nodes running Couchbase Server
    - Services: isolated set of processes dedicated to particular tasks

Couchbase Services

Couchbase implements the data access methods through a set of dedicated services, with the Data Service at its center.

- Data service and Key/Value engine: the principal component of the data service architecture is the key-value management system known as KV Engine.
    - Couchstore and Couchbase Magma, High Data-Density Storage: Magma is a high density storage engine.
    - Managed object cache: the managed object cache of each node hashes the document into an in-memory hash table based on the key.
    - Document expiration: documents are set to expire using a time to live setting
    - Memory management: Couchbase has a background task called the item pager
    - Compression: end-to-end data document compression is available across all features of the database using Snappy. The compression modes are: off, passive and active.
    - Compaction: you can reclaim empty gaps in all data files by performing compaction
    - Mutations: mutations happen at a document level
- Query service: it is an engine for processing SLQ++ queries.
    - ACID Transactions in SQL++: Couchbase supports definition of ACID transactions within SQL++.
    - Cost-based Query Optimization: query service uses a cost-based query optimizer to take advantage of indexes that are available.
- Index service: primary: indexes collection using the document key, secondary: indexes an object using a key-value, composite/covered: multiple fields stored in an index, functional: secondary index that allows functional expressions, array: an index of array elements, adaptive: secondary array index for all or some fields of a document, flex: for queries containing compound selection criteria.
- Index Advisor: it is a built-in query command that asks the database which index to use.
    - Query consistency: Couchbase indexes are updated asynchronously after the data has been changed by the application.
    - Memory-optimized Indexes (MOI): they use a skip list structure, optimizing memory consumption and concurrent processing of index updates and scans.
- Search service: engine for performing Full-text searches on the JSON data stored in a collection.
- Eventing service: it supports custom server-side functions that are triggered using an Event-Condition-Action model.
- Analytics: provides an ad hoc querying capability without the need for indexes, bringing a hybrid operational and analytical processing model for real-time and operational analytics on the JSON data.
- Mobile and the edge App Services: Couchbase Mobile includes Couchbase Lite and Sync Gateway.