

MongoDB and CouchDB

The
Pragmatic
Programmers

Instructor: Dr. GP Saggese - gsaggese@umd.edu

- All concepts in slides
- MongoDB tutorial
- Web
 - <https://www.mongodb.com/>
 - Official docs
 - pymongo
- Book
 - Seven Databases in Seven Weeks, 2e

Seven Databases in Seven Weeks

Second Edition

A Guide to Modern
Databases and the
NoSQL Movement

Luc Perkins
with Eric Redmond and Jim R. Wilson

Series editor: Bruce A. Tate
Development editor: Jacquelyn Carter



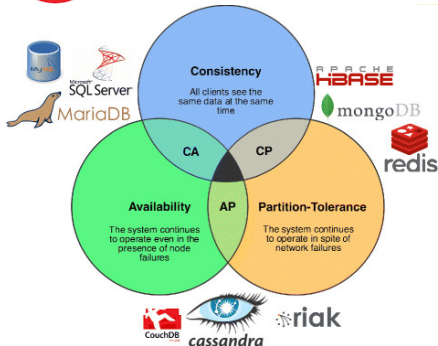
- *Couchbase*

Couchbase

- NoSQL document-oriented DB (like MongoDB)
- Couchbase = merge of CouchDB and membase
 - *CouchDB*
 - Open source document store
 - HTTP RESTful API to add, update, delete documents
 - Support all 4 ACID properties
 - *membase*
 - Distributed key-value store (like Redis)
 - Designed to scale both up and down
 - Highly available and partition tolerant
- Uses HTTP protocol to query and interact with objects in the DB
 - No query language
 - Objects stored in *buckets*
 - Collection of JSON docs, with no special relation to

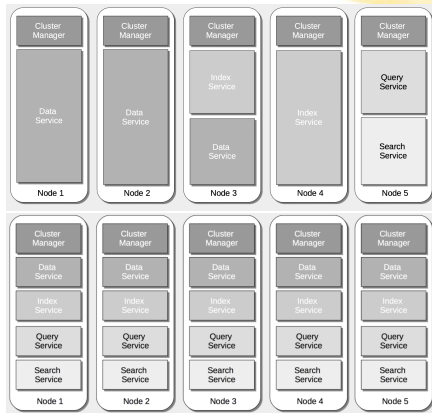


Couchbase



Architecture

- Every Couchbase node consists of **different services**:
 - Data service
 - Index service
 - Query service
 - Cluster manager component
- Services can run on separate nodes of the cluster, if needed
- **Data replication**
 - Across nodes of a cluster
 - Across data centers
- **Data service**
 - Writes data *asynchronously* to disk after acknowledging to the client
 - Optionally *synchronous*: ensure data is written to more than one server before acknowledging a write



Queries

- **Can create multiple views over documents**
 - Views are optimized / indexed by Couchbase for fast queries
 - Re-indexed when underlying documents changes
 - Can do full-text searches using the indexes
- **Perform well when:**
 - There are infrequent changes to the structure of documents
 - Know in advance what kinds of queries you want to execute
- **Query**
 - Uses a custom query language called N1QL (“nickel”)
 - Extends SQL to JSON documents
 - Queries over multiple documents using (server-side) joins
- **Map-reduce support**
 - (Map) First define a view with the columns of the document your are interested in
 - (Reduce) Optionally define aggregate functions over the data