MongoDB VS CouchDB

A Comparative Analysis

Group 7 – T20A
Akanksha Patil
Samarth Sehgal
Mohammad Saif

TODAY'S DISCUSSION

- Introduction
- Feature 1 : Storage Architecture
- Feature 2 : CAP Theorem
- Feature 3 : Indexing
- Conclusion

INTRODUCTION

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

- Startup of 10gen, originated in 2007, and managed by MongoDB Inc.
- Written in C++
- Makes use of BSON

INTRODUCTION

CouchDB is an open-source document-oriented NoSQL database, that uses multiple formats and protocols to store, transfer, and process its data.

- Released in 2005, Developed and managed by Apache Software Foundation.
- Implemented in Erlang.
- Makes use of JSON.

COMPARISON OVERVIEW

DATA MODEL

- MongoDB : document-oriented, uses BSON
- CouchDB : document-oriented, uses JSON

INTERFACE

- MongoDB : uses binary and custom protocol over TCP/IP
- CouchDB : HTTP/REST –based interface

OBJECT STORAGE

- MongoDB : database contains collections and collections contain documents
- CouchDB : database contains documents

SPEED

- MongoDB : performs faster reads
- CouchDB : slower read speed compared to MongoDB

MOBILE SUPPORT

- MongoDB : no mobile support
- CouchDB : provides support for Apple IOS and Android

QUERY METHOD

- MongoDB : Map – Reduce for creating collections and object-based query language
- CouchDB : Map - Reduce

REPLICATION

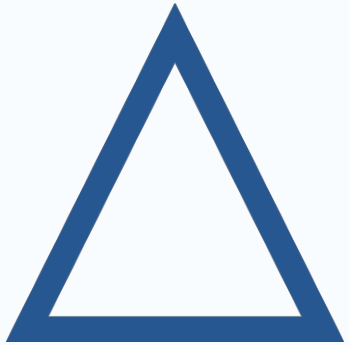
- MongoDB : supports master-slave replication
- CouchDB : supports both master-master and master-slave replication

CONCURRENCY

- MongoDB : update in-place
- CouchDB : follows MVCC

KEY FEATURES

STORAGE ARCHITECTURE



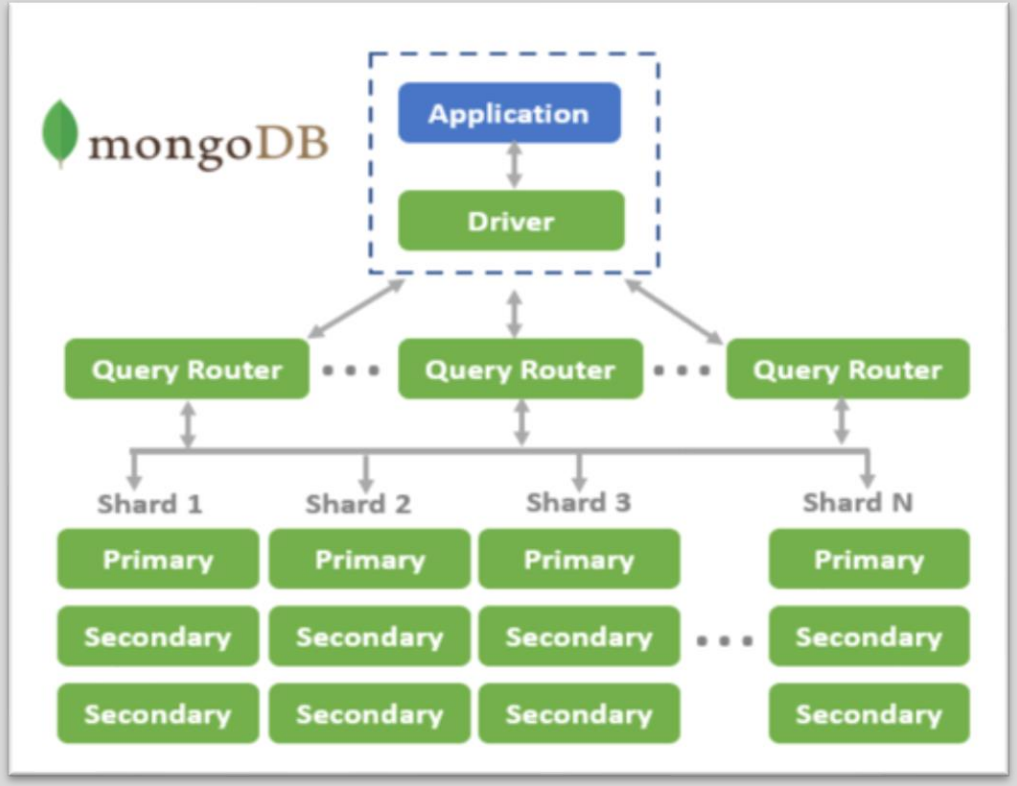
CAP TRADE-OFF

INDEXING

STORAGE ARCHITECTURE

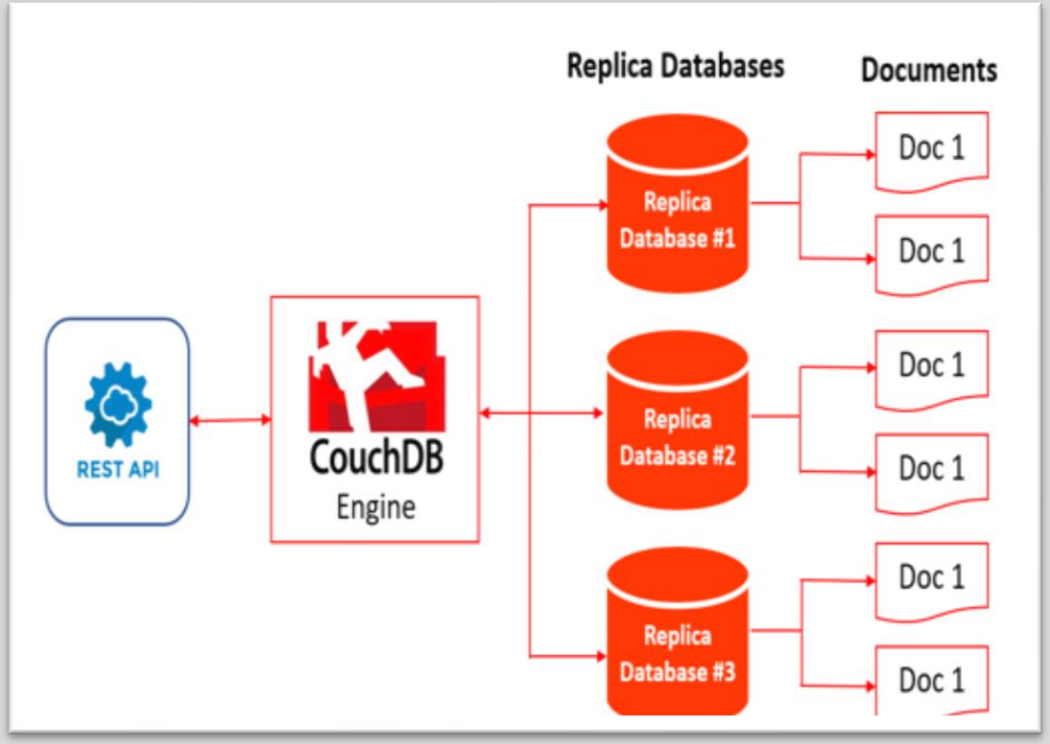
FEATURE 1

mongoDB



- Uses BSON that supports document storage and data interchange.
- NoSQL, schema-free database
- Database can be on multiple servers.
- Makes use of replication (master – slave).
- Incorporates sharding which makes use of scaling processes horizontally.
- Offers Load Balancing

CouchDB



- Uses JSON for data storage
- Document – type structure with schema
- Unlike regular relational database, does not have data and relationships in table
- Databases can be on multiple servers
- Makes use of replication (master – master and master – slave)
- Uses RESTful HTTP API to help applications read, edit and delete documents
- Lockless mechanism makes it highly scalable

VERDICT

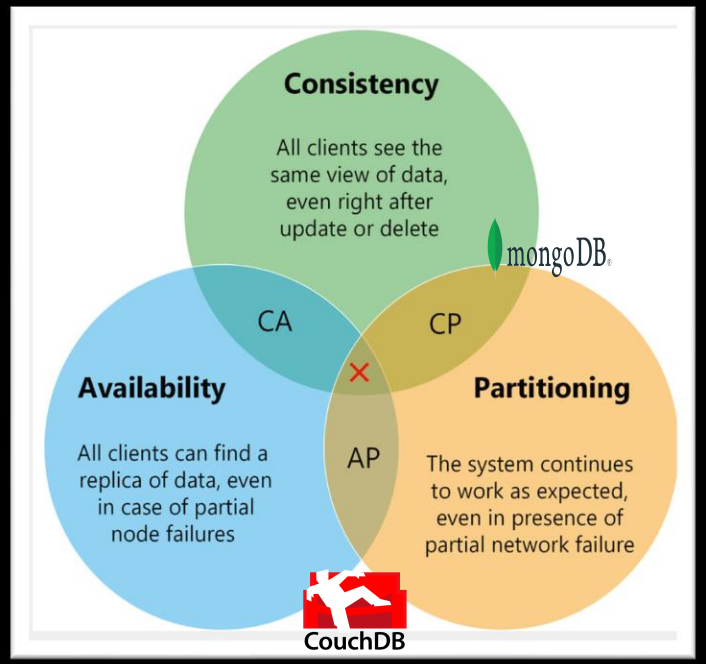
If one is making a transition from a relational database to NoSQL database, MongoDB is the best choice for such applications.

CouchDB as JSON is more efficient and faster as it requires less computation time, and the data size is smaller.

CAP TRADE-OFF

FEATURE 2

CAP TRADE-OFF



Three primary concerns you must balance when choosing a data management system.

Consistency: Each client always has same view of data

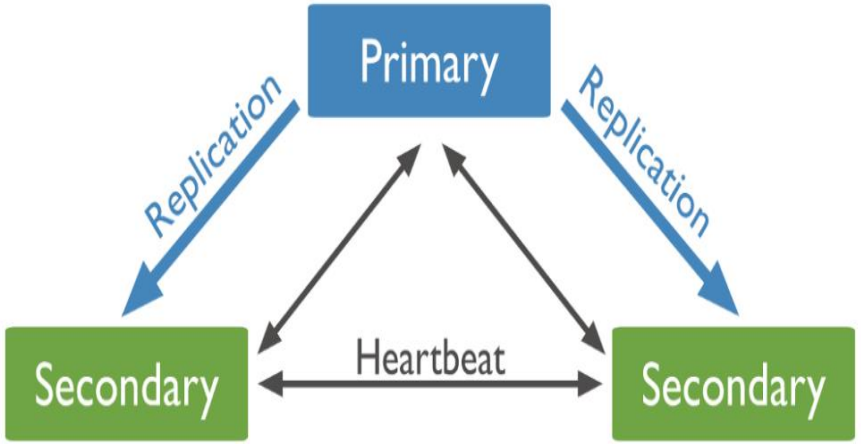
Availability: All clients can always read and write

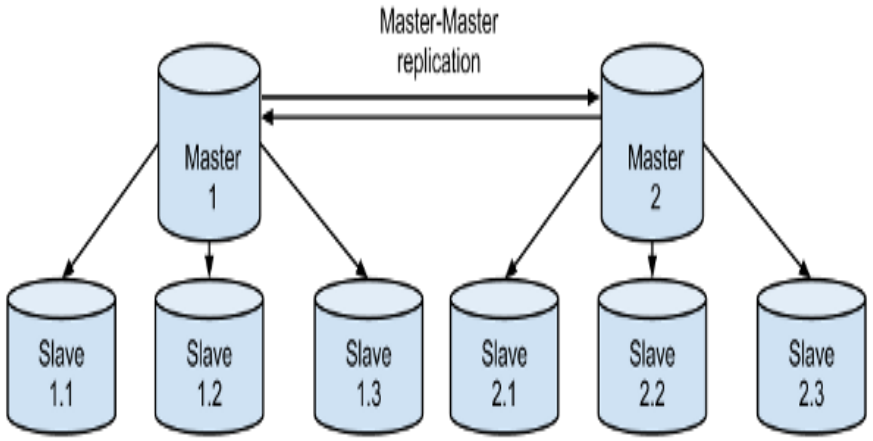
Partition Tolerance: System works well across physical partitions.

MongoDB favors consistency and partitioning while CouchDB favors availability and partitioning.

- Uses “strict consistency”
- Traditional update-in-place
- Both read and write from primary node

- Uses “Eventual Consistency”
- MVCC based
- Creates versions of data





VERDICT

mongoDB®

If consistency is what you're after, MongoDB is what you should go for

CouchDB

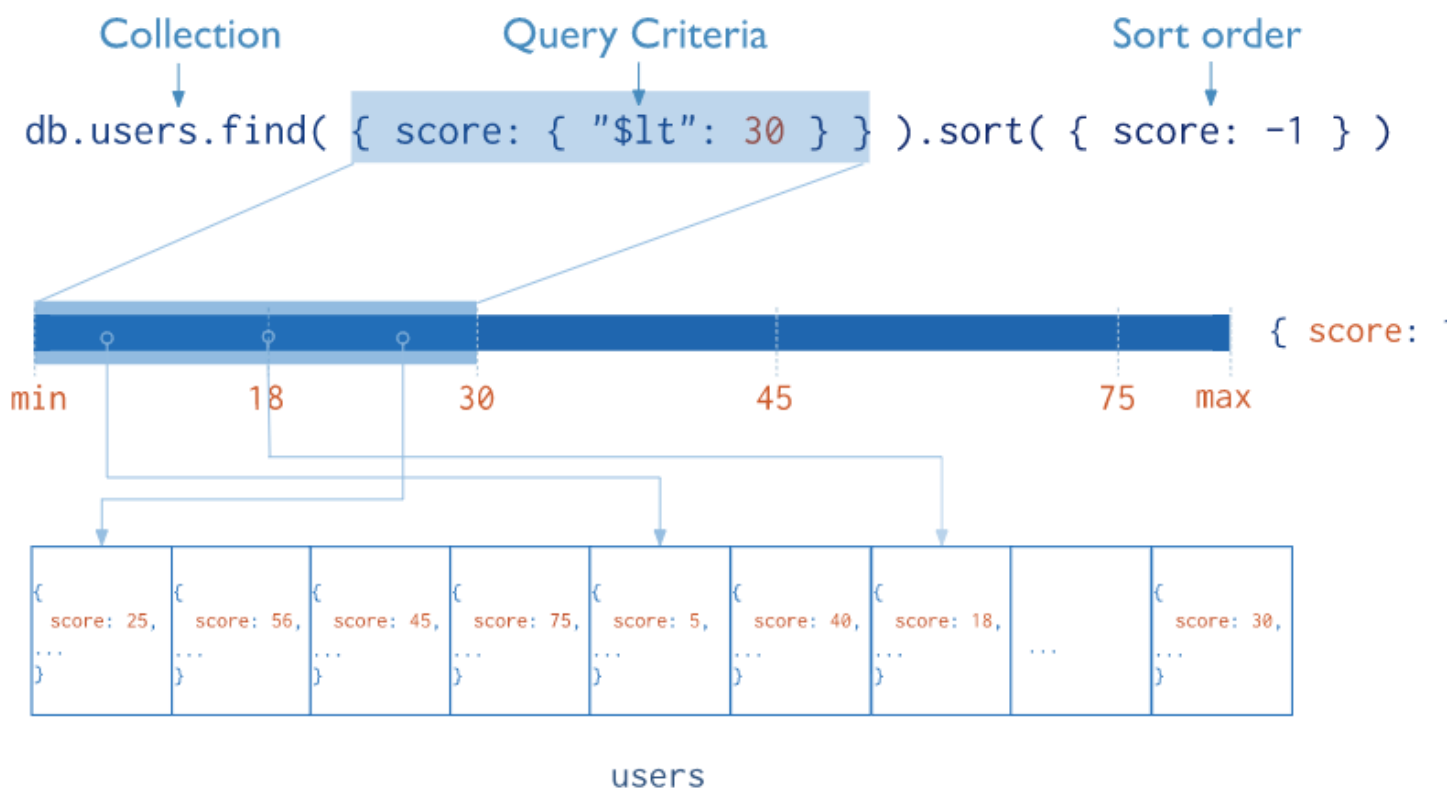
Prefer CouchDB when high availability is more important, even if some clients are seeing data which is slightly out of date.

INDEXING

FEATURE 3

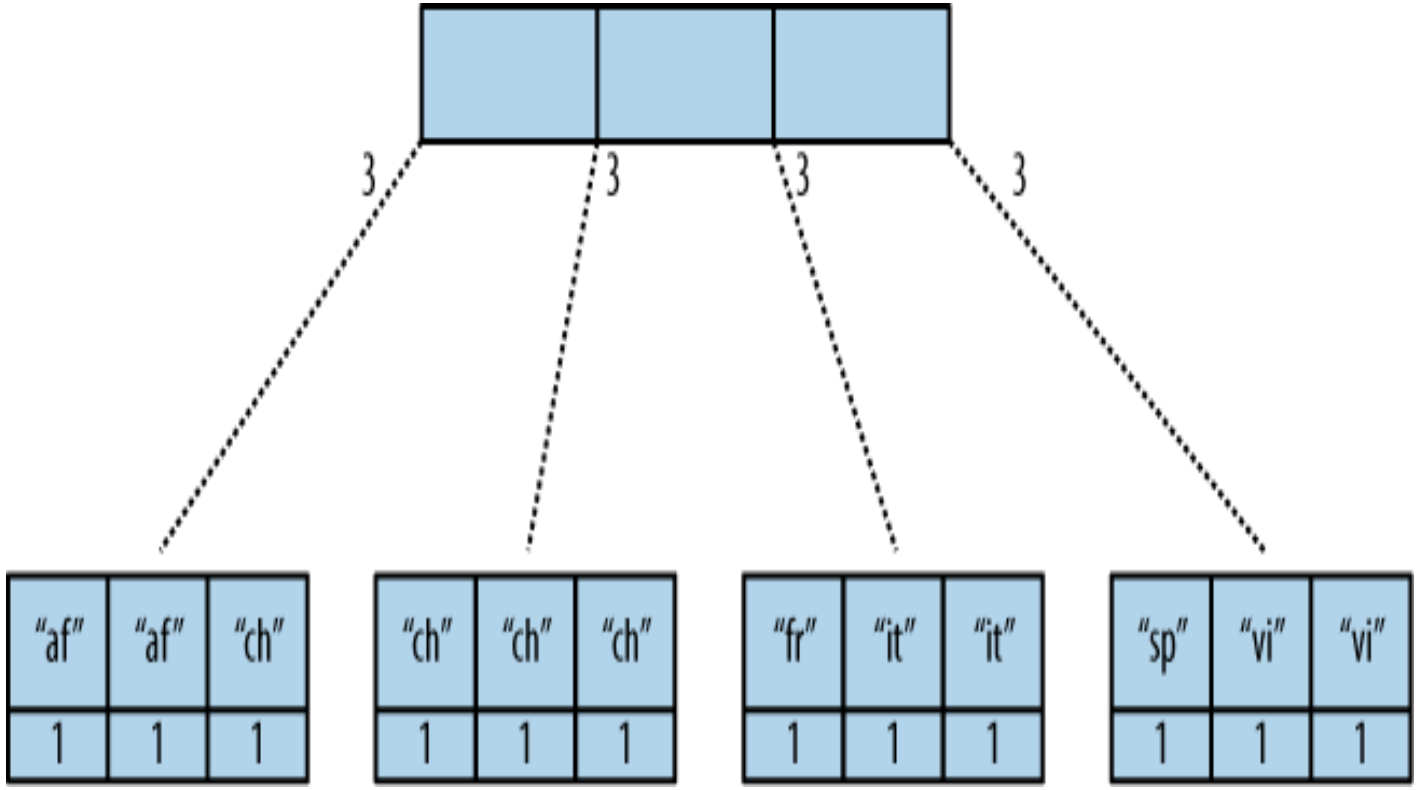
MongoDB	CouchDB
Implemented at collection level	Implemented via views
MongoDb query used for creation	Needs javascript function for creation
Query utilises index for search	Query is run on top of Views

MONGODB INDEX



- Stored at Collection level.
- Small portion of data in traversal form.
- Fields are ordered by value.

COUCHDB INDEX



- Create a view using map function
- This returns view with key and value and is ordered by key
- Write queries that run on this view
- Subsequent queries because of BTree retrieval

JAVASCRIPT MAP FUNCTION:

```
function(doc) {
  if(doc.score && doc.title) {
    emit(doc.score, doc.title);
  }
}
```

BENEFITS AND DRAWBACKS

mongoDB®

- Simple syntax and easy to understand implementation.
- Needs to be explicitly created by the user.
- Needs an assumption of required fields before creation.

CouchDB

- Indexes are created on the creation of a view.
- Different views can lead to index duplication.
- Currently doesn’t support geospatial index.

SUMMARIZING

Storage Architecture

CouchDB is clearly better in terms of less computation time, but if you are transitioning from relational database, MongoDB is a good option to adapt easily.

CAP Theorem

If your system has lots of read operations, go for a strongly consistent MongoDB platform. If you favor availability over consistency, CouchDB is a clear choice.

Indexing

If you have a requirement for manually creating index field for faster read operations, MongoDB would be a clear choice. If you need automatic indexing, go for CouchDB.

Mobility

If you need multi platform support for your database system, CouchDB is a clear winner.

Thank You