+

=

Apache CouchDB relax

CouchDB

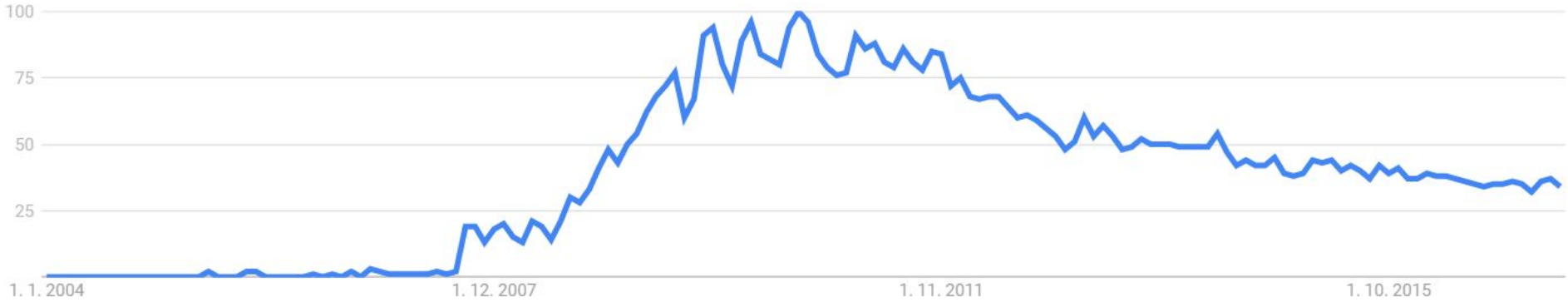
Mário Kudoláni, Michael Le, Peter Petkanič

Introduction

- COUCH = **C**luster **O**f **U**nreliable **C**ommodity **S**oftware
- Motto: “*Apache CouchDB has started. Time to Relax.*”
- Strongly inspired by web architecture
  - Relational databases are overshoot for some basic applications
  - “Database for Web” - blogs and other typical web apps
- Created by Damien Katz in April 2005, now under the Apache Foundation
- Last release: 2.0.0 in September 2016

Introduction

- Used to be quite popular



Google Trends for “CouchDB” keyword

Introduction

- Document Store database
  - No database schema
  - No fixed data structure
- Written in Erlang
- Focused on off-line availability
  - Suitable for use on mobile devices
- Heads to modularity and scalability

Basic Concepts

Document Store

- Uses JSON semi-structured document format
  - **Motivation:** Many documents has similar semantics but slightly different syntax
  - Relational databases are too rigid
- Document consists of:
  - Named fields (string, number, date, ordered list, associative map)
  - Attachements
  - Metadata (DocID, Sequence ID)
- Access using RESTful API

Views and View Models

- Sometimes, we need to filter, aggregate, create reports on data to get interesting data
  - This facility is provided by **Views**
    - Special type of documents
    - Not rebuilt every time
- And sometimes we want to add some order to documents, “a mask”
  - We use **View Models** for adding a structure to our documents

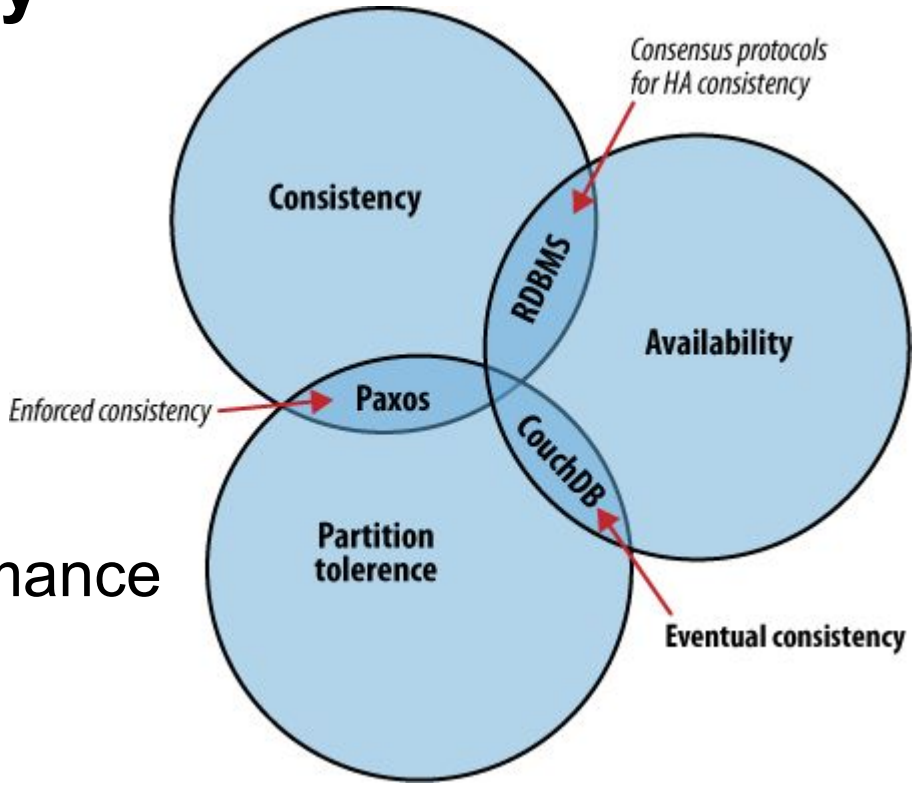
Views

- Also helps to build efficient indices
  - Uses B-Tree
- Satisfy efficient lookups
- Can be used in MapReduce fashion

Under the Hood

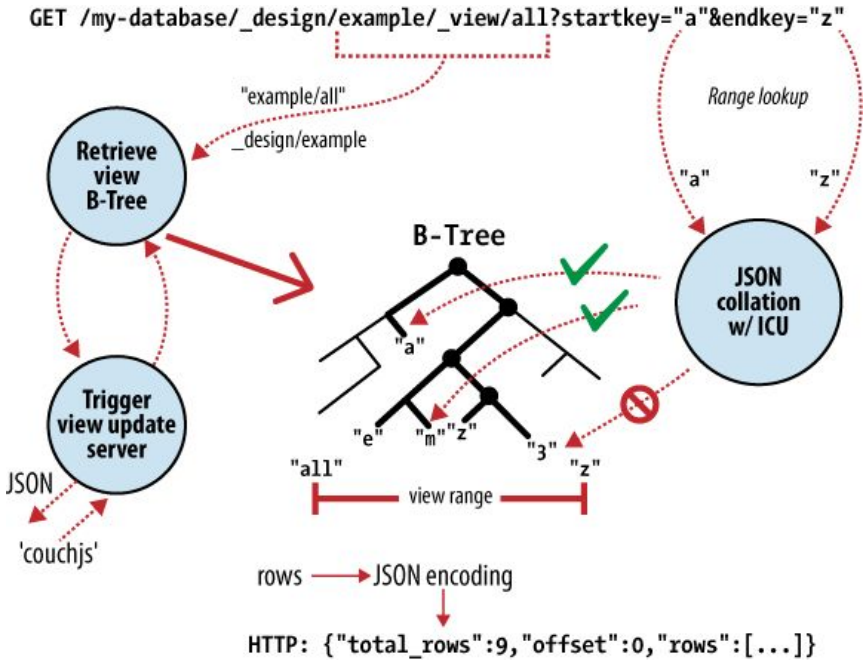
Eventual Consistency

- CAP theorem
  - Consistency
  - Availability
  - Partition tolerance
- CDB sacrifices immediate consistency in favor of performance



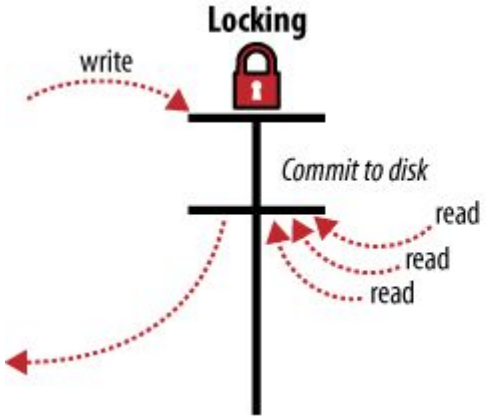
Local Consistency

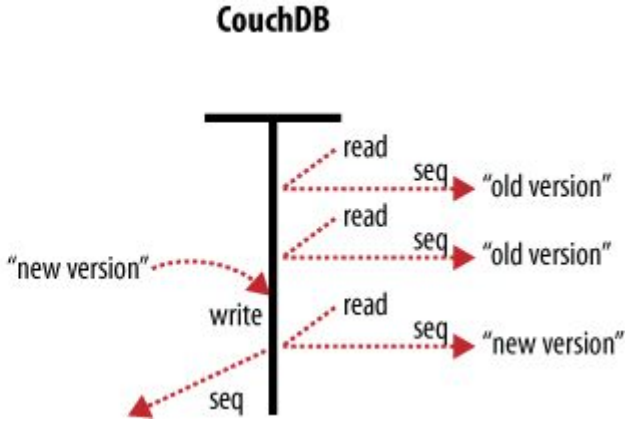
- CDB uses B-tree storage engine
- Map/Reduce functions are used to compute the results of a view
- Document access by key or key range only



Database update

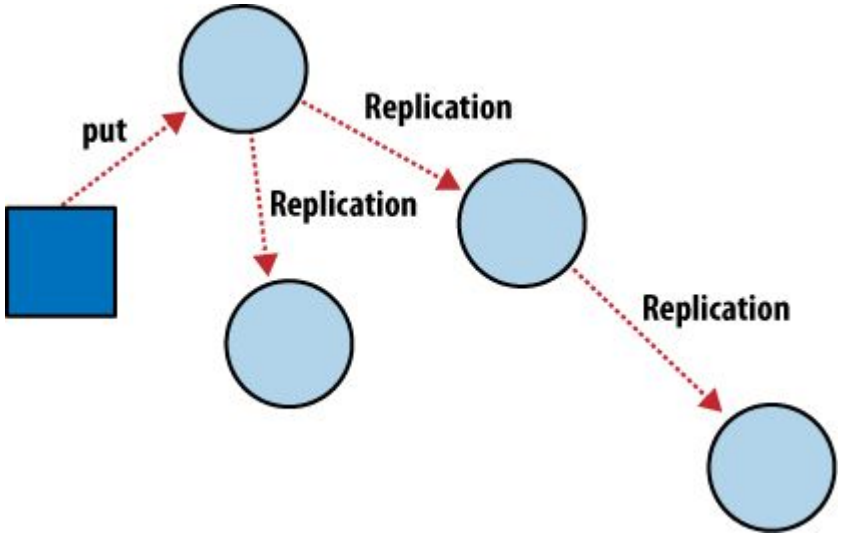
- RDBS
  - Table locking - waste of CPU
- CouchDB
  - MVCC (Multi-Version Concurrency Control)
  - Validation functions





Distributed database

- Incremental replication
  - Shared nothing cluster
  - Each node is independent and self-sufficient
- Conflict detection and Resolution
  - Document most recent Version
  - CDB save losing version



Security

- Admin Party
  - Any request to be made by anyone
- Basic Authentication
  - Note very secure
- Cookie Authentication
  - ‘/_session’ API returns token (by default valid for 10 minutes)

Practical Usage

Comparison with MongoDB - Overwiew

CouchDB

- Availability (Ease of use)
- Master-Master replication
- Map-reduce
- Safer
- HTTP/REST
- Versioning
- Erlang

MongoDB

- Consistency
- Master-Slave replication
- SQL-like querying syntax
- Faster
- Binary
- Data changes too much
- C++

Comparison with MongoDB - Syntax

CouchDB

```
db.save('_design/telemetryViews', {
      altitude: {
        map: function (doc) {
            emit(doc.time, doc.altitude);
        }
      }
  });

db.view('telemetryViews/altitude', function(err, res) {
    console.log('Altitude data:');
    res.forEach(function(key, row, id) {
        console.log('%s: %s', key, row);
    });
});
```

MongoDB

```
TelemetryDbModel
.find()
.sort('time')
.select('time altitude')
.exec(function(err, data) {
    if (err) return console.error(err);
    console.log(data);
});
```

Comparison with MongoDB - Conclusion

CouchDB

- Great UI
- Static views

MongoDB

- Easier syntax
- Dynamic queries

FUTON

- Web based API
- Very intuitive and informative
- CRUD database, document
- Manage security (accounts)
- Replicate database
- ...

Demo

Simple Database

Create new database

```
curl -X PUT http://127.0.0.1:5984/[db]
```

List databases

```
curl -X GET http://127.0.0.1:5984/_all_dbs
```

Insert new document

```
curl -X PUT http://127.0.0.1:5984/[db]/[uid]
-d '{[document content]}'
```

Database update conflict (DEMO in FUTON)

Simple Database

Query using Mango (like MongoDB)

http://docs.couchdb.org/en/2.0.0/api/database/find.html

```
curl -X POST http://127.0.0.1:5984/[db]/_find
-d '{[conditions]}'
```

```
{
  "selector": {
    "price": {
      "$lt": 1000
    }
  },
  "fields": ["item", "price"]
}
```

Simple Database

View/Map (DEMO in FUTON)

Pricelist in CZK

```
function(doc) {
    if(doc.item && doc.price && doc.rank) {
      emit(doc.item, doc.price * 24);
    }
}
```

Simple Database

View/Map (DEMO in FUTON)

Rank

```
function (doc) {
    if(doc.item && doc.rank.length > 0) {
        emit(sum(doc.rank) / doc.rank.length, doc.item);
    }
}
```

```
curl -X GET
http://127.0.0.1:5984/_design/[doc]/_view/[vie
w]?[start]key=[value]
```

More resources about CouchDB

- CouchDB official documentation
  http://docs.couchdb.org/
- Google Trends
  https://trends.google.com/trends/explore?date=all&q=couchdb
- CouchDB: The Definitive Guide
  http://guide.couchdb.org/editions/1/en/index.html
- Lennon Joe: Beginning with CouchDB (book)