




Couchbase

Eugenie Kňazeva

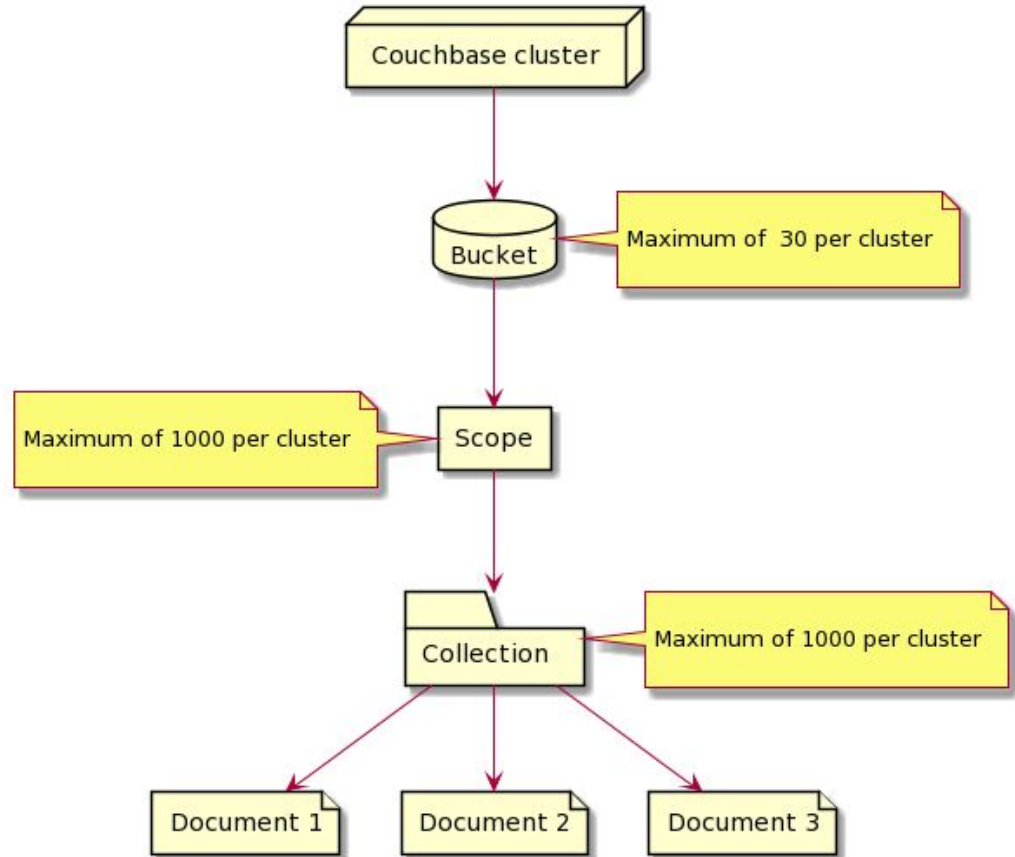
- 
- multimodelová databáze:
 - dokumenty(JSON)
 - klíč-hodnota
 - prostorová (spatial) data
 - podpora replikace
 - každý vrchol umí i zapisovat i číst (na rozdíl třeba od MongoDB)
 - podpora událostí (eventing, triggers)
 - pro dotazování poskytuje
 - SQL like queries
 - full-text search
 - indexování - key-value access, N1QL, MapReduce
 - speciální rychlé API pro CRUD operace nad Key-value databází
 - poskytuje integraci SQL a konektory pro Hadoop, Spark, Elasticsearch aj.
 - psaní dotazů je možné přes *curl operace*, SDK pro konkrétní jazyk, nebo pomocí klienta *cbc*



Datový model

- data se primárně ukládají jako JSON dokumenty
- je možné ukládat i ne JSON data
 - integer, string, array, boolean, null
 - vlastní typy
- querying a full-text search je dostupný pouze pro data ve formátu JSON, MapReduce je dostupný v omezené míře i pro ne JSON data
- datové struktury
 - Map - jako JavaScript object
 - List - jako JavaScript array
 - Set
- ID se nenastavují automaticky

Struktura dat





Klíč-hodnota model

- zakomponován jako součást dokumentové databáze
- ID dokumentu = klíč, dokument a jeho obsah = hodnota
- pro rychlé a optimalizované CRUD operace
- v příkladech používáme cbc

Operace v SDK:

- `upsert(docid, document)`
- `insert(docid, document)`
- `replace(docid, document)`
- `get(docid)`
- `remove(docid)`



Klíč-hodnota příklady

create:

```
cbc create -u Administrator -P password -U couchbase://localhost/travel-sample \  
  --scope='inventory' \  
  --collection='hotel' \  
  --mode insert hotel-123 <./tmp/hotel-123.json
```

read:

```
cbc cat hotel-123 -u Administrator -P password -U couchbase://localhost/travel-sample \  
  --scope='inventory' \  
  --collection='hotel'
```

read nad částmi dokumentů:

```
cbc-subdoc -u Administrator -P password -U couchbase://localhost/travel-sample subdoc > get  
airport_1254 --path geo
```



Klíč-hodnota příklady

update s upsert (existuje i mode replace):

```
cbc create -u Administrator -P password -U couchbase://localhost/travel-sample \  
  --scope='inventory' \  
  --collection='hotel' \  
  --mode upsert hotel-123 <./tmp/hotel-123.json
```

update části dokumentu:

```
cbc-subdoc -u Administrator -P password -U couchbase://localhost/travel-sample subdoc> dict-upsert  
airport_1254 --path pets_ok=true
```



Klíč-hodnota příklady

delete:

```
cbc rm -u Administrator -P password -U couchbase://localhost/travel-sample airport_1254
```

delete části dokumentu:

```
cbc-subdoc -u Administrator -P password -U couchbase://localhost/travel-sample subdoc> remove  
hotel-123 --path url
```




Prostorový model

- pomocí rozšíření **GeoCouch** (již jako součást Couchbase)
- používá v dokumentech **GeoJSON** formát
- dotazování pomocí curl dotazů
- po založení databáze je třeba přidat tzv. design document, který je možné zadat pomocí Design Document Function
- výpis v různých formátech
- stale, count, limit, skip

```
{
  "type": "FeatureCollection",
  "features": [{
    "type": "Feature",
    "geometry": {
      "type": "Point",
      "coordinates": [102.0, 0.5]
    },
    "properties": {
      "prop0": "value0"
    }
  }, {
    "type": "Feature",
    "geometry": {
      "type": "Polygon",
      "coordinates": [
        [
          [100.0, 0.0],
          [101.0, 0.0],
          [101.0, 1.0],
          [100.0, 1.0],
          [100.0, 0.0]
        ]
      ]
    },
    "properties": {
      "prop0": "value0",
      "prop1": {
        "this": "that"
      }
    }
  }
]}
```



GeoCouch - ukázka

create db:

```
curl -X PUT http://127.0.0.1:5984/places
```

set design document:

```
curl -X PUT -d '{"spatial":{"points":"function(doc) {\n  if (doc.loc) {\n    emit([\n      type: \"Point\",\n      coordinates: [doc.loc[0],\n      doc.loc[1]]\n    ], [doc._id, doc.loc]);\n  };\n}'} http://127.0.0.1:5984/places/_design/main
```

put test data:

```
curl -X PUT -d '{"loc": [-122.270833, 37.804444]}' http://127.0.0.1:5984/places/oakland  
curl -X PUT -d '{"loc": [10.898333, 48.371667]}' http://127.0.0.1:5984/places/augsburg
```

query:

```
curl -X GET --globoff 'http://localhost:5984/places/_design/main/_spatial/points?start_range=[0,0]&end_range=[180,90]'
```

result:

```
{\"id\":\"augsburg\",\"key\":[[10.89833299999999916,10.89833299999999916],[48.37166700000000219,48.37166700000000219]],  
geometry\":{\"type\":\"Point\",\"coordinates\":[10.89833299999999916,48.37166700000000219]},\"value\":[\"augsburg\",[10.8983329999  
9999916,48.37166700000000219]]}
```



N1QL Queries

- vlastní dotazovací jazyk N1QL (“nickel”) - rozšíření SQL o dotazování nad JSON dokumenty
- při použití SDK (třeba pro Node.js) lze dotazy i parametrizovat
- podporuje subqueries a speciální window functions

```
SELECT *           // parts of document to return
FROM tutorial      // the data bucket, or data store to work with
WHERE title = "Mr." // conditions the document must satisfy
```



N1QL Queries - příklady

```
SELECT children[0].fname AS child_name
FROM tutorial
WHERE fname='Dave'
```

```
SELECT fname, email
FROM tutorial
WHERE email LIKE '%@yahoo.com'
```

```
SELECT fname, children
FROM tutorial
WHERE ANY/EVERY child IN tutorial.children
SATISFIES child.age > 10 END
```

AND, OR, NOT

```
SELECT fname || " " || lname AS full_name
```



N1QL Queries - příklady

```
SELECT fname, email
FROM tutorial
USE KEYS ["dave", "ian"]
```

```
SELECT fname, age
FROM tutorial
ORDER BY age
LIMIT 2 OFFSET 4
```

```
SELECT relation, COUNT(*) AS count
FROM tutorial
GROUP BY relation
HAVING COUNT(*) > 1
```

```
AVG, MIN, MAX
INNER, LEFT [OUTER] JOIN
NEST, UNNEST
UPDATE
```



N1QL Queries - příklady

```
SELECT usr.personal_details, orders
FROM users_with_orders usr
USE KEYS "Elinor_33313792"
    JOIN orders_with_users orders
    ON KEYS ARRAY s.order_id FOR s
    IN usr.shipped_order_history
END
```

```
{
  "results": [
    {
      "orders": {...},
      "personal_details": {...}
    },
    {
      "orders": {...},
      "personal_details": {...}
    }
  ]
}
```



Děkuji za pozornost!

Zdroje:

- Couchbase: <https://www.couchbase.com/>
- dokumentace Couchbase: <https://docs.couchbase.com/home/index.html>
- N1QL tutorial: <https://query-tutorial.couchbase.com/tutorial/#1>
- GeoJSON: <https://geojson.org/> , <https://www.rfc-editor.org/rfc/pdf/rfc7946.txt.pdf>
- GeoCouch: <https://github.com/couchbase/geocouch>,
<https://github.com/couchbase/geocouch/blob/master/gc-couchdb/README.md>
- Couchbase Client Commandline Utility:
https://docs.couchbase.com/sdk-api/couchbase-c-client/md_doc_cbc.html
- Key-value operations: <https://docs.couchbase.com/server/current/guides/kv-operations.html>