Lab 2
MongoDB

[S20] Data Modeling and Databases II

Introduction

MongoDB is a document-oriented DB, which means:

- It is a subclass of key-value databases
  - However, MongoDB relies on the structure of the data to extract metadata for optimization
- It is different from RDB in how objects are stored
  - In RDB, an object can be stored in multiple tables
  - In Document DB, the object is always stored in a single instance and every object can be different from other
  - In MongoDB documents are similar to JSON objects
- It supports CRUD operations

MongoDB main features

- Ad hoc queries
- Indexing
- Replication
- Load balancing
- File storage
  - See grid filesystem
- Aggregation
  - Built-in MapReduce
- Capped collections (circular queues)

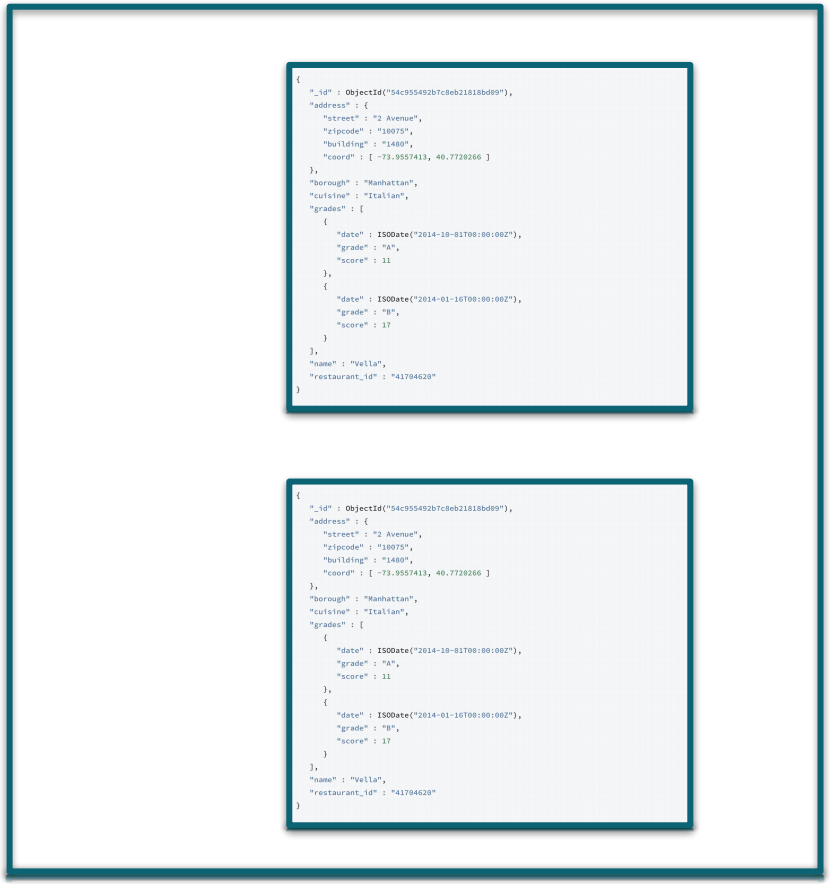
Documents

- A record in MongoDB is a document, which is a data structure composed of field and value pairs. MongoDB documents are similar to JSON objects.

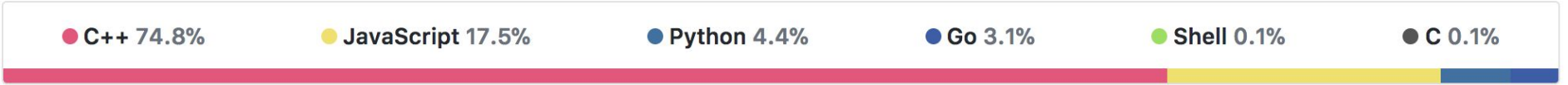
```
{
   "_id" : ObjectId("54c955492b7c8eb21818bd09"),
   "address" : {
      "street" : "2 Avenue",
      "zipcode" : "10075",
      "building" : "1480",
      "coord" : [ -73.9557413, 40.7720266 ]
   },
   "borough" : "Manhattan",
   "cuisine" : "Italian",
   "grades" : [
      {
         "date" : ISODate("2014-10-01T00:00:00Z"),
         "grade" : "A",
         "score" : 11
      },
      {
         "date" : ISODate("2014-01-16T00:00:00Z"),
         "grade" : "B",
         "score" : 17
      }
   ],
   "name" : "Vella",
   "restaurant_id" : "41704620"
}
```

Collections

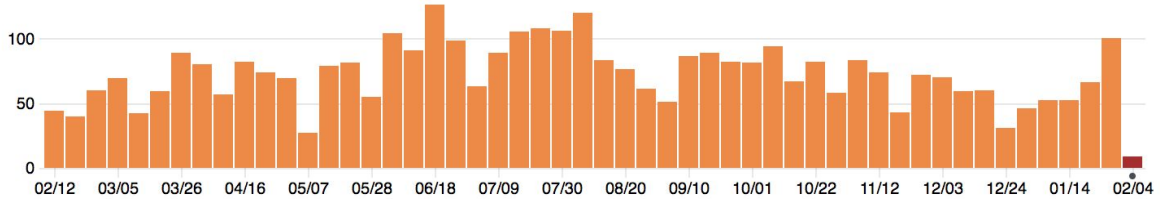
- MongoDB stores documents in collections. Collections are analogous to tables in relational databases. Unlike a table, however, a collection does not require its documents to have the same schema.



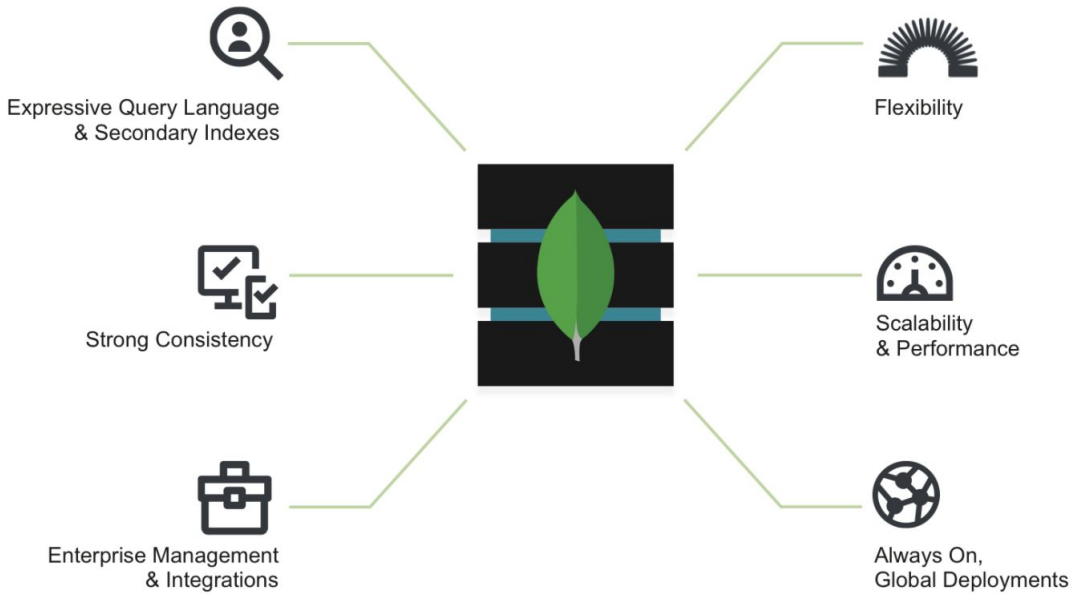
MongoDB, a closer look



- Written in C++
- Actively maintained (last commit was an hour ago at the moment of writing)
- Average commit frequency is above 50 per week



MongoDB Architecture



Data as Documents

- Data is stored in BSON
- BSON documents contain fields, fields contain values (including arrays, binary data and sub documents)
- Closely aligned to the structure of objects in programming languages
  - Faster for developers to map data
- Keep all data for a given record in a single document

Data as Documents

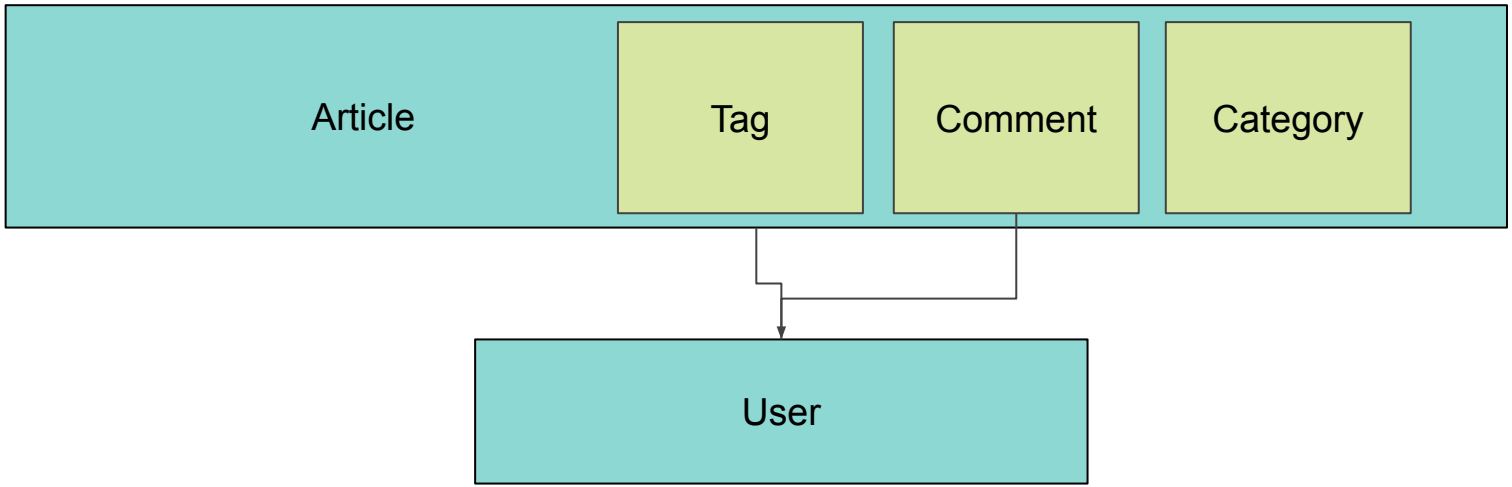
- Consider for example a data model for blogging application:
  - In RDB, it would involve having separate tables for Users, Categories, Articles, Comments etc
  - In MongoDB this could be modelled by two collections: one for Users and one for Articles:

Data as Documents

- Consider for example a data model for blogging application:
  - In RDB, it would involve having separate tables for Users, Categories, Articles, Comments etc
  - In MongoDB this could be modelled by two collections: one for Users and one for Articles
- Pros:
  - Data is more localized, no need to join
  - A single read can return the whole document
  - Higher scalability
- Cons:
  - What happens if you have many transactions?

Data as Documents

- Consider for example a data model for blogging application:
  - In RDB, it would involve having separate tables for Users, Categories, Articles, Comments etc
  - In MongoDB this could be modelled by two collections: one for Users and one for Articles:



Dynamic Data

- MongoDB allows variations in data structure
    - MongoDB allows developers to evolve data schema as the application becomes more mature
    - If a new field is needed to be introduced to a document, it can be introduced without affecting other documents and without taking the system offline

Dynamic Data

- A problem with having dynamic data - there may be situations when a strict schema must be guaranteed
- MongoDB provides schema validation within the database via syntax derived from the IETF JSON Schema standard
- Schema validation allows developers to create a flexible rules on which to save or reject a document
- Structure can be imposed on a subset of fields

Dynamic Data

Benefits of data validations

- Application logic simplification. With a strictly defined definition of what a collection looks like, there isn't a need for the application to handle the guarantees and associated errors.

- Control of data. Client applications must adhere to the rules set forth by the collection. No need to worry about data being updated or inserted that has incorrect field names or data types.

- Governmental compliance. There are applications and data models in a variety of industries and locales that require data to be stored in specific formats. For example, the EU General Data Protection Regulation in the European Union.

Query Model

Query types:

- **Key-value queries** return results based on any field in the document, often the primary key
- **Range queries** return results based on values defined as inequalities (e.g. greater than, less than or equal to, between) return results based on proximity criteria, intersection and inclusion as specified by a point, line, circle or polygon
- Geospatial return results based on proximity criteria
- **Search queries** queries return results in relevance order and in faceted groups, based on text arguments using Boolean operators (e.g., `AND`, `OR`, `NOT`)
- **Aggregation Pipeline** queries return aggregations and transformations of documents and values
- **JOINs and graph traversals**

Sharding

- Sharding is a way to achieve horizontal scalability
- MongoDB supports following:

- **Range Sharding**. Documents are partitioned across shards according to the shard key value
- **Hash Sharding**. Documents are distributed according to an MD5 hash of the shard key value
- **Zone Sharding**. Provides the ability for DBAs and operations teams to define specific rules governing data placement in a sharded cluster

Consistency and availability model

- MongoDB provides ACID properties at the document level.
  - The ACID ensure complete isolation as a document is updated; any errors cause the operation to roll back so that clients receive a consistent view of the document.
- MongoDB's Write Concerns allow to commit to the application only after they have been flushed to the journal file on disk
- Each query can specify the appropriate write concern, such as writing to at least two replicas in one data center and one replica in a second data center

Indexing

- MongoDB includes support for many types of secondary indexes that can be declared on any field in the document
- The list of supported indexes:
  - Unique Indexes
  - Compound Indexes
  - Array Indexes
  - TTL Indexes
  - Geospatial Indexes
  - Partial Indexes
  - Sparse Indexes
  - Text Search Indexes.

MongoDB, components & utils

```
COMPONENTS

  mongod - The database server.
  mongos - Sharding router.
  mongo  - The database shell (uses interactive javascript).

UTILITIES

  mongodump          - Create a binary dump of the contents of a database.
  mongorestore       - Restore data from the output created by mongodump.
  mongoexport        - Export the contents of a collection to JSON or CSV.
  mongoimport        - Import data from JSON, CSV or TSV.
  mongofiles         - Put, get and delete files from GridFS.
  mongostat          - Show the status of a running mongod/mongos.
  bsondump           - Convert BSON files into human-readable formats.
```

Installing MongoDB

- Binaries are available for all major platforms (Linux, OS X, Windows)
  - Easy way
  - https://docs.mongodb.com/tutorials/
- You can also build from sources
  - https://github.com/mongodb/mongo
  - https://github.com/mongodb/mongo/wiki/Build-Mongodb-From-Source
  - Hard way, but you can grab the latest version
- Go ahead and install MongoDB on your laptop (you can work in groups)
- Run with `mongod`

Importing data

Once you are done, import data from the following link:

https://raw.githubusercontent.com/mongodb/docs-assets/primer-dataset/primer-dataset.json

With the following command:

```
mongoimport --db test --collection restaurants --drop --file ~/downloads/primer-dataset.json
```

To import data into a mongod instance running on a different host or port, specify the hostname or port by including the `--host` and the `--port` options in your mongoimport command

Connecting to MongoDB from Python

- Install Python interface with
  - `pip install pymongo`
- Import MongoClient
  - `from pymongo import MongoClient`
- Connect to running MongoDB instance
  - `client = MongoClient("mongodb://hostname:port")`
  - `client = MongoClient("mongodb://localhost") # will connect to localhost and default port 27017`
- Access database of interest
  - `db = client['db_name']`

Query data

- Find data inside collection:
  - `cursor = db.collection_name.find({key1: value, key2: value2 ...})`
  - if no parameters are supplied, all data will be returned (as in SELECT *)
  - cursor is a generator which can be iterated through using standard Python syntax

Exercise 1

Create functions to execute the following tasks:

Query all Indian cuisines

Query all Indian and Thai cuisines

Find a restaurant with the following address: 1115 Rogers Avenue, 11226

Insert data

- Insert data into collection:
  - `result = db.collection_name.insert_one({key1: value, key2: value2 ...})`
    - Returns id of the inserted record
  - `result = db.collection_name.insert_many([{key1: value, key2: value2 ...}, {...}, ...])`
    - Returns ids on inserted records

Exercise 2

Create a function to execute the following task:

Insert into the database following restaurant:

- Address: 1480 2 Avenue, 10075, -73.9557413, 40.7720266
- Borough: Manhattan
- Cuisine: Italian
- Name: Vella
- Id: 41704620
- Grades:
    - A, 11, 01 Oct, 2014

Delete data

- Delete data from collection:
  - `result = db.collection_name.delete_one({key1: value, key2: value2 ...})`
    - Deletes single record, returns number of deleted records
  - `result = db.collection_name.delete_many({key1: value, key2: value2 ...})`
    - Deletes all items matching the condition

Exercise 3

Create functions to execute the following tasks:

Delete from the database a single Manhattan located restaurant

Delete from the database all Thai cuisines

Exercise 4

Create functions to execute the following tasks:

Query all restaurants on the Rogers Avenue street, for each of them do the following:

- if it has more than one C grades → delete this restaurant
- otherwise, add another C grade to this restaurant