DB - Study Point Assignment 3 - Mongo DB

Mathias Drejer, Tobias Linge
Robert Pallesen

April 2023

The following is to be answered in the assignment

- What is sharding in mongoDB?
  Sharding is a method for distributing data across multiple machines. MongoDB uses sharding to support deployments with large data sets.
  Database systems with large data sets can challenge the capacity of a single server.
  There are two methods for addressing system growth: vertical and horizontal scaling.

  Vertical scaling means increasing the capacity of a single server, such as more computer power (addition of a more powerful CPU, more RAM, more storage space etc.).

  Horizontal scaling means diving the system data set and load over multiple servers, adding additional servers to needed capacity. While the overall speed or capacity of a single machine may not be high, each machine handles a subset of the overall workload, potentially providing better efficiency than a single high-speed/high-capacity server.

  Sharding is a form of horizontal scaling, because we're adding multiple servers to handle additional data.

- What are the different components required to implement sharding?
  To implement sharding in you need to have 5 components. **Shard cluster, config servers, shards, routers and balancing tool.**

  1. **Shard cluster** are the instances that work together to store and manage data across multiple machines.

  2. **Config servers** are special MongoDB servers that stores important informationabout how data is organized across all the different MongoDB servers in a sharded cluster. They keep track of things like which server has which data nd which data should be stored on which server.

  3. **Shards** are the individual MongoDB instances that store a subset of the data in the cluster. Each shard can be deployed as a standalone server or even as a replica set to provide redundancy.

  4. **Routers** are the access points for the client application to connect to the sharded cluster. They receive queries from client applications and route them to the appropriate shard(s), based on the metadata stored on the config servers.

  5. **Balancing tools** is a built-in tool in MongoDB that automatically distributes data evenly across the shards. It monitors the distribution of data and can move data between shards as needed to maintain balance and the most optimal performance.

- Explain architecture of sharding in mongoDB?

  Sharding in MongoDB is a technique that horizontally partitions data across multiple servers to allow for scalable data storage and processing for large-scale applications. The architecture of sharding in MongoDB consists of three main components: shard servers, config servers, and query routers.

  Shard servers are instances of MongoDB that store a subset of the data in the cluster. Each shard server is responsible for storing and managing a portion of the data, determined by a shard key.

  Config servers store metadata about the cluster, including information about the shard key range and which data is stored on each shard server.

  Query routers, or mongos instances, receive queries from the application and route them to the appropriate shard server based on the shard key range.

- Provide implementation of map and reduce function

```
        from pymongo import MongoClient

client = MongoClient('mongodb://127.0.0.0:27017/')
db = client['twitter']

map_func = '''function() {
    emit(null, { text: this.text, likes: this.likes });
}'''

reduce_func = '''function(key, values) {
    values.sort(function(a, b) {
        return b.likes - a.likes;
    });
    return values.slice(0, 10);
}'''

result = db.tweets.map_reduce(map_func, reduce_func, "top10tweets")
for doc in result.find():
    print(doc)
```

- Provide execution command for running MapRecude or the aggregate way of doing the same

```
    db.tweets.mapReduce(
  function() {
    emit(null, { text: this.text, likes: this.likes });
  },
  function(key, values) {
    values.sort(function(a, b) {
      return b.likes - a.likes;
    });
    return values.slice(0, 10);
  },
  { out: ”top10tweets” }
);

db.tweets.aggregate([
  { $project: { text: 1, likes: 1 } },
  { $sort: { likes: -1 } },
  { $limit: 10 }
]);
```

- Provide top 10 recorded out of the sorted result. (hint: use sort on the result returned by MapRecude or the aggregate way of doing the same)

  - FCBlive(11 times)
  - IWCI(5 times)
  - EspanyolFCB(4 times)
  - BPL(3 times)
  - NapalEarthquake(3 times)
  - job (2 times)
  - Nepal(2 times)
  - Xavi500(2 times)
  - ATTHack(2 times)
  - WBA(2 times)

Optional questions:

- Show what happens to the data when one shard is turned off
- Show what happens to the data when the shard rejoins
- Explain how you could introduce redundancy to the setup above