DISTRIBUTED DATABASE

PROJECT FINAL REPORT

By Lingqi Shi,Dongjie Zhao

JUNE 17, 2019

## Project background

With the rapid development of social economy and technology, people's material life enriched, followed by people's psychological problems, more and more people begin to pay close attention to mental health, increasing juvenile psychological pressure too much stress can lead to mental health problems, reading therapy is a kind of by reading books or other contact information material, to help readers to relieve negative emotions, thus achieve the balanced state of body and mind.Based on this problem, the project developed a platform for reading sharing and recommendation to reduce the psychological problems of teenagers through reading.

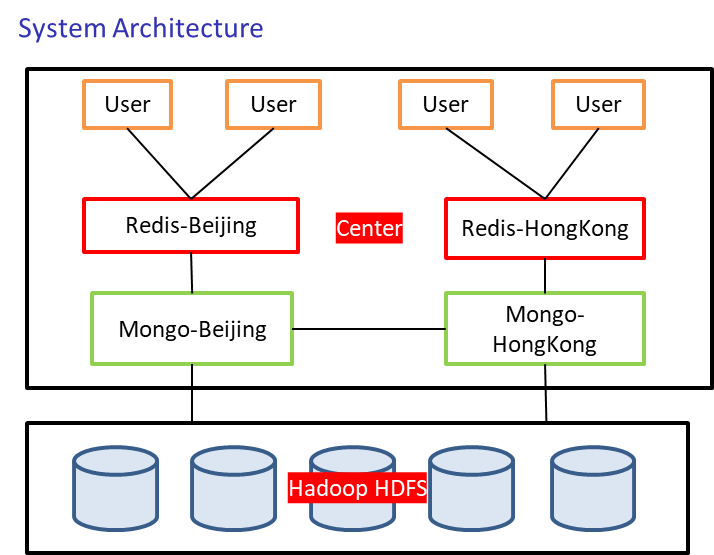
"Histories make men wise; poets witty; the mathematics subtle; natural philosophy deep; moral grave; logic and rhetoric able to contend.This is the famous British materialist philosopher and scientist Francis Bacon a famous paragraph, that reading plays a very important role in shaping people's character.Liu xiang, a Chinese writer in the han dynasty, said, "books are like medicine.In fact, in addition to "into character" and "treatment of the fool", reading does have the effect of curing disease, especially for psychosocial factors caused by diseases, such as depression, anxiety, panic, worry, etc. The first theoretical work on reading therapy was born in mainland China. It is wang bo's reading therapy.

## Existing solutions

**MongoDB** is a product between relational database and non-relational database. Among non-relational databases, MongoDB has the most abundant functions and most resembles relational database. MongoDBis a typical Nosql data structure, which is more flexible and efficient. Meanwhile, MongoDB itself provides an efficient caching mechanism.

**Cache** provides more efficient system performance.Both Redis and memcached are caches and run in memory, greatly increasing the speed of high-volume web access.Memcached, however, provides simple data structures, such as string storage.Redis, on the other hand, offers a number of data structures, such as string, list, set, hashset, sorted set. That makes things so much easier.

#### System Architecture



**Cache applicable scenario**  
1. The operation of user query data is far greater than updating and inserting data.

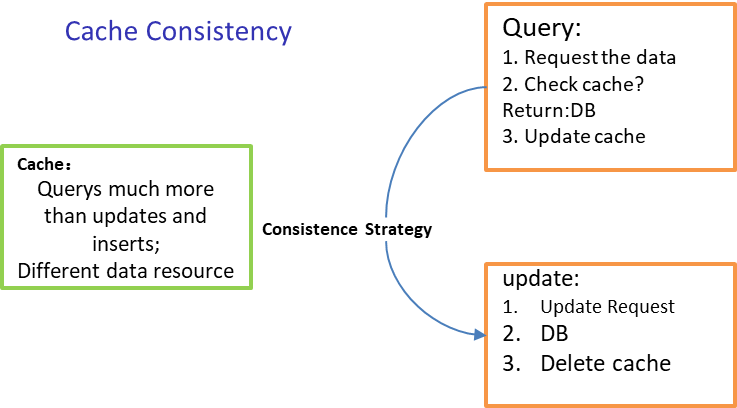
1. The system reads data from different data sources, such as MYSQL and HDFS.

**When querying data**

The server first queries the local cache and returns if there is one.Otherwise, query the database.If the database exists, update the local cache and return.

**Update and write data**

The user writes the request to the server.System execution of SQL to database. A delete request is also sent to the local cache.The decision to delete rather than update was made primarily to ensure consistency.Database management is to add, delete and check, transparent to users.



#### Solutions to Choose

Docker

* The Driving force behind the containerized applications.
* consistency,cross platforms and fast configuration
* light-weight and easy

MongoDB

* NoSQL Database
* More flexible
* Hierarchical structure by subdocuments
* opensource
* scalability and performace

Redis

* In-memory key-value store
* Memcached better for small and static data
* Redis: more structural and powerful

#### System Implementation

**Server-end**

1. Basic Data Generation

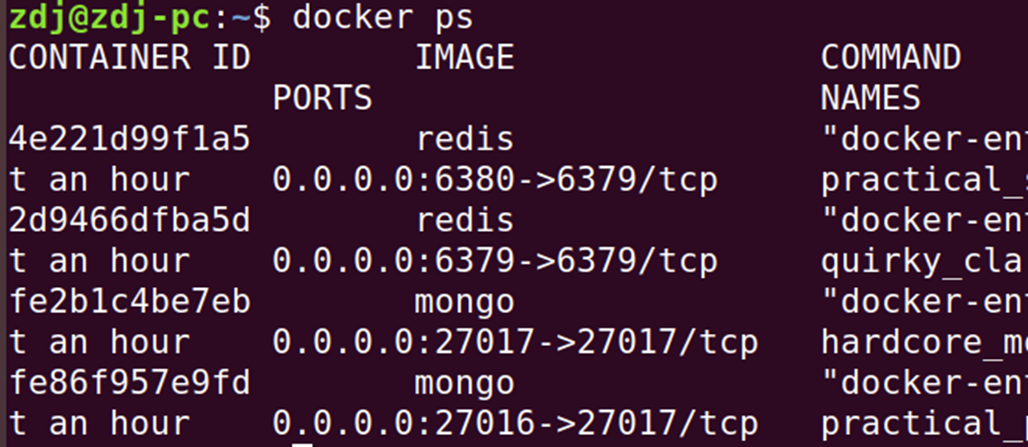
* Import to Mongo
* Caculate be-read and popular-rank (everyday) table based on the existing data
* Redis
* Effective data manipulation

1. Data generation

* set the test nums
* run the python file

1. Mongo and Redis

* Docker image
* two containter for different ports for both redis
* tools to monitor the db status
* adminmongo and redisboard



1. Data import and calculate

* import three base tables into mongo
* considering the partition and duplication
* calculate be-read and popular-rank table and update in the redis

1. Data munipulate

* python
* related libs, including pymongo, redis
* excute the commands from client-end

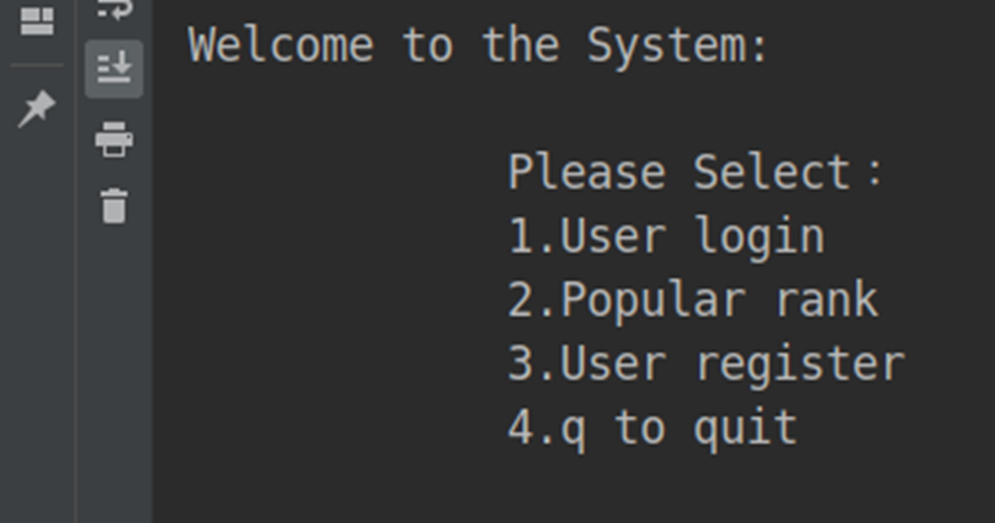
**Client-end**

1. Command-line ui

* Unconsious of the distribute environments
* Friendly hints
* Basic operations

1. System entry

* user login
* popular
* user register
* Quit

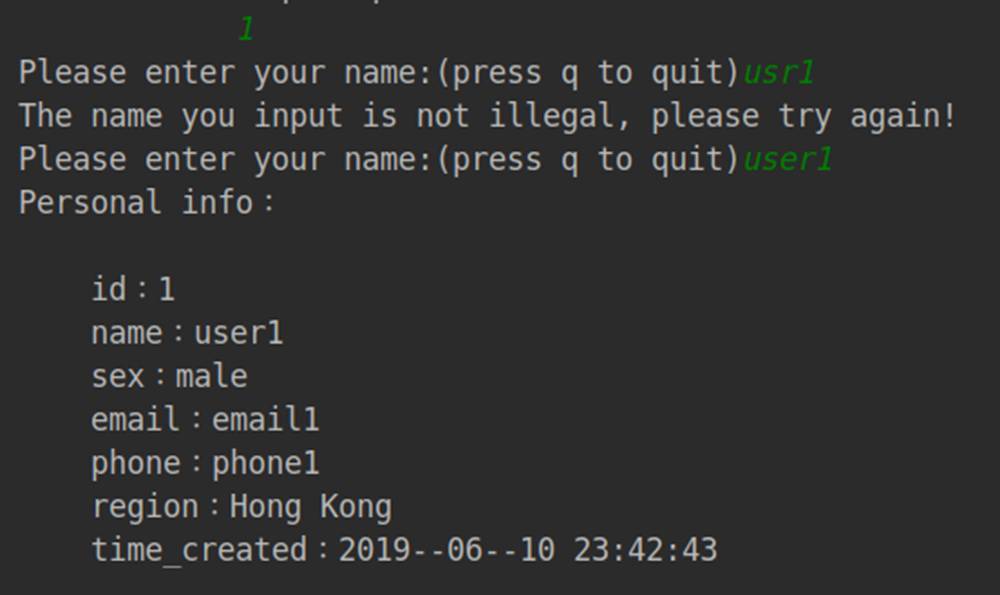
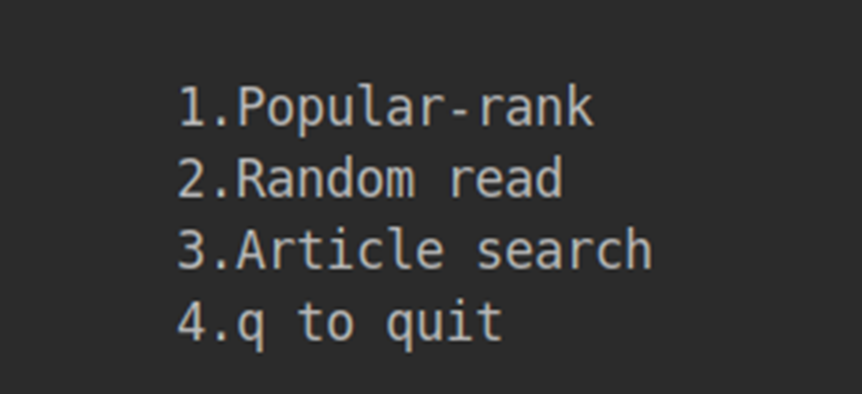


1. user login

* by username

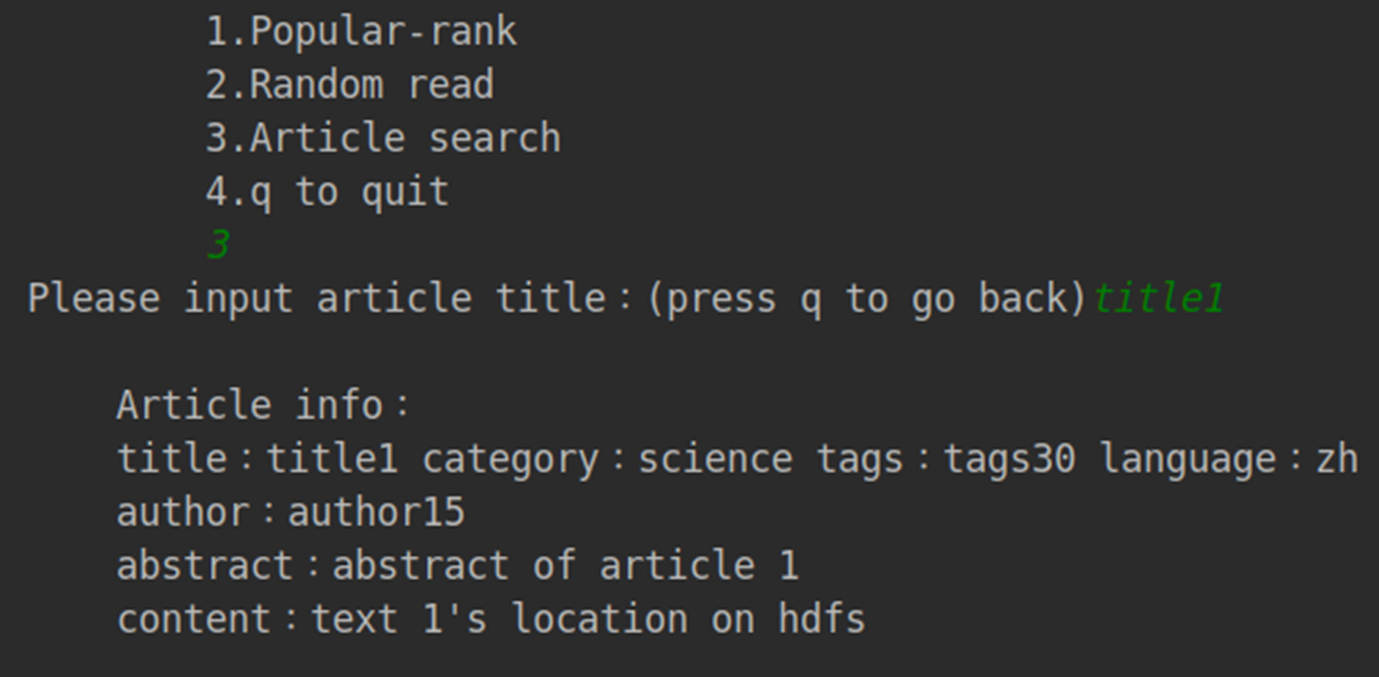
1. after login

* 1.Popular-rank
* 2.Random read
* 3.Article search
* 4.q to quit

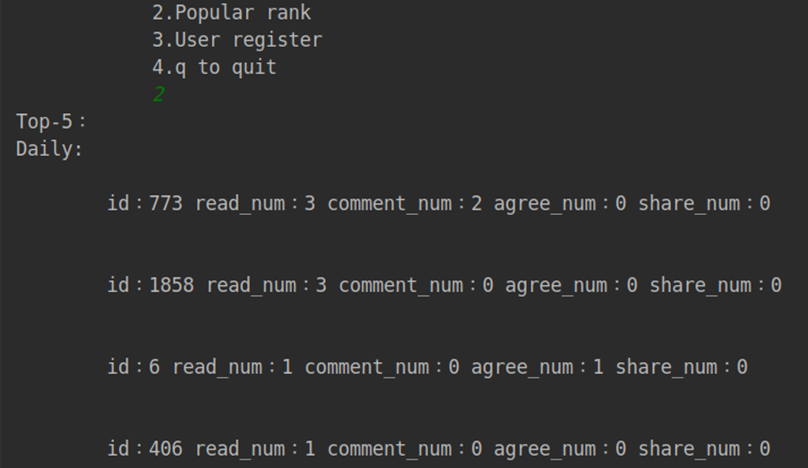
 

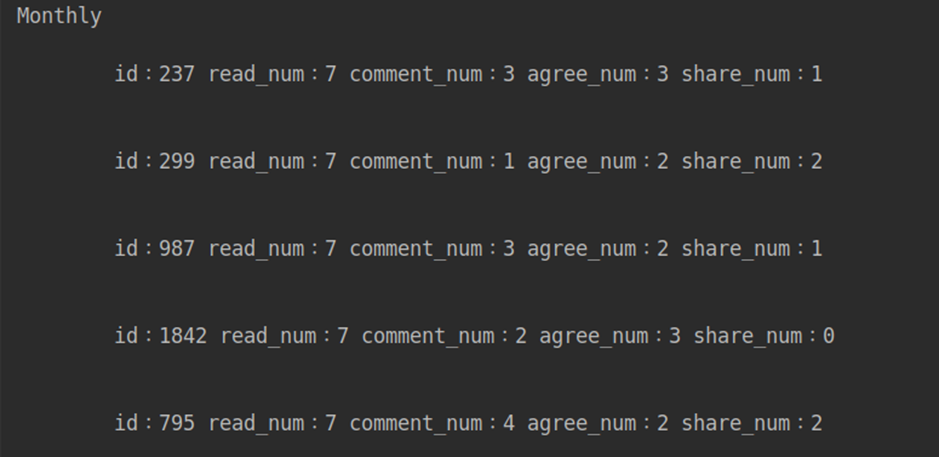
1. Article search

* by title



1. Popular rank

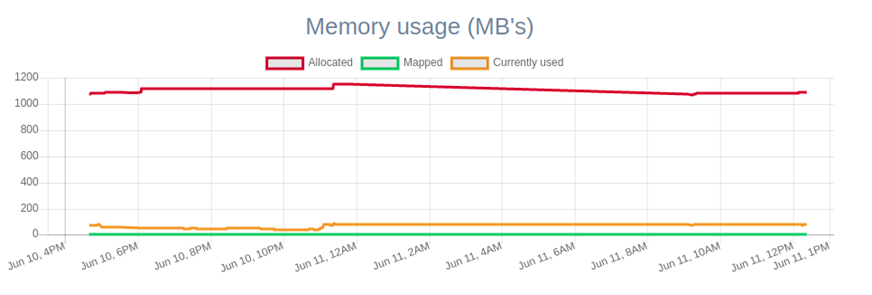


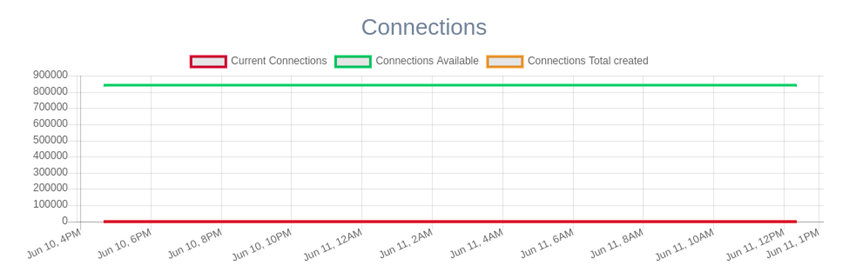


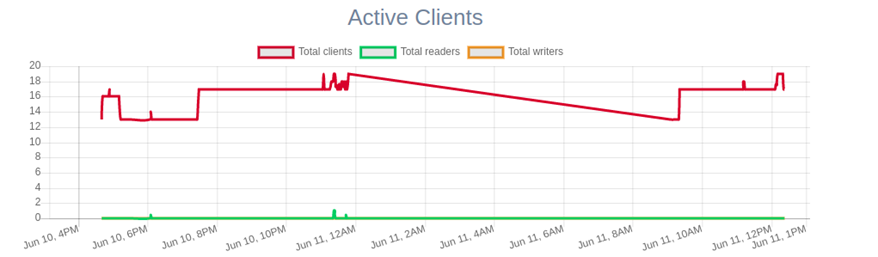
1. Database

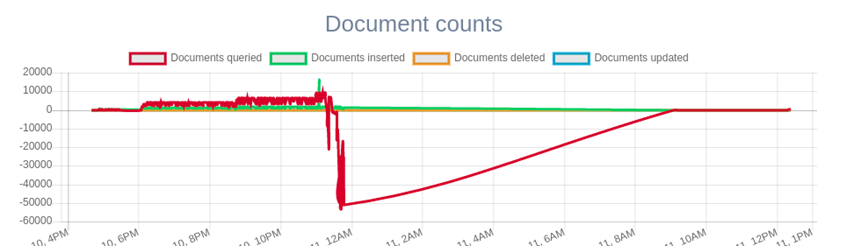
* mongo tables
* mongo status





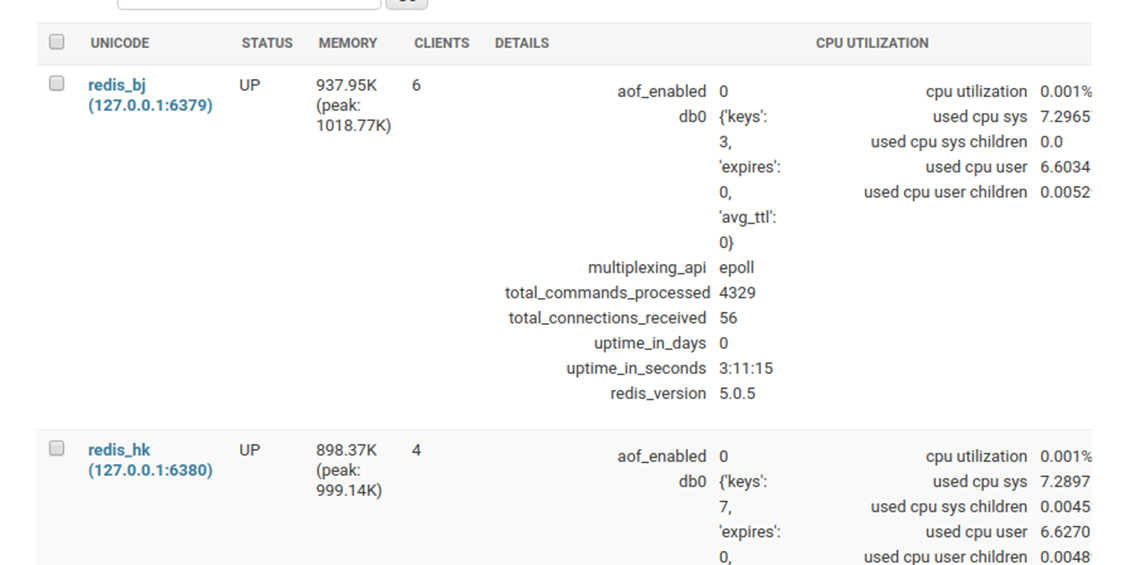






1. Redis

* first check
* Update



#### Conclusion

By setting up the distributed database system, we understand how to realize the distributed database system, further understand its principle and implementation process, and basically complete the task. And lay a solid foundation for further engaging in related tasks in the future.