

SIMPLIFIED DISTRIBUTED DATABASE

Exploring the Redis Replication Clock

By Muzhou Liu
For CS6650, SP2024

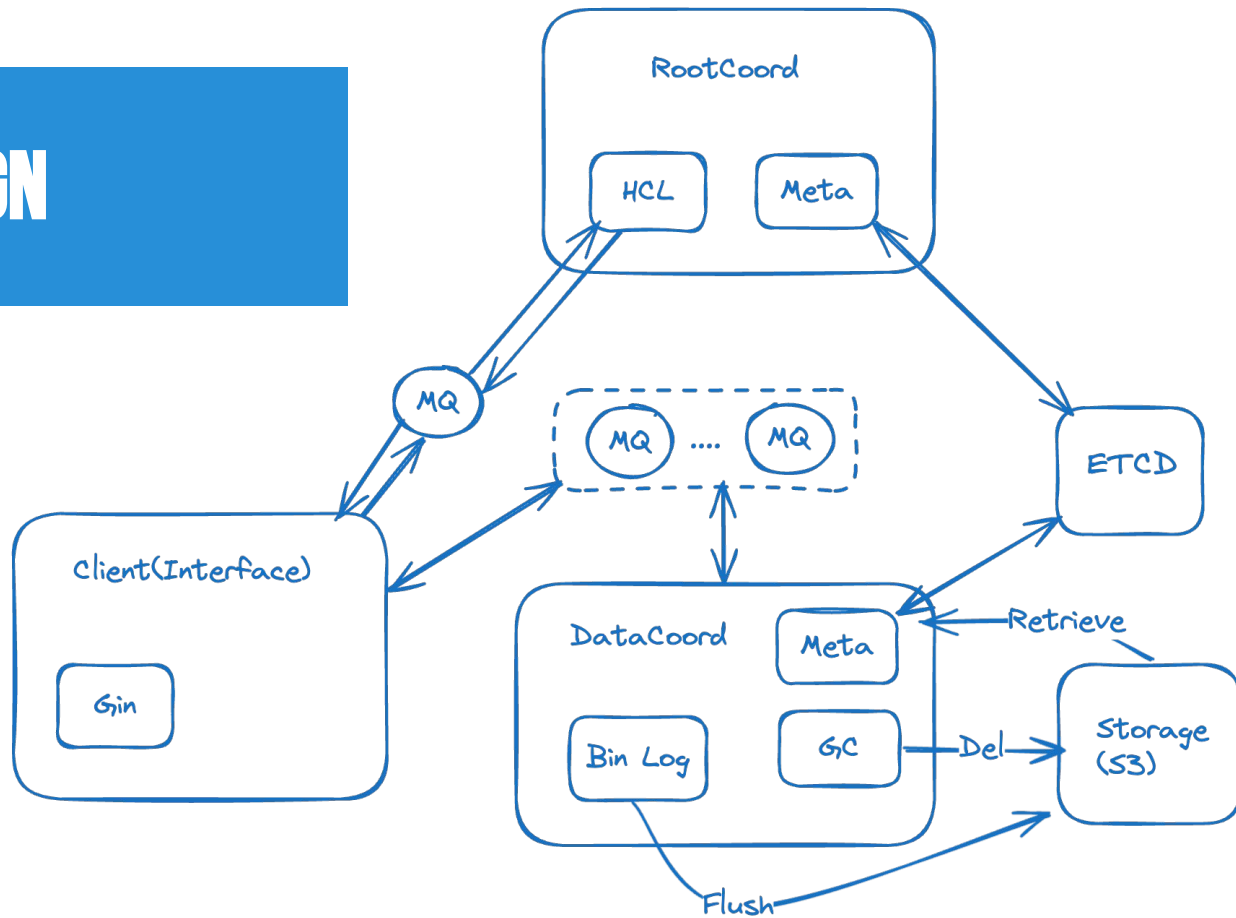


01

ABOUT

Want to explore more on a distributed database and trying to implement a simplified version to better understand the whole process

DESIGN



TIME STAMP

For each message that the client creates, it apply for a timestamp from the Root Coordinator and put it into MQ message body for Data Coordinator.

```
type atomicTsObject struct { 4 usages
    physical time.Time
    logical  int64
}
```

```
type timeStampNode struct { 6 usages
    key      string
    kv       *EtcdNode
    TS0      unsafe.Pointer
    lastSavedTime atomic.Value
}
```

...	Query Ts	...	Service Ts	...	Query Ts	Service Ts
-----	----------	-----	------------	-----	----------	------------

For Query task, we add a service Ts to ensure we are getting the latest Info of the stored data. Service Ts is periodically (50ms) update to meta data and Query will wait for the closest Service Ts to return the result.

EVALUATION

Generated a Sequence of operation and added some forced fail in operation. In the end, check the log, compare the Metadata and stored Json file on S3 to check correctness

SL

You



thread1.json

File

help me generate a JSON file that generates 100 operations like the format in the file with the following instruction

job can be "create", "update", "delete"

fileID can be 1 to 5

data is a string of digits less than 100

keep forceFail can be "true" in totally 10 times and please keep these "true" in the middle

< 3 / 3 >



ChatGPT

Let's proceed by creating a JSON file with 100 operations based on the format you've shared. We'll randomize the values for ``job``, ``fileID``, and ``data``, and ensure that the ``forceFail`` is ``"true"`` for 10 occurrences, strategically placed in the middle of the dataset.

I'll generate this JSON file for you now.

The JSON file with 100 operations has been created as per your instructions. You can download it using the link below:

[Download the operations JSON file \[~\]](#)

SL

You

generate two more file like this and name them thread2 and thread3



ChatGPT

The two additional JSON files, named ``thread2.json`` and ``thread3.json``, have been created.

You can download them using the links below:

- [Download thread2.json](#)
- [Download thread3.json \[~\]](#)

FUTURE WORKS

Implement the WAL (write after log) for 2 purpose

- Better verify the correctness of the system
- Help restore the system from disruption

THANKS

<https://github.khoury.northeastern.edu/mzh1111/Simple-Distributed-D-B-HCL>

<https://github.com/milvus-io/milvus>

CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**.