

Muxi Lai

15521144359 | 1721261216@qq.com | peking

EDUCATION

Xidian University

Sep 2020 - Jun 2024

Software Engineering Bachelor Computer Science and Technology

XiAn

- GPA: 3.7/4.0
- Related courses: Operating Systems, Distributed Systems, Databases

SUMMARY

- Familiar with go language development, studied mit6.824
- Studied mit6.s081
- know some database knowledge, studied cmu15-445, tinysql, tinykv
- Read part of the source code of etcd-raft
- Good communication skills and teamwork ability, love learning, strong self-learning ability, strong ability to find information and read documents.

PROJECT EXPERIENCE

bustup

Learned cmu15-445 and implemented the following features :

- Added buffer pool to bustup database, implemented lru replacement policy and database page management.
- Added Extendable Hashing to bustup as the main data structure to store indexes and use it to implement hash join.
- Added multiple executor functions to bustup, using the volcano model, such as insert, delete, seqScan, aggregate, hash join, nested join, etc.
- Added lock manager and concurrency control to bustup

tinykv

tinykv is used as the underlying row kv storage for tinysql (distributed relational database). The following functions are currently implemented:

1. raft basic functions: leader election, append log, snapshot, etc.

raft optimization: such as returning conflict points to speed up log sending, using snapshot tagging to reduce the number of network io, implementing configuration changes (single step change) and leader conversion.

2. distributed kv layer, build raftstore to use raft, which implements: message processing, ready application, truncated log, application snapshot, configuration change and leader conversion and region split.

Optimization points are: batching optimization of messages, i.e., write multiple requests to a message and then reply uniformly to reduce network io.

3. Transaction layer: implements the percolator algorithm of kv storage layer, including : prewrite, commit, resolve lock, check status, etc.

tinysql

tinysql is a distributed relational database, the underlying use tinykv into row storage, the current implementation of the following functions :

1, the encoding of table info, index info storage, such as unique index using the following encoding: key: tablePrefix{tableID}_indexPrefix{indexID}; value : rowID

2, parser using goyacc and pingcap homebrew lexer.go for sql text parsing

3、 schema uses Google F1's asynchronous change algorithm

4, optimizer layer using System R framework (logical optimization -> physical optimization), physical optimization using information statistics as a cost analysis

5, the executor layer uses the volcano model + vector model (transition from the volcano model to the vector model), and implements some functions such as: hash join, hash aggregate, etc.

6, transaction layer with percolator as the implementation, and tinykv percolator with the implementation of prewrite, commit, resolve lock, check txn, snapshot get and other functions

PRACTICAL EXPERIENCE

ByteDance

Jan 2023 - Present

Recommended Architecture Development Intern data-arch-ies

peking

Douyin recommends architecture development interns

- Responsible for video multi-queue optimization for predict
- Responsible for the development of video isolation, that is, parallel operations on multiple videos, including recall and rough sort for recommender systems

Dolphindb

Database Development Intern

- Responsible for reverse subscription and port multiplexing of streaming data
- Responsible for the development of high-availability flow tables, such as realizing high availability of flow tables based on raft
- Responsible for development related to data playback, such as specifying additional field sorting for heterogeneous flow table playback, not restricting heterogeneous flow table partitions, and querying memory leaks during playback, etc.