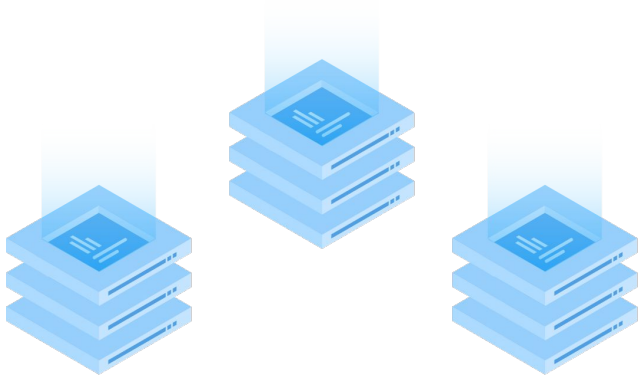
# Lesson 05: High Performance DDL

**Presented by Xie Tengjin**

PingCAP TiDB



# Before we begin

- Context: TiDB DDL Architecture
- Goal :  Learn the architecture and the optimizations of TiDB DDL module
- Outline:
  - Overview
  - Optimizations
    - Parallel Optimization
    - Instant Optimization
  - Homework
- Readings:
  - DDL source code reading
  - DDL schema change implementation
  - DDL schema change optimization

# Part I: Overview

# Data Definition Language

SQL Statements

- **Data Definition Statements**
- Data Manipulation Statements
- Transactional and Locking Statements
- Replication Statements
- Prepared Statements
- Compound Statement Syntax
- Database Administration Statements
- Utility Statements

# Data Definition Language

Used to **add/remove/change** the schema object in database. (Including schema, table, view, sequence, etc.)

- SQL Language
  - CREATE...
  - ALTER…
  - DROP…

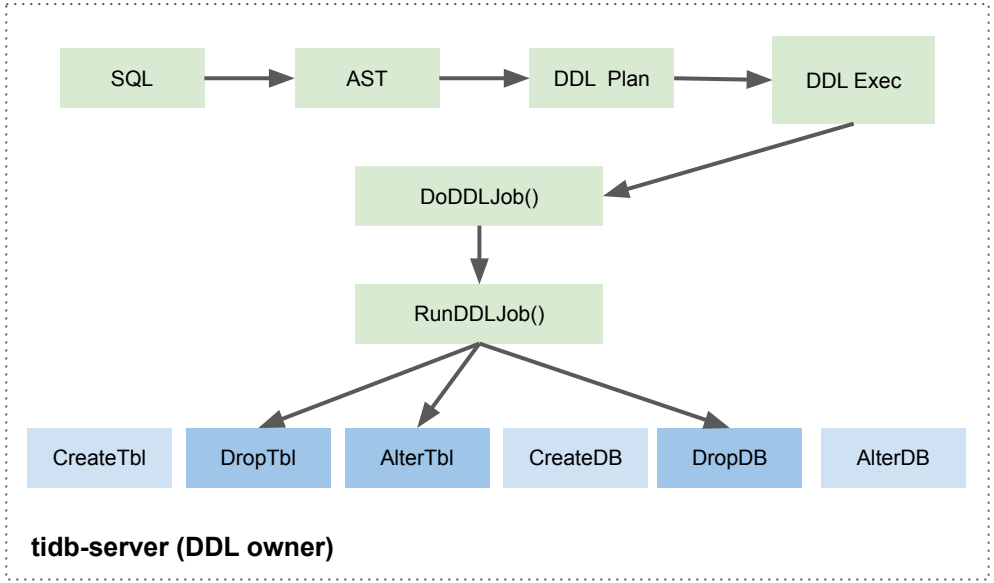
- DDL Golang Interface (code path: tidb/ddl/ddl.go)
  - CreateSchema
  - DropSchema
  - CreateTable
  - CreateView
  - DropTable
  - ...

# Data Definition Language

```go
// DDL is responsible for updating schema in data store and maintaining in-memory InfoSchema cache.
type DDL interface {
    CreateSchema(ctx sessionctx.Context, name model.CIStr, charsetInfo *ast.CharsetOpt) error
    AlterSchema(ctx sessionctx.Context, stmt *ast.AlterDatabaseStmt) error
    DropSchema(ctx sessionctx.Context, schema model.CIStr) error
    CreateTable(ctx sessionctx.Context, stmt *ast.CreateTableStmt) error
    CreateView(ctx sessionctx.Context, stmt *ast.CreateViewStmt) error
    DropTable(ctx sessionctx.Context, tableIdent ast.Ident) (err error)
    RecoverTable(ctx sessionctx.Context, recoverInfo *RecoverInfo) (err error)
    DropView(ctx sessionctx.Context, tableIdent ast.Ident) (err error)
    CreateIndex(ctx sessionctx.Context, tableIdent ast.Ident, keyType ast.IndexKeyType, indexName model.CIStr,
        columnNames []*ast.IndexPartSpecification, indexOption *ast.IndexOption, ifNotExists bool) error
    DropIndex(ctx sessionctx.Context, tableIdent ast.Ident, indexName model.CIStr, ifExists bool) error
    AlterTable(ctx sessionctx.Context, tableIdent ast.Ident, spec []*ast.AlterTableSpec) error
    TruncateTable(ctx sessionctx.Context, tableIdent ast.Ident) error
    RenameTable(ctx sessionctx.Context, oldTableIdent, newTableIdent ast.Ident, isAlterTable bool) error
    LockTables(ctx sessionctx.Context, stmt *ast.LockTablesStmt) error
    UnlockTables(ctx sessionctx.Context, lockedTables []model.TableLockTpInfo) error
    CleanupTableLock(ctx sessionctx.Context, tables []*ast.TableName) error
    UpdateTableReplicaInfo(ctx sessionctx.Context, physicalID int64, available bool) error
    RepairTable(ctx sessionctx.Context, table *ast.TableName, createStmt *ast.CreateTableStmt) error
    CreateSequence(ctx sessionctx.Context, stmt *ast.CreateSequenceStmt) error
```

# Execution Flow

DDL execution flow



# Components

Component & Role

- TiDB: update and load schema
- TiKV: store schema information
- PD: notify schema changes

# Concepts

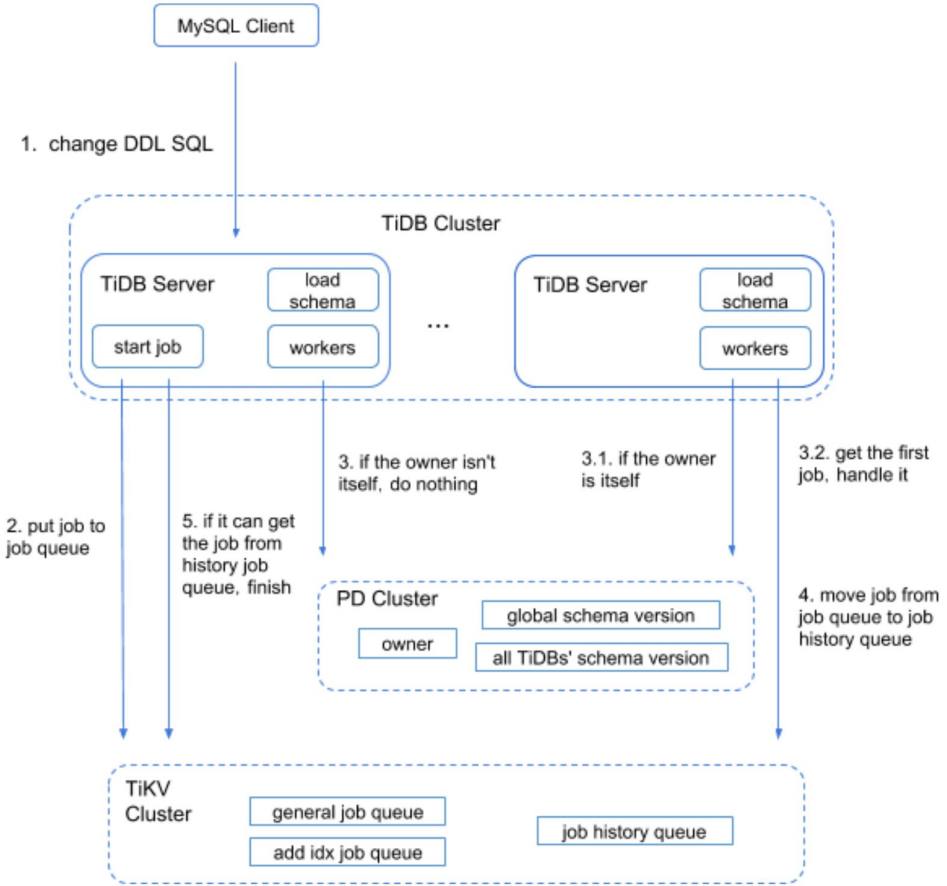
DDL Owner

- A role for the TiDB node
- At most 1 owner for each cluster
- Responsible to execute the DDL schama change

Job & Worker

- Each kind of DDL statement a represent by a “DDL Job”
- DDL jobs are initialized and put to a **job queue**
- The workers are responsible to execute the “DDL Job” by dequeuing

# Execution Flow



# Concepts

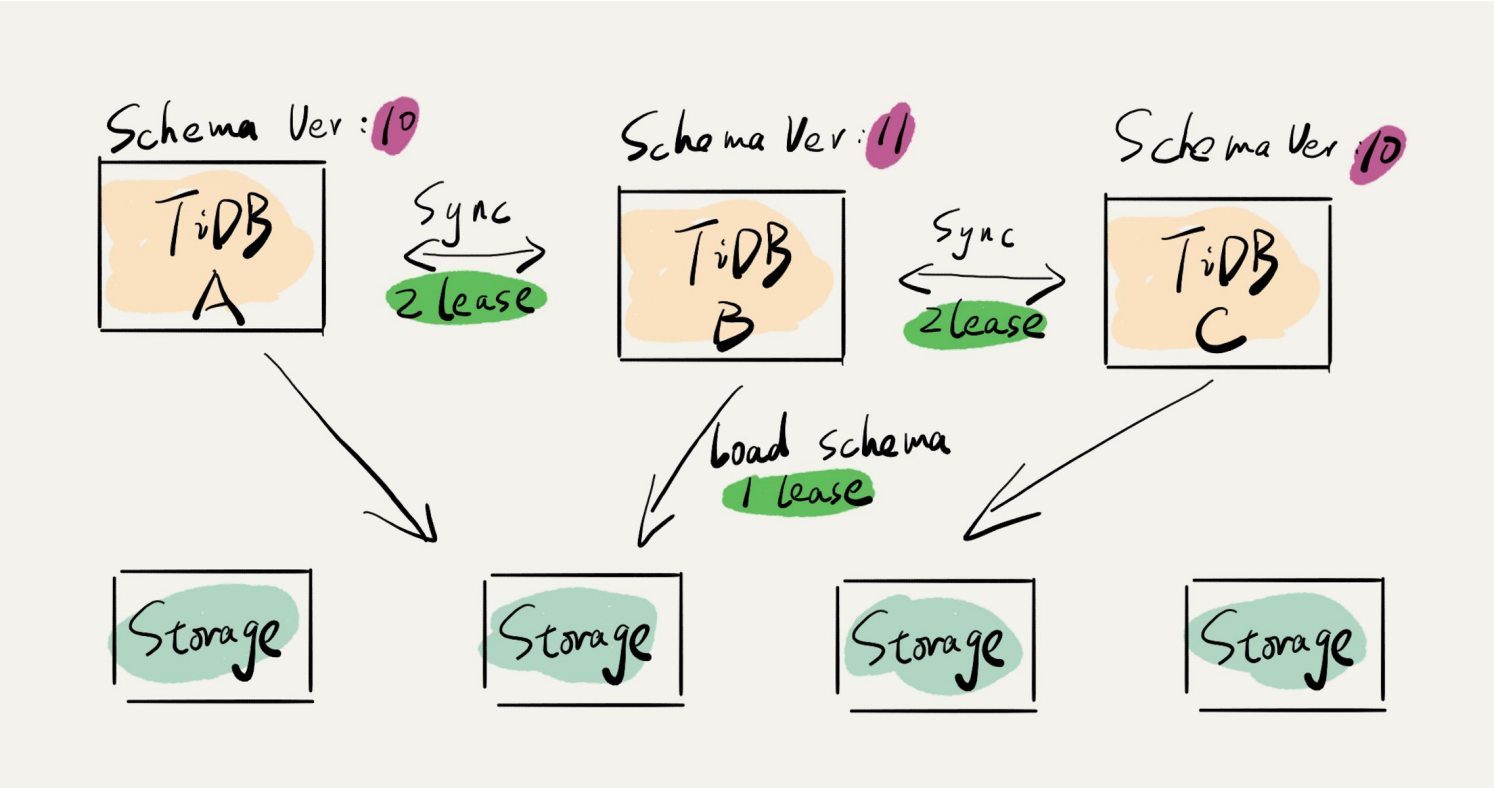
Schema Version

- Bind to a specific ‘snapshot’, represent a state of meta information
- Each DDL changes trigger schema version += n

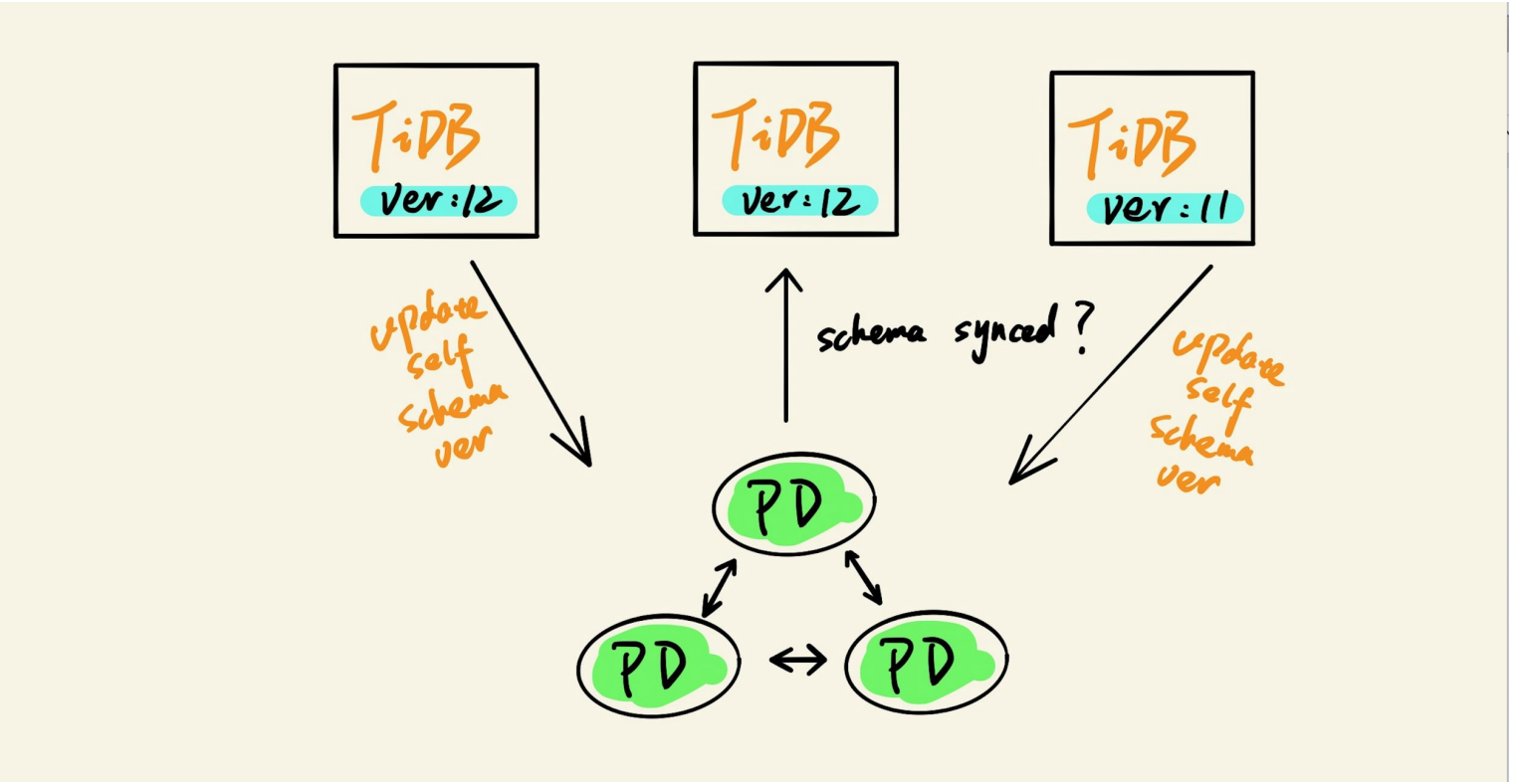
Schema Version Lease

- In a cluster, there are at most 2 schema versions
- In each lease, every node reload the schema information automatically
- The failed node delete itself from the cluster

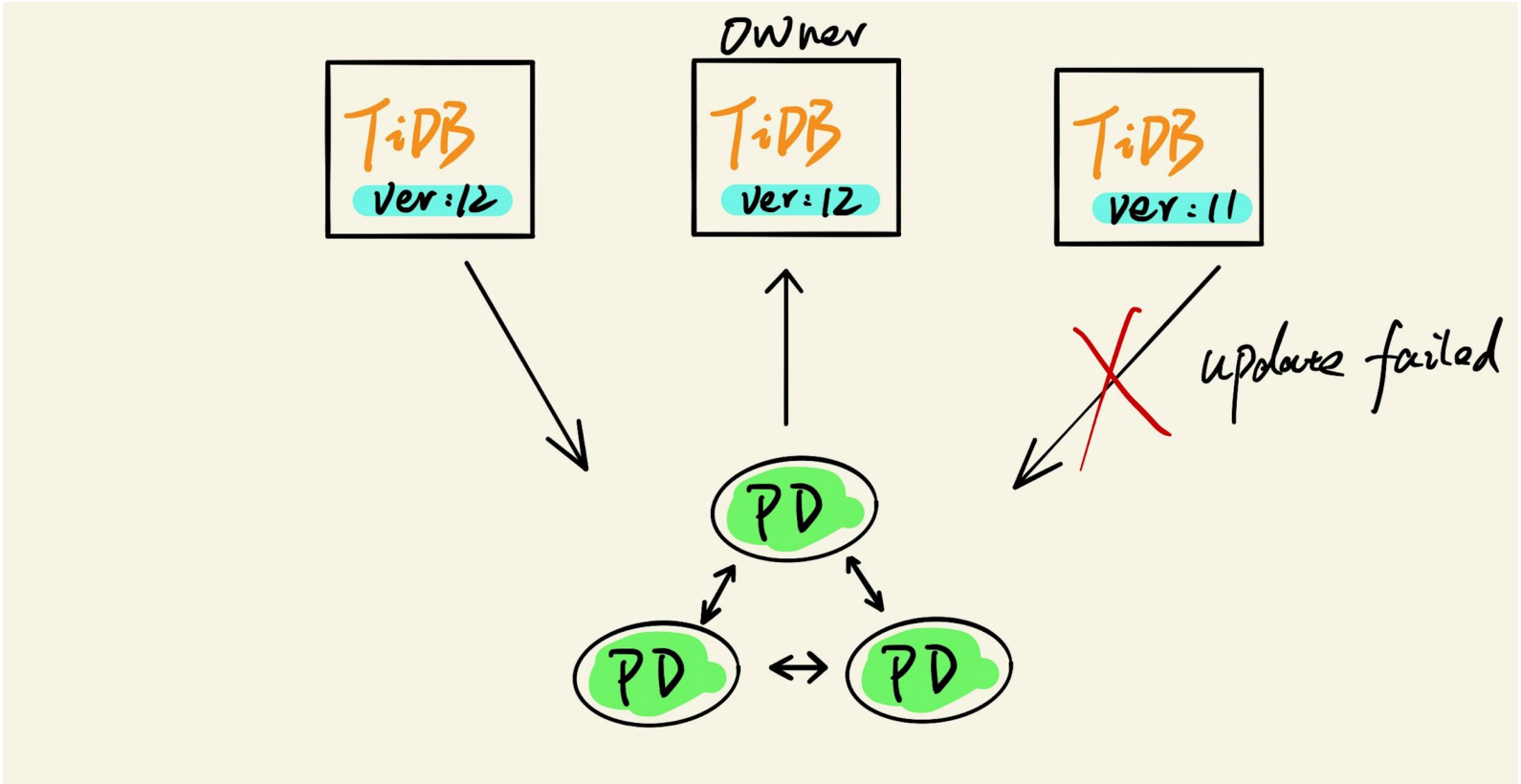
# An Example


Schema Ver : 10
TiDB A
Sync
2 lease
Schema Ver : 11
TiDB B
Sync
2 lease
Schema Ver : 10
TiDB C
Load schema
1 lease
Storage
Storage
Storage
Storage

# An Example


TiDB
ver:12
TiDB
ver:12
TiDB
ver:11
update self schema ver
schema synced ?
update self schema ver
PD
PD
PD

# An Example


Owner
TiDB
ver:12
TiDB
ver:12
TiDB
ver:11
update failed
PD
PD
PD

# The Consistency Problem

- DDL means the change of schema objects
- How to ensure the correctness of this change across multiple TiDB servers?
  - Drop Index & Insert
  - Add Index & Delete

# Concepts

"Online, Asynchronous Schema Change in F1"

Online DDL (no effect on other SQL statements)

- None
- Delete-Reorganization
- **Delete-Only**
- Write-Reorganization
- **Write-Only**
- Public

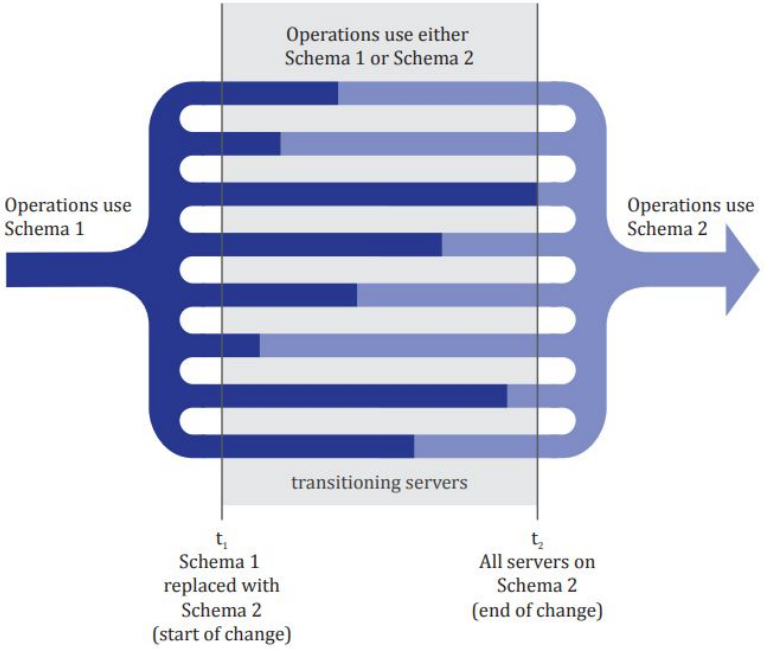



Figure 2: A view of the schema change process from Schema 1 to Schema 2 over time. Lines in the center of the figure represent individual F1 servers as they load Schema 2.

# Example

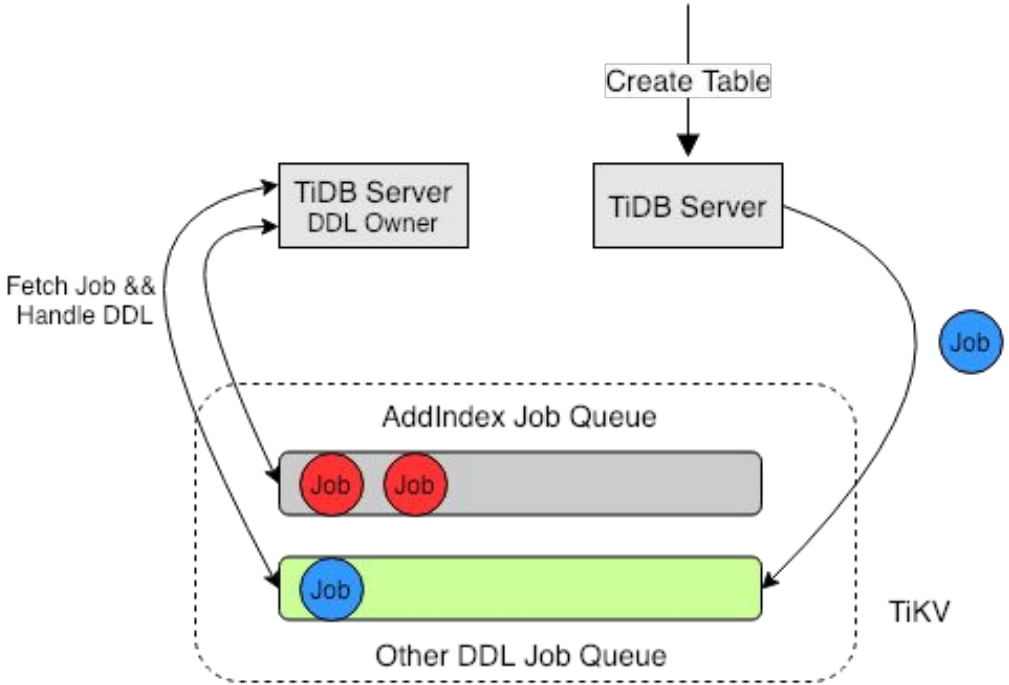
| Drop Index (DDL owner TiDB) | Insert (DDL non-owner TiDB) |
|---|---|
| Public | Public |
| Write-Only | Public |
| Write-Only | Write-Only |
| Delete-Only | Write-Only |
| Delete-Only | Delete-Only |
| Delete-Reorganization **(drop data)** | Delete-Only |
| Delete-Reorganization | Delete-Reorganization |
| None | Delete-Reorganization |
| None | None |

# Part II: Optimizations

# Parallelization Optimization

Parallel jobs:

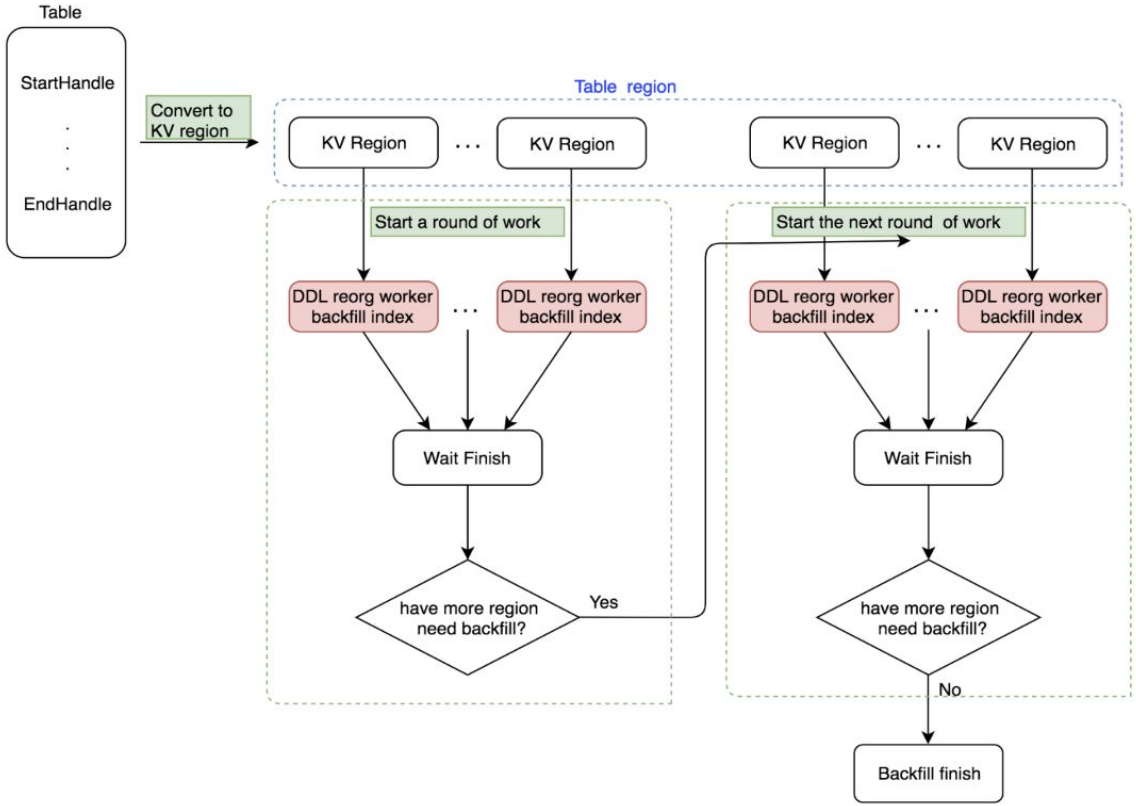
- Add index jobs
- Other jobs
- More?



# Parallelization Optimization

Parallel tasks:

- Add index task can be split into multiple ranges
- @@tidb_ddl_reorg_worker_cnt

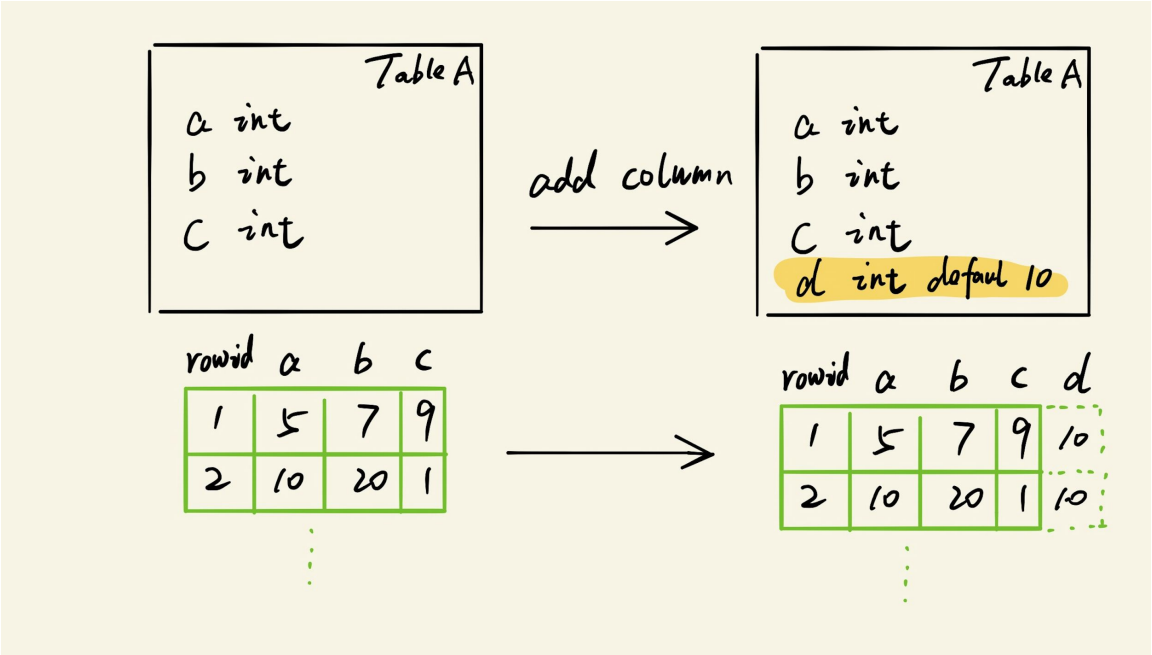


# Instant Optimization

O(1) add column

Store the default value to meta

- No need to reorganize

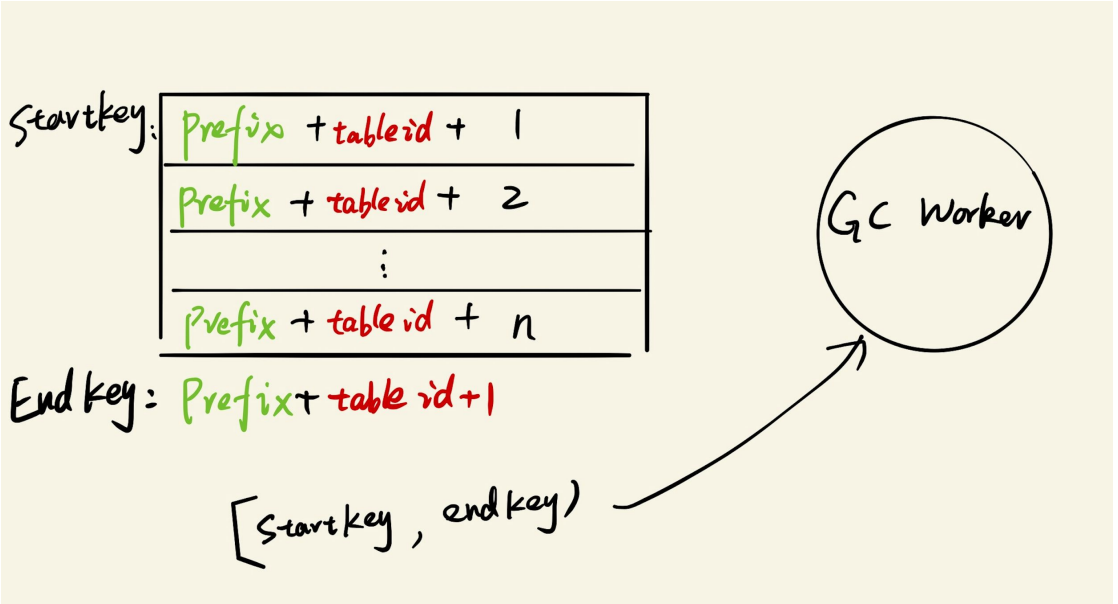


# Instant Optimization

Async Drop

Drop a database/table/index

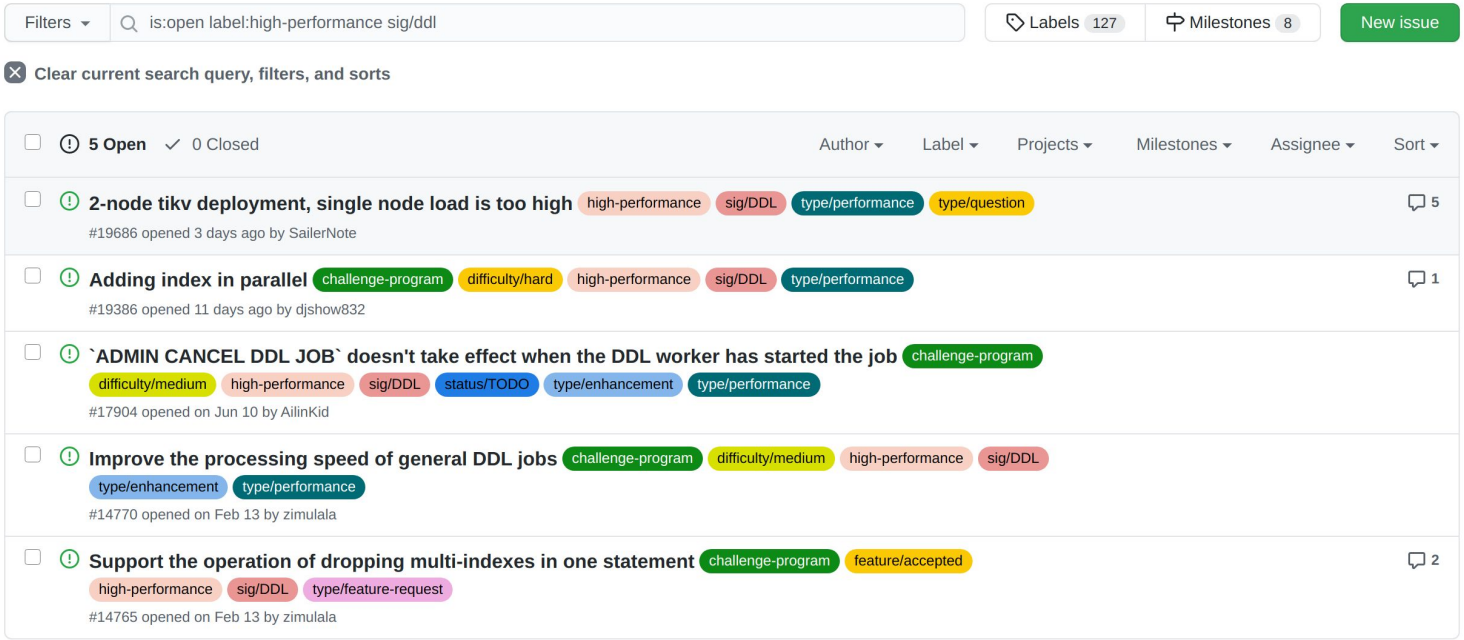
- mysql.gc_delete_range
- the real job is done by background GC worker



# Part III: Homework

# Homework

GitHub Issue: is:open label:high-performance sig/ddl



# Homework (1/6)

Score : 300

Description: Improve the processing speed of general DDL jobs

When the TiDB and DDL owner receiving the DDL requests are not on a TiDB, even if there is no data in the table corresponding to this DDL operation, it will take more than a second.

**GitHub issue:** issue-14770

# Homework (2/6)

Score : 600

Description: Support multiple table rename

mysql> RENAME TABLE t1 to t1_old, t2 to t1; <--- should work

ERROR 1105 (HY000): can't run multi schema change

**GitHub issue:** issue-9384

# Homework (3/6)

Score : 1200

Description: Support the operation of dropping multi-indexes in one statement

create table t(a int, b int, key idx1(a), key idx2(b));

alter table t drop index idx1, drop index idx2;

ERROR 8200 (HY000): Unsupported multi schema change

**GitHub issue:** issue-14765

# Homework (4/6)

Score : 1500

Description: `ADMIN CANCEL DDL JOB` doesn't take effect when the DDL worker has started the job

The cancelled state of job cannot be perceived by the DDL worker immediately.

**GitHub issue:** issue-17904

# Homework (5/6)

Score : 5100

Description: Adding index in parallel

Now adding index is processed only on the DDL owner. When the table is huge, it takes too much time. We can leverage the computing capability of the whole cluster to accomplish it.

**GitHub issue:** issue-19386

# Homework (6/6)

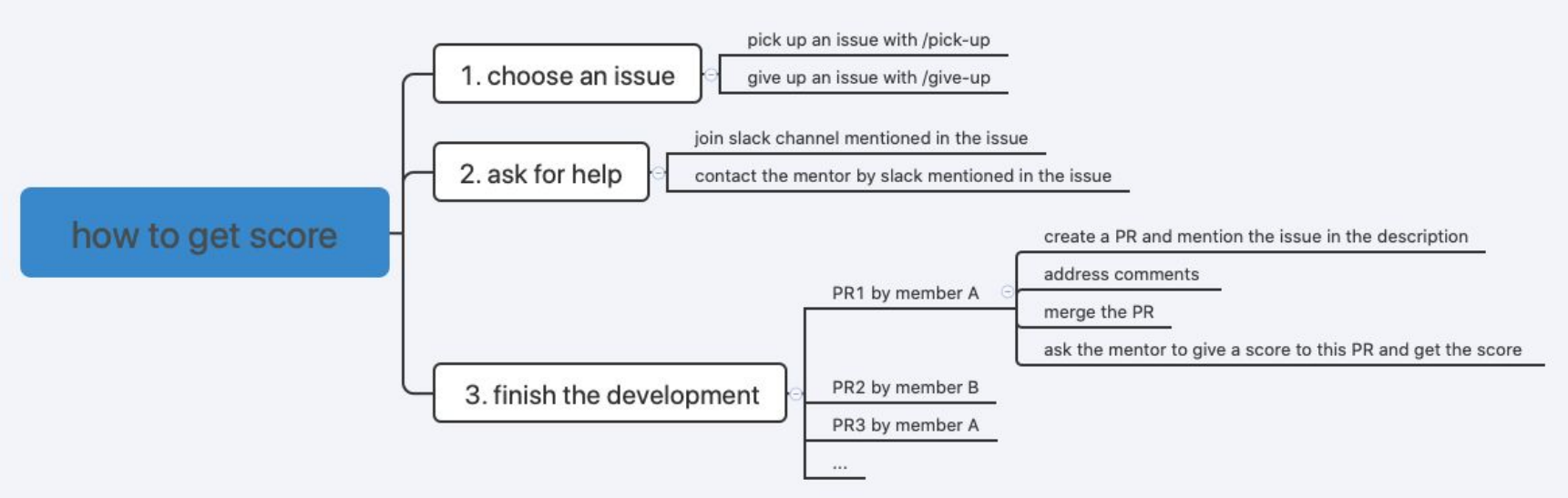
Score : 2100

Description: Design and implement a scheduling algorithm for ddl jobs

The internal ddl jobs should be put to a different job queue, in order to avoid blocking existing jobs because of the ‘in place’ algorithm.

**GitHub issue:** issue-19397

# 作业认领方式


how to get score
1. choose an issue
pick up an issue with /pick-up
give up an issue with /give-up
2. ask for help
join slack channel mentioned in the issue
contact the mentor by slack mentioned in the issue
3. finish the development
PR1 by member A
create a PR and mention the issue in the description
address comments
merge the PR
ask the mentor to give a score to this PR and get the score
PR2 by member B
PR3 by member A
...

# 作业认领相关命令

/pick-up

- 作用：issue 评论中回复认领 issue，如果是多人协作完成，派一个代表 pick 即可，对外只是标记这个任务已经有人在处理了。pick-up 完毕后，该 issue 会自动打上 picked 标签
- 权限：anyone
- 认领后：七天无动态认为该同学无法完成该任务，将自动 give-up

/give-up

- 作用：issue 评论中回复放弃当前认领的任务，give up 完毕后，该 issue 的 picked 标签会被移除
- 权限：当前挑战者

关联 PR 和 issue，PR 描述中按照以下方式之一关联 issue

- Issue Number: close #xxx
- Issue Number: #xxx

# 课程答疑与学习反馈


扫描左侧二维码填写报名信息，加入课程学习交流群，课程讲师在线答疑，学习效果 up up！

# 更多课程


想要了解更多关于 TiDB 运维、部署以及 TiDB 内核原理相关课程，可以扫描左侧二维码，或直接进入 http://university.pingcap.com 查看

# Thank you!