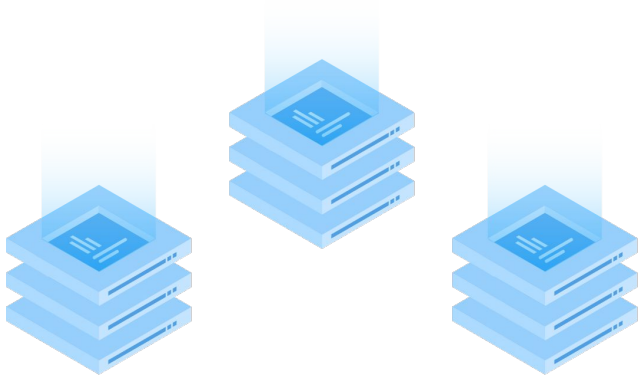
Source code reading of TiDB Transaction

Li Xia | PingCAP

2019/6

PingCAP TiDB



lixia@pingcap.com
@紫沐夏_go
https://github.com/zimulala

Agenda

- Principle introduction
- Implementation process
- 2 Phase Commit
- Checkers and common errors

PingCAP

Part I - Principle introduction

ACID

- **Atomicity**
  - Each transaction is treated as a single "unit", which *either succeeds completely, or fails completely*
- **Consistency**
  - Any data written to the database must be valid according to *all defined rules*.
- **Isolation**
  - Isolation ensures that *concurrent execution of transactions leaves the database in the same state* that would have been obtained if the transactions were executed sequentially
- **Durability**
  - Once a transaction has been committed, it will *remain committed even in the case of a system failure*

Read uncommitted

Session A	*Session B*

begin;
select account from account where id = 1
// will get 1000	

	begin;
	update account set account=account+500 where id = 1
	// not commit here

select account from account where id = 1 // will get 1500 (**Dirty read**)	
	rollback;

Read committed

Session A	*Session B*

begin;
select account from account where id=1;
// get 1000	

	begin; update account set account = account+500 where id=1; **commit**;
select account from account where id = 1; // get 1500 (**Non-repeatable reads**) **commit**;	

Repeatable read

Session A	*Session B*

begin;
select account from account where id=1;
// get 1000	

	begin; update account set account = account+500 where id=1; **commit**;
select account from account where id = 1; // get 1000 **commit**;	

Repeatable read

Session A	*Session B*

begin;
select id from account;
// get id(1), id(2)	

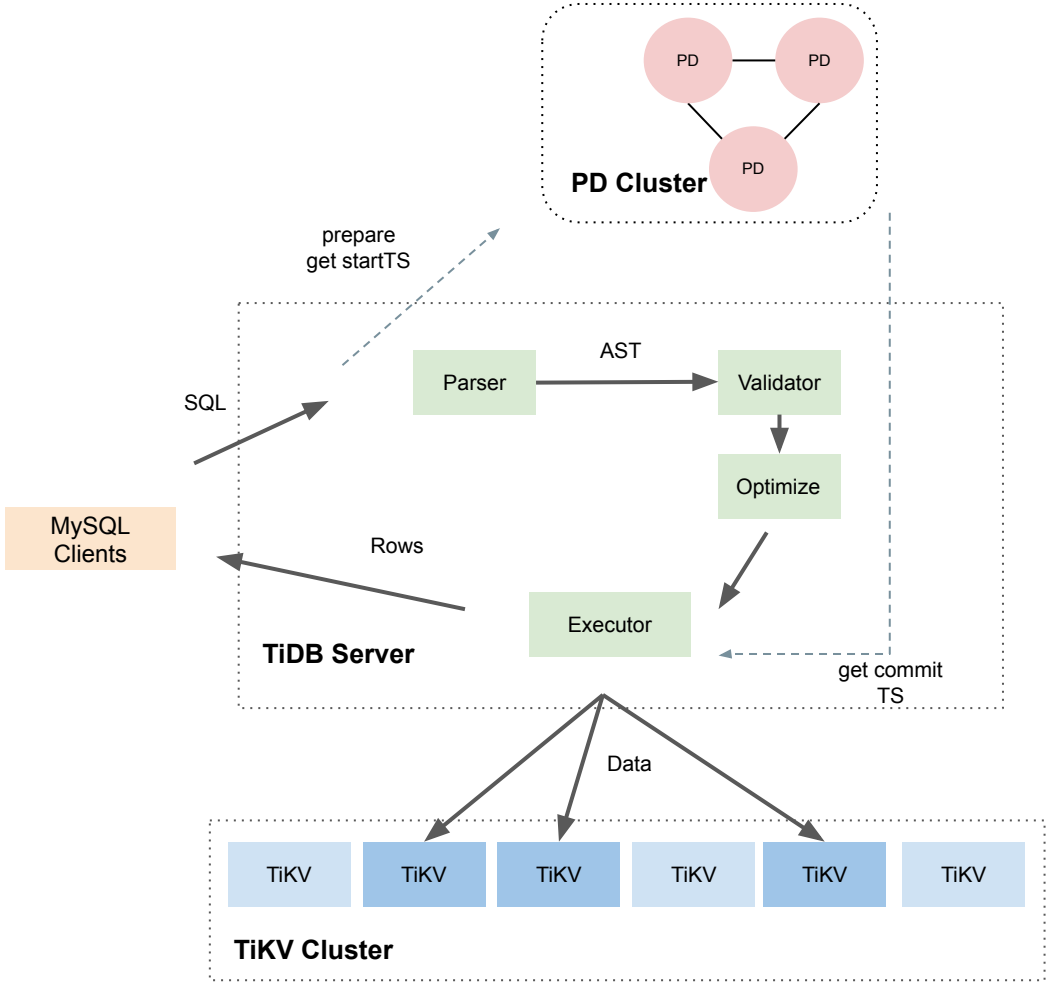
	begin; insert into account values(3,"Dada",5000); **commit**;
select id from account; // get id(1), id(2)	
insert into account values(3,"Dada",5000); // ERROR 1062 (23000): Duplicate entry '3' for key 'PRIMARY' (**Phantom reads**)	

PingCAP

Part II - Implementation process

About Transaction

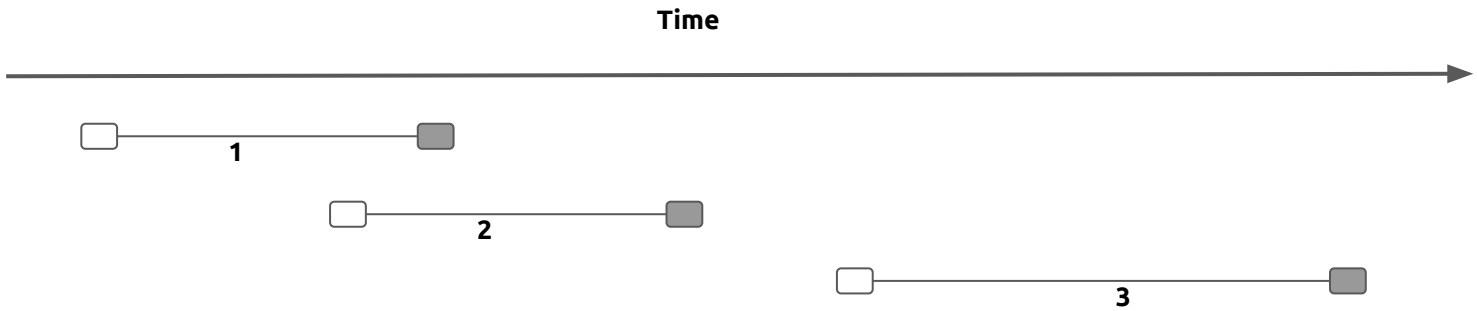
- TiDB SQL Layer
  a. The MySQL Client sends a SQL request.
  b. When a TiDB server receives a request, it will reach the TiDB SQL layer for execution.



PingCAP

TSO -- TimeStamp Oracle

- Percolator transaction requires TSO: a start ts and a commit ts for each transaction.



- get start TS:
  - getTxnFuture, wait BeginWithStartTS
  - getTimestampWithRetry store.Begin()
- get commit TS: getTimestampWithRetry

TSO -- TimeStamp Oracle

- Point get transaction use max Int64 instead of a real TSO
- Get start ts asnychronously with parse & compile
- PD client send TSO request in batch

Transaction cache

- union store
  - MemBuffer
  - snapshot
- e.g.

```
Insert into test (i) values (1);

Begin;

Insert into test (i) values (2);

Select * from test; -- results in TiKV（1） + results in txn（2）
```

latch

- Config variable
  - txn-local-latches
- LatchesScheduler
  - Lock
  - Unlock

PingCAP

Part III - 2 Phase Commit

2 Phase Commit

Bob have $10, Joe have $2, Bob will give Joe $7.

key	data	lock	write
Bob	5: $10		
			6: data @5
Joe	5: $2		
			6: data @5

Phase#1 : Prewrite

key	data	lock	write
Bob	5: $10		
			6: data @5
	7:$3	7:I'm primary	
Joe	5: $2		
			6: data @5
	7:$9	7:primary @ Bob	

Phase#2: Primary Commit (Sync)

Bob have **$3**, Joe have **$9** now.

key	data	lock	write
Bob	5: $10		
			6: data @5
	7:$3	~~7: I'm primary~~	
			8: data @7
Joe	5: $2		
			6: data @5
	7:$9	7: primary @ Bob	

Phase#2: Secondary Commit (Async)

Bob have **$3**, Joe have **$9** now.

key	data	lock	write
Bob	5: $10		
			6: data @5
	7:$3	~~7: I'm primary~~	
			8: data @7
Joe	5: $2		
			6: data @5
	7:$9	~~7: primary @ Bob~~	
			8: data @7

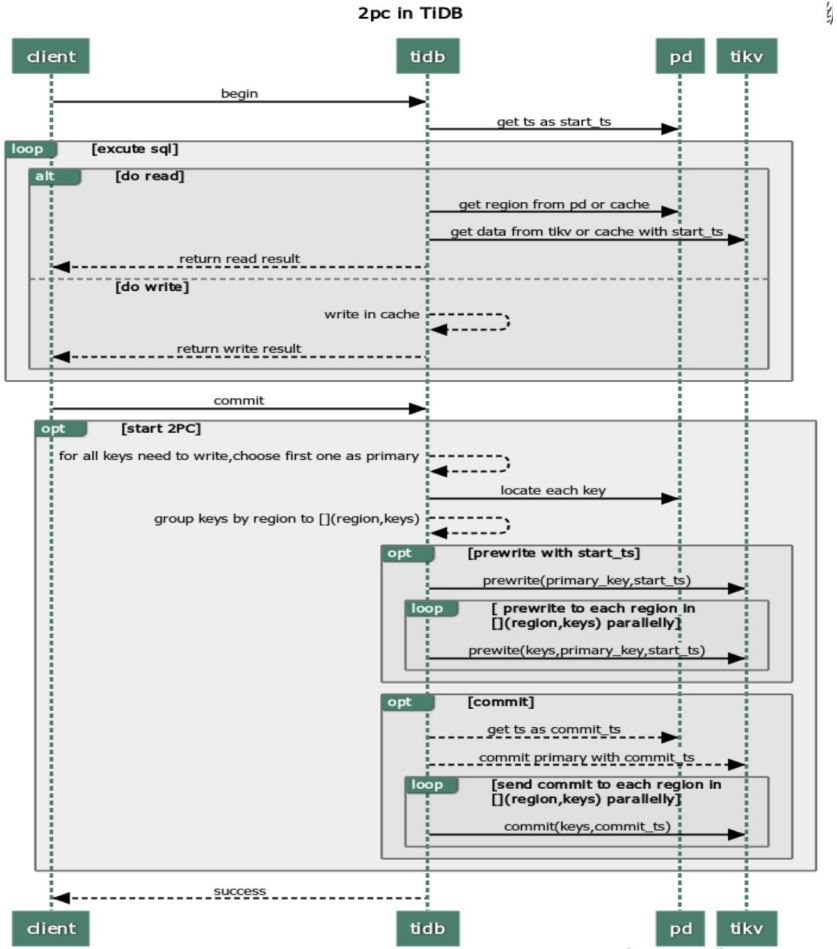
PingCAP

Get region from PD

- GroupKeysByRegion
  - LocateKey, findRegionByKey
    - searchCachedRegion
    - loadRegion
    - insertRegionToCache
- onRegionError
  - GetRPCContext
  - UpdateLeader
  - markNeedCheck
  - ...

Transaction in TiKV client

- Commit in txn.go
  a. newTwoPhaseCommitter
  b. initKeysAndMutations
  c. executeAndWriteFinishBinlog
     - prewriteKeys
     - getTimestampWithRetry
     - commitKeys
     - do cleanupKeys in some cases



PingCAP

Part IV - Checkes and common errors

Checkers

- CheckVisibility
  - get, batchGet
- IsExpired
  - commit
- Schema checker

Common errors

- Send request to TiKV
  - WriteConflict
  - ServerIsBusy
  - PD/TiKV timeout
  - Undetermined
  - …
- Retry transaction
  - WriteConflict
  - InfoSchemaChanged
- Others
  - TxnTooLarge

PingCAP

Thank You !