Replicating MySQL Data to TiDB

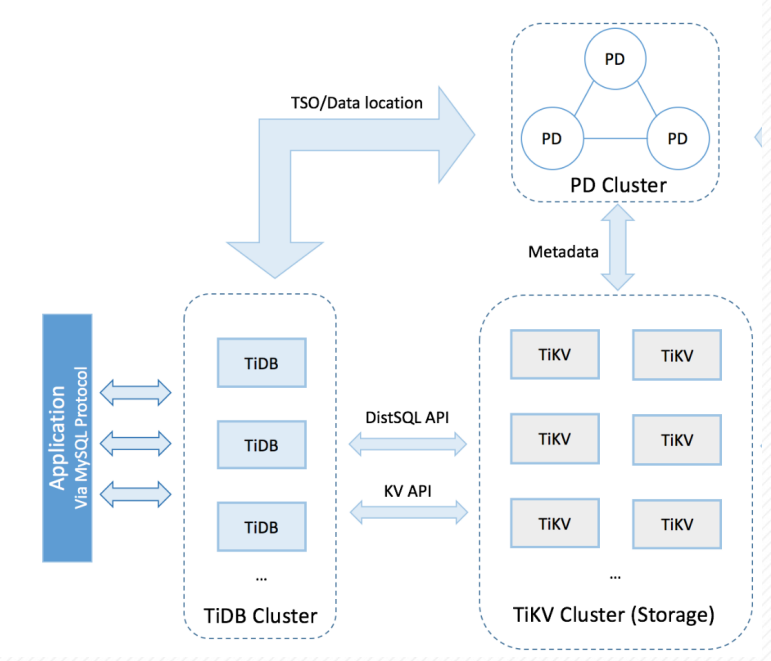
For Near Real-Time Analytics

28-30 MAY 2019
AUSTIN, TEXAS

Who Are We

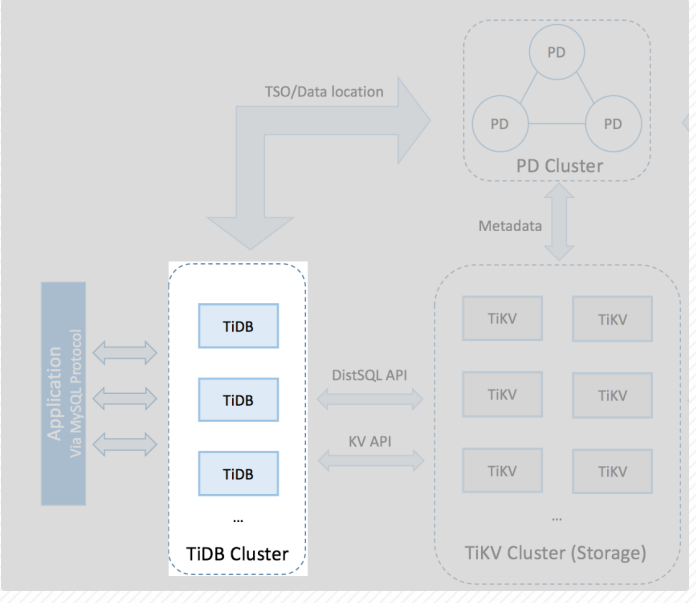
- Jervin Real, Architect
- Francisco Bordenave, Architect

TiDB at a Glance



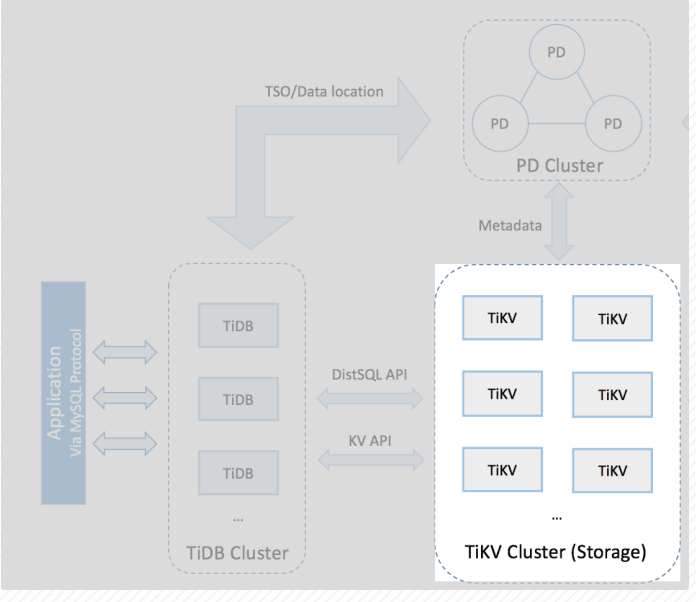
TiDB at a Glance

- TiDB cluster server process SQL and interacts with clients
  - MySQL Compatibility



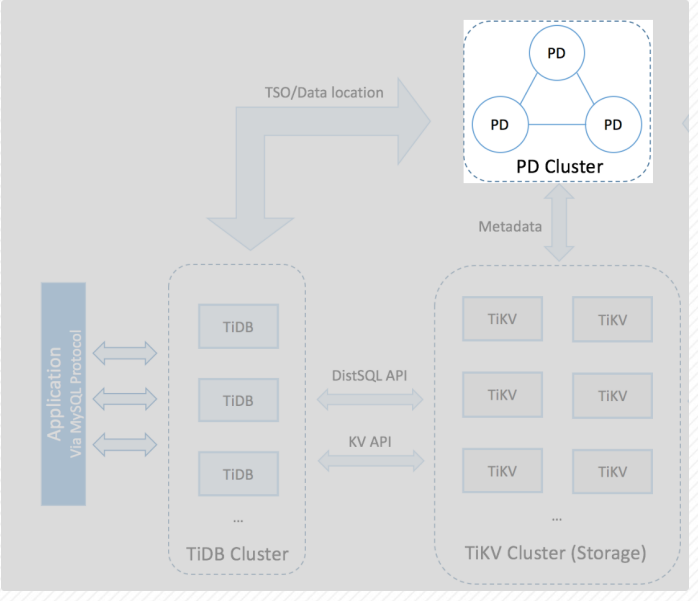
TiDB at a Glance

- TiKV cluster distributed storage, RocksDB



TiDB at a Glance

- PD cluster stores metadata, load balancing and monitor storage



MySQL Compatibility

- MySQL clients and drivers
- DML, DDL compatibility - some limitations
- Easy to start querying DB

Replication Goals

- Consistency
  - Resume from previous checkpoint
  - Verifying consistency
- Performance
  - Can we avoid replication lag?
- Observability
  - Is the process broken?
  - Are we querying stale data?

Toolbox

Ready To Use Tools

Toolbox

- Data Migration (full suite)
  - dm-master
  - dm-worker
    - mydumper
    - loader
    - syncer
- tidb-lightning
- DIY

mydumper

- https://github.com/maxbube/mydumper fork
  - Additional customizations
- Uncompressed dump only for loader

```
mydumper -B percona -o . -C -t 3 -h 10.3.4.4 \
   -u tidb -p tidb -v 3
```

loader (1)

- Loads mydumper data in parallel, ~1MB chunk per load, resumable
- Uncompressed dump only
- `myloader` is possible too
  - Be aware of trx size limits (i.e. use `-q 1`)*
  - Not resumable
- Database/table filters/routing.

```
loader -d ./  -h 127.0.0.1 -P 4000 -u root -t 3 \
   -rm-checkpoint -L info
```

```
myloader -d ./ -B percona -h 127.0.0.1 -u root \
   -P 4000 -t 3 -q 1
```

*https://github.com/tikv/tikv/issues/2986

loader (2)

- Database/table filters/routing.

```
{
  "log-level": "info",
  "log-file": "",
  "status-addr": ":8272",
  "pool-size": 3,
  "dir": "./",
  "db": {
    "host": "127.0.0.1",
    "user": "root",
    "port": 4000,
    "sql-mode": "@DownstreamDefault"
  },
  "checkpoint-schema": "tidb_loader",
  "config-file": "",
  "route-rules": null,
  "do-table": null,
  "do-db": null,
  "ignore-table": null,
  "ignore-db": null,
  "rm-checkpoint": false
}
```

loader Status

- Available from `tidb_loader.checkpoint` by default

```
mysql> select * from tidb_loader.checkpoint \G
*************************** 1. row ***************************
         id: 13917f
   filename: percona.history.sql
  cp_schema: percona
   cp_table: history
     offset: 55998837
    end_pos: 2685020785
create_time: 2019-05-29 02:56:14
update_time: 2019-05-29 03:08:53
*************************** 2. row ***************************
         id: 13917f
   filename: percona.search.sql
  cp_schema: percona
   cp_table: search
     offset: 59999013
    end_pos: 812604151
create_time: 2019-05-29 02:56:14
update_time: 2019-05-29 03:08:52
...
```

syncer (1)

- Resumes from loader (or any replication coordinates)
- Runs per table, per event (not at transaction level)
- Allows routing/filtering per table
- Allows DML filtering
- Multiple worker threads, but still in commit order.
    - Eventually consistent

```
# cat config.toml
...
# [[skip-dmls]]
# db-name = "foo"
# tbl-name = "bar"
# type = "delete"
...


syncer -config config.toml
```

syncer (2)

```
{
  "log-level": "info",
  "log-file": "syncer.log",
  "log-rotate": "day",
  "status-addr": ":8271",
  "server-id": 1234567890,
  "meta": "./syncer.meta",
  "persistent-dir": "",
  "flavor": "mysql",
  "worker-count": 16,
  "batch": 100,
  "max-retry": 100,
  "replicate-do-table": null,
  "replicate-do-db": null,
  "replicate-ignore-table": null,
  "replicate-ignore-db": null,
  "skip-ddls": [],
  "skip-dmls": null,
  "route-rules": null,
...
```

```
...
  "from": {
    "host": "10.3.4.4",
    "user": "tidb",
    "port": 3306
  },
  "to": {
    "host": "127.0.0.1",
    "user": "root",
    "port": 4000
  },
  "enable-gtid": false,
  "auto-fix-gtid": false,
  "disable-detect": false,
  "safe-mode": false,
  "config-file": "config.toml",
  "stop-on-ddl": false,
  "execute-ddl-timeout": "3h",
  "execute-dml-timeout": "1m",
  "execute-queue-length": 5000,
  "enable-ansi-quotes": false,
  "timezone": ""
}
```

syncer Status

- HTTP Interface

```
me@tidb:~/git/dm/syncer# curl localhost:8271/metrics \
    | egrep '[syncer_binlog_file|syncer_binlog_pos]\{node'

syncer_binlog_file{node="master"} 3
syncer_binlog_file{node="syncer"} 3
syncer_binlog_pos{node="master"} 7.31884075e+08
syncer_binlog_pos{node="syncer"} 7.32175424e+08
```

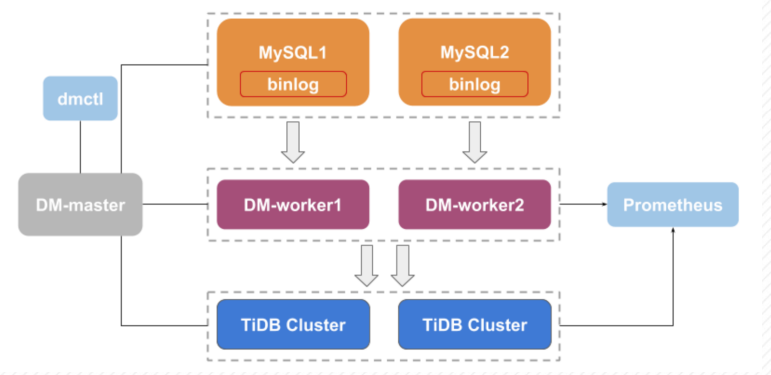
- Log file

```
2019/05/24 16:09:35 syncer.go:951: [info] [syncer]total events = 407220,
  tps = 719.666667, master-binlog = (mysql-bin.000009, 220613462),
  master-binlog-gtid=, syncer-binlog = (mysql-bin.000006, 306307142),
  syncer-binlog-gtid =
```

- TiDB checkpoint table when used with `dm-woker`

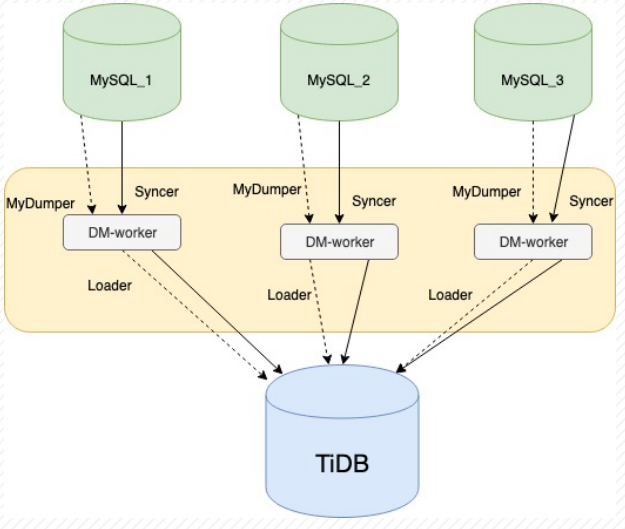
dm - Data Migration Tool

- dm-master, dm-worker
  - One dm-worker per replication source
  - Sharded table to single table



dm - Data Migration Tool

- dm-worker = (mydumper + loader + syncer)
  - binlog-[ignore|replicate|do]-[db|table]
  - skip-dmls (db|table|type)



lighting - super fast loader tool

- tikv-importer + tidb-lighting
- Parse mydumper or CSV files and convert into kv files
- Loads data directly to tikv server
- Disable compaction and set cluster in ‘import’ mode (makes writes way faster!)
    - In case of import failure cluster needs to be set back to ‘normal’ mode
- During load cluster service is limited

How Others Do It

Replicating to Another Platform

To Cassandra

- Apache Spark
- Striim https://www.striim.com/integrations/mysql-cassandra/
- Tungsten replicator

All solutions requires some type of ETL process. None of them are easy to implement as requires model de-normalization

To Clickhouse

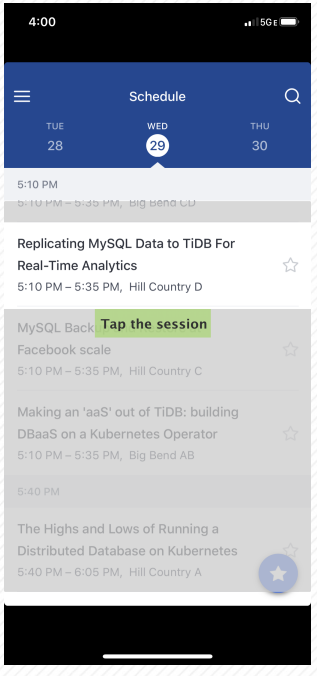
- Altinity https://github.com/altinity/clickhouse-mysql-data-reader
- Tungsten replicator (of course)
- CSV to clickhouse-client (batch)
- DIY for tricky situations
  - See how at 5:40PM at Bid Bend CD
  - Jervin's next talk!

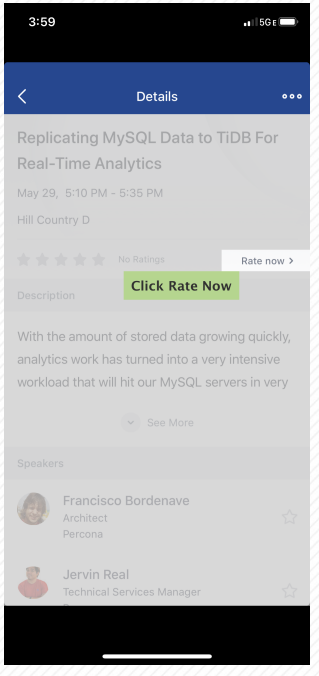
MariaDB ColumnStore

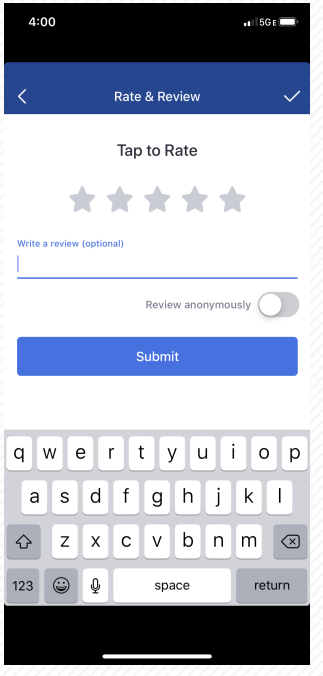
- Straight forward is distributed with MariaDB out of the box (not fully true)

Some nice benchmarks:
https://www.percona.com/blog/2017/03/17/column-store-database-benchmarks-mariadb-columnstore-vs-clickhouse-vs-apache-spark/

Rate Our Talk







Thank you for joining us this year!

... and thanks to our sponsors!

Questions?