PingCAP

A Brief Introduction of TiDB

Dongxu (Edward) Huang
CTO, PingCAP

PingCAP

About me

- Dongxu (Edward) Huang, Cofounder & CTO of PingCAP
- PingCAP, based in Beijing, China.
- Infrastructure software engineer, open source hacker
- Codis / TiDB / TiKV
- Golang / Python / Rust

PingCAP

What would you do when…

- RDBMS is becoming the performance bottleneck of your backend service
- The amount of data stored in RDBMS is overwhelming
- You want to do some complex queries on a sharding cluster
  - e.g. simple JOIN or GROUP BY
- Your application needs ACID transaction on a sharding cluster

TiDB Project - Goal

- SQL is necessary
- Transparent sharding and data movement
- 100% OLTP + 80% OLAP
  - Transaction + Complex query
- Compatible with MySQL, at most cases
- 24/7 availability, even in case of datacenter outages
  - Thanks to Raft consensus algorithm
- Open source, of course.

TiDB

A Distributed SQL Database

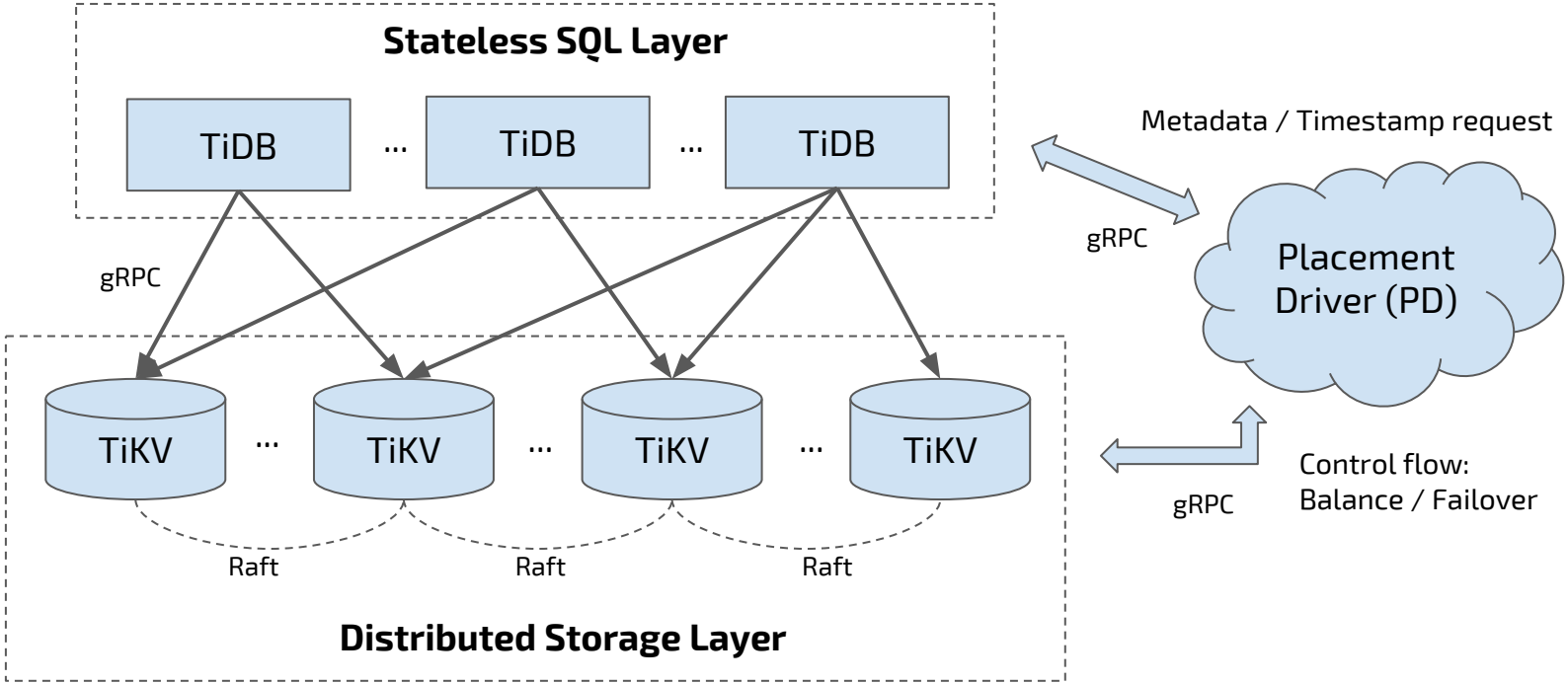
PingCAP

Agenda

- Technical overview of TiDB / TiKV
  - Storage
  - Distributed SQL
  - Tools
- Real-world cases and benchmarks
- Demo

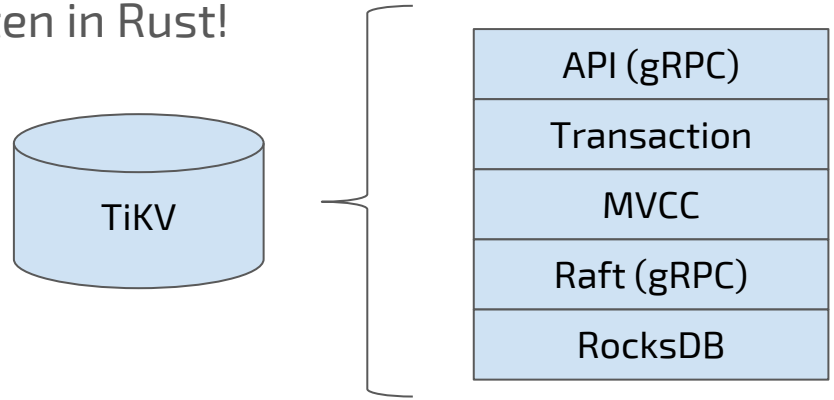
PingCAP

Architecture



Storage stack 1/2

- TiKV is the underlying storage layer
- Physically, data is stored in RocksDB
- We build a Raft layer on top of RocksDB
  - What is Raft?
- Written in Rust!

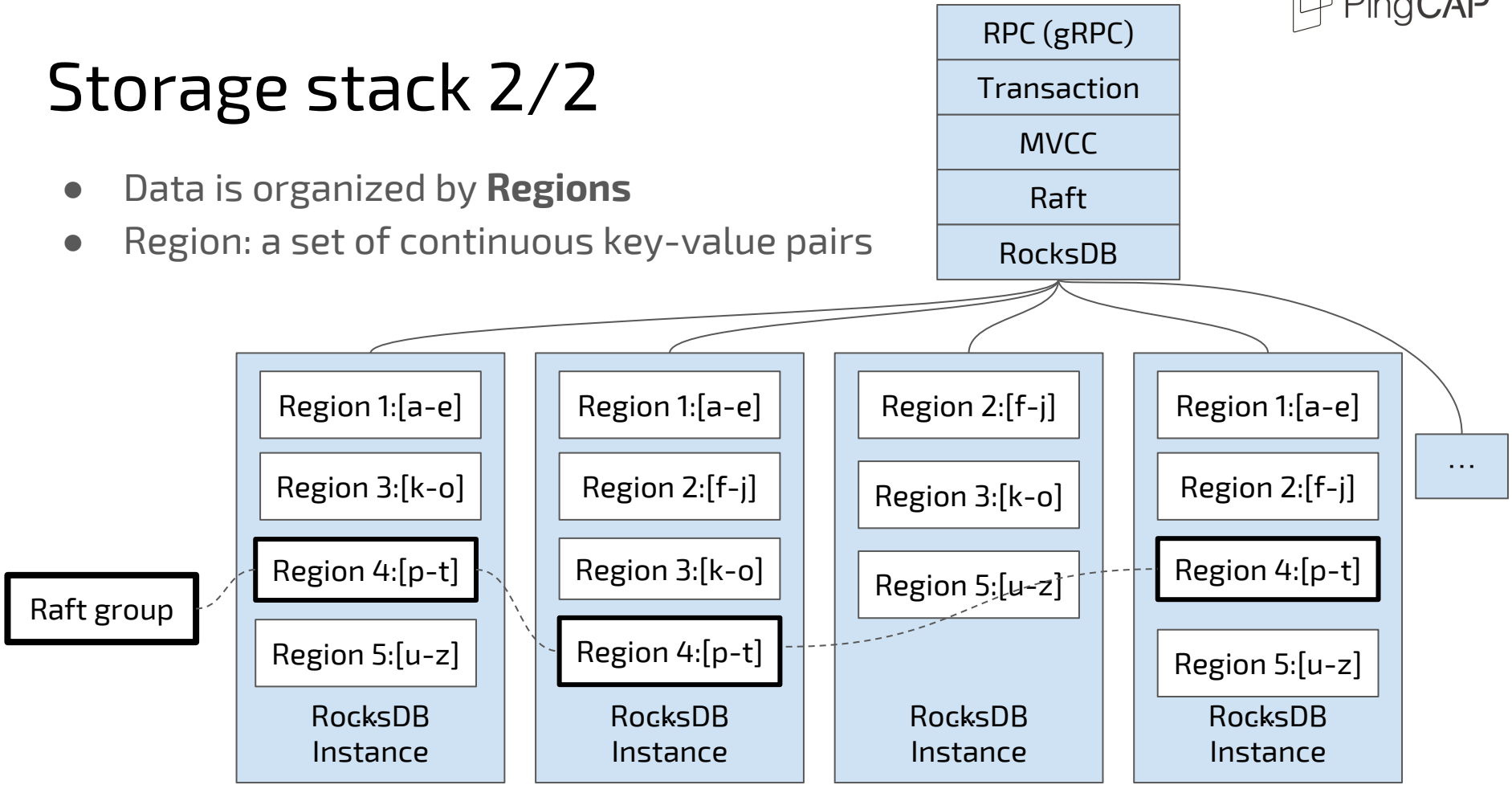


Transactional KV API
(https://github.com/pingcap/tidb/blob/master/cmd/benchkv/main.go)

Raw KV API
(https://github.com/pingcap/tidb/blob/master/cmd/benchraw/main.go)

Storage stack 2/2

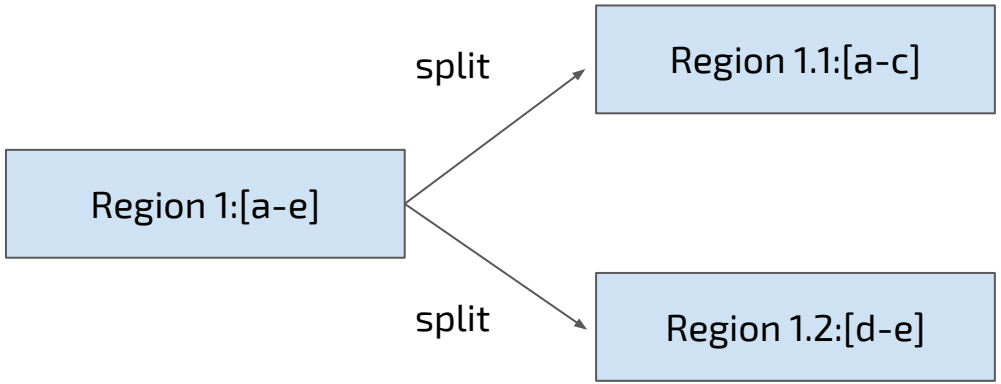
- Data is organized by **Regions**
- Region: a set of continuous key-value pairs



PingCAP

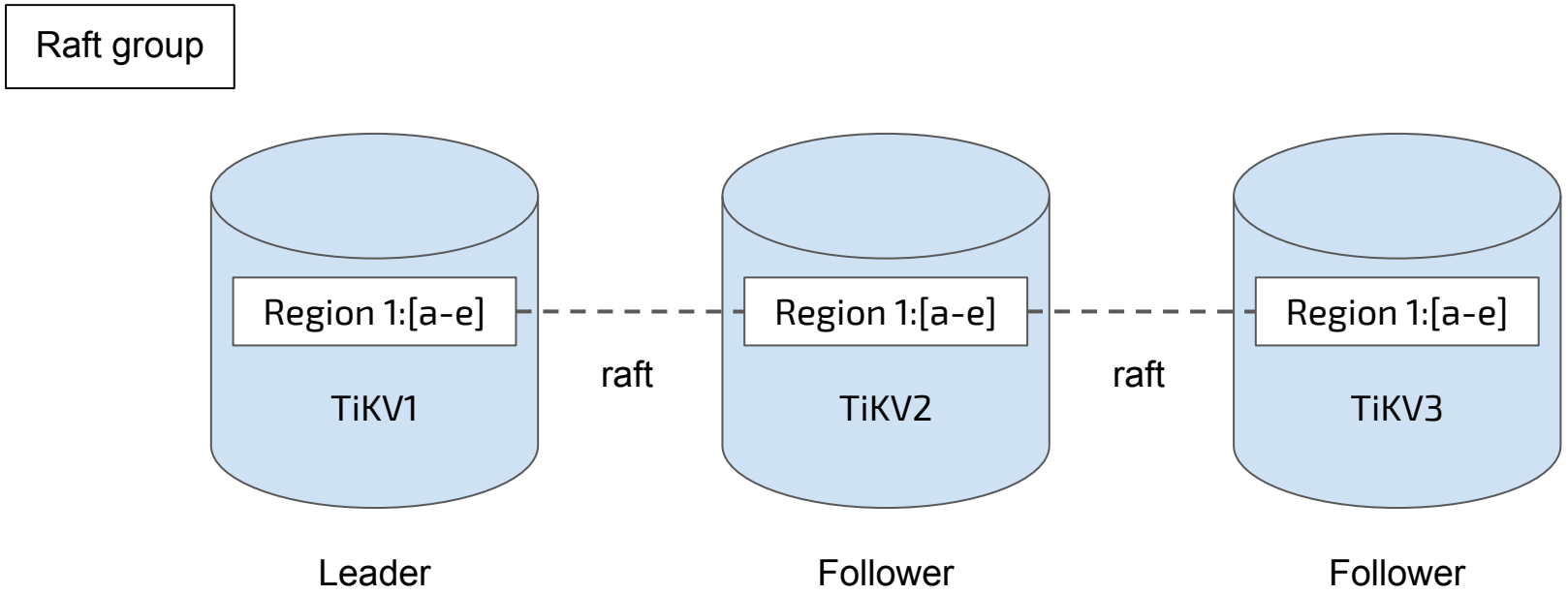
Dynamic Multi-Raft

- What's Dynamic Multi-Raft?
  - Dynamic split / merge
- Safe split / merge



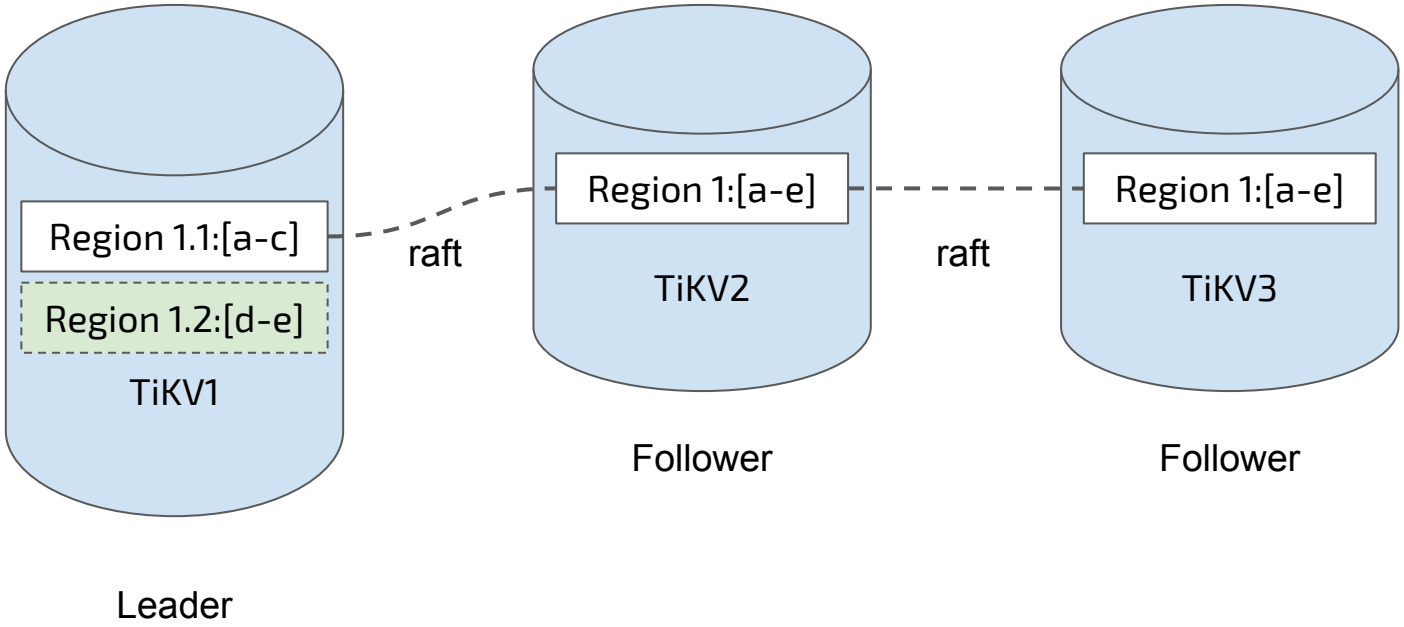
PingCAP

Safe Split: 1/4

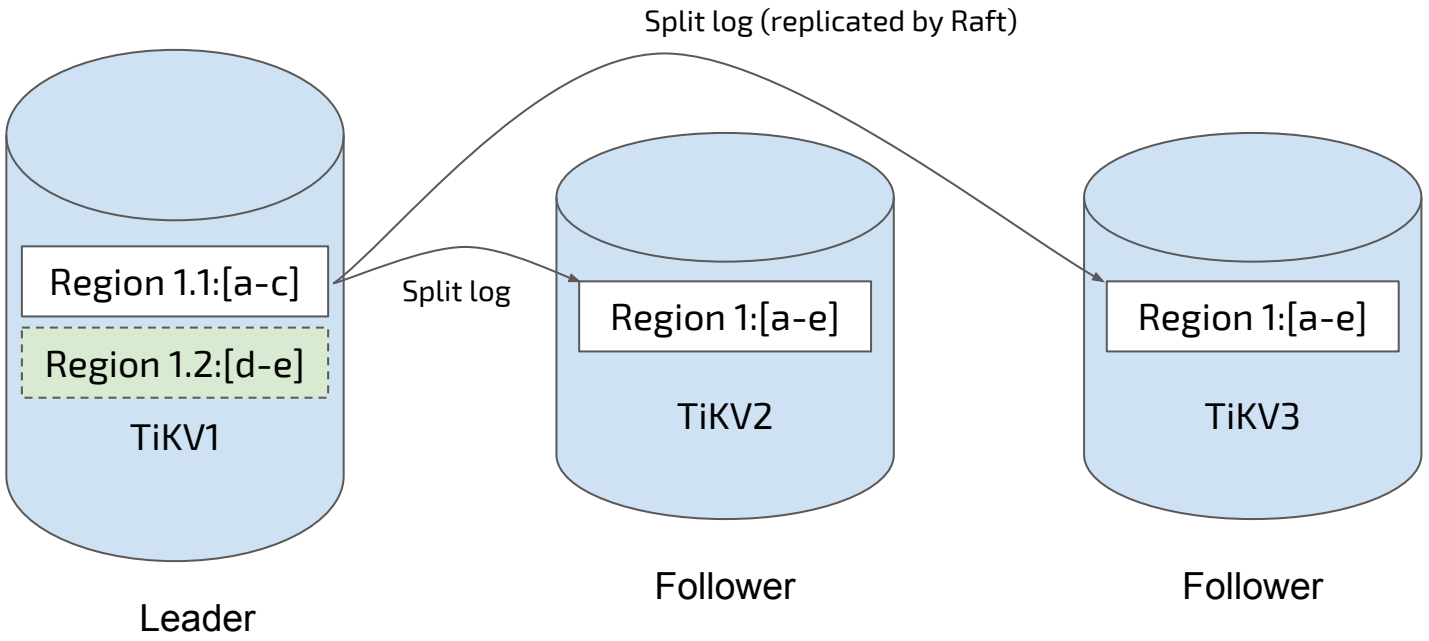


PingCAP

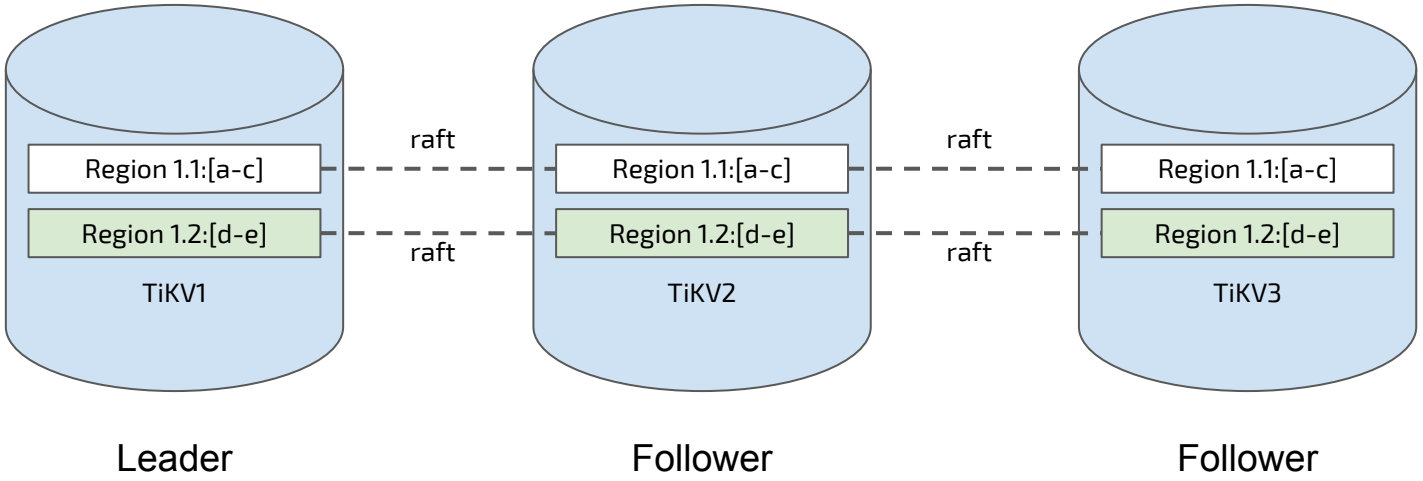
Safe Split: 2/4



Safe Split: 3/4

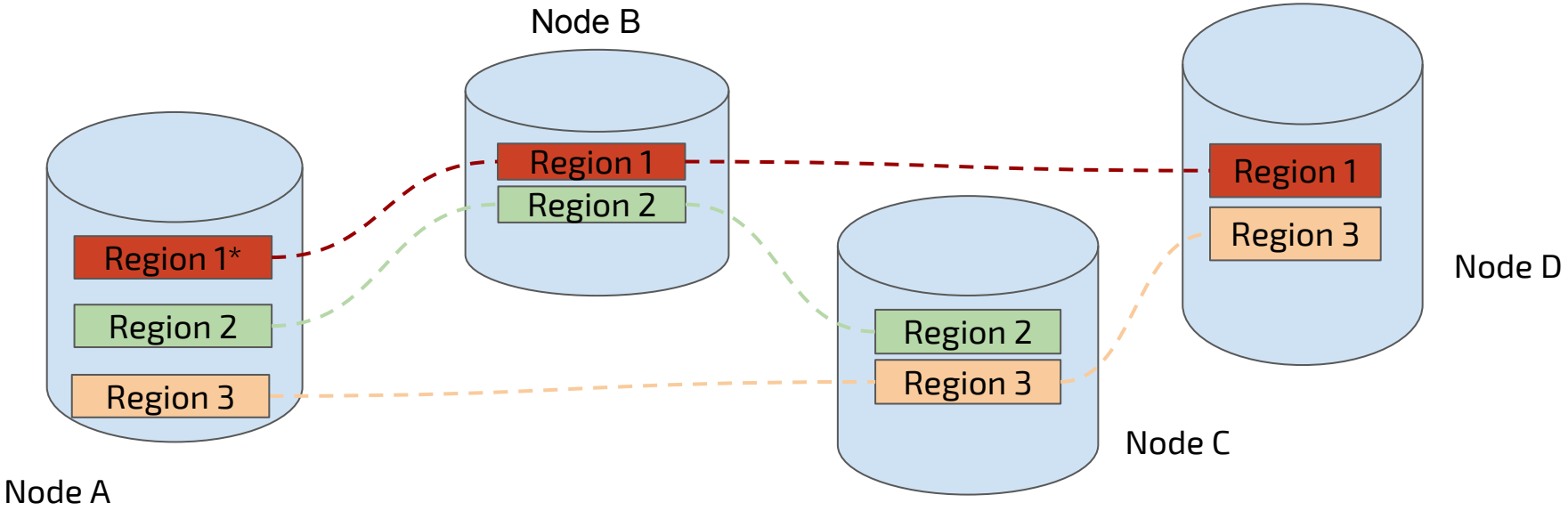


Safe Split: 4/4



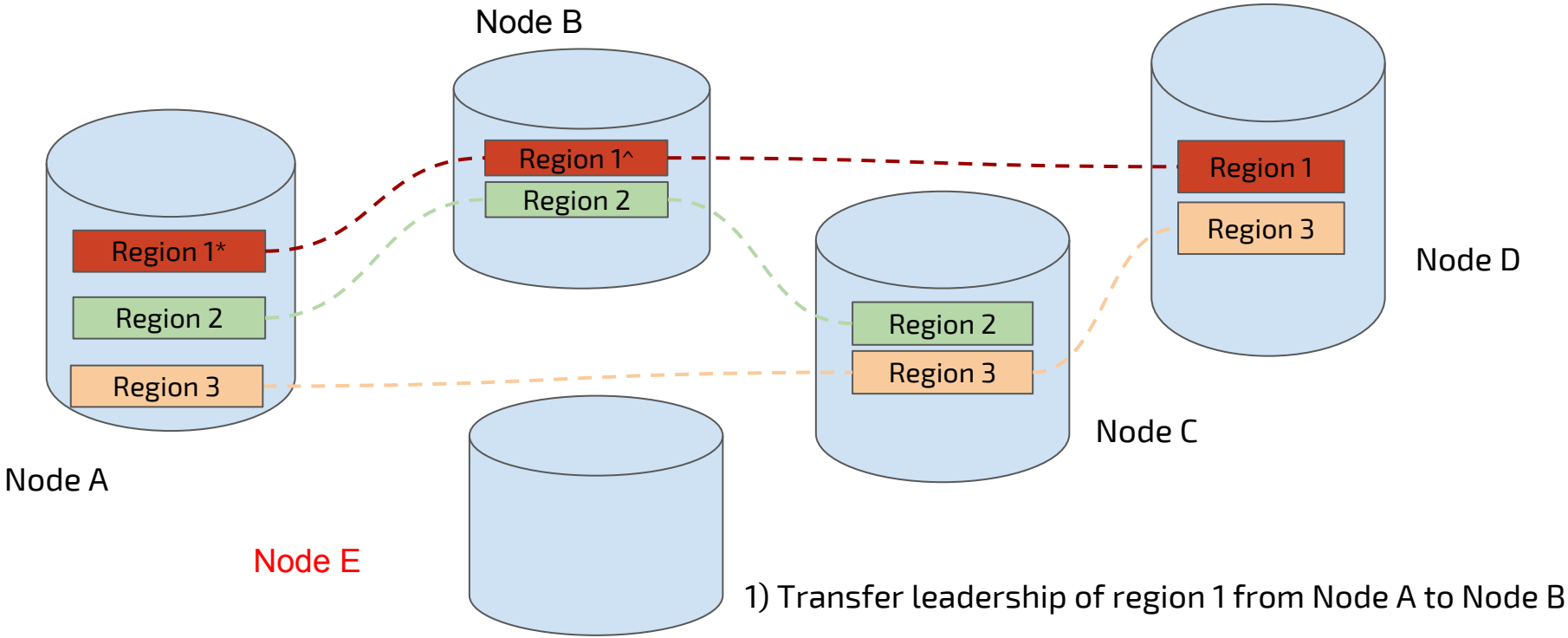
PingCAP

Scale-out (initial state)



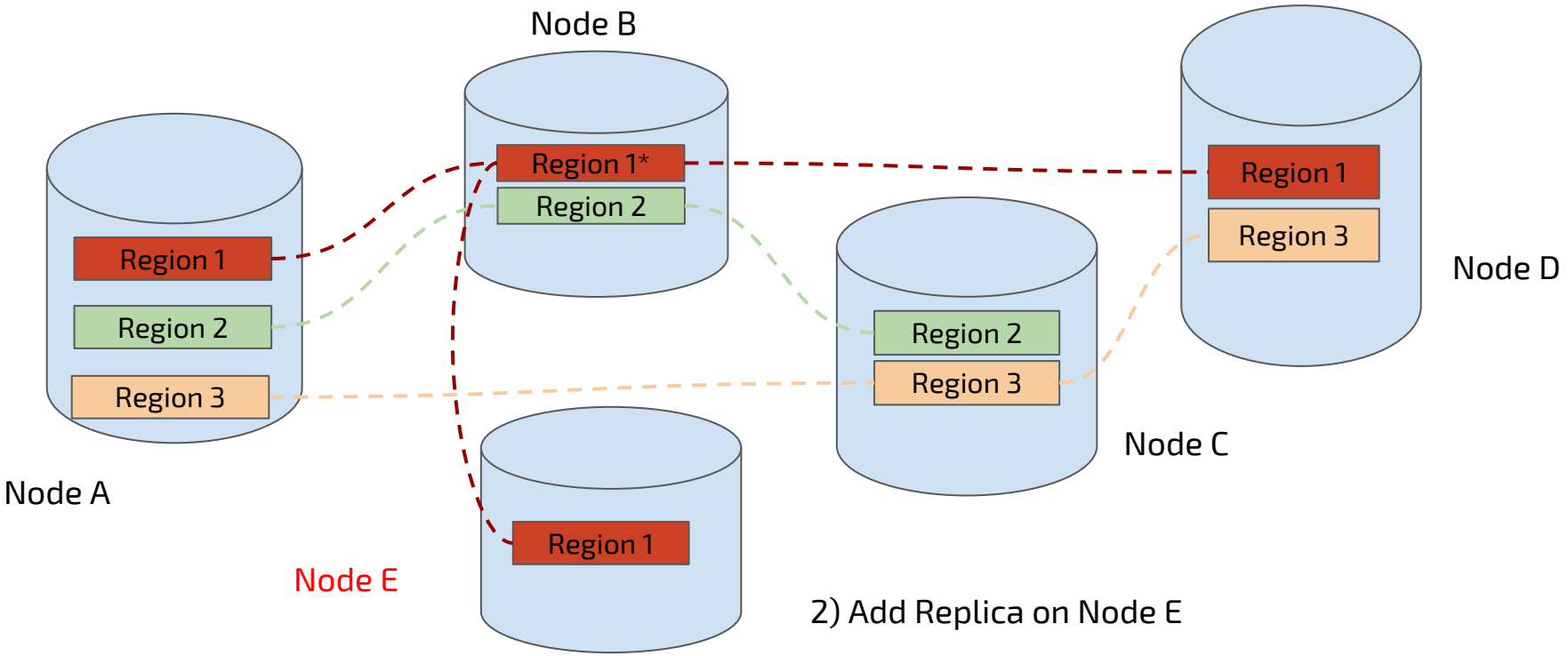
PingCAP

Scale-out (add new node)



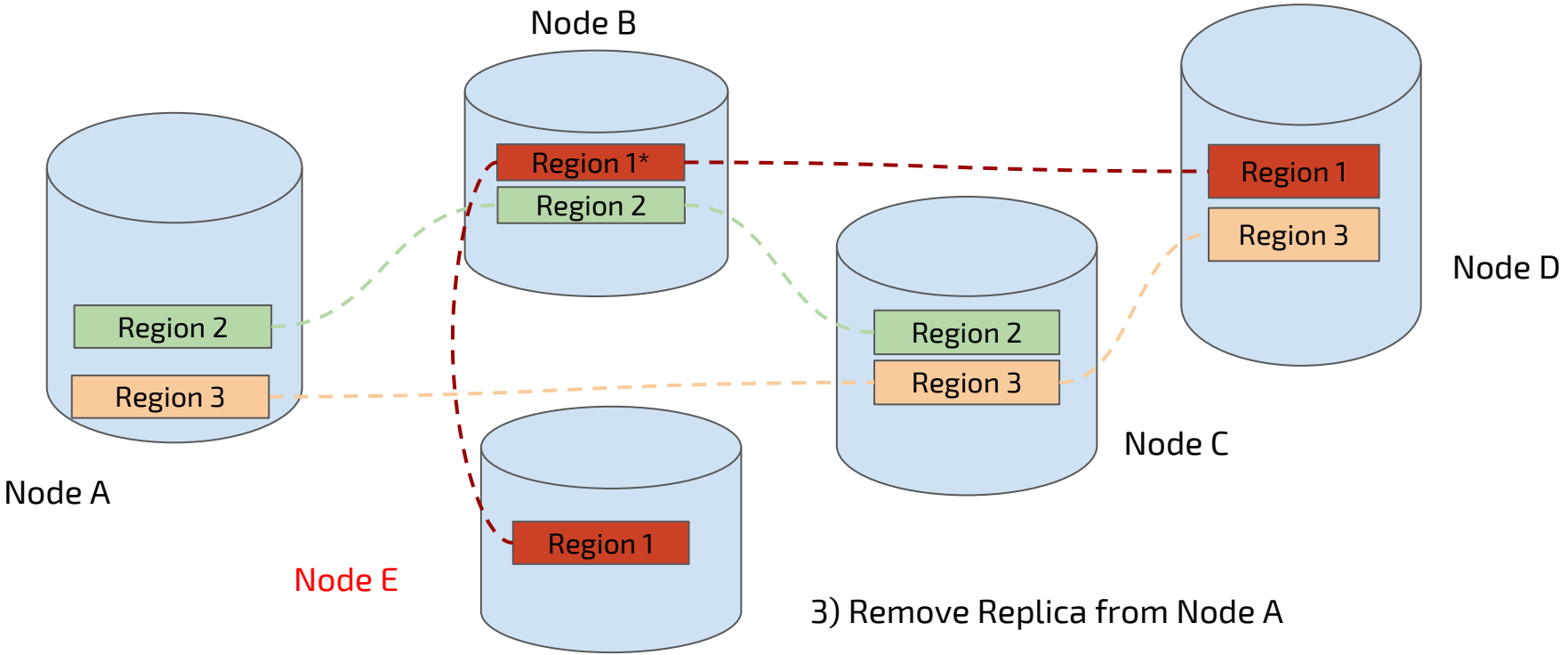
PingCAP

Scale-out (balancing)



PingCAP

Scale-out (balancing)



PingCAP

ACID Transaction

- Based on Google Percolator
- ‘Almost’ decentralized 2-phase commit
  - Timestamp Allocator
- Optimistic transaction model
- Default isolation level: Repeatable Read
- External consistency: Snapshot Isolation + Lock
    - SELECT … FOR UPDATE

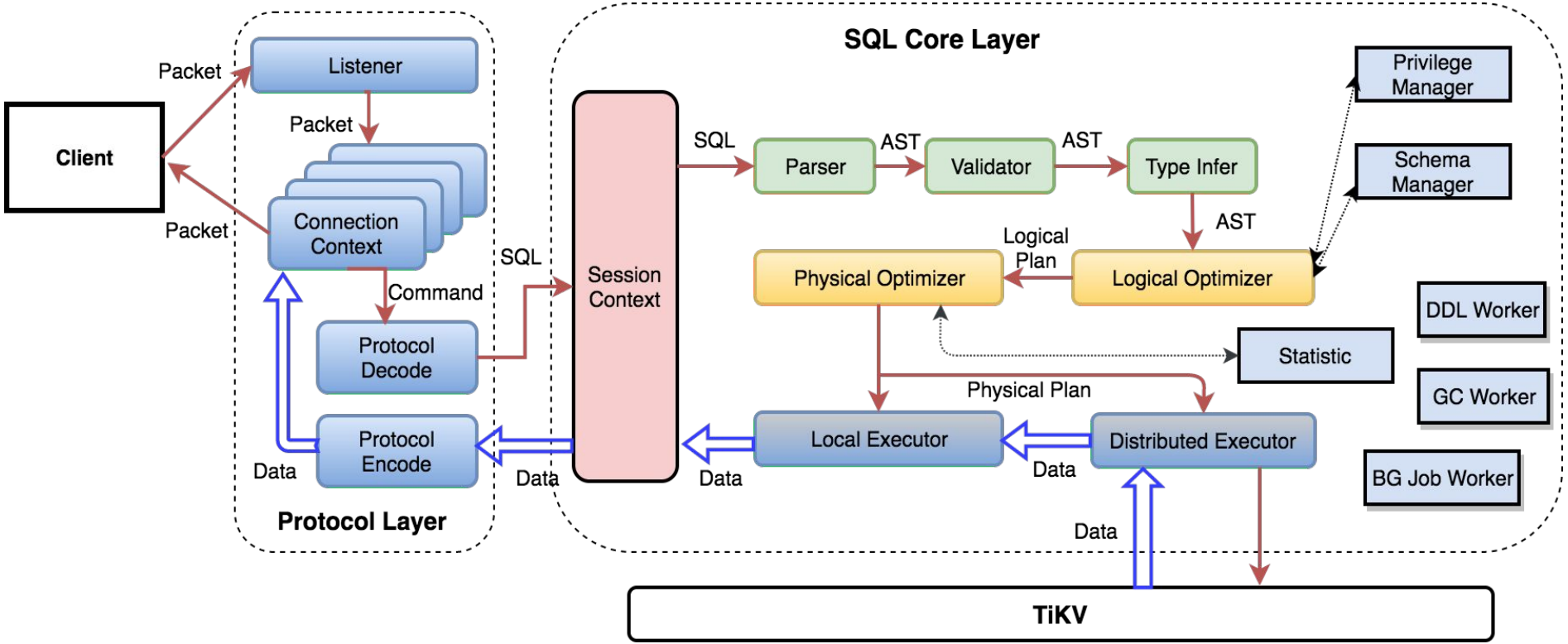
PingCAP

Distributed SQL

- Full-featured SQL layer
- Predicate pushdown
- Distributed join
- Distributed cost-based optimizer (Distributed CBO)

PingCAP

TiDB SQL Layer overview



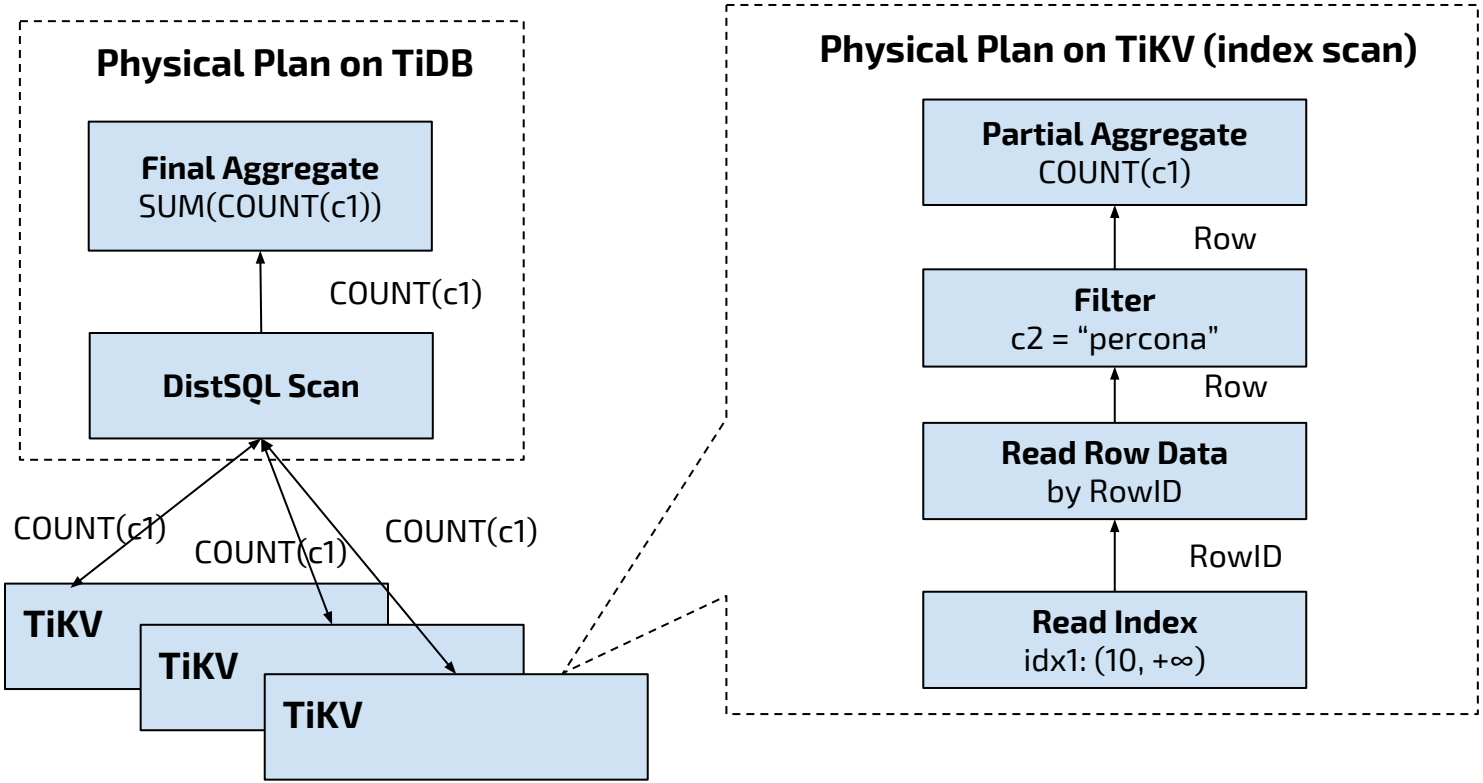
PingCAP

What happens behind a query

```
CREATE TABLE t (c1 INT, c2 TEXT, KEY idx_c1(c1));

SELECT COUNT(c1) FROM t WHERE c1 > 10 AND c2 = ‘percona’;
```

Query Plan


Physical Plan on TiDB
Final Aggregate
SUM(COUNT(c1))
COUNT(c1)
DistSQL Scan
COUNT(c1)
COUNT(c1)
COUNT(c1)
TiKV
TiKV
TiKV
Physical Plan on TiKV (index scan)
Partial Aggregate
COUNT(c1)
Row
Filter
c2 = “percona”
Row
Read Row Data
by RowID
RowID
Read Index
idx1: (10, +∞)

PingCAP

What happens behind a query

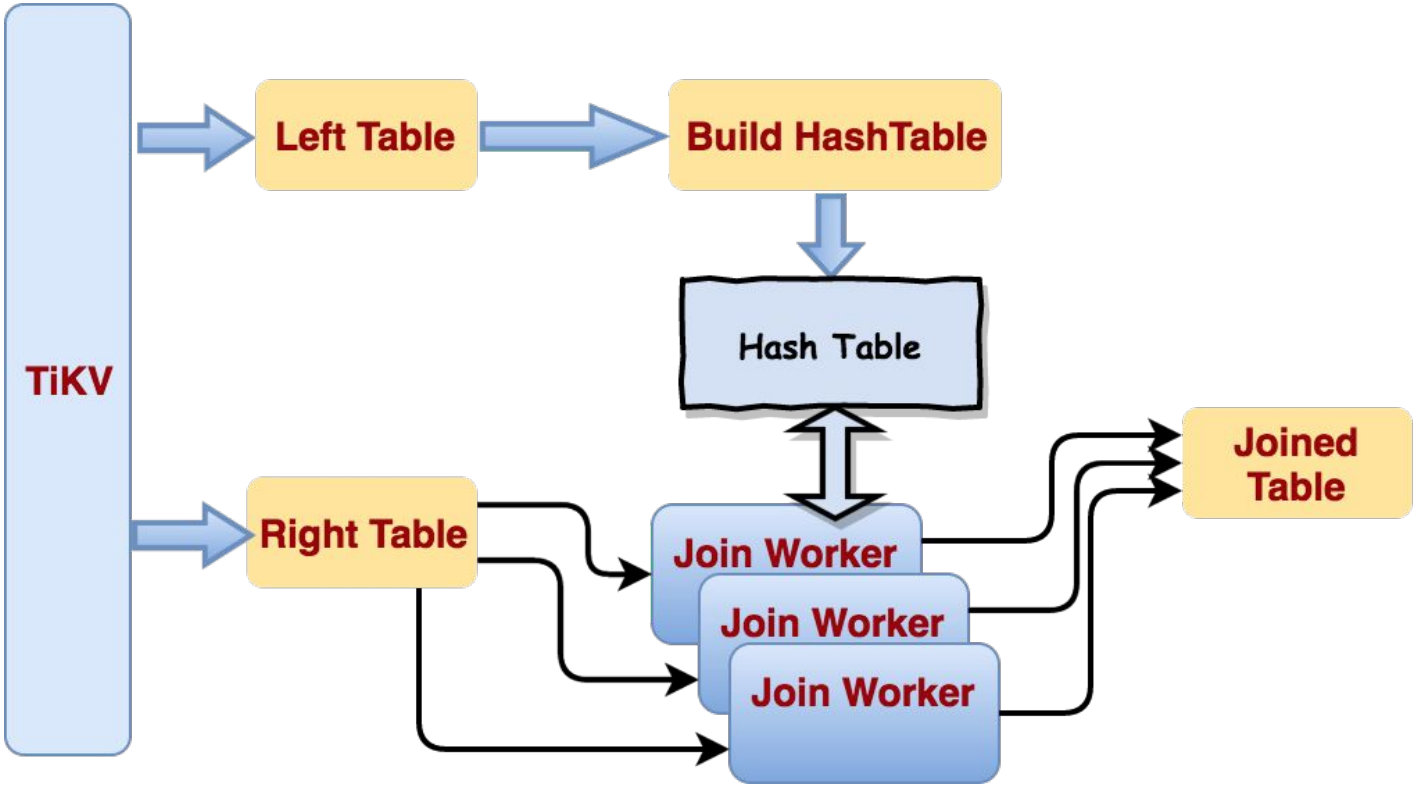
```
CREATE TABLE left (id INT, email TEXT, KEY idx_id(id));

CREATE TABLE right (id INT, email TEXT, KEY idx_id(id));

SELECT * FROM left join right WHERE left.id = right.id;
```

PingCAP

Distributed Join (HashJoin)



PingCAP

Supported Distributed Join Type

- Hash Join
- Sort merge Join
- Index-lookup Join

PingCAP

No silver bullet (anti-patterns for TiDB SQL)

- Join between large tables without index or any hints
- Get distinct values from large tables without index
- Sort without index
- Result set is too large (forget LIMIT N?)

PingCAP

Best practices

- **Random**, massive, read / write workload
- No hot small table
- Use transaction, but not much conflicts

PingCAP

Tools matter

- Syncer
- TiDB-Binlog
- Mydumper/MyLoader(loader)

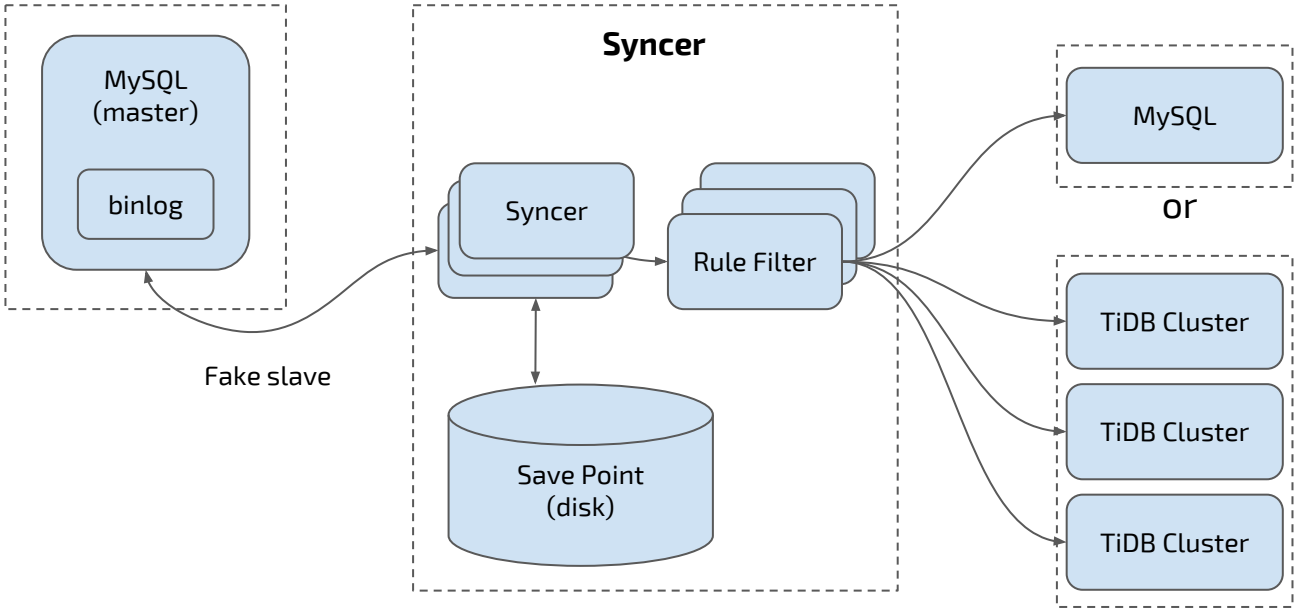
Open sourced, too.

https://github.com/pingcap/tidb-tools

PingCAP

Syncer

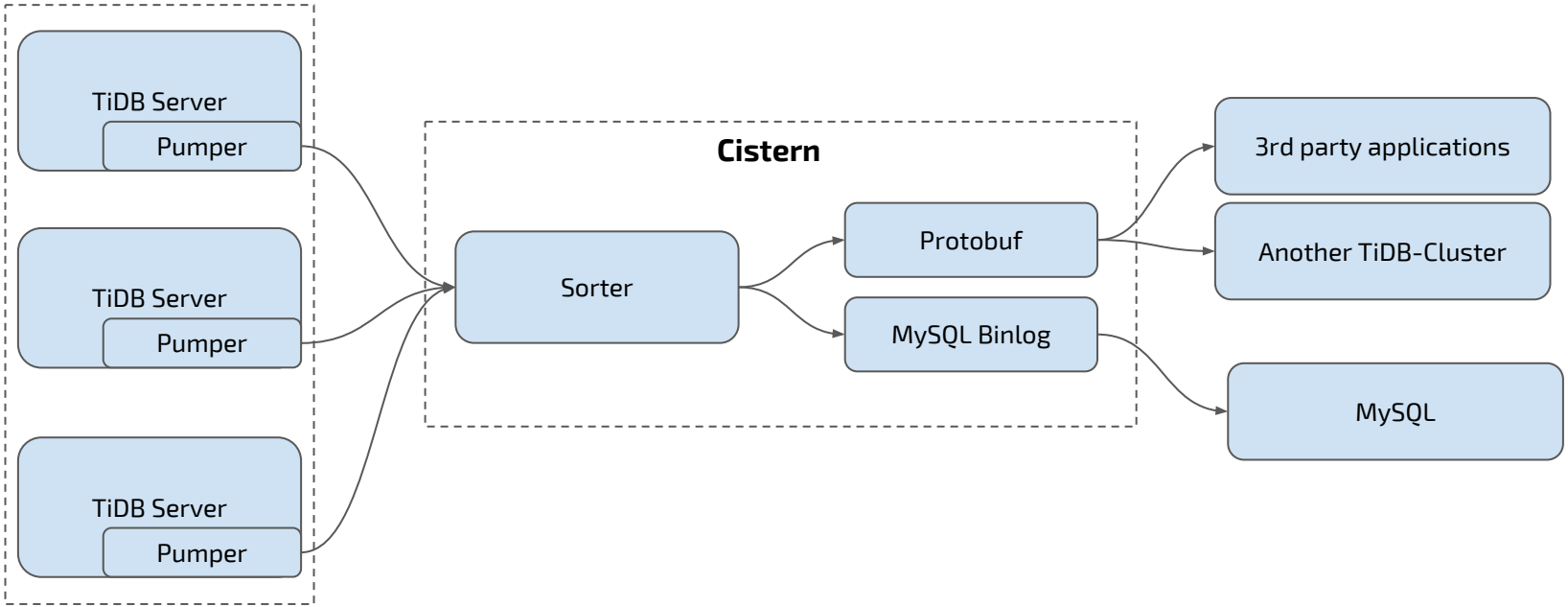
- Synchronize data from MySQL in real-time
- Hook up as a MySQL replica



PingCAP

TiDB-Binlog

- Subscribe the incremental data from TiDB
- Output Protobuf formatted data or MySQL Binlog format(WIP)



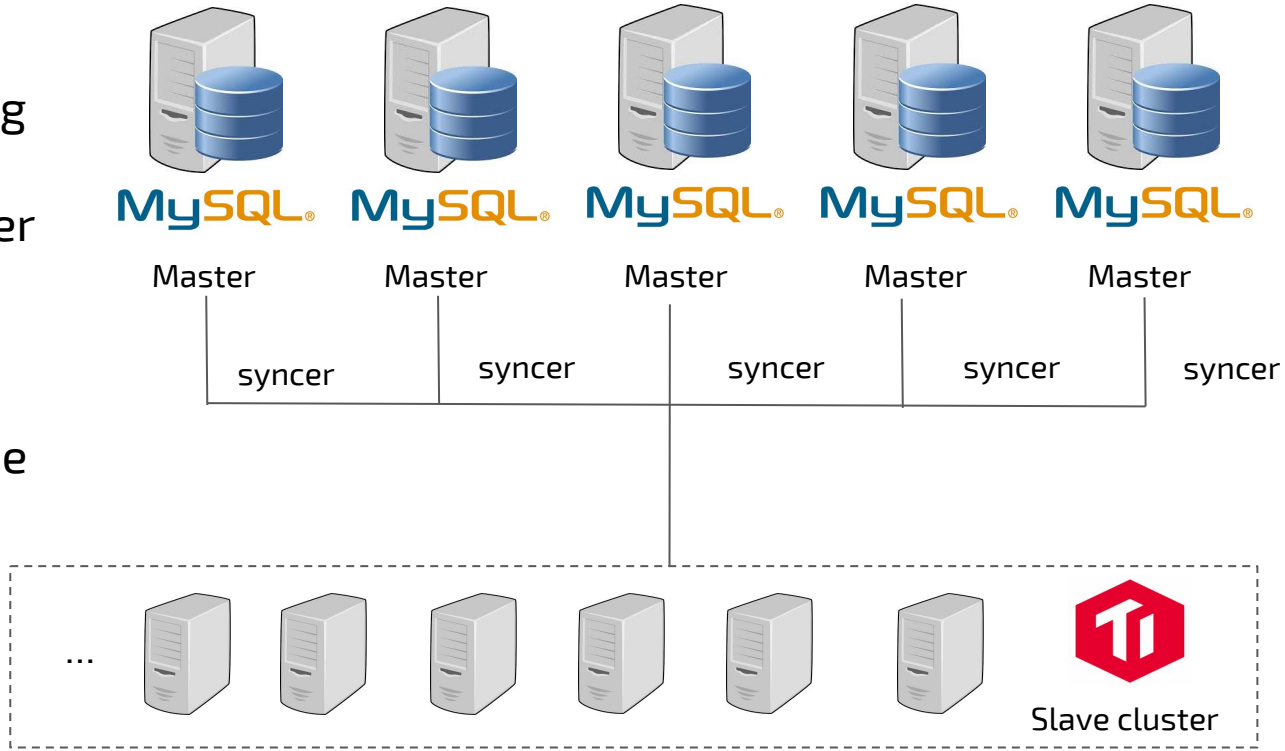
PingCAP

MyDumper / Loader

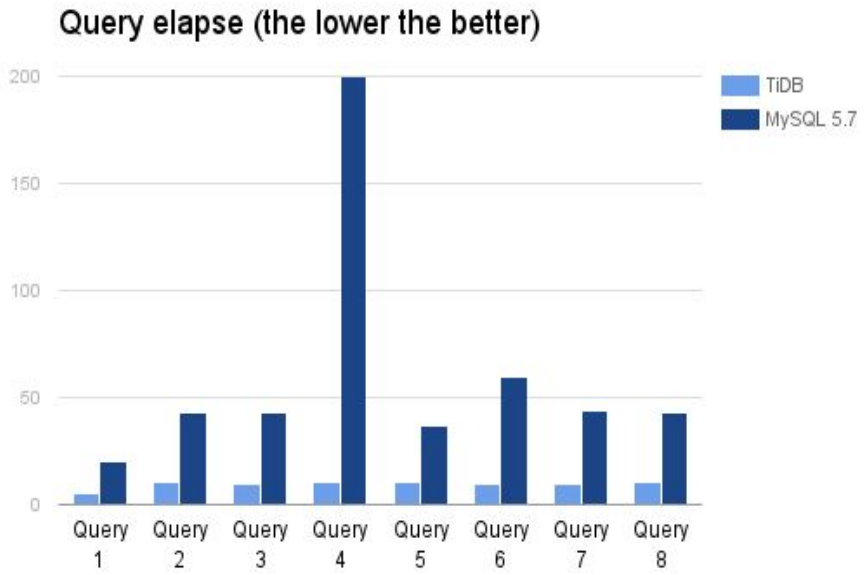
- Backup/restore in parallel
- Works for TiDB too
- Actually, we don't have our own data migration tool for now

Use case 1: OLTP + OLAP

- One of the most popular bike sharing companies in China
- 7-nodes TiDB cluster for order storage (OLTP).
- Hook up as MySQL Replica, synchronize data to a 10-nodes TiDB cluster for Ad-hoc OLAP.



Use case 1: Ad-hoc OLAP

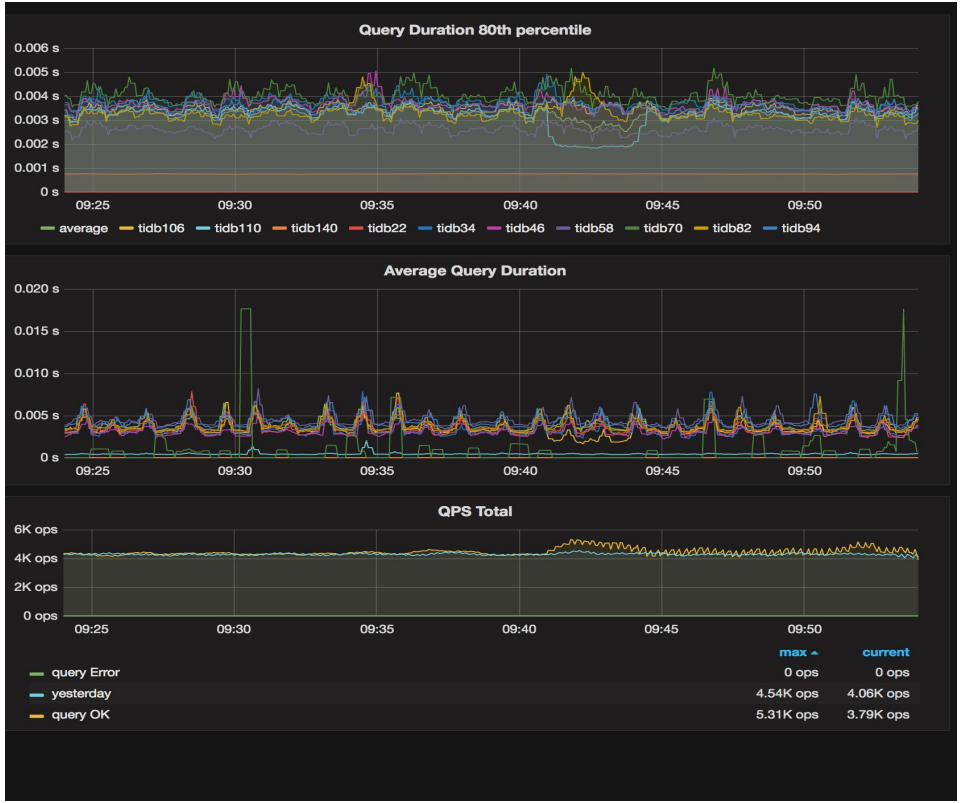


TiDB Elapse (3 nodes)	MySQL Elapse
5.07699437s	19.93s
10.524703077s	43.23s
10.077812714s	43.33s
10.285957629s	>20 mins
10.462306097s	36.81s
9.968078965s	1 min 0.27 sec
9.998030375s	44.05s
10.866549284s	43.18s

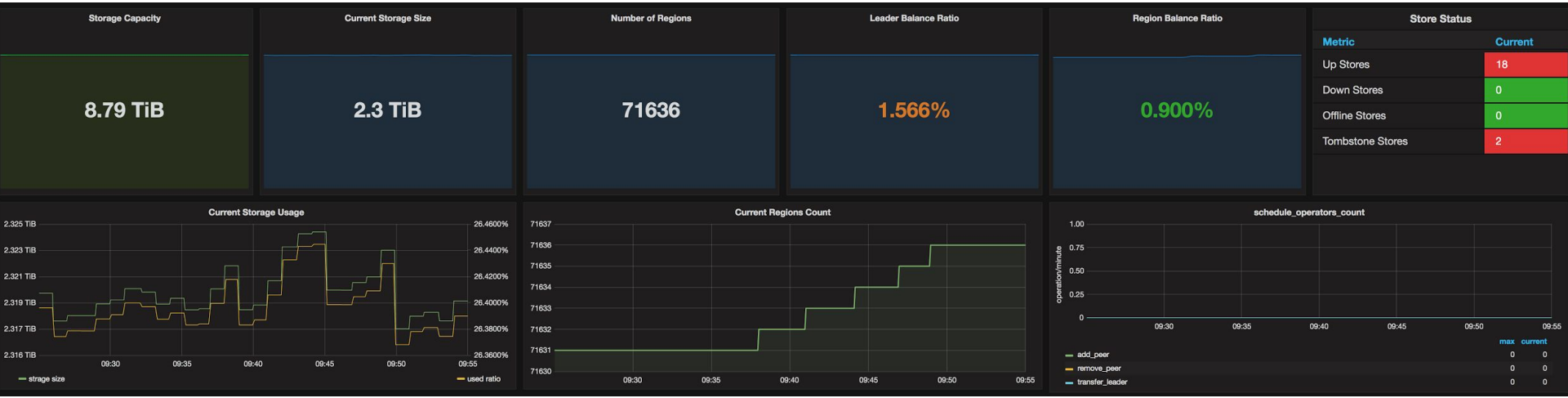
PingCAP

Use case 2: Distributed OLTP

- One of the biggest MMORPG game in China.
- 2.2 T, 18 nodes.
- Drop-in replacement for MySQL
- Distributed OLTP



PingCAP


Storage Capacity
8.79 TiB
Current Storage Size
2.3 TiB
Number of Regions
71636
Leader Balance Ratio
1.566%
Region Balance Ratio
0.900%
Store Status
Metric
Current
Up Stores
18
Down Stores
0
Offline Stores
0
Tombstone Stores
2
Current Storage Usage
2.325 TiB
2.323 TiB
2.321 TiB
2.319 TiB
2.317 TiB
2.316 TiB
26.4600%
26.4400%
26.4200%
26.4000%
26.3800%
26.3600%
09:30
09:35
09:40
09:45
09:50
09:55
strage size
used ratio
Current Regions Count
71637
71636
71635
71634
71633
71632
71631
71630
09:30
09:35
09:40
09:45
09:50
09:55
schedule_operators_count
operator/minute
1.00
0.75
0.50
0.25
0
09:30
09:35
09:40
09:45
09:50
09:55
max
current
add_peer
0
0
remove_peer
0
0
transfer_leader
0
0

PingCAP

Sysbench

OS	linux (ubuntu 14.04)
CPU	28 ECUs, 8 vCPUs, 2.8 GHz, Intel Xeon E5-2680v2
RAM	16 G
DISK	80 G (SSD)

Notice: 3 replicas

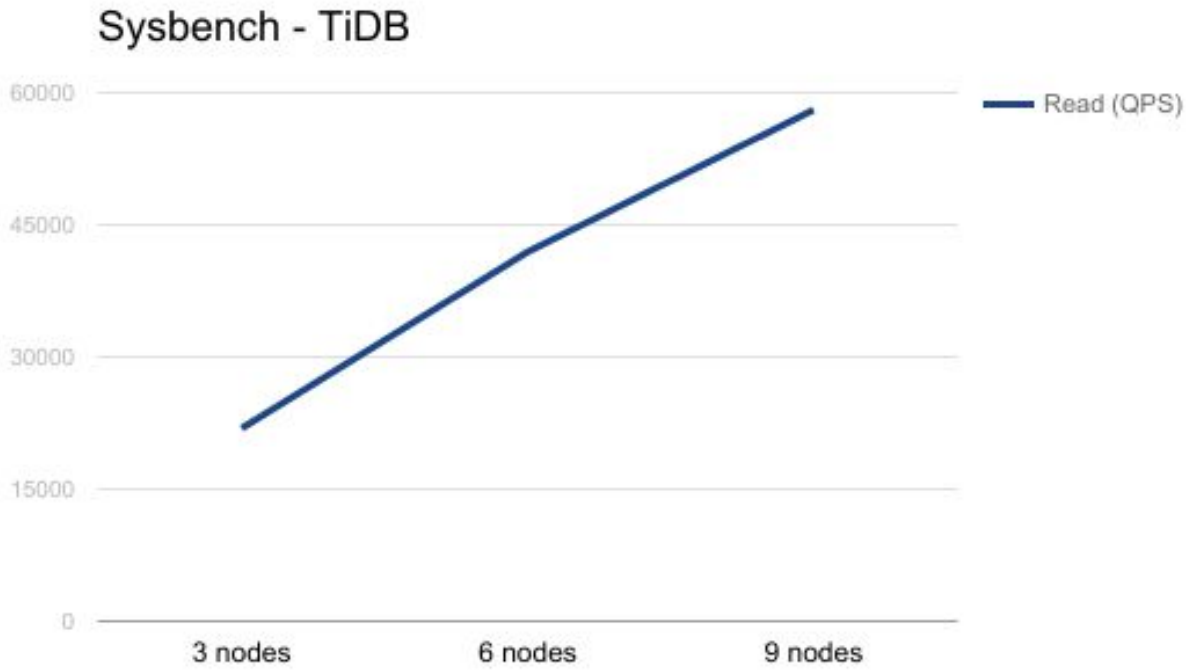
PingCAP

Sysbench (Read)

	table count	table size	sysbench threads	qps	latency(avg/.95)
3 nodes	16	1M rows	256	21899.59	11.69ms / 19.87ms
6 nodes	16	1M rows	256	41928.84	6.10ms / 10.96ms
9 nodes	16	1M rows	256	58044.80	4.41ms / 7.36ms

PingCAP

Sysbench (Read)



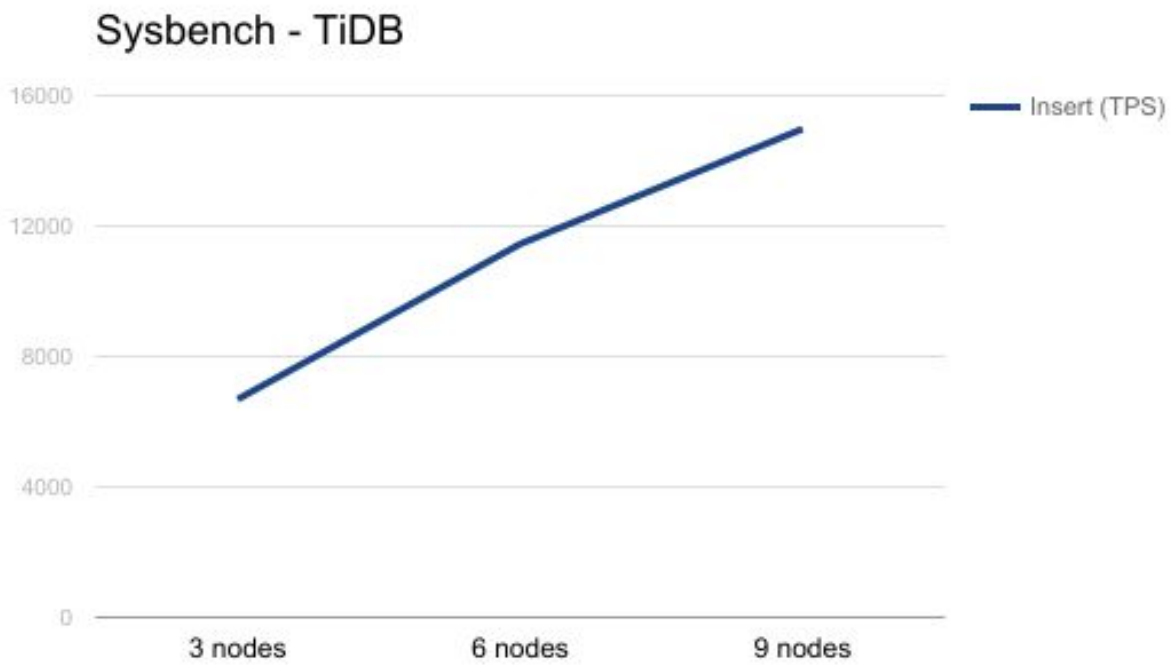
PingCAP

Sysbench (Insert)

	table count	table size	sysbench threads	TPS	latency(avg/.95)
3 nodes	16	1M rows	256	6686.59	38.28ms / 78.21ms
6 nodes	16	1M rows	256	11448.08	22.36ms / 44.61ms
9 nodes	16	1M rows	512	14977.01	34.18ms / 86.85ms

PingCAP

Sysbench (Insert)



PingCAP

Roadmap

- TiSpark: Integrate TiKV with SparkSQL
- Better optimizer (Statistic && CBO)
- Json type and document store for TiDB
  - MySQL 5.7.12+ X-Plugin
- Integrate with Kubernetes
  - Operator by CoreOS

Thanks

https://github.com/pingcap/tidb

https://github.com/pingcap/tikv

Contact me:

huang@pingcap.com

PingCAP