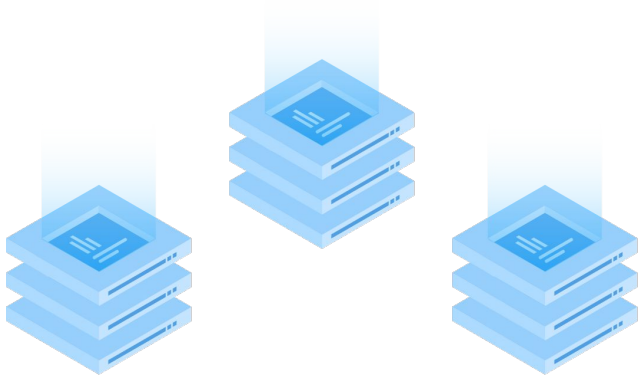
Deep Dive into the TiDB SQL Layer

zhangjian@pingcap.com

PingCAP TiDB



TiDB Community Slack Channel
https://pingcap.com/tidbslack/

About Me

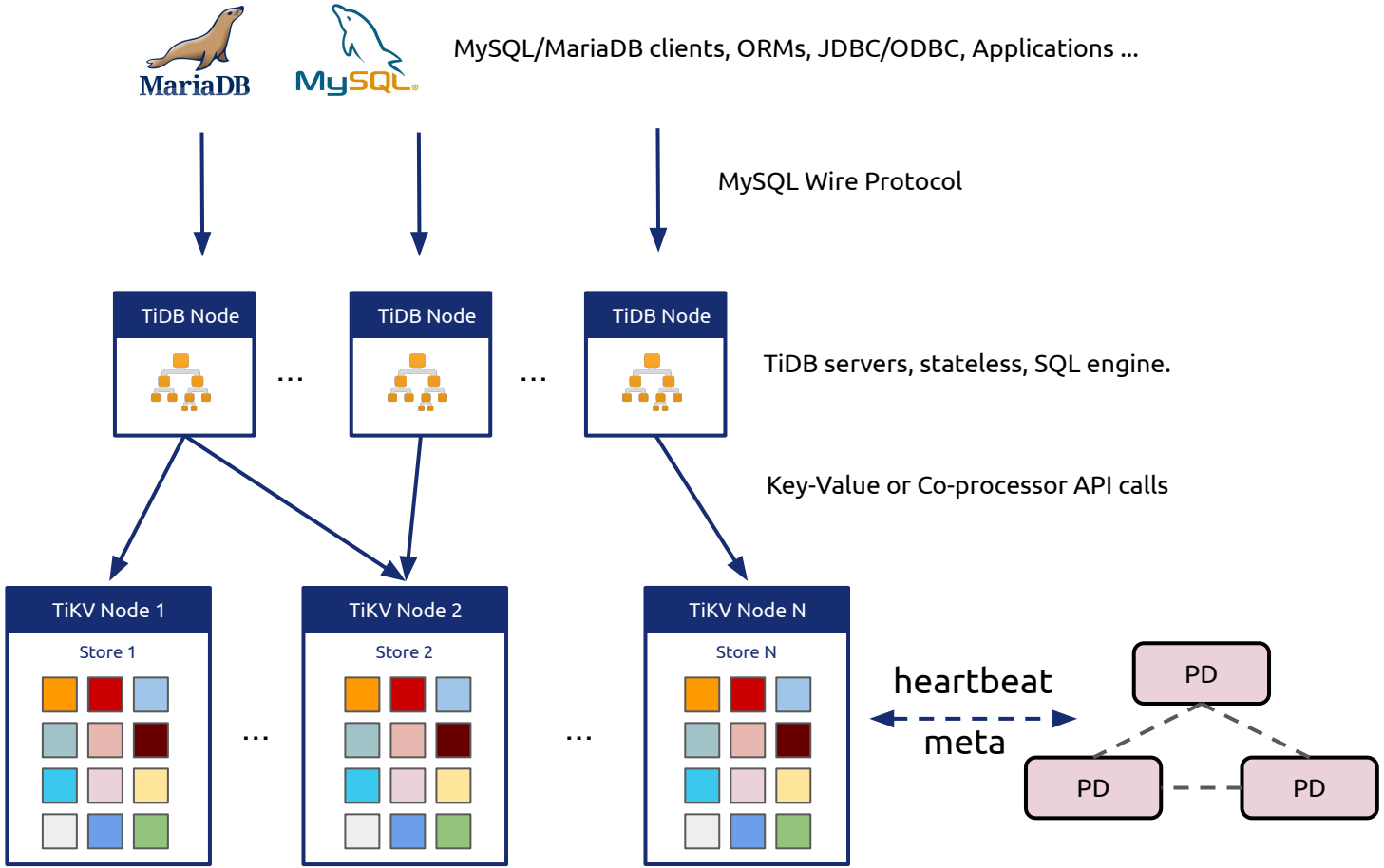
- Zhang Jian (张建)
- Was: Engineer, Runtime Team, Alibaba MaxCompute
- Now: Tech Lead, TiDB SQL Engine Team
- Focus on:
  - SQL Optimization
  - Distributed Computation

Topics

1. Intro to TiDB
2. Query Optimization
3. Statistics
4. Query Execution
5. Future Work

PingCAP

Part I - Intro to TiDB


MariaDB
MySQL
MySQL/MariaDB clients, ORMs, JDBC/ODBC, Applications ...
MySQL Wire Protocol
TiDB Node
TiDB Node
TiDB Node
TiDB servers, stateless, SQL engine.
Key-Value or Co-processor API calls
TiKV Node 1
Store 1
TiKV Node 2
Store 2
TiKV Node N
Store N
heartbeat
meta
PD
PD
PD

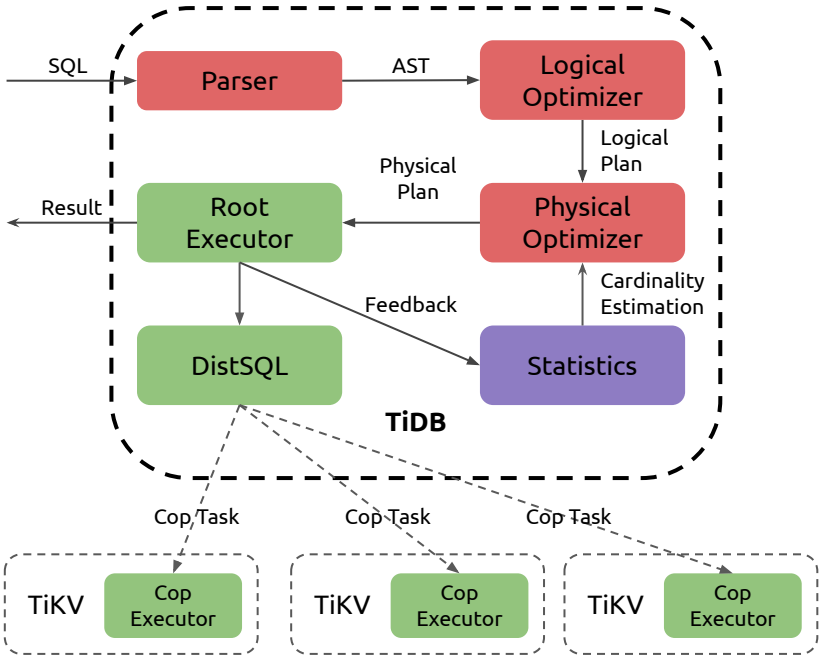
TiDB SQL Layer

Query Optimization

- Logical Optimizer
- Physical Optimizer
- Statistics

Query Execution

- Root Executor
- DistSQL
- Coprocessor Executor



PingCAP

Part II - Query Optimization

Query Optimization

A Query Execution Plan Example:

```
TiDB(root@127.0.0.1:test) > explain select count(*) from t1, t2 where t1.a=t2.a;
+--------------------------+----------+------+-------------------------------------------------------------------+
| id                       | count    | task | operator info                                                     |
+--------------------------+----------+------+-------------------------------------------------------------------+
| StreamAgg_13             | 1.00     | root | funcs:count(1)                                                    |
| └─MergeJoin_33           | 12500.00 | root | inner join, left key:test.t1.a, right key:test.t2.a               |
|   ├─IndexReader_25       | 10000.00 | root | index:IndexScan_24                                                |
|   | └─IndexScan_24       | 10000.00 | cop  | table:t1, index:a, range:[NULL,+inf], keep order:true, stats:pseudo |
|   └─IndexReader_28       | 10000.00 | root | index:IndexScan_27                                                |
|     └─IndexScan_27       | 10000.00 | cop  | table:t2, index:a, range:[NULL,+inf], keep order:true, stats:pseudo |
+--------------------------+----------+------+-------------------------------------------------------------------+
6 rows in set (0.00 sec)
```

Query Optimization

Query Optimization

- **search**, to find candidates
- **cost**, to choose the best

Query Optimization

Phase 1: Logical Optimization

- Equal
- Beneficial

```
func logicalOptimize(logic LogicalPlan) LogicalPlan {
    for _, rule := range ruleList {
        logic = rule.optimize(logic)
    }
    return logic
}
```

Logical Rules

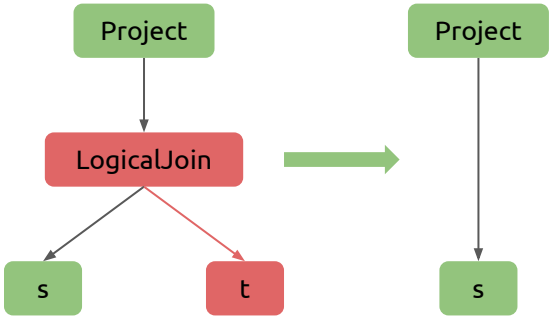
- Column Pruning
- Partition Pruning
- Group By Elimination
- Max/Min Elimination
- Project Elimination
- **Outer Join Elimination**
- Outer Join Simplification
- Subquery Decorrelation
- Predicate Push Down
- Aggregate Push Down
- TopN/Limit Push Down
- Join Reorder
- ...

Rule Example: Outer Join Elimination

- Only outer columns are used
- Join Key on inner side is unique

```
SELECT s.*
FROM s LEFT JOIN t
ON s.a = t.unique_key

SELECT * FROM s;
```

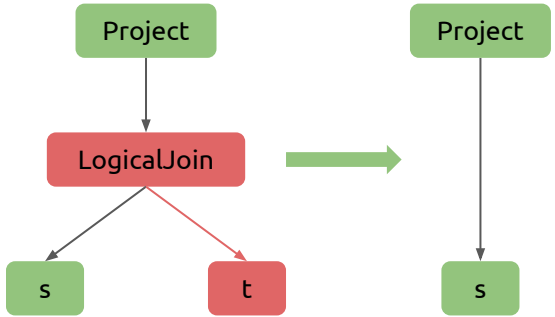


Rule Example: Outer Join Elimination

- ~~Only outer columns are used~~
- ~~Join Key on inner side is unique~~
- Only **distinct** outer columns are used

```
SELECT SUM( DISTINCT  s.a)
FROM s LEFT JOIN t
ON s.b = t.b;

SELECT SUM( DISTINCT  s.a) FROM s;
```



Query Optimization

Phase 1: Logical Optimization

- Equal
- Beneficial

Phase 2: Physical Optimization

Physical Optimization

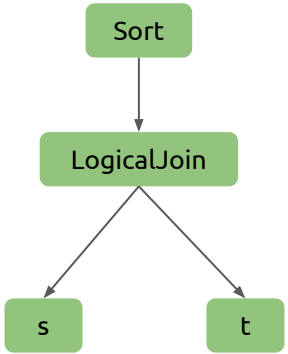
Physical Property:

- data order
- task type

The Dynamic Programming:

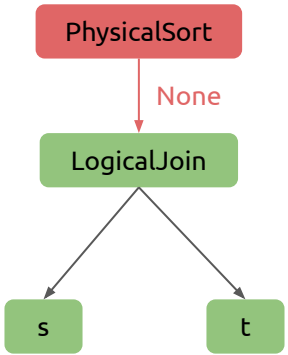
- (Logical Plan , Required Physical Property) -> Physical Plan

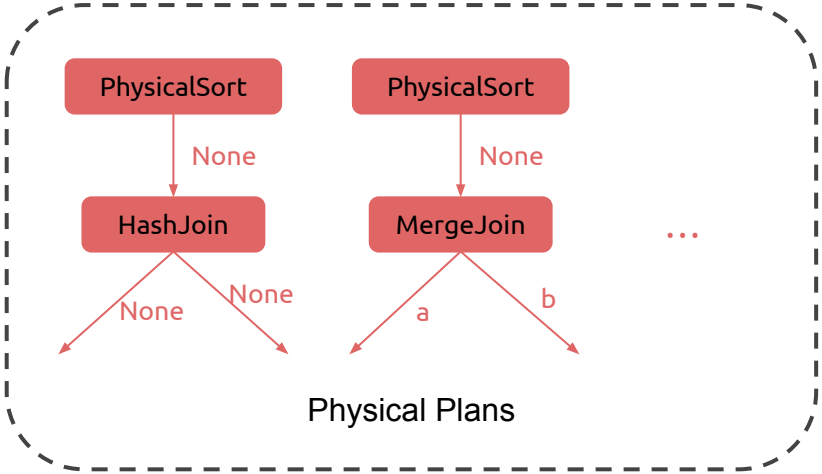
Physical Optimization




```
select *
from s
join t
on s.a = t.b
order by s.a
```

Physical Optimization





PingCAP

Part III - Statistics

Statistics

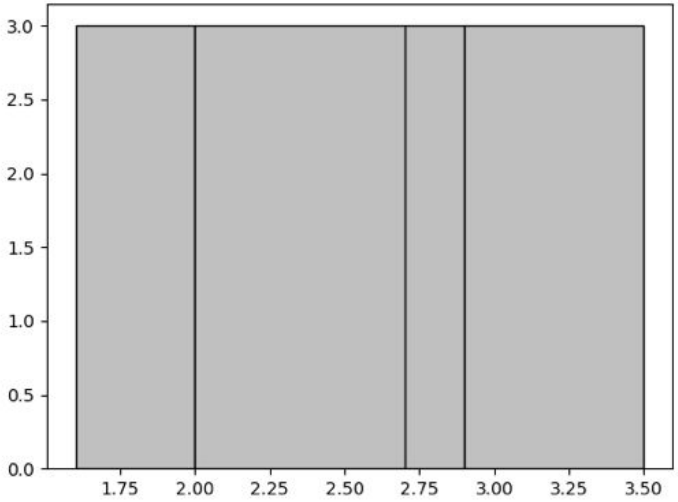
Cardinality Estimation

- **WHERE** s.a > 0
- **ON** s.a = t.b
- **GROUP BY** s.a, s.b

Statistics

What kinds of statistics does TiDB need?

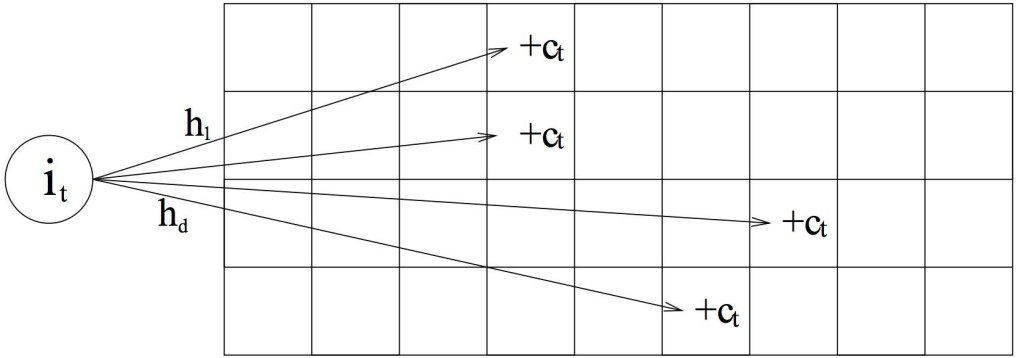
- Equi-depth Histogram



Statistics

What kinds of statistics does TiDB need?

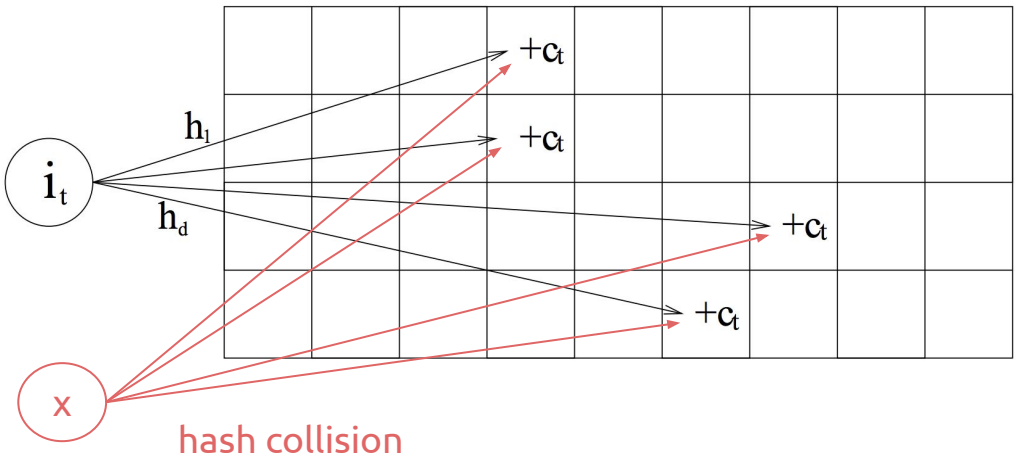
- CM-Sketch



Statistics

What kinds of statistics does TiDB need?

- CM-Sketch **with TopN**



TopN Statistic

Data	Count
x1	10000
x2	9990
...	...

Comparing to MySQL

TiDB is not a fork :-) Both systems will have unique benefits.

Major differences:

1. TiDB has more operators to consider
   (hash join, merge join, hash aggregation etc.)
2. The cost model must consider network cost
3. Implements most MySQL optimizations, but currently not
   index merge, loose index scan
4. Optimizer never dives the index (only considers histograms + CM-Sketch)
5. Approach is more layered (necessitated by #1)

PingCAP

Part IV - Query Execution

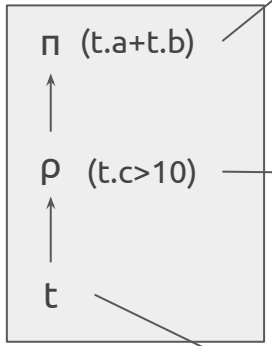
Query Execution

Since TiDB 2.0: Vectorized/Batch Model

- Based on the Iterator Model
- Column Oriented Memory Layout
- Vectorized Evaluation
- Less cache/branch/TLB miss, more efficiency

Query Execution

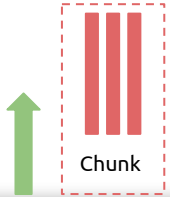
```
select t.a+t.b
from t
where t.c>10
```




```
input = child.Next()
output = proj(input, t.a+t.b)
return output
```

```
while |output|<batchSize:
    input = child.Next()
    output += filter(input, t.c>10)
return output
```

```
while |output|<batchSize:
    output += scan(t)
return output
```



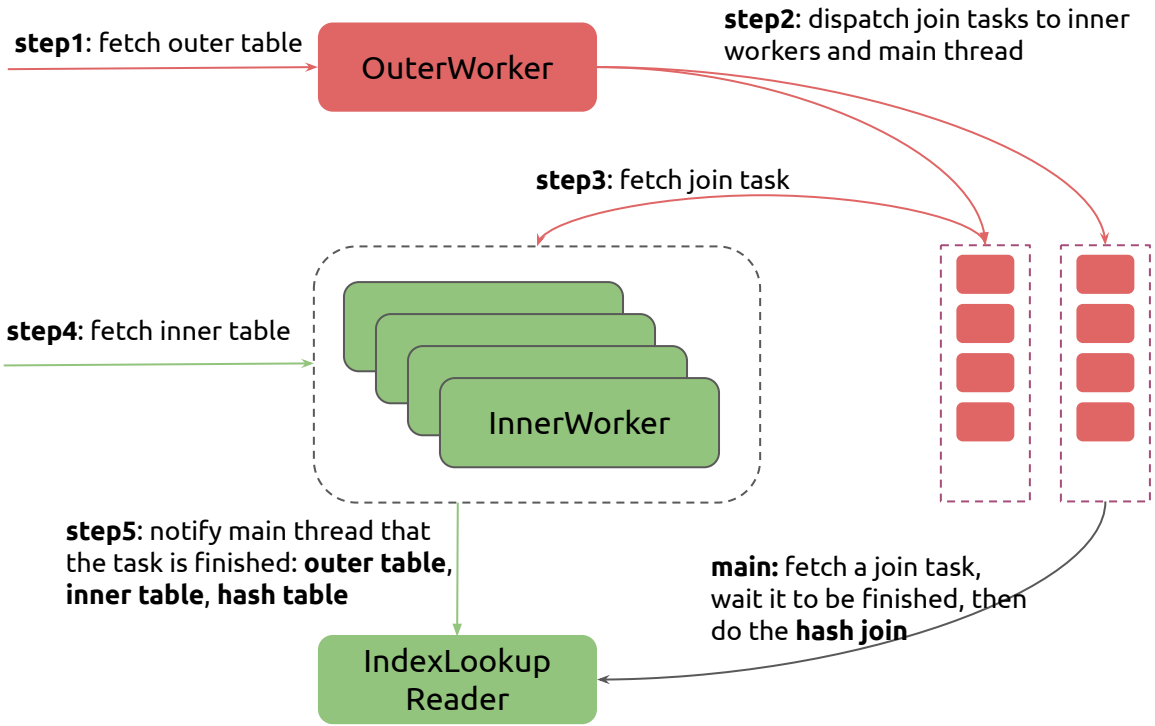
Query Execution

Parallel physical operator implementations:

- Hash Join
- Index Join
- Hash Aggregate
- …

Under development: https://github.com/pingcap/tidb/projects/11

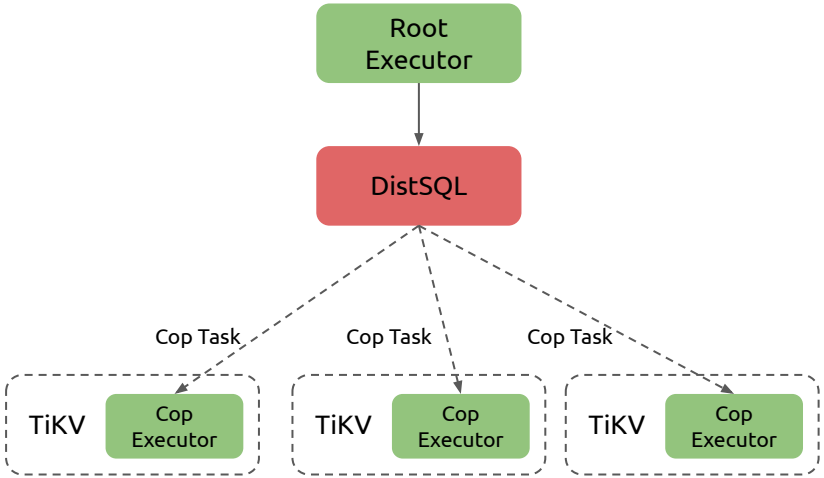
Query Execution/Index Join



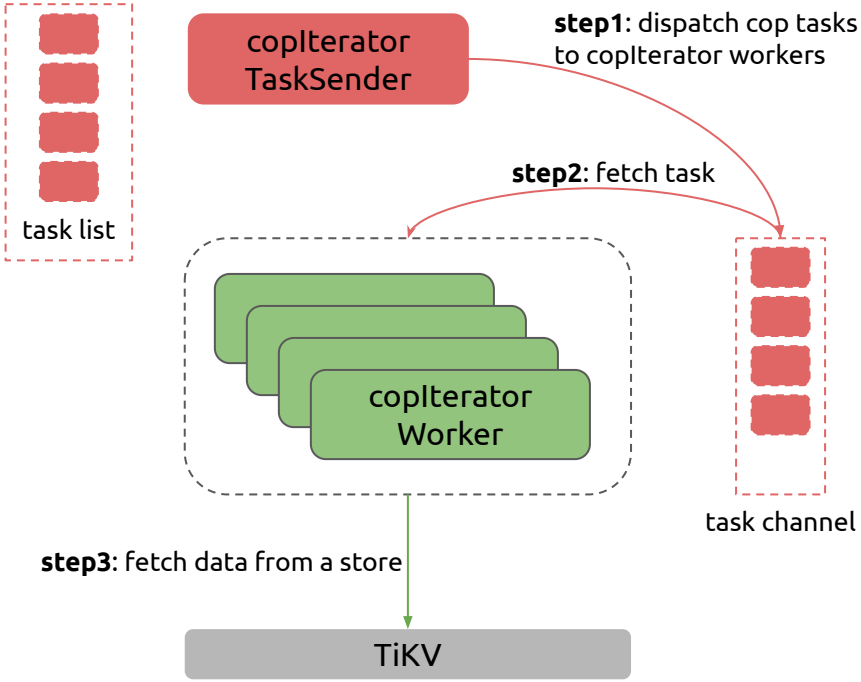
Query Execution/DistSQL

DistSQL:

1. Send Cop request
2. Receive Cop response
3. Handle TiKV Region errors



Query Execution/DistSQL



copIterator

PingCAP TiDB

PingCAP

Part V - Future Work

Future Work

Towards a Self-Driving Database

- Cascades Query Planner
- Multi-Column Statistics
- SQL Tuning Advisor
- SQL Plan Evolution
- Multi-Index Scan

Query Execution

- Parallel Control & Scheduling
- Memory Control

PingCAP TiDB

PingCAP

Thank You !

TiDB Community Slack Channel
https://pingcap.com/tidbslack/