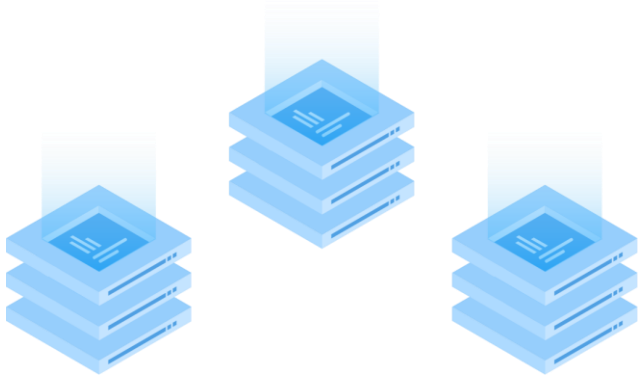
Index Advisor Overview

Presented by ZouHuan

PingCAP TiDB



PingCAP

Part I – Proposal Overview

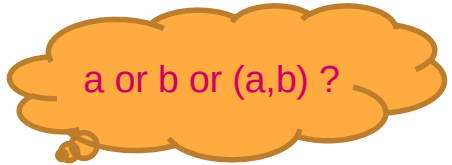
Our Goals

- **Inputs**
  Query Set (multiple queries, only **select** statement)
  Database (multiple tables)
- **Outputs**
  Recommended Index
- **Criterion**
  Execution time of the query set without index– t_none
  Execution time of the same query set with index – t_index

$$t\_index < t\_none$$

Our Tools ➢ Inputs

- Query Set




```
SELECT * FROM t1 WHERE a >1

SELECT * FROM t1 WHERE a =1 ORDER BY b

SELECT dt.d_year, item.i_brand_id brand_id ,item.i_brand brand, sum(ss_ext_sales_price) sum_agg
FROM date_dim dt, store_sales, item
WHERE dt.d_date_sk = store_sales.ss_sold_date_sk and store_sales.ss_item_sk = item.i_item_sk
  and item.i_manufact_id = 436 and dt.d_moy=12
GROUP BY dt.d_year, item.i_brand, item.i_brand_id
ORDER BY dt.d_year, sum_agg desc, brand_id
LIMIT 100;
```

- Database Information ? ? ? ?

- **For single query**, what's the potential index?
- **For query set**, how to build global candidate index set from local indexes?

Our Tools ➢ SQL Optimizer

- What's the possible potential index?
  - **Infoschema oriented** -- Brute Fore and Ignore Enumeration (BFI Enumeration)

    for table who has 10 cols,
    [1 column index]: 10 combinations
    [2 columns index]: 10*9 combinations
    [3 columns index]: 10*9*8 combinations
    ……
  - **Query oriented** – Smart Enumeration

    SELECT a FROM t1 WHERE b=1 and c = 1 ORDER BY d

    - (b, c, d) and (b, c, d) are the most desirable
    - consider the cost of adding index to tables into consideration
  - **Combine BFI Enumeration and Smart Enumeration**

    BFI Enumeration: one and two column index
    Smart Enumeration: three and four column index

Our Tools ➢ SQL Optimizer

- What's the criterion to evaluate potential indexes?
  - **For Single Query -- Physical Plan**

assume all possible potential indexes are available to physical optimizer
SQL optimizer output physical plan
indexes contained in physical plan are regarded as local candidate indexes for current query

  - **For Query Set – bestTask.Cost()**

original_cost: best task's cost without virtual indexes on table info
virtual-cost: best task's cost with virtual indexes on table info
virutal index's benefits: (virtual_cost – original_cost) / index_size

  - **Criterion**

Global index: always favor local virtual indexes with high benefits

Our Tools ➢ SQL Optimizer

- Gaps between local and global optimal index set

[Query 1]: {a:100}
[Query 2]: {b: 80}, {c: 80}, {d: 80}, {a,c: 80}
[Query 3]: {e: 60}, {d,e: 60}

Requirements: global indexes size is at most 3

{a}, {b}, {c} or

{a}, {b}, {d} or

{a}, {e}, {d,e} or

…..

cannot evaluate each index's contribution to the overall task independently!

Bridge the gap between the local and global
Swap and Re-Evaluate

```
sort(GCIS, GCIS.Cost_Efficiency, Descend_Order)
Recommend_Set = GCIS[:N]
Remaining_Set = GCIS[N+1:]
Best_Cost = Evaluate_QueryBatch(QuerySet, Recommend_Set)
for {
    V_Recommend_Set, V_Remaining_Set = SwapElements(Recommend_Set, Remaining_Set, M)
    Variant_Cost = Evaluate_QueryBatch(QuerySet, V_Recommend_Set)
    if Variant_Cost < Best_Cost{
        Best_Cost = Variant_Cost
        Recommend_Set = V_Recommend_Set
        Remaining_Set = V_Remaining_Set
    }
    ....
    //Swap and Evaluate，直到超时或者运行满一定的轮数,break
}
```

PingCAP

Part II – Code Overview

Integrate with TiDB

- What's the desirable output of TiDB?

  - local selected index: obtained from Physical Plan
  - Index's benefits: obtained from beskTask.Cost()

- How can we get the desirable output ?

  For local selected index:

  - they can be extracted from physical plan
  - the following statement execution step is not necessary

  For index's benefits:

  - they requires two cost: 1) cost without virtual indexes; 2) cost with virtual indexes
  - difference between the two cost: their indexes information are different

Q:

How can we get the physical plan, cost from the execution flow of TiDB server?
How can we insert virtual index information and let TiDB do normal SQL Optimization?

Get internal result from TiDB

- Session Variable – tidb_enable_index_advisor

  If current session is in index advisor mode, do:

  - **session.execute:**

    skip statement execution step

  - **optimizer.physicalOptimize:**

    return bestTask instead of bestTask.plan()

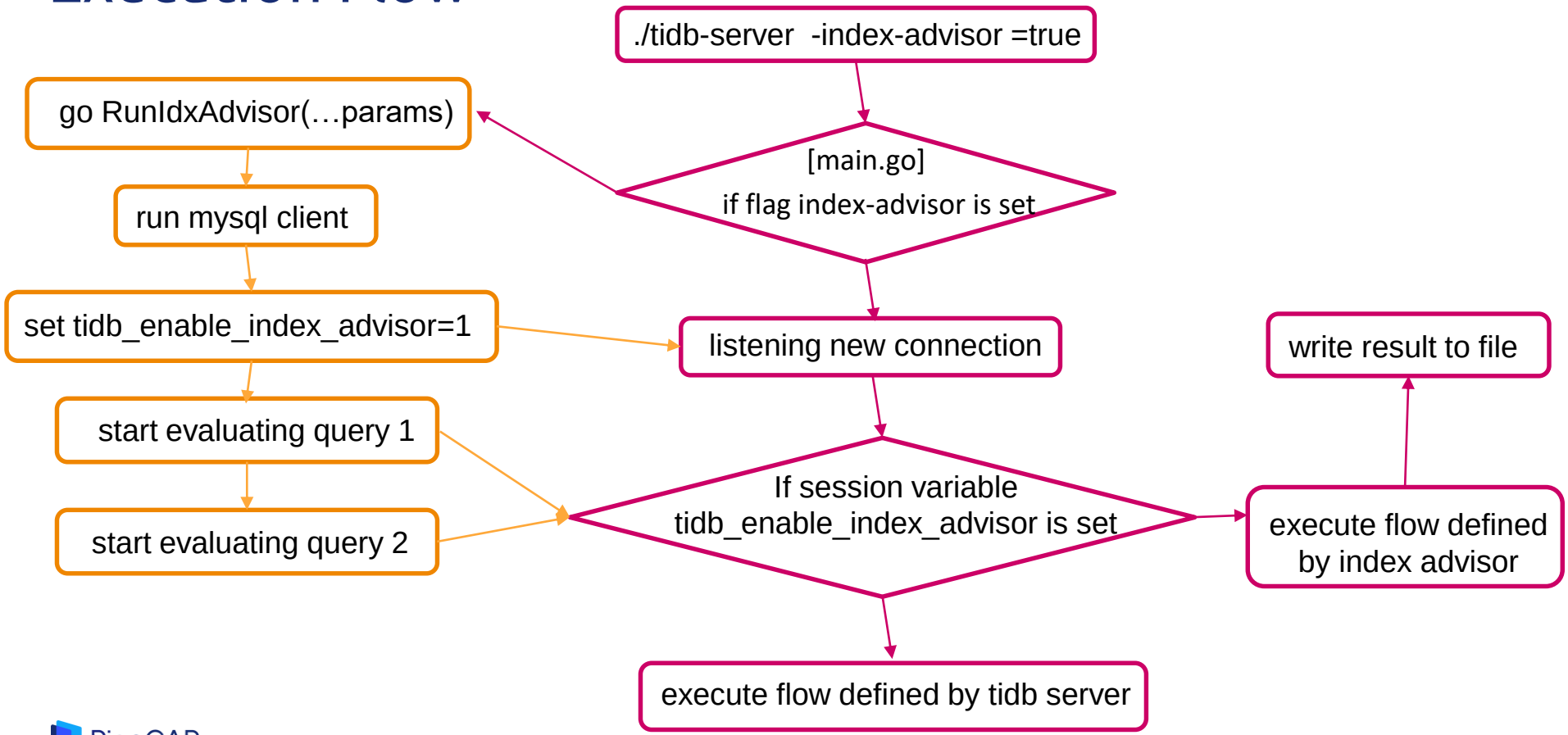
  - **compiler.compile:**

    build virtual infoschema according to original infoschema and given query

    get virtual physical plan with passed-in virtual infoschema

- New SQL statement like *'explain'*

Execution Flow

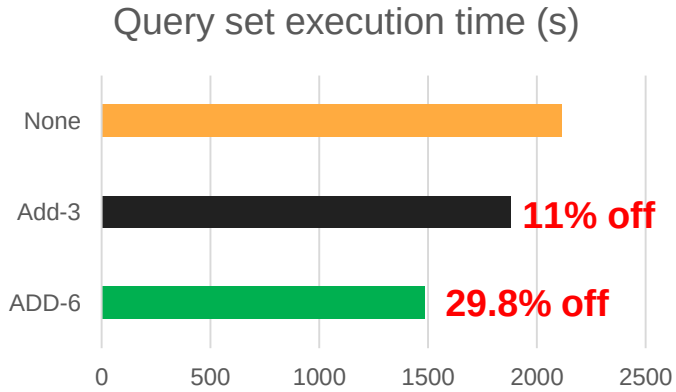


PingCAP

Part III – Experiments

TPCH Benchmark

- 21 Queries (15.sql excluded)
- 32 GB Memory
- 16 Cores

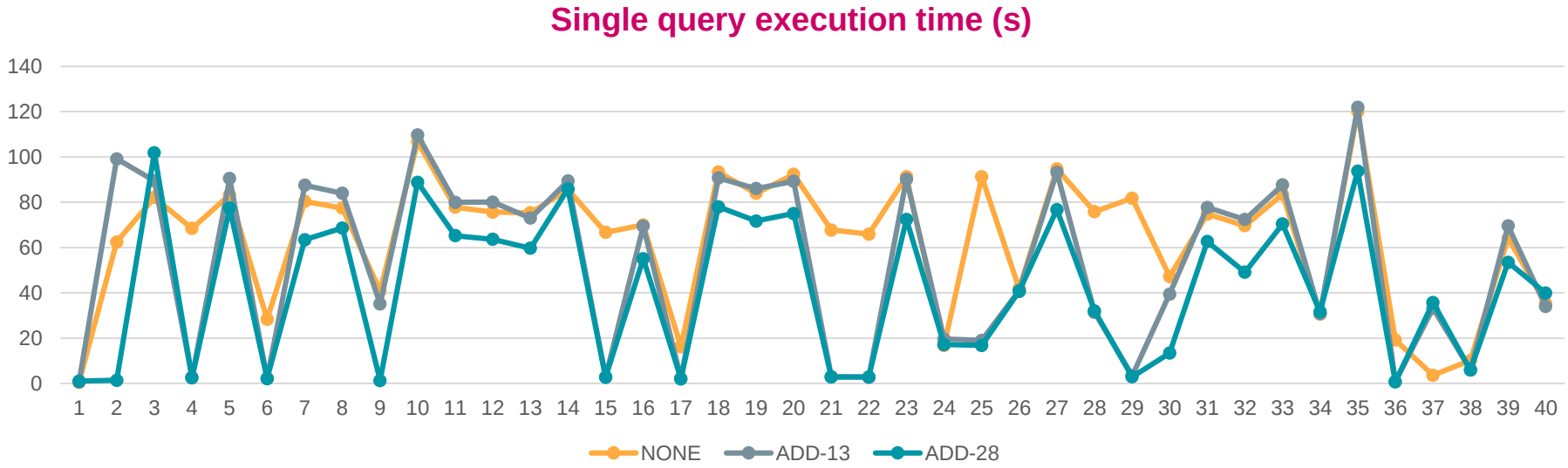


Recommend Index	Add Index Cost
S_NATIONKEY	1.02 s
L_PARTKEY	1264.87 s
N_NAME	0.02 s
L_PARTKEY L_QUANTITY	1260.52 s
P_BRAND P_CONTAINER	34.367981 s
L_SHIPDATE	1014.79 s

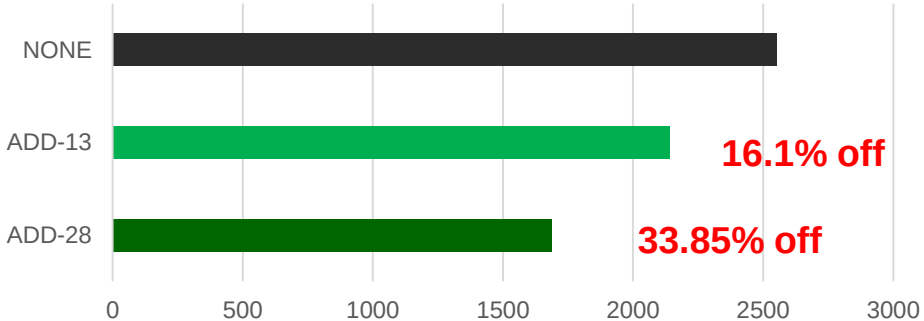
PingCAP

TPCDS Benchmark

- 40 Queries (3.sql and 85.sql excluded)
- 64 GB Memory
- 16 Cores



Query set execution time (s)



PingCAP

PingCAP

Thank You !