Query Optimizer
Chen Luo,
Zhanxiang(Patrick) Huang,
Bowie Chen
Overview

▷ **Current Simple Optimizer**
  ○ Hard-coded rules, hard to extend

▷ **New Optimizer**
  ○ Cascade framework, object-oriented

Diagram:

1. **Queries** → **Parser** → **Parse Tree** → **Binder** → **Optimizer** → **Peloton Plan**
Goals

▷ 75%
  ○ Support TPC-C Benchmark
  ○ Single table queries & 2 way joins

▷ 100%
  ○ Support all functionalities of executors
  ○ Implementation rules & Multi-way joins

▷ 125%
  ○ Transformation rules
SELECT T.id1 + T.id2, name
FROM TEAMMATE as T, Course as C
WHERE T.cid = C.id AND id < 3
SelectStatement {
  SelectElements: PlusExpression (id1+id2),
  TupleValueExpression
    (name)
  FROM: TableRef T, TableRef C
  WHERE: ComparisonExpression
    (T.cid = C.id AND id < 3)
}
Binder: Bind object ids to expression

SelectStatement {
  SelectElements: PlusExpression (id1 + id2),
  TupleValueExpression (name)
  FROM: TableRef T, TableRef C
  WHERE: ComparisonExpression
    (T.cid = C.id AND id < 3)
}

① Collect Context From Tables
② Annotated with the corresponding table id, column id and table alias
Binder

- Need to
- And some text
- But remember not to overload your slides with content

Your audience will listen to you or read the content, but won’t do both.
Cascade Framework

▷ Top-down optimization
  ○ Speed up by memorization

▷ Properties and rules as object
  ○ Highly extensible
Cascade Framework

▷ Top-down optimization
  ○ Speed up by memorization

▷ Properties and rules as object
  ○ Highly extensible
Single Table Optimization

Select a, MAX(b) FROM table GROUP BY a ORDER BY a
Select a, MAX(b) FROM table GROUP BY a ORDER BY a
Select a, MAX(b) FROM table GROUP BY a ORDER BY a
Single Table Optimization

Select a, MAX(b) FROM table
GROUP BY a ORDER BY a

Output Plan
Sort(a)

Aggregate (Group 0)
SortGroupBy
Sort(a)

HashGroupBy
None

PhysicalOrderBy
Sort(a)

Get(table)
IndexScan(table)
SeqScan(table)
Single Table Optimization

Select a, MAX(b) FROM table
GROUP BY a ORDER BY a
Demo
Test

▷ **Unit Test**
  ○ Correctness of physical plan & query results

▷ **TPC-C Benchmark**
  ○ Support all TPC-C queries
Future Works

▷ Predicate Pushdown
  ○ Modify predicates

▷ Join Transformation Rules
  ○ $A \bowtie (B \bowtie C) \rightarrow (A \bowtie B) \bowtie C$

▷ Integrate stats and cost Models
Thanks!

Any questions?