ORACLE

Optimizer Trace for Hypergraph Optimizer

JSON-based

Øystein Grøvlen

MySQL Optimizer Team Lead

Oracle

May 26, 2026

Øystein Grøvlen

MySQL Optimizer Team Lead, Oracle

Safe harbor statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, timing, and pricing of any features or functionality described for Oracle's products may change and remains at the sole discretion of Oracle Corporation.

MySQL Optimizer Trace

- EXPLAIN shows the selected plan
- Optimizer trace shows WHY the plan was selected

```
SET optimizer_trace='enabled=on';
SELECT * FROM t1, t2 WHERE f1=1 AND f1=f2 AND f2>0;
SELECT trace FROM information_schema.optimizer_trace
  INTO OUTFILE filename LINES TERMINATED BY '';
SET optimizer_trace='enabled=off';
```

QUERY	SELECT * FROM t1, t2 WHERE f1=1 AND f1=f2 AND f2>0;
TRACE	{ "steps": [{ "join_preparation": { "select#": 1,... } ... } ...] }
MISSING_BYTES_BEYOND_MAX_MEM_SIZE	0
INSUFFICIENT_PRIVILEGES	0

Optimizer Trace

Classical Join Optimizer

32

Basis for tools

joinopttrace

```
Table AccessType:IndexName Rows/Cost TotalRows/TotalCost
------------------------------------------------------
`customer` scan 2367/244.2 478.9/244.2
  `lineitem` scan 2495/119491 1190000/119735
    `orders` eq_ref:PRIMARY 1/120108 59742/239843 *** NEW BEST PLAN ***
  `orders` scan 2502/101621 478.9/101865
    `lineitem` ref:PRIMARY 1.0155/168.35 486.3/102033 *** NEW BEST PLAN ***
`lineitem` scan 2495/255.5 2495/255.5
  `customer` scan 2367/119716 1190000/119972 PRUNED(cost)
  `orders` eq_ref:PRIMARY 1/873.25 2115.1/1128.8
    `customer` eq_ref:PRIMARY 1/740.27 427.92/1869 *** NEW BEST PLAN ***
`orders` scan 2502/255.7 2121/255.7
  `customer` eq_ref:PRIMARY 1/742.35 429.12/998.05
    `lineitem` ref:PRIMARY 1.0155/150.86 435.76/1148.9 *** NEW BEST PLAN ***
  `lineitem` ref:PRIMARY 1.0155/745.63 2153.8/1001.3 PRUNED(heuristic)
```

Hypergraph Optimizer

Current optimizer trace

```
…
"Enumerating subplans:",
"",
"Found node orders [rows=1500000]",
" - using selectivity 0.500 (750000 rows) from range scan on index i_o_orderdate to cover ((orders.o_orderDATE < DATE'1995-03-24'))",
" - {INDEX_RANGE_SCAN, cost=3.96e+6, init_cost=0, rows=750000} [i_o_orderdate range] is first alternative, keeping",
" - {SORT, cost=4.69e+6, init_cost=4.69e+6, rows=750000, order=6} [i_o_orderdate range, sort(7)] is potential alternative, keeping",
" - {INDEX_RANGE_SCAN, cost=3.96e+6, init_cost=0, rows=750000, order=6} [i_o_orderdate ordered range] is better than all previous alternatives, replacing all",
" - {INDEX_RANGE_SCAN, cost=3.96e+6, init_cost=0, rows=750000, order=7} [i_o_orderdate ordered range] is not better than existing path {INDEX_RANGE_SCAN, cost=3.
" - {INDEX_SCAN, cost=2.31e+6, init_cost=0, rows=750000, order=4} [PRIMARY] is better than previous {INDEX_RANGE_SCAN, cost=3.96e+6, init_cost=0, rows=750000, or
" - PRIMARY is applicable for ref access",
" - (lineitem.l_orderkey = orders.o_orderkey) is subsumed by ref access on orders.o_orderkey",
" - {EQ_REF, cost=4.75, init_cost=0, rows=0.5, parm={lineitem}, order=4} [PRIMARY] is potential alternative, keeping",
" - {INDEX_SCAN, cost=2.31e+6, init_cost=0, rows=750000, order=5} [PRIMARY] is not better than existing path {INDEX_SCAN, cost=2.31e+6, init_cost=0, rows=750000,
" - PRIMARY is applicable for ref access",
" - (lineitem.l_orderkey = orders.o_orderkey) is subsumed by ref access on orders.o_orderkey",
" - {EQ_REF, cost=4.75, init_cost=0, rows=0.5, parm={lineitem}, order=5} [PRIMARY] is not better than existing path {EQ_REF, cost=4.75, init_cost=0, rows=0.5, pa
" - i_o_custkey is applicable for ref access (using 1/2 key parts only)",
" - (customer.c_custkey = orders.o_custkey) is subsumed by ref access on orders.o_custkey",
" - {REF, cost=53, init_cost=0, rows=5, parm={customer}} [i_o_custkey] is potential alternative, keeping",
" - i_o_custkey is applicable for ref access",
" - (lineitem.l_orderkey = orders.o_orderkey) is subsumed by ref access on orders.o_orderkey",
" - (customer.c_custkey = orders.o_custkey) is subsumed by ref access on orders.o_custkey",
" - {EQ_REF, cost=2.19, init_cost=0, rows=3.33e-6, parm={lineitem, customer}} [i_o_custkey] is potential alternative, keeping",
" - {INDEX_SCAN, cost=8.01e+6, init_cost=0, rows=750000, order=6} [i_o_orderdate] is not better than existing path {INDEX_SCAN, cost=2.31e+6, init_cost=0, rows=7
" - {INDEX_SCAN, cost=8.01e+6, init_cost=0, rows=750000, order=7} [i_o_orderdate] is not better than existing path {INDEX_SCAN, cost=2.31e+6, init_cost=0, rows=7
" - current access paths for {orders}: {INDEX_SCAN, cost=2.31e+6, init_cost=0, rows=750000, order=4}, {EQ_REF, cost=4.75, init_cost=0, rows=0.5, parm={lineitem},
"",
"Found node customer [rows=150000]",
" - {INDEX_SCAN, cost=350115, init_cost=0, rows=29717} [PRIMARY] is first alternative, keeping",
" - PRIMARY is applicable for ref access",
" - (customer.c_custkey = orders.o_custkey) is subsumed by ref access on customer.c_custkey",
" - {EQ_REF, cost=5.5, init_cost=0, rows=0.198, parm={orders}} [PRIMARY] is potential alternative, keeping",
…
```

Structured trace

One JSON object for each proposed access path

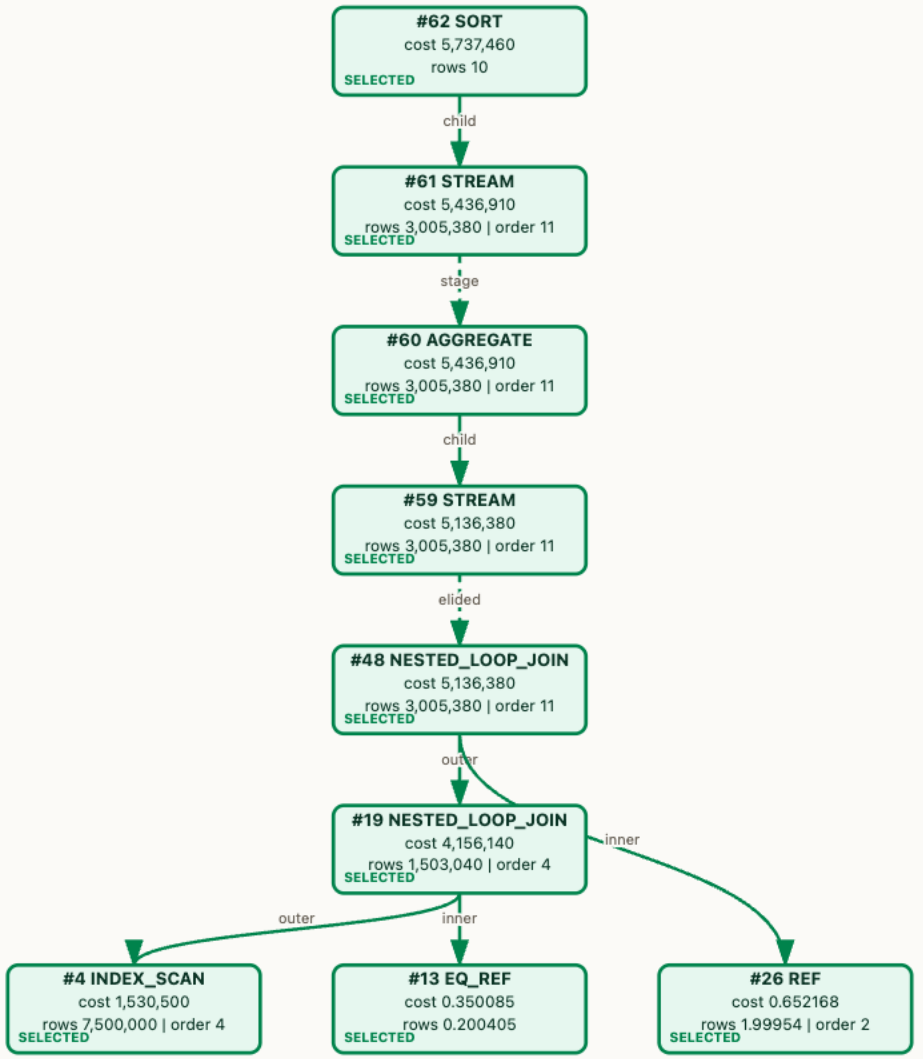

```
{
  "id": 0,
  "type": "INDEX_RANGE_SCAN",
  "cost": 2.62565e+06,
  "init_cost": 0,
  "rows": 7.5e+06,
  "safe_for_rowid_once": false,
  "unsafe_for_rowid": false,
  "description_for_trace": "i_o_orderdate range",
  "index": "i_o_orderdate",
  "keeping": true,
  "cause": "first"
},
{
  "id": 1,
  "type": "SORT",
  "cost": 2.05045e+07,
  "init_cost": 2.05045e+07,
  "rows": 7.5e+06,
  "order": 6,
  "safe_for_rowid_once": false,
  "unsafe_for_rowid": false,
  "description_for_trace": "i_o_orderdate range, sort(7)",
  "child": {
    "id": 0,
    "type": "INDEX_RANGE_SCAN"
  },
  "keeping": true,
  "cause": "potential_alternative”
},
```



```
{
  "left": "{customer}",
  "right": "{orders}",
  "connect_subgraph": [
    {
      "type": "hash_join",
      "left": "{orders}",
      "right": "{customer}",
      "condition": "(customer.c_custkey = orders.o_custkey)",
      "access_paths": [
        {
          "id": 14,
          "type": "HASH_JOIN",
          "cost": 2.57021e+06,
          "init_cost": 184491,
          "rescan_cost": 2.38572e+06,
          "rows": 1.50304e+06,
          "safe_for_rowid_once": false,
          "unsafe_for_rowid": false,
          "outer": {
            "id": 4,
            "type": "INDEX_SCAN"
          },
          "inner": {
            "id": 12,
            "type": "TABLE_SCAN"
          },
          "keeping": true,
          "cause": "first"
        },
        …
```

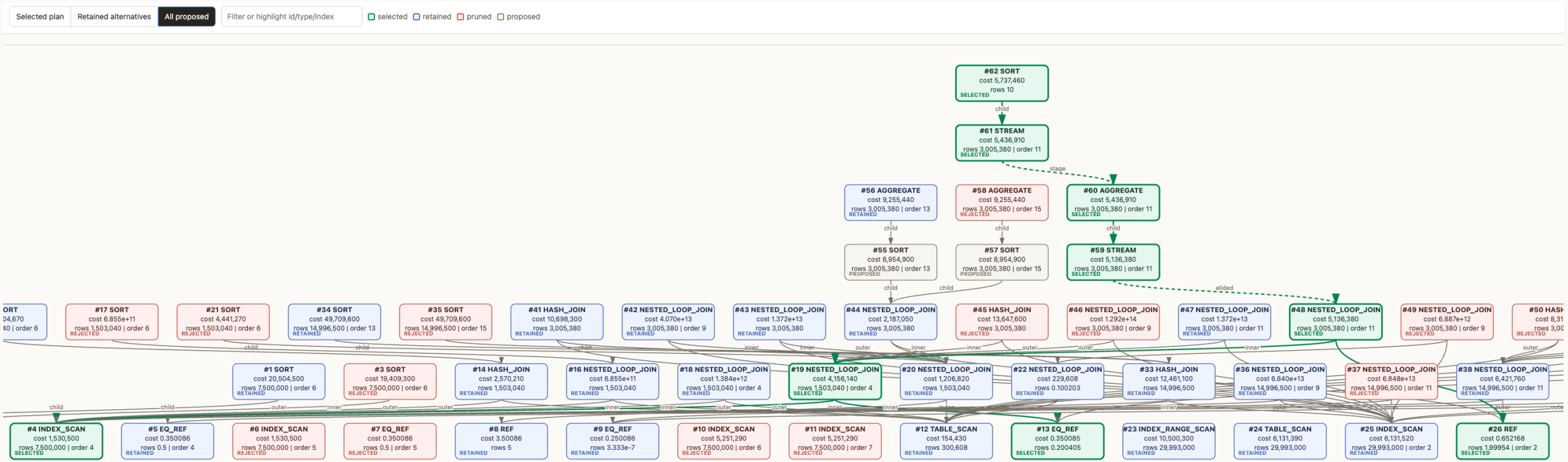
Hypergaph Optimizer

Visualization – Best plan



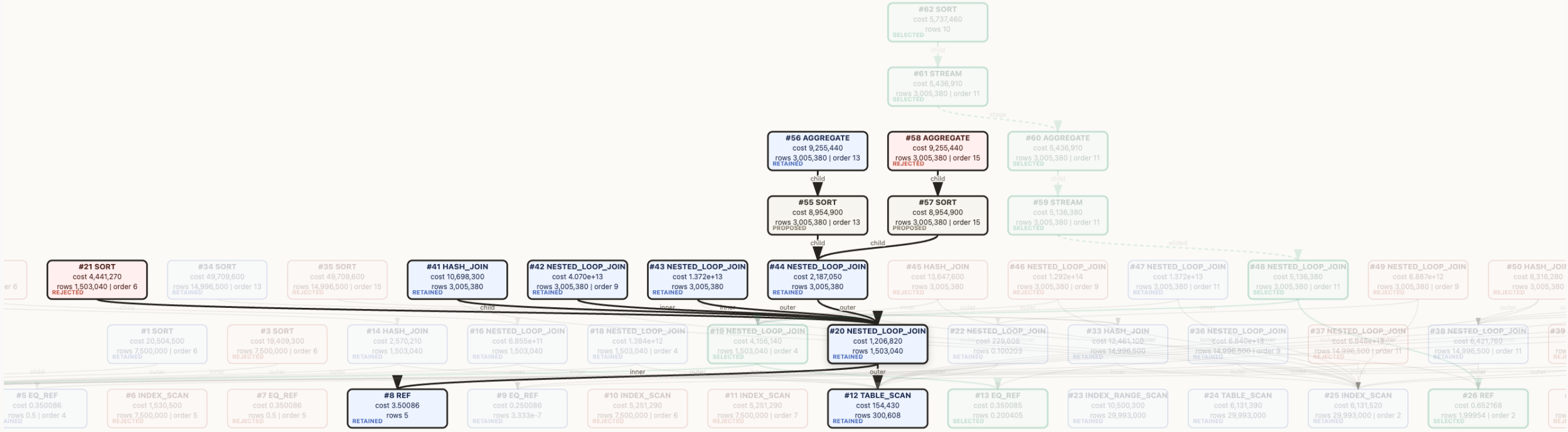
Hypergaph Optimizer

Visualization – All proposed subplans



Hypergaph Optimizer

Visualization – Subplans containing one specific access path



Call for contributions

- Presentation is based on an outdated prototype
  - Needs to be updated and extended
- Good way to learn the inner workings of the hypergraph optimizer
- Ideas and contributions on tools for visualization are welcomed!

Thank you


ORACLE