

ORCA的Agg实现

Agg+子连接

```
1  EXPLAIN
2      SELECT count((SELECT SUM(a) from test_a))
3          FROM test_b
4          group by b;
```

设置 `optimizer_enforce_subplans` 为true，不扁平化转Join行计划

```
1  -- Enforce correlated execution in the optimizer
2  SET optimizer_enforce_subplans=on;
```

采取相关执行，也就是子连接得到subplan广播+物化的方式，可以得到下面的执行计划，

```

1  Gather Motion 3:1 (slice1; segments: 3) (cost=0.00..1324044.43 rows=2
width=8)
2    -> HashAggregate (cost=0.00..1324044.43 rows=1 width=8)
3        Group Key: test_b.b
4    -> Redistribute Motion 3:3 (slice2; segments: 3)
(cost=0.00..1324044.39 rows=334 width=8)
5        Hash Key: test_b.b
6    -> Seq Scan on test_b (cost=0.00..1324044.39 rows=334
width=8)
7        SubPlan 1
8        -> Materialize (cost=0.00..431.00 rows=2
width=4)
9            -> Broadcast Motion 3:3 (slice3;
segments: 3) (cost=0.00..431.00 rows=2 width=4)
10                -> Seq Scan on test_a
(cost=0.00..431.00 rows=1 width=4)
11  Optimizer: Pivotal Optimizer (GPORCA)

```

```

1  Algebrized query:
2  +--CLogicalGbAgg( Global ) Grp Cols: ["b" (1)][Global], Minimal Grp
Cols: [], Generates Duplicates :[ 0 ]
3      |--CLogicalGet "test_b" ("test_b"), columns: ["a" (0), "b" (1), "c"
(2), "d" (3), "ctid" (4), "xmin" (5), "cmin" (6), "xmax" (7), "cmax"
(8), "tableoid" (9), "gp_segment_id" (10)] key sets: {[4,10]}
4      +--CScalarProjectList
5          +--CScalarProjectElement "count" (23)
6          +--CScalarAggFunc (count , Distinct: false , Aggregate Stage:
Global)
7              |--CScalarValuesList
8              |   +--CScalarSubquery["sum" (22)]
9              |       +--CLogicalGbAgg( Global ) Grp Cols: [][Global],
Minimal Grp Cols: [], Generates Duplicates :[ 0 ]
10             |           |--CLogicalGet "test_a" ("test_a"), columns: ["a"
(11), "b" (12), "c" (13), "d" (14), "ctid" (15), "xmin" (16), "cmin"
(17), "xmax" (18), "cmax" (19), "tableoid" (20), "gp_segment_id" (21)]
key sets: {[4,10]}
11             |               +--CScalarProjectList
12             |                   +--CScalarProjectElement "sum" (22)
13             |                       +--CScalarAggFunc (sum , Distinct: false ,
Aggregate Stage: Global)
14             |                           |--CScalarValuesList
15             |                           |   +--CScalarIdent "a" (11)
16             |                           |       |--CScalarValuesList
17             |                           |       |--CScalarValuesList
18             |                           |       +--CScalarValuesList
19             |                           |--CScalarValuesList
20             |                           |--CScalarValuesList
21             +--CScalarValuesList

```

```

1 2023-06-05 14:26:01:781043 CST,THD000,TRACE,"Xform: CXformGbAgg2Apply
2 Input:
3 +--CLogicalGbAgg( Global ) Grp Cols: ["b" (1)][Global], Minimal Grp
  Cols: [], Generates Duplicates :[ 0 ]   origin: [Grp:14, GrpExpr:0]
4   |--CLogicalGet "test_b" ("test_b"), Columns: ["a" (0), "b" (1), "c"
    (2), "d" (3), "ctid" (4), "xmin" (5), "cmin" (6), "xmax" (7), "cmax"
    (8), "tableoid" (9), "gp_segment_id" (10)] Key sets: {[4,10]}   origin:
    [Grp:0, GrpExpr:0]
5   +--CScalarProjectList   origin: [Grp:13, GrpExpr:0]
6     +--CScalarProjectElement "count" (23)   origin: [Grp:12,
    GrpExpr:0]
7       +--CScalarAggFunc (count , Distinct: false , Aggregate Stage:
    Global)   origin: [Grp:11, GrpExpr:0]
8         |--CScalarValuesList   origin: [Grp:10, GrpExpr:0]
9         | +--CScalarSubquery["sum" (22)]   origin: [Grp:9,
    GrpExpr:0]
10        | +--CLogicalGbAgg( Global ) Grp Cols: [][Global],
    Minimal Grp Cols: [], Generates Duplicates :[ 0 ]   origin: [Grp:8,
    GrpExpr:0]
11        | |--CLogicalGet "test_a" ("test_a"), Columns: ["a"
    (11), "b" (12), "c" (13), "d" (14), "ctid" (15), "xmin" (16), "cmin"
    (17), "xmax" (18), "cmax" (19), "tableoid" (20), "gp_segment_id" (21)]
    Key sets: {[4,10]}   origin: [Grp:1, GrpExpr:0]
12        | +--CScalarProjectList   origin: [Grp:7, GrpExpr:0]
13        | +--CScalarProjectElement "sum" (22)   origin:
    [Grp:6, GrpExpr:0]
14        | +--CScalarAggFunc (sum , Distinct: false ,
    Aggregate Stage: Global)   origin: [Grp:5, GrpExpr:0]
15        | |--CScalarValuesList   origin: [Grp:3,
    GrpExpr:0]
16        | | +--CScalarIdent "a" (11)   origin:
    [Grp:2, GrpExpr:0]
17        | | |--CScalarValuesList   origin: [Grp:4,
    GrpExpr:0]
18        | | |--CScalarValuesList   origin: [Grp:4,
    GrpExpr:0]
19        | | +--CScalarValuesList   origin: [Grp:4,
    GrpExpr:0]
20        | |--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
21        | |--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
22        | +--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
23 Output:
24 Alternatives:
25 0:
26 +--CLogicalGbAgg( Global ) Grp Cols: ["b" (1)][Global], Minimal Grp
  Cols: [], Generates Duplicates :[ 0 ]
27   |--CLogicalLeftOuterCorrelatedApply (Reqd Inner Cols: "sum" (22))
28   | |--CLogicalGet "test_b" ("test_b"), Columns: ["a" (0), "b" (1),
    "c" (2), "d" (3), "ctid" (4), "xmin" (5), "cmin" (6), "xmax" (7), "cmax"
    (8), "tableoid" (9), "gp_segment_id" (10)] Key sets: {[4,10]}   origin:
    [Grp:0, GrpExpr:0]

```

```

29 | |--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal Grp Cols:
    |, Generates Duplicates :[ 0 ]
30 | | |--CLogicalGet "test_a" ("test_a"), Columns: ["a" (11), "b"
    (12), "c" (13), "d" (14), "ctid" (15), "xmin" (16), "cmin" (17), "xmax"
    (18), "cmax" (19), "tableoid" (20), "gp_segment_id" (21)] Key sets:
    {[4,10]} origin: [Grp:1, GrpExpr:0]
31 | | +--CScalarProjectList origin: [Grp:7, GrpExpr:0]
32 | | +--CScalarProjectElement "sum" (22) origin: [Grp:6,
    GrpExpr:0]
33 | | +--CScalarAggFunc (sum , Distinct: false , Aggregate
    Stage: Global) origin: [Grp:5, GrpExpr:0]
34 | | |--CScalarValuesList origin: [Grp:3, GrpExpr:0]
35 | | | +--CScalarIdent "a" (11) origin: [Grp:2,
    GrpExpr:0]
36 | | |--CScalarValuesList origin: [Grp:4, GrpExpr:0]
37 | | |--CScalarValuesList origin: [Grp:4, GrpExpr:0]
38 | | +--CScalarValuesList origin: [Grp:4, GrpExpr:0]
39 | +--CScalarConst (1)
40 +--CScalarProjectList
41 +--CScalarProjectElement "count" (23)
42 +--CScalarAggFunc (count , Distinct: false , Aggregate Stage:
    Global)
43 |--CScalarValuesList
44 | +--CScalarIdent "sum" (22)
45 |--CScalarValuesList
46 |--CScalarValuesList
47 +--CScalarValuesList

```

如果不转Apply, 进行agg拆分

```

1 2023-06-05 14:26:01:781243 CST,THD000,TRACE,"Xform: CXformSplitGbAgg
2 Input:
3 +--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal Grp Cols: [],
    Generates Duplicates :[ 0 ] origin: [Grp:8, GrpExpr:0]
4 |--CLogicalGet "test_a" ("test_a"), Columns: ["a" (11), "b" (12), "c"
    (13), "d" (14), "ctid" (15), "xmin" (16), "cmin" (17), "xmax" (18),
    "cmax" (19), "tableoid" (20), "gp_segment_id" (21)] Key sets: {[4,10]}
    origin: [Grp:1, GrpExpr:0]
5 +--CScalarProjectList origin: [Grp:7, GrpExpr:0]
6 +--CScalarProjectElement "sum" (22) origin: [Grp:6, GrpExpr:0]
7 +--CScalarAggFunc (sum , Distinct: false , Aggregate Stage:
    Global) origin: [Grp:5, GrpExpr:0]
8 |--CScalarValuesList origin: [Grp:3, GrpExpr:0]
9 | +--CScalarIdent "a" (11) origin: [Grp:2, GrpExpr:0]
10 |--CScalarValuesList origin: [Grp:4, GrpExpr:0]
11 |--CScalarValuesList origin: [Grp:4, GrpExpr:0]
12 +--CScalarValuesList origin: [Grp:4, GrpExpr:0]
13 Output:
14 Alternatives:
15 0:

```

```

16 +--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal Grp Cols: [],
    Generates Duplicates :[ 1 ]
17   |--CLogicalGbAgg( Local ) Grp Cols: [][Local], Minimal Grp Cols: [],
    Generates Duplicates :[ 1 ]
18   | |--CLogicalGet "test_a" ("test_a"), Columns: ["a" (11), "b" (12),
    "c" (13), "d" (14), "ctid" (15), "xmin" (16), "cmin" (17), "xmax" (18),
    "cmax" (19), "tableoid" (20), "gp_segment_id" (21)] Key sets: {[4,10]}
    origin: [Grp:1, GrpExpr:0]
19   | +--CScalarProjectList
20   |   +--CScalarProjectElement "ColRef_0024" (24)
21   |   +--CScalarAggFunc (sum , Distinct: false , Aggregate Stage:
    Local)
22   |       |--CScalarValuesList origin: [Grp:3, GrpExpr:0]
23   |       | +--CScalarIdent "a" (11) origin: [Grp:2, GrpExpr:0]
24   |       |--CScalarValuesList origin: [Grp:4, GrpExpr:0]
25   |       |--CScalarValuesList origin: [Grp:4, GrpExpr:0]
26   |       +--CScalarValuesList origin: [Grp:4, GrpExpr:0]
27   +--CScalarProjectList
28   +--CScalarProjectElement "sum" (22)
29   +--CScalarAggFunc (sum , Distinct: false , Aggregate Stage:
    Global)
30   |--CScalarValuesList
31   | +--CScalarIdent "ColRef_0024" (24)
32   |--CScalarValuesList
33   |--CScalarValuesList
34   +--CScalarValuesList

```

最终选取的物理计划:

```

1 Physical plan:
2 +--CPhysicalMotionGather(master) rows:2 width:12 rebinds:1
    cost:1324044.434251 origin: [Grp:8, GrpExpr:7]
3   +--CPhysicalHashAgg( Global ) Grp Cols: ["b" (1)], Minimal Grp Cols:
    ["b" (1)], Generates Duplicates :[ 0 ] rows:2 width:12 rebinds:1
    cost:1324044.434192 origin: [Grp:8, GrpExpr:6]
4   |--CPhysicalMotionHashDistribute HASHED: [ CScalarIdent "b" (1),
    nulls colocated ], opfamilies: (1977,1.0), rows:1000 width:8
    rebinds:1 cost:1324044.393890 origin: [Grp:10, GrpExpr:2]
5   | +--CPhysicalCorrelatedLeftOuterNLJoin rows:1000 width:8
    rebinds:1 cost:1324044.385544 origin: [Grp:10, GrpExpr:1]
6   | |--CPhysicalTableScan "test_b" ("test_b") rows:2
    width:38 rebinds:1 cost:431.000017 origin: [Grp:0, GrpExpr:1]
7   | |--CPhysicalSpool (Streaming) rows:2 width:34 rebinds:1
    cost:431.000256 origin: [Grp:1, GrpExpr:3]
8   | | +--CPhysicalMotionBroadcast rows:2 width:34
    rebinds:1 cost:431.000248 origin: [Grp:1, GrpExpr:2]
9   | | +--CPhysicalTableScan "test_a" ("test_a") rows:2
    width:34 rebinds:1 cost:431.000025 origin: [Grp:1, GrpExpr:1]
10  | +--CScalarConst (1) origin: [Grp:9, GrpExpr:0]
11  +--CScalarProjectList origin: [Grp:15, GrpExpr:0]

```

```

12      +--CScalarProjectElement "count" (22)  origin: [Grp:14,
GrpExpr:0]
13      +--CScalarAggFunc (count , Distinct: false , Aggregate
Stage: Global)  origin: [Grp:13, GrpExpr:0]
14      |--CScalarValuesList  origin: [Grp:12, GrpExpr:0]
15      |  +--CScalarIdent "a" (11)  origin: [Grp:11, GrpExpr:0]
16      |--CScalarValuesList  origin: [Grp:4, GrpExpr:0]
17      |--CScalarValuesList  origin: [Grp:4, GrpExpr:0]
18      +--CScalarValuesList  origin: [Grp:4, GrpExpr:0]

```

Having+子连接

```

1  explain
2  SELECT b
3      FROM test_b
4      group by b
5      having count(b) > (select sum(a) from test_a);
6
7  -- 将groupby+count(b)计算结果作为关系
8  -- 之后用count(b)>sublink 进行过滤(Select)
9  -- translate query等价于
10 explain
11 select b (
12     SELECT b, count(b) as cnt_b
13     from test_b
14     group by b
15 ) where cnt_b > (select sum(a) from test_a);

```

最初的

```

1  Algebrized query:
2  +--CLogicalSelect
3      |--CLogicalGbAgg( Global ) Grp Cols: ["b" (1)][Global], Minimal Grp
Cols: [], Generates Duplicates :[ 0 ]
4      |  |--CLogicalGet "test_b" ("test_b"), Columns: ["a" (0), "b" (1),
"c" (2), "d" (3), "ctid" (4), "xmin" (5), "cmin" (6), "xmax" (7), "cmax"
(8), "tableoid" (9), "gp_segment_id" (10)] Key sets: {[4,10]}
5      |  +--CScalarProjectList
6      |      +--CScalarProjectElement "count" (11)
7      |      +--CScalarAggFunc (count , Distinct: false , Aggregate
Stage: Global)
8      |          |--CScalarValuesList
9      |          |  +--CScalarIdent "b" (1)
10     |          |--CScalarValuesList
11     |          |--CScalarValuesList
12     |          +--CScalarValuesList
13     +--CScalarCmp (>)
14         |--CScalarIdent "count" (11)
15         +--CScalarSubquery["sum" (23)]
16             +--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal Grp
Cols: [], Generates Duplicates :[ 0 ]

```

```

17      |--CLogicalGet "test_a" ("test_a"), Columns: ["a" (12), "b"
      (13), "c" (14), "d" (15), "ctid" (16), "xmin" (17), "cmin" (18), "xmax"
      (19), "cmax" (20), "tableoid" (21), "gp_segment_id" (22)] Key sets:
      {[4,10]}
18      +--CScalarProjectList
19      +--CScalarProjectElement "sum" (23)
20      +--CScalarAggFunc (sum , Distinct: false , Aggregate
      Stage: Global)
21      |--CScalarValuesList
22      | +--CScalarIdent "a" (12)
23      |--CScalarValuesList
24      |--CScalarValuesList
25      +--CScalarValuesList

```

DISTINCT+子连接

```

1  EXPLAIN
2      SELECT distinct (SELECT SUM(a) from test_a)
3      FROM test_b;

```

```

1  Algebrized query:
2  +--CLogicalGbAgg( Global ) Grp Cols: ["sum" (23)][Global], Minimal Grp
      Cols: [], Generates Duplicates :[ 0 ]
3      |--CLogicalProject
4      | |--CLogicalGet "test_b" ("test_b"), Columns: ["a" (0), "b" (1),
      "c" (2), "d" (3), "ctid" (4), "xmin" (5), "cmin" (6), "xmax" (7), "cmax"
      (8), "tableoid" (9), "gp_segment_id" (10)] Key sets: {[4,10]}
5      | +--CScalarProjectList
6      | +--CScalarProjectElement "sum" (23)
7      | +--CScalarSubquery["sum" (22)]
8      | +--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal
      Grp Cols: [], Generates Duplicates :[ 0 ]
9      | |--CLogicalGet "test_a" ("test_a"), Columns: ["a"
      (11), "b" (12), "c" (13), "d" (14), "ctid" (15), "xmin" (16), "cmin"
      (17), "xmax" (18), "cmax" (19), "tableoid" (20), "gp_segment_id" (21)]
      Key sets: {[4,10]}
10     | +--CScalarProjectList
11     | +--CScalarProjectElement "sum" (22)
12     | +--CScalarAggFunc (sum , Distinct: false ,
      Aggregate Stage: Global)
13     | |--CScalarValuesList
14     | | +--CScalarIdent "a" (11)
15     | |--CScalarValuesList
16     | |--CScalarValuesList
17     | +--CScalarValuesList
18     +--CScalarProjectList

```

```

1  2023-06-06 18:20:50:422707 CST,THD000,TRACE,"Xform: CXformSplitGbAgg
2  Input:

```

```

3  +--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal Grp Cols: [],
  Generates Duplicates :[ 0 ]   origin: [Grp:8, GrpExpr:0]
4    |--CLogicalGet "test_a" ("test_a"), Columns: ["a" (11), "b" (12), "c"
  (13), "d" (14), "ctid" (15), "xmin" (16), "cmin" (17), "xmax" (18),
  "cmax" (19), "tableoid" (20), "gp_segment_id" (21)] Key sets: {[4,10]}
  origin: [Grp:1, GrpExpr:0]
5    +--CScalarProjectList   origin: [Grp:7, GrpExpr:0]
6      +--CScalarProjectElement "sum" (22)   origin: [Grp:6, GrpExpr:0]
7        +--CScalarAggFunc (sum , Distinct: false , Aggregate Stage:
  Global)   origin: [Grp:5, GrpExpr:0]
8          |--CScalarValuesList   origin: [Grp:3, GrpExpr:0]
9            | +--CScalarIdent "a" (11)   origin: [Grp:2, GrpExpr:0]
10           |--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
11           |--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
12           +--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
13  Output:
14  Alternatives:
15  0:
16  +--CLogicalGbAgg( Global ) Grp Cols: [][Global], Minimal Grp Cols: [],
  Generates Duplicates :[ 1 ]
17    |--CLogicalGbAgg( Local ) Grp Cols: [][Local], Minimal Grp Cols: [],
  Generates Duplicates :[ 1 ]
18    | |--CLogicalGet "test_a" ("test_a"), Columns: ["a" (11), "b" (12),
  "c" (13), "d" (14), "ctid" (15), "xmin" (16), "cmin" (17), "xmax" (18),
  "cmax" (19), "tableoid" (20), "gp_segment_id" (21)] Key sets: {[4,10]}
  origin: [Grp:1, GrpExpr:0]
19    | +--CScalarProjectList
20    |   +--CScalarProjectElement "ColRef_0024" (24)
21    |   +--CScalarAggFunc (sum , Distinct: false , Aggregate Stage:
  Local)
22    |       |--CScalarValuesList   origin: [Grp:3, GrpExpr:0]
23    |       | +--CScalarIdent "a" (11)   origin: [Grp:2, GrpExpr:0]
24    |       |--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
25    |       |--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
26    |       +--CScalarValuesList   origin: [Grp:4, GrpExpr:0]
27    +--CScalarProjectList
28      +--CScalarProjectElement "sum" (22)
29      +--CScalarAggFunc (sum , Distinct: false , Aggregate Stage:
  Global)
30          |--CScalarValuesList
31          | +--CScalarIdent "ColRef_0024" (24)
32          |--CScalarValuesList
33          |--CScalarValuesList
34          +--CScalarValuesList

```



```
1 SELECT count((SELECT SUM(a) from test_a)) -- Sublink1
2 FROM test_b
3 group by b
4 having count(b) > (select sum(a) from test_a) -- Sublink2;
```

