tantor

pg_stat_advisor - optimizing PostgreSQL planner statistics with recommendations

Ilia Evdokimov, Tantor Labs LLC

Contents

- issues with statistics
- **pg_stat_advisor** overview
- future plans

Compare estimated and actual rows

For the best plan estimated rows should be similar with actual rows:

```
EXPLAIN ANALYZE
SELECT * FROM my_tbl
WHERE fld_1 = 500 AND fld_2 = 100;
>>>
     Gather  (cost=1000.00..124002.73 rows=1 width=12) (actual
time=5.778..135.219 rows=100 loops=1)"
       Workers Planned: 2"
       Workers Launched: 2"
       ->  Parallel Seq Scan on my_tbl  (cost=0.00..122975.12 rows=115
width=12) (actual time=86.647..129.191 rows=33 loops=3)"
             Filter: ((fld_1 = 500) AND (fld_2 = 100))"
             Rows Removed by Filter: 333300"
     Planning Time: 0.160 ms"
     Execution Time: 135.260 ms"
```

Prepare test table

Create table with disabled autovacuum to update statistics manually:

```
CREATE TABLE my_tbl(fld_1 INTEGER, fld_2 BIGINT)
WITH (autovacuum_enabled = false);
>>>
    CREATE TABLE
```

Evaluate data cardinality

Evaluate test data before INSERT:

```
SELECT
     COUNT(DISTINCT T.fld_1)::float/10000000,
     COUNT(DISTINCT T.fld_2)::float/10000000
FROM (
     SELECT i/100 AS fld_1, i/500 AS fld_2
     FROM generate_series(1, 10000000) s(i)
) T;
>>>
      0.01      0.002       <-- low cardinality
```

Insert values

Fill the table with diverse data:

```
INSERT INTO my_tbl (fld_1, fld_2)
SELECT
     i/100 as fld_1,
     i/500 as fld_2
FROM generate_series(1, 10000000) s(i);
>>>
     INSERT 0 1000000
```

Compare estimated and actual rows

Without ANALYZE

```
EXPLAIN ANALYZE
SELECT * FROM my_tbl
WHERE fld_1 = 500 AND fld_2 = 100;
>>>
     Gather  (cost=1000.00..124002.73 rows=276 width=12) (actual
time=5.778..135.219 rows=100 loops=1)"
       Workers Planned: 2"
       Workers Launched: 2"
       ->  Parallel Seq Scan on my_tbl  (cost=0.00..122975.12 rows=115
width=12) (actual time=86.647..129.191 rows=33 loops=3)"
             Filter: ((fld_1 = 500) AND (fld_2 = 100))"
             Rows Removed by Filter: 333300"
     Planning Time: 0.160 ms"
     Execution Time: 135.260 ms"
```

Compare estimated and actual rows

After ANALYZE

```
ANALYZE my_tbl;
>>>
     ANALYZE

EXPLAIN ANALYZE
SELECT * FROM my_tbl
WHERE fld_1 = 500 AND fld_2 = 100;
>>>
     Gather  (cost=1000.00..117554.23 rows=1 width=12) (actual
time=6.061..134.014 rows=100 loops=1)"
       Workers Planned: 2"
       Workers Launched: 2"
       ->  Parallel Seq Scan on my_tbl  (cost=0.00..116554.12 rows=1
width=12) (actual time=86.053..128.202 rows=33 loops=3)"
             Filter: ((fld_1 = 500) AND (fld_2 = 100))"
             Rows Removed by Filter: 333300"
     Planning Time: 0.128 ms"
     Execution Time: 134.057 ms"
```

default_statistics_target

```
SHOW default_statistics_target;
>>>
    default_statistics_target
   ---------------------------
    100                      <-- this is default value
   (1 row)

SELECT reltuples FROM pg_class WHERE relname = 'my_tbl';
>>>
    reltuples
   -----------
    10000000
   (1 row)
```

Compare estimated and actual rows

After default_statistics_target = 1000

```
SHOW default_statistics_target;
>>>
     1000

EXPLAIN ANALYZE
SELECT * FROM my_tbl
WHERE fld_1 = 500 AND fld_2 = 100;
>>>
     Gather  (cost=1000.00..117554.23 rows=1 width=12) (actual
time=6.061..134.014 rows=100 loops=1)"
       Workers Planned: 2"
       Workers Launched: 2"
       ->  Parallel Seq Scan on my_tbl  (cost=0.00..116554.12 rows=1
width=12) (actual time=86.053..128.202 rows=33 loops=3)"
             Filter: ((fld_1 = 500) AND (fld_2 = 100))"
             Rows Removed by Filter: 333300"
     Planning Time: 0.128 ms"
     Execution Time: 134.057 ms"
```

Extended statistics

dependencies	Evaluates functional dependencies between columns
ndistinct	Counts unique combinations of values in columns
mcv	Analyzes the most frequent value combinations in columns

```
CREATE STATISTICS stat_tbl (dependencies)
ON fld_1, fld_2
FROM my_tbl;
>>>
      CREATE STATISTICS

ANALYZE my_tbl;
>>>
      ANALYZE
```

Results

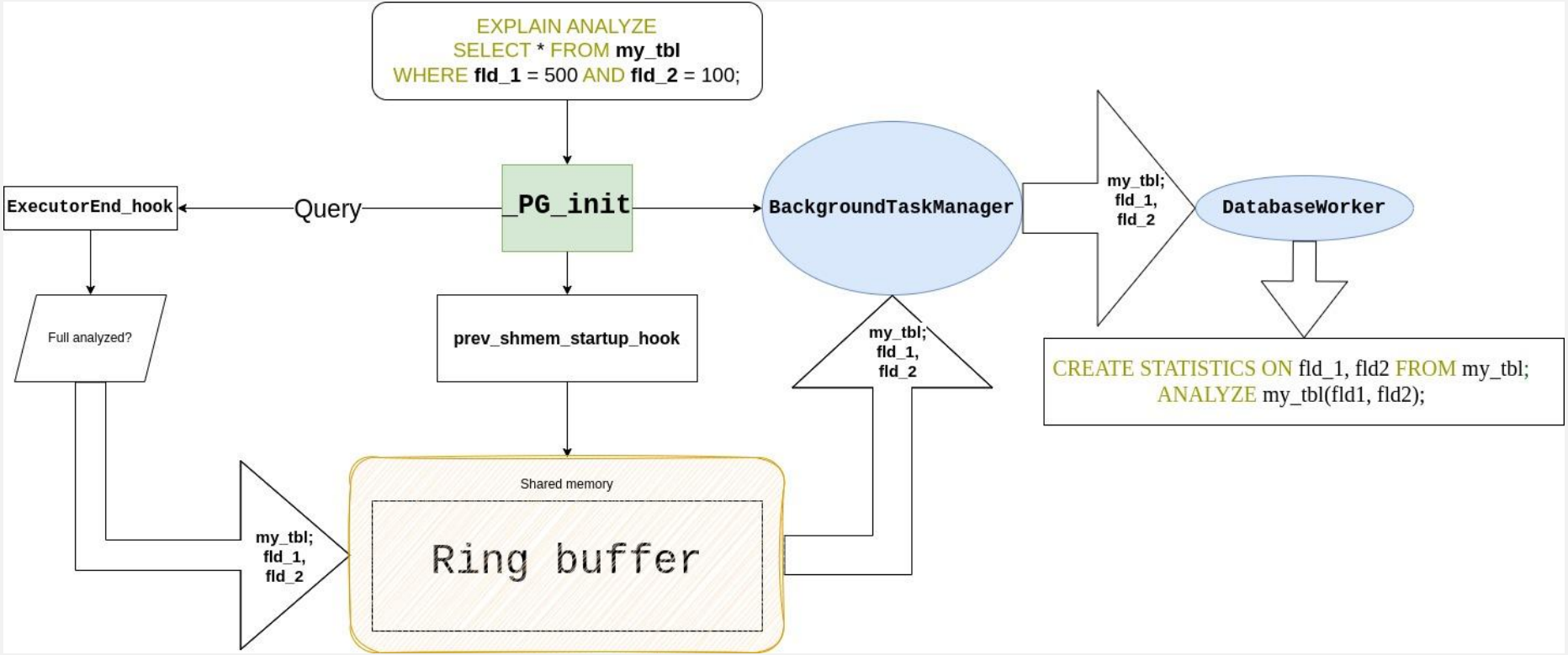
```
EXPLAIN ANALYZE
SELECT * FROM my_tbl
WHERE fld_1 = 500 AND fld_2 = 100;
>>>
    Gather  (cost=1000.00..24324.71 rows=100 width=12) (actual
        time=102.944..114.427 rows=100 loops=1)"
      Workers Planned: 2"
      Workers Launched: 2"
      ->  Parallel Seq Scan on my_tbl  (cost=0.00..23311.01 rows=1 width=12)
    (actual time=68.527..101.757 rows=33 loops=3)"
            Filter: ((fld_1 = 500) AND (fld_2 = 100))"
            Rows Removed by Filter: 666634"
    Planning Time: 0.109 ms"
    Execution Time: 116.551 ms"
```

About pg_stat_advisor

What is the Purpose of pg_stat_advisor?

- Suggests performing ANALYZE operations on non-analyzed tables.
- Creates extended statictics and analyzes table based on specific criteria.

pg_stat_advisor internals



How use pg_stat_advisor

Add library to postgresql.conf for current session:

```
shared_preload_libraries = 'pg_stat_advisor'
```

Set extension parameters in postgresql.conf or current session:

- pg_stat_advisor.ring_buffer_capacity
- pg_stat_advisor.suggest_statistics_threshold

Configure ANALYZE

```
EXPLAIN ANALYZE SELECT * FROM my_tbl WHERE fld_1 = 500 AND fld_2 = 100;
>>>
    NOTICE:  pg_stat_advisor suggestion: 'ANALYZE my_tbl'
    Gather  (cost=1000.00..124002.73 rows=276 width=12) (actual
               time=5.778..135.219 rows=100 loops=1)"
          Workers Planned: 2"
          Workers Launched: 2"
          ->  Parallel Seq Scan on my_tbl  (cost=0.00..122975.12 rows=115
    width=12) (actual time=86.647..129.191 rows=33 loops=3)"
                Filter: ((fld_1 = 500) AND (fld_2 = 100))"
                Rows Removed by Filter: 333300"
        Planning Time: 0.160 ms"
        Execution Time: 135.260 ms"
```

Configure suggest_statistics_threshold

```
-- estimated_row / actual rows < pg_stat_advisor.suggest_statistics_threshold
SET pg_stat_advisor.suggest_statistics_threshold = 0.01;
>>>
     SET

EXPLAIN ANALYZE SELECT * FROM my_tbl WHERE fld_1 = 500 AND fld_2 = 100;
>>>
     Gather  (cost=1000.00 .24311.11 rows=1 width=12) (actual
     time=104.973..113.246 rows=100 loops=1)"
       Workers Planned: 2"
       Workers Launched: 2"
       ->  Parallel Seq Scan on my_tbl  (cost=0.00..23311.01 rows=1 width=12)
     (actual time=68.527..101.757 rows=33 loops=3)"
             Filter: ((fld_1 = 500) AND (fld_2 = 100))"
             Rows Removed by Filter: 666634"
     Planning Time: 0.109 ms"
     Execution Time: 116.551 ms"
```

Results of applying statistics

```
EXPLAIN ANALYZE
SELECT * FROM my_tbl
WHERE fld_1 = 500 AND fld_2 = 100;
>>>
    Gather  (cost=1000.00..24324.71 rows=100 width=12) (actual
        time=102.944..114.427 rows=100 loops=1)"
      Workers Planned: 2"
      Workers Launched: 2"
      ->  Parallel Seq Scan on my_tbl  (cost=0.00..23311.01 rows=1 width=12)
    (actual time=68.527..101.757 rows=33 loops=3)"
            Filter: ((fld_1 = 500) AND (fld_2 = 100))"
            Rows Removed by Filter: 666634"
    Planning Time: 0.109 ms"
    Execution Time: 116.551 ms"

SELECT stxname, stxkeys, stxkind FROM pg_statistic_ext;
>>>          stxname         |stxkeys | stxkind
     ------------------------+--------+-----------
     my_tbl_fld_1_fld_2_stat | 1 2    | {d, f, m}
```

pg_stat_advisor does NOT process

- Part-analyzed, TEMP tables;
- UPDATE, DELETE queries;
- Nested Loop, Merge Join, Hash Join nodes
- Columns with ndistinct = 1

TODO

- Kind of statistics
- Remove useless extended statistics
- Adjust default_statistics_target

Thank You for Your Attention

ilia-evdokimov

EvdokimovIlia

ilidock95