A Close Look At Stats: What You Can Do With Them !

pgConf.eu Tallinn 2016

Cédric Villemain cedric@2ndQuadrant.fr

Nov. 4, 2016

Cédric Villemain

2ndQuadrant®
PostgreSQL

PostgreSQL Expertise & Development
Training
24x7 Support & RDBA

PostgreSQL Platinum Sponsor

Live stats

- activity
- vacuum progress
- shared buffers usage
- operating system cache usage

Live stats

- activity
- vacuum progress
- shared buffers usage
- operating system cache usage

Live stats

- activity
- vacuum progress
- shared buffers usage
- operating system cache usage

Live stats

- activity
- vacuum progress
- shared buffers usage
- operating system cache usage

Slow query, the waiter & the locker

```
SELECT a.pid, a.datname, a.application_name
     , now() - a.query_start as duration
     , a.query, a.wait_event_type, a.wait_event
     , blocking.pid as blocked_by_pid
     , blocking.query as blocked_by_query
FROM pg_stat_activity a
   , LATERAL (select pid, query from pg_stat_activity
              where pid = ANY (pg_blocking_pids(a.pid)))
              as blocking
WHERE now() - query_start > '3 s'
AND state = 'active'
; \watch 1
```

Slow query, the waiter & the locker

```
-[ RECORD 1 ]----+----------------------------------------
pid              | 4713
datname          | cedric
application_name | psql
duration         | 00:01:56.166649
query            | lock TABLE demo in ACCESS EXCLUSIVE MODE;
wait_event_type  | Lock
wait_event       | relation
blocked_by_pid   | 414
blocked_by_query | lock table demo;
-[ RECORD 2 ]----+----------------------------------------
pid              | 5047
datname          | cedric
application_name | psql
duration         | 00:01:44.422787
query            | select * from demo;
wait_event_type  | Lock
wait_event       | relation
blocked_by_pid   | 414
blocked_by_query | lock table demo;
-[ RECORD 3 ]----+----------------------------------------
pid              | 5047
datname          | cedric
application_name | psql
duration         | 00:01:44.422787
query            | select * from demo;
wait_event_type  | Lock
wait_event       | relation
blocked_by_pid   | 4713
blocked_by_query | lock TABLE demo in ACCESS EXCLUSIVE MODE;
```

VACUUM Progress Reporting

(auto)-VACUUM is still running since 5 minutes!
Time to have a look at: pg_stat_progress_vacuum

VACUUM Progress Reporting

```
with guc as (
  SELECT current_setting('vacuum_cost_delay')::interval   as v_c_d
       , current_setting('vacuum_cost_limit')::float   as v_c_l
       , current_setting('vacuum_cost_page_hit')::float   as v_c_p_h
       , current_setting('vacuum_cost_page_dirty')::float as v_c_p_d
)
SELECT v.pid, v.datname, v.relid::regclass
     , now() - a.query_start as duration
     , wait_event_type, wait_event
     , CASE
            WHEN v.phase = 'initializing' AND (now() - a.query_start) > '3 s'
            THEN 'Unexpected long initialization phase!!!'
            WHEN v.phase = 'scanning heap'
            THEN 'Still ' || heap_blks_total - heap_blks_scanned
              || ' blocks to heap scan (min/max: ' ||
v_c_d * ( (heap_blks_total-heap_blks_scanned) / (v_c_l / v_c_p_h) )
    || ' / ' ||
v_c_d * ( (heap_blks_total-heap_blks_scanned) / (v_c_l/v_c_p_d) )
                || ')'
            ELSE phase
       END
FROM guc, pg_stat_progress_vacuum v
JOIN pg_stat_activity a using (pid)
; \watch 1
```

VACUUM Progress Reporting

```
-[ RECORD 1 ]------+-------------------------------------------------------
pid                | 10117
datname            | cedric
relid              | pgbench_accounts
duration           | 00:00:03.246483
wait_event_type    |
wait_event         |
phase              | Still 176541 blocks to heap scan
                  (min/max: 00:01:28.2705 / 00:29:25.41)
pid                | 10117
datid              | 16385
datname            | cedric
relid              | 24809
phase              | scanning heap
heap_blks_total    | 177372
heap_blks_scanned  | 831
heap_blks_vacuumed | 0
index_vacuum_count | 0
max_dead_tuples    | 11184810
num_dead_tuples    | 57
```

shared buffers usage

Is there enought cache ? Well used ?

```
SELECT trunc(
            100 * sum(blks_hit) / sum(blks_hit + blks_read)
          , 2) as hit_miss_ratio
FROM pg_stat_database;
 hit_miss_ratio
----------------
          95.16
```

shared buffers usage

We can have much more details.

```
CREATE EXTENSION pg_buffercache;
select usagecount
     , isdirty
     , count(*)
     , trunc(100 * count(*) / sum(count(*)) over (),2) as "%"
from pg_buffercache group by 1, 2 order by 1, 2;
 usagecount | isdirty | count |   %
------------+---------+-------+-------
          0 | f       |  2649 | 16.16
          1 | f       |  2726 | 16.63
          2 | f       |  3513 | 21.44
          3 | f       |   161 |  0.98
          4 | f       |  1261 |  7.69
          5 | f       |   169 |  1.03
            |         |  5905 | 36.04
```

operating system cache usage

Is it well used ?

```
select datname
     , blk_read_time, blks_read, blks_hit
     , blk_read_time / blks_read as avg_read_duration
from pg_stat_database where datname = current_database()
 and blks_read+blks_hit > 0;
-[ RECORD 1 ]-----+-------------------
datname           | cedric
blk_read_time     | 11662.568
blks_read         | 3128145
blks_hit          | 61294388
avg_read_duration | 0.00372826962944493
```

operating system cache usage

And more details!

```
CREATE EXTENSION pgfincore;
SELECT relpath, segment
     , pg_size_pretty(rel_os_pages * os_page_size)
     , case when rel_os_pages > 0
            THEN trunc(pages_mem * 100 / rel_os_pages, 2)
            else 0 end as "% in memory"
FROM pgfincore( 'pgbench_accounts');
      relpath        | segment | pg_size_pretty | % in memory
---------------------+---------+----------------+-------------
 base/16385/24815    |       0 | 1024 MB        |      100.00
 base/16385/24815.1  |       1 | 258 MB         |      100.00
 base/16385/24815.2  |       2 | 0 bytes        |            0
```

Accumulated stats: don't miss them!

- background writer
- checkpointer
- backends

Accumulated stats: don't miss them!

- background writer
- checkpointer
- backends

Accumulated stats: don't miss them!

- background writer
- checkpointer
- backends

background writer

```
select now()-stats_reset as since
     , buffers_clean
     , maxwritten_clean
from pg_stat_bgwriter ;

-[ RECORD 1 ]----+------------------------
since            | 6 days 13:17:34.694006
buffers_clean    | 159196
maxwritten_clean | 1372

show bgwriter_lru_maxpages;
 bgwriter_lru_maxpages
-----------------------
 100
```

checkpointer

```
select now()-stats_reset as since
     , checkpoints_timed, checkpoints_req
     , checkpoint_write_time, checkpoint_sync_time
     , buffers_checkpoint
     , round(checkpoint_write_time/buffers_checkpoint::float) as avg_block_write
     , round(checkpoint_sync_time/buffers_checkpoint::float) as avg_block_sync
from pg_stat_bgwriter ;

-[ RECORD 1 ]---------+-----------------------
since                 | 6 days 13:34:14.177471
checkpoints_timed     | 727
checkpoints_req       | 11
checkpoint_write_time | 989382
checkpoint_sync_time  | 276964
buffers_checkpoint    | 150462
avg_block_write       | 7
avg_block_sync        | 2
```

backends

```
select now()-stats_reset as since
     , buffers_backend, buffers_backend_fsync
     , round( buffers_backend
           / (EXTRACT(EPOCH FROM (now()-stats_reset))))
         || ' blks/sec' as avg_write
from pg_stat_bgwriter ;

-[ RECORD 1 ]---------+-----------------------
since                 | 6 days 13:53:02.941036
buffers_backend       | 1935640
buffers_backend_fsync | 0
avg_write             | 3 blks/sec
```

Accumulated stats again: don't miss them!

- pg_stat(_xact)_user_tables
- pg_stat(_xact)_user_indexes
- pg_stat(_xact)_user_functions
- pg_statio_user_sequences
- pg_statio_user_tables
- pg_statio_user_indexes

Accumulated stats again: don't miss them!

- pg_stat(_xact)_user_tables
- pg_stat(_xact)_user_indexes
- pg_stat(_xact)_user_functions
- pg_statio_user_sequences
- pg_statio_user_tables
- pg_statio_user_indexes

pg_stat_...

```
            View "pg_catalog.pg_stat_user_tables"
      Column        |           Type           | Modifiers
--------------------+--------------------------+-----------
 relid              | oid                      |
 schemaname         | name                     |
 relname            | name                     |
 seq_scan           | bigint                   |
 seq_tup_read       | bigint                   |
 idx_scan           | bigint                   |
 idx_tup_fetch      | bigint                   |
 n_tup_ins          | bigint                   |
 n_tup_upd          | bigint                   |
 n_tup_del          | bigint                   |
 n_tup_hot_upd      | bigint                   |
 n_live_tup         | bigint                   |
 n_dead_tup         | bigint                   |
 n_mod_since_analyze | bigint                   |
 last_vacuum        | timestamp with time zone |
 last_autovacuum    | timestamp with time zone |
 last_analyze       | timestamp with time zone |
 last_autoanalyze   | timestamp with time zone |
 vacuum_count       | bigint                   |
 autovacuum_count   | bigint                   |
 analyze_count      | bigint                   |
 autoanalyze_count  | bigint                   |
```

pg_statio_...

```
View "pg_catalog.pg_statio_user_tables"
     Column       |  Type   | Modifiers
------------------+---------+-----------
 relid            | oid     |
 schemaname       | name    |
 relname          | name    |
 heap_blks_read   | bigint  |
 heap_blks_hit    | bigint  |
 idx_blks_read    | bigint  |
 idx_blks_hit     | bigint  |
 toast_blks_read  | bigint  |
 toast_blks_hit   | bigint  |
 tidx_blks_read   | bigint  |
 tidx_blks_hit    | bigint  |
```

Content stats: at least !!!

- pg_stat(istic)s
- pg_class

Let start with the easy part

```
select relname
     , relpages, relallvisible, reltuples
     , round(reltuples / relpages) as tup_density
     , age(relfrozenxid), reloptions
from pg_class where relname = 'pgbench_accounts';
-[ RECORD 1 ]-+---------
relname       | pgbench_accounts
relpages      | 164117
relallvisible | 164117
reltuples     | 1.14225e+07
tup_density   | 70
age           | 11568
reloptions    | fillfactor=100
```

Let start with the easy part

```
select count(*) from pgbench_accounts ;
  count
----------
 10000000
```

Mmmhh, so we already have around 10% error in the number of tuples
(Yes...I did some aborted transactions to confuse PostgreSQL...)

And now... we have real stats!

```
select attname
     , null_frac, avg_width
     , n_distinct, correlation
from pg_stats where tablename = 'pgbench_accounts'
order by 1 ;
```

attname	null_frac	avg_width	n_distinct	correlation
abalance	0	4	23	0.997801
aid	0	4	-1	0.998481
bid	0	4	100	0.998508
filler	0	85	1	1

Filler attribute

```
select attname
     , most_common_vals, most_common_freqs
     , histogram_bounds
     , most_common_elems, most_common_elem_freqs
     , elem_count_histogram
from pg_stats where tablename = 'pgbench_accounts'
and attname = 'filler';

-[ RECORD 1 ]----------+-----------
attname                | filler
most_common_vals       | {"    [...]     "}
most_common_freqs      | {1}
histogram_bounds       |
most_common_elems      |
most_common_elem_freqs |
elem_count_histogram   |
```

Aid attribute

```
select attname
     , most_common_vals, most_common_freqs
     , histogram_bounds
     , most_common_elems, most_common_elem_freqs
     , elem_count_histogram
from pg_stats where tablename = 'pgbench_accounts'
and attname = 'aid';

-[ RECORD 1 ]----------+-------------------------------
attname                | aid
most_common_vals       |
most_common_freqs      |
histogram_bounds       | {1293,101501,205586,309517,410106,505
most_common_elems      |
most_common_elem_freqs |
elem_count_histogram   |
```

Bid attribute

```
select attname
     , most_common_vals, most_common_freqs
     , histogram_bounds
     , most_common_elems, most_common_elem_freqs
     , elem_count_histogram
from pg_stats where tablename = 'pgbench_accounts'
and attname ='bid';

-[ RECORD 1 ]----------+-------------------------------
attname                | bid
most_common_vals       | {45,75,34,17,83,36,31,35,68,22,88,70,2
most_common_freqs      | {0.0119667,0.0116667,0.0115333,0.0115,
histogram_bounds       |
most_common_elems      |
most_common_elem_freqs |
elem_count_histogram   |
```

Abalance

```
select attname
     , most_common_vals, most_common_freqs
     , histogram_bounds
     , most_common_elems, most_common_elem_freqs
     , elem_count_histogram
from pg_stats where tablename = 'pgbench_accounts'
and attname ='abalance';
-[ RECORD 1 ]----------+-----------------------------------
attname                | abalance
most_common_vals       | {0}
most_common_freqs      | {0.9986}
histogram_bounds       | {-4830,-3522,-3066,-3055,-3041,-2974,-
most_common_elems      |
most_common_elem_freqs |
elem_count_histogram   |
```

Do the planner job

```
select reltuples
     , round(reltuples * (1-0.9986)) as estimate
from pg_class
where relname = 'pgbench_accounts';
  reltuples  |  estimate
-------------+------------
 1.05184e+07 | 14726
```

Check

```
explain select *
from pgbench_accounts
where abalance != 0;
         QUERY PLAN
-----------------------------------------------
 Seq Scan on pgbench_accounts
     (cost=0.00..295597.09 rows=14726 width=97)
   Filter: (abalance <> 0)
```

Planner job again

```
select reltuples
     , round(reltuples * (1-0.9986)/41) as estimate
 from pg_class where relname = 'pgbench_accounts';

  reltuples   | estimate
--------------+----------
 1.05184e+07 |      359
```

Check again

```
explain select * from pgbench_accounts
 where abalance between -4830 and -3522;
         QUERY PLAN
-----------------------------------------------
 Seq Scan on pgbench_accounts
     (cost=0.00..321893.10 rows=358 width=97)
   Filter: ((abalance >= '-4830'::integer)
        AND (abalance <= '-3522'::integer))
```

Check further

```
explain select * from pgbench_accounts
 where abalance between -1004831 and -3522;
         QUERY PLAN
-------------------------------------------------
 Seq Scan on pgbench_accounts
     (cost=0.00..321893.10 rows=358 width=97)
   Filter: ((abalance >= '-1004831'::integer)
        AND (abalance <= '-3522'::integer))
```

And how to plan with more conditions

```
explain select * from pgbench_accounts
 where abalance!=0 and bid = 64;
         QUERY PLAN
-----------------------------------------------
 Seq Scan on pgbench_accounts
     (cost=0.00..321893.10 rows=141 width=97)
   Filter: ((abalance <> 0) AND (bid = 64))
```

Estimate bid

```
select reltuples
     , round(reltuples * 0.0096 ) as estimate
from pg_class where relname = 'pgbench_accounts';
  reltuples  | estimate
-------------+----------
 1.05184e+07 |   100977
```

Estimate bid AND abalance

```
select reltuples
     , round(reltuples * (0.0096 * (1-0.9986))) as estimate
from pg_class where relname = 'pgbench_accounts';

  reltuples   | estimate

 1.05184e+07 |      141
```

And Check

```
select count(*)
from pgbench_accounts where abalance!=0 and bid = 64;
 count
-------
    96
```

Help my stats !

```
CREATE TABLE ts AS
  SELECT d::TIMESTAMP
  FROM GENERATE_SERIES('2016-01-01','2016-12-31'
                       , '10 mins'::INTERVAL) d(d);

  ANALYZE ts;
```

What about this simple query

```
EXPLAIN SELECT * FROM ts
WHERE DATE_TRUNC('month', d) = '2016-11-01';
         QUERY PLAN
-----------------------------------------------
 Seq Scan on ts
    (cost=0.00..1021.41 rows=263 width=8)
   Filter: (date_trunc('month'::text, d)
        = '2016-11-01 00:00:00'::timestamp without time zone)
```

Check

```
select count(*) from ts
WHERE DATE_TRUNC('month', d) = '2016-11-01';
 count
-------
  4320
```

Help the planner

```
CREATE INDEX ON ts USING brin ((DATE_TRUNC('month',d)));
analyze ts;
set enable_bitmapscan to off;
```

Really ?

```
EXPLAIN SELECT * FROM ts
 WHERE DATE_TRUNC('month', d) = '2016-11-01';
         QUERY PLAN
-------------------------------------------------
 Seq Scan on ts
     (cost=0.00..1021.41 rows=4324 width=8)
   Filter: (date_trunc('month'::text, d)
        = '2016-11-01 00:00:00'::timestamp without time zone)
```

Why did it work ?

```
 select tablename, attname
 from pg_stats where tablename like 'ts%';;
     tablename      |  attname
--------------------+------------
 ts                 | d
 ts_date_trunc_idx | date_trunc
```

Ho ho, so there is something going on here

```
select attname
     , most_common_vals, most_common_freqs
     , histogram_bounds
     , most_common_elems, most_common_elem_freqs
     , elem_count_histogram
from pg_stats where tablename = 'ts_date_trunc_idx'
and attname ='date_trunc';
-[ RECORD 1 ]----------+-------------------------------
attname                | date_trunc
most_common_vals       | {"2016-08-01 00:00:00","2016-07-01 00
most_common_freqs      | {0.0861667,0.0859,0.0849667,0.0845333
histogram_bounds       |
most_common_elems      |
most_common_elem_freqs |
elem_count_histogram   |
```

Not everything covered...

you also have pg_stat_statements, and extensions giving you even more stats about your query plans.

Any questions ?

Please ask !