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

pgConf.eu Tallinn 2016

Cédric Villemain cedric@2ndQuadrant.fr

Nov. 4, 2016
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
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;
(auto)-VACUUM is still running since 5 minutes!
Time to have a look at: pg_stat_progress_vacuum
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

<table>
<thead>
<tr>
<th>pid</th>
<th>10117</th>
</tr>
</thead>
<tbody>
<tr>
<td>datname</td>
<td>cedric</td>
</tr>
<tr>
<td>relid</td>
<td>pgbench_accounts</td>
</tr>
<tr>
<td>duration</td>
<td>00:00:03.246483</td>
</tr>
<tr>
<td>wait_event_type</td>
<td></td>
</tr>
<tr>
<td>wait_event</td>
<td></td>
</tr>
<tr>
<td>phase</td>
<td>Still 176541 blocks to heap scan</td>
</tr>
</tbody>
</table>

(min/max: 00:01:28.2705 / 00:29:25.41)

<table>
<thead>
<tr>
<th>pid</th>
<th>10117</th>
</tr>
</thead>
<tbody>
<tr>
<td>datid</td>
<td>16385</td>
</tr>
<tr>
<td>datname</td>
<td>cedric</td>
</tr>
<tr>
<td>relid</td>
<td>24809</td>
</tr>
<tr>
<td>phase</td>
<td>scanning heap</td>
</tr>
<tr>
<td>heap_blks_total</td>
<td>177372</td>
</tr>
<tr>
<td>heap_blks_scanned</td>
<td>831</td>
</tr>
<tr>
<td>heap_blks_vacuumed</td>
<td>0</td>
</tr>
<tr>
<td>index_vacuum_count</td>
<td>0</td>
</tr>
<tr>
<td>max_dead_tuples</td>
<td>11184810</td>
</tr>
<tr>
<td>num_dead_tuples</td>
<td>57</td>
</tr>
</tbody>
</table>
Is there enough 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;

<table>
<thead>
<tr>
<th>usagecount</th>
<th>isdirty</th>
<th>count</th>
<th>%</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>f</td>
<td>2649</td>
<td>16.16</td>
</tr>
<tr>
<td>1</td>
<td>f</td>
<td>2726</td>
<td>16.63</td>
</tr>
<tr>
<td>2</td>
<td>f</td>
<td>3513</td>
<td>21.44</td>
</tr>
<tr>
<td>3</td>
<td>f</td>
<td>161</td>
<td>0.98</td>
</tr>
<tr>
<td>4</td>
<td>f</td>
<td>1261</td>
<td>7.69</td>
</tr>
<tr>
<td>5</td>
<td>f</td>
<td>169</td>
<td>1.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5905</td>
<td>36.04</td>
</tr>
</tbody>
</table>
operating system cache usage

Is it well used?

```sql
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

Cédric Villemain  cedric@2ndQuadrant.fr  A Close Look At Stats: What You Can Do  Nov. 4, 2016  12 / 46
operating system cache usage

And more details!

```sql
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');
```

<table>
<thead>
<tr>
<th>relpath</th>
<th>segment</th>
<th>pg_size_pretty</th>
<th>% in memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>base/16385/24815</td>
<td>0</td>
<td>1024 MB</td>
<td>100.00</td>
</tr>
<tr>
<td>base/16385/24815.1</td>
<td>1</td>
<td>258 MB</td>
<td>100.00</td>
</tr>
<tr>
<td>base/16385/24815.2</td>
<td>2</td>
<td>0 bytes</td>
<td>0</td>
</tr>
</tbody>
</table>
```
Accumulated stats: don’t miss them!

- background writer
- checkpoint
- backends
Accumulated stats: don’t miss them!

- background writer
- checkpoint
- backends
Accumulated stats: don’t miss them!

- background writer
- checkpoint
- backends
background writer

```sql
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
```

```sql
show bgwriter_lru_maxpages;
  bgwriter_lru_maxpages
------------------------
  100
```
checkpointer

```sql
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 ]------------------------------------------

<table>
<thead>
<tr>
<th>since</th>
<th>6 days 13:34:14.177471</th>
</tr>
</thead>
<tbody>
<tr>
<td>checkpoints_timed</td>
<td>727</td>
</tr>
<tr>
<td>checkpoints_req</td>
<td>11</td>
</tr>
<tr>
<td>checkpoint_write_time</td>
<td>989382</td>
</tr>
<tr>
<td>checkpoint_sync_time</td>
<td>276964</td>
</tr>
<tr>
<td>buffers_checkpoint</td>
<td>150462</td>
</tr>
<tr>
<td>avg_block_write</td>
<td>7</td>
</tr>
<tr>
<td>avg_block_sync</td>
<td>2</td>
</tr>
</tbody>
</table>
## backends

```sql
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_user_tables

**View "pg_catalog.pg_stat_user_tables"**

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>relid</td>
<td>oid</td>
<td></td>
</tr>
<tr>
<td>schemaname</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>relname</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>seq_scan</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>seq_tup_read</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>idx_scan</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>idx_tup_fetch</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_tup_ins</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_tup_upd</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_tup_del</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_tup_hot_upd</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_live_tup</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_dead_tup</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>n_mod_since_analyze</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>last_vacuum</td>
<td>timestamp with time zone</td>
<td></td>
</tr>
<tr>
<td>last_autovacuum</td>
<td>timestamp with time zone</td>
<td></td>
</tr>
<tr>
<td>last_analyze</td>
<td>timestamp with time zone</td>
<td></td>
</tr>
<tr>
<td>last_autoanalyze</td>
<td>timestamp with time zone</td>
<td></td>
</tr>
<tr>
<td>vacuum_count</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>autovacuum_count</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>analyze_count</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>autoanalyze_count</td>
<td>bigint</td>
<td></td>
</tr>
</tbody>
</table>
View "pg_catalog.pg_statio_user_tables"

<table>
<thead>
<tr>
<th>Column</th>
<th>Type</th>
<th>Modifiers</th>
</tr>
</thead>
<tbody>
<tr>
<td>relid</td>
<td>oid</td>
<td></td>
</tr>
<tr>
<td>schemaname</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>relname</td>
<td>name</td>
<td></td>
</tr>
<tr>
<td>heap_blks_read</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>heap_blks_hit</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>idx_blks_read</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>idx_blks_hit</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>toast_blks_read</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>toast_blks_hit</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>tidx_blks_read</td>
<td>bigint</td>
<td></td>
</tr>
<tr>
<td>tidx_blks_hit</td>
<td>bigint</td>
<td></td>
</tr>
</tbody>
</table>
Content stats: at least !!!

- `pg_stat(istic)s`
- `pg_class`
Let start with the easy part

```sql
select relname,
    relpages, relallvisible, reltuples,
    round(reltuples / relpages) as tup_density,
    age(refrozenxid), reloptions
from pg_class
where relname = 'pgbench_accounts';
```

<table>
<thead>
<tr>
<th>relname</th>
<th>pgbench_accounts</th>
</tr>
</thead>
<tbody>
<tr>
<td>relpages</td>
<td>164117</td>
</tr>
<tr>
<td>relallvisible</td>
<td>164117</td>
</tr>
<tr>
<td>reltuples</td>
<td>1.14225e+07</td>
</tr>
<tr>
<td>tup_density</td>
<td>70</td>
</tr>
<tr>
<td>age</td>
<td>11568</td>
</tr>
<tr>
<td>reloptions</td>
<td>fillfactor=100</td>
</tr>
</tbody>
</table>
Let start with the easy part

```sql
select count(*) from pgbench_accounts;

<table>
<thead>
<tr>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>10000000</td>
</tr>
</tbody>
</table>
```

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

Cédric Villemain cedric@2ndQuadrant.fr A Close Look At Stats: What You Can Do Nov. 4, 2016 23 / 46
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 ;
```

<table>
<thead>
<tr>
<th>attname</th>
<th>null_frac</th>
<th>avg_width</th>
<th>n_distinct</th>
<th>correlation</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalance</td>
<td>0</td>
<td>4</td>
<td>23</td>
<td>0.997801</td>
</tr>
<tr>
<td>aid</td>
<td>0</td>
<td>4</td>
<td>-1</td>
<td>0.998481</td>
</tr>
<tr>
<td>bid</td>
<td>0</td>
<td>4</td>
<td>100</td>
<td>0.998508</td>
</tr>
<tr>
<td>filler</td>
<td>0</td>
<td>85</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
### Filler attribute

```sql
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';
```

<table>
<thead>
<tr>
<th>attname</th>
<th>filler</th>
</tr>
</thead>
<tbody>
<tr>
<td>most_common_vals</td>
<td>{&quot; [...]  &quot;}</td>
</tr>
<tr>
<td>most_common_freqs</td>
<td>{1}</td>
</tr>
<tr>
<td>histogram_bounds</td>
<td></td>
</tr>
<tr>
<td>most_common_elems</td>
<td></td>
</tr>
<tr>
<td>most_common_elem_freqs</td>
<td></td>
</tr>
<tr>
<td>elem_count_histogram</td>
<td></td>
</tr>
</tbody>
</table>
### Aid attribute

```sql
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';
```

<table>
<thead>
<tr>
<th>attname</th>
<th>aid</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>{1293,101501,205586,309517,410106,505560}</td>
</tr>
</tbody>
</table>
Bid attribute

```sql
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';
```

<table>
<thead>
<tr>
<th>attname</th>
<th>bid</th>
</tr>
</thead>
<tbody>
<tr>
<td>most_common_vals</td>
<td>{45,75,34,17,83,36,31,35,68,22,88,70,2}</td>
</tr>
<tr>
<td>most_common_freqs</td>
<td>{0.0119667,0.0116667,0.0115333,0.0115,0}</td>
</tr>
<tr>
<td>histogram_bounds</td>
<td></td>
</tr>
</tbody>
</table>
### Abalance

```sql
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';
```

<table>
<thead>
<tr>
<th>attname</th>
<th>abalance</th>
</tr>
</thead>
<tbody>
<tr>
<td>most_common_vals</td>
<td>{0}</td>
</tr>
<tr>
<td>most_common_freqs</td>
<td>{0.9986}</td>
</tr>
<tr>
<td>histogram_bounds</td>
<td>{-4830,-3522,-3066,-3055,-3041,-2974,-2663,-2337,-2315,-2039,-1542,-1267,-1145,-801,-493,-416,-281,-154,-151,389,439,493,630,686,1442,1925,2179,2239,2406,2849,2857,3279,3284,3476,3491,3957,4104,4261,4364,4403,4453,4673}</td>
</tr>
</tbody>
</table>
select reltuples
    , round(reltuples * (1-0.9986)) as estimate
from pg_class
where relname = 'pgbench_accounts';

reltuples | estimate
-----------+------------
1.05184e+07 | 14726
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)
select reltuples,
  round(reltuples * (1-0.9986)/41) as estimate
from pg_class where relname = 'pgbench_accounts';

<table>
<thead>
<tr>
<th>reltuples</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.05184e+07</td>
<td>359</td>
</tr>
</tbody>
</table>
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))
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))
```
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

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

<table>
<thead>
<tr>
<th>reltuples</th>
<th>estimate</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.05184e+07</td>
<td>141</td>
</tr>
</tbody>
</table>
And Check

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

count
-------

96
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
```
CREATE INDEX ON ts USING brin ((DATE_TRUNC('month',d)));
analyze ts;
set enable_bitmapscan to off;
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?

```sql
select tablename, attname
from pg_stats where tablename like 'ts%';
```

<table>
<thead>
<tr>
<th>tablename</th>
<th>attname</th>
</tr>
</thead>
<tbody>
<tr>
<td>ts</td>
<td>d</td>
</tr>
<tr>
<td>ts_date_trunc_idx</td>
<td>date_trunc</td>
</tr>
</tbody>
</table>
Ho ho, so there is something going on here

```sql
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 ]-----------------------------------------------
attnname | date_trunc
most_common_vals | {"2016-08-01 00:00:00","2016-07-01 00:00:00","2016-03-01 00:00:00","2016-01-01 00:00:00","2016-05-01 ...
most_common_freqs | {0.0861667,0.0859,0.0849667,0.0845333,0.0843667,0.0829667,0.0825667,0.0822667,0.0820333,0.082,0.0819333,0.0803}
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!