

PostgreSQL Performance Troubleshooting

By: Karl Arao

(best viewed in MS Word -> View -> Web Layout)

Table of Contents

STEP 1) GET DBSTATE OUTPUT	1
STEP 2) GET EXPLAIN PLAN OF THE SLOW SQL.....	3
\TIMING	7
EXPLAIN ANALYZE + COST	8
STEP 3) MONITORING POSTGRES WORKLOAD	10
Zip ASH DATA FOR REMOTE TROUBLESHOOTING	14

Step 1) Get dbstate output

Script by Abel Macias, Dimas Chbane

- Connect using psql

```
psql -h localhost
```

- List all databases

```
\l          # this list all databases
```

Example output:

```
kristofferson.a.arao=# \l
```

Name		Owner	List of databases	
Ctype	Access privileges		Encoding	Collate

```

-----+-----+-----+-----
-+-----+-----+-----+-----
  kristofferson.a.arao | kristofferson.a.arao | UTF8      | en_US.UTF-8
| en_US.UTF-8 |
  postgres              | postgres              | UTF8      | en_US.UTF-8
| en_US.UTF-8 |
  template0             | postgres              | UTF8      | en_US.UTF-8
| en_US.UTF-8 | =c/postgres          +
|              |                       |           |
|              | postgres=CTc/postgres |           |
  template1             | postgres              | UTF8      | en_US.UTF-8
| en_US.UTF-8 | =c/postgres          +
|              |                       |           |
|              | postgres=CTc/postgres |           |
(4 rows)

```

- Show current database

```

kristofferson.a.arao=# SELECT current_database();
      current_database
-----
  kristofferson.a.arao
(1 row)

```

- Connect to the database

```

\c <db name>      # connect to a database

Example output:

kristofferson.a.arao=# \c postgres
You are now connected to database "postgres" as user
"kristofferson.a.arao".

```

- Run the pg_dbstate.sql script

```

\i pg_dbstate.sql      # run the sql script

postgres=# \i pg_dbstate.sql
Timing is on.
Time: 1.709 ms

```

```
Time: 1.345 ms
Time: 0.192 ms
Tuples only is on.
Expanded display is on.
Time: 0.247 ms
Time: 0.470 ms
Time: 5.222 ms
Time: 3.889 ms
Time: 1.759 ms
Time: 12.733 ms
Expanded display is off.
```

- Rename the dbstate.txt to the database name

```
ls dbstate.txt pg_dbstate.sql
dbstate.txt    pg_dbstate.sql

mv dbstate.txt dbstate.txt.postgresql
```

Step 2) Get explain plan of the slow SQL

- Connect using psql

```
psql -h localhost
```

- Connect to the database

```
\c <db name>      # connect to a database

Example output:

kristofferson.a.arao=# \c postgres
You are now connected to database "postgres" as user
"kristofferson.a.arao".
```

- Run the EXPLAIN command with the slow SQL and spool to text

```
# example artificial slow query below with pg_sleep to 5 seconds

explain (analyze,verbose, costs,buffers, FORMAT text)  SELECT
schemaname, tablename
  FROM pg_tables, pg_sleep(5)
 WHERE schemaname <> 'pg_catalog';

# run with \o <filename> to spool to text
```

Example output:

```
postgres=# \o explain_text.txt
postgres=#
postgres=# explain (analyze,verbose, costs,buffers, FORMAT text)
SELECT schemaname, tablename
postgres-#      FROM pg_tables, pg_sleep(5)
postgres-#      WHERE schemaname <> 'pg_catalog';
postgres=#
postgres=# \o
```

- Query Plan output

```

                                QUERY PLAN
-----
Nested Loop  (cost=1.14..18.32 rows=58 width=128) (actual time=5000.398..5000.435
rows=7 loops=1)
  Output: n.nspname, c.relname
  Buffers: shared hit=13
   -> Function Scan on pg_catalog.pg_sleep  (cost=0.00..0.01 rows=1 width=0)
      (actual time=5000.082..5000.082 rows=1 loops=1)
        Output: pg_sleep.pg_sleep
        Function Call: pg_sleep('5'::double precision)
   -> Hash Join  (cost=1.14..17.73 rows=58 width=128) (actual time=0.311..0.344
rows=7 loops=1)
      Output: c.relname, n.nspname
      Inner Unique: true
      Hash Cond: (c.relnamespace = n.oid)
      Buffers: shared hit=13
       -> Seq Scan on pg_catalog.pg_class c  (cost=0.00..16.27 rows=69 width=72)
          (actual time=0.028..0.279 rows=69 loops=1)
            Output: c.relname, c.relnamespace, c.reltypes, c.reloftype,
c.relowner, c.relam, c.relfilenode, c.reltablespace, c.relpages, c.reltuples,
c.relallvisible, c.reltoastrelid, c.relhasindex, c.relisshred, c.relpersistence,
```

```

c.relkind, c.relnatts, c.relchecks, c.relhasoids, c.relhasrules, c.relhastriggers,
c.relhassubclass, c.relrowsecurity, c.relfrowsecurity, c.relispopulated,
c.relreplident, c.relispartition, c.relrewrite, c.relrozenxid, c.relminmxid,
c.relacl, c.reloptions, c.relpartbound
    Filter: (c.relkind = ANY ('{r,p}':"char"[]))
    Rows Removed by Filter: 273
    Buffers: shared hit=12
-> Hash (cost=1.07..1.07 rows=5 width=68) (actual time=0.024..0.024
rows=5 loops=1)
    Output: n.nspname, n.oid
    Buckets: 1024 Batches: 1 Memory Usage: 9kB
    Buffers: shared hit=1
-> Seq Scan on pg_catalog.pg_namespace n (cost=0.00..1.07 rows=5
width=68) (actual time=0.010..0.015 rows=5 loops=1)
    Output: n.nspname, n.oid
    Filter: (n.nspname <> 'pg_catalog'::name)
    Rows Removed by Filter: 1
    Buffers: shared hit=1
Planning Time: 0.574 ms
Execution Time: 5000.522 ms
(27 rows)

```

- Paste the explain plan to <https://explain.depesz.com> , click Submit

New explain

Optional title for plan:

s4

Paste your explain/explain analyze here:

QUERY PLAN

```
Nested Loop (cost=1.14..18.32 rows=58 width=128) (actual time=5000.398..5000.435 rows=7 loops=1)
Output: n.nspname, c.relname
Buffers: shared hit=13
-> Function Scan on pg_catalog.pg_sleep (cost=0.00..0.01 rows=1 width=0) (actual time=5000.082..5000.082 rows=1 loops=1)
Output: pg_sleep.pg_sleep
Function Call: pg_sleep('5':double precision)
-> Hash Join (cost=1.14..17.73 rows=58 width=128) (actual time=0.311..0.344 rows=7 loops=1)
Output: c.relname, n.nspname
Inner Unique: true
Hash Cond: (c.relnamespace = n.oid)
Buffers: shared hit=13
-> Seq Scan on pg_catalog.pg_class c (cost=0.00..16.27 rows=69 width=72) (actual time=0.028..0.279 rows=69 loops=1)
Output: c.relname, c.relnamespace, c.reltyp, c.reloftype, c.relowner, c.relam, c.relfileid, c.reltablespace, c.relpages, c.reltuples, c.relallvisible, c.reltoastrelid,
c.relhastindex, c.reliashared, c.relpersistence, c.relkind, c.relnatts, c.relchecks, c.relhastoids, c.relhastrules, c.relhasttriggers, c.relhassubclass, c.relnrowsecurity, c.relnforcerowsecurity,
c.relispopulated, c.relreplicated, c.relispartition, c.relrewrite, c.relfrozenxid, c.relinminxid, c.relacl, c.reloptions, c.relpartbound
Filter: (c.relkid = ANY ('(r,p)::"char"[]))
Rows Removed by Filter: 273
Buffers: shared hit=12
-> Hash (cost=1.07..1.07 rows=5 width=68) (actual time=0.024..0.024 rows=5 loops=1)
Output: n.nspname, n.oid
Buckets: 1024 Batches: 1 Memory Usage: 9kB
Buffers: shared hit=1
-> Seq Scan on pg_catalog.pg_namespace n (cost=0.00..1.07 rows=5 width=68) (actual time=0.010..0.015 rows=5 loops=1)
Output: n.nspname, n.oid
Filter: (n.nspname <> 'pg_catalog':name)
Rows Removed by Filter: 1
Buffers: shared hit=1

Planning Time: 0.574 ms
Execution Time: 5000.522 ms
(27 rows)
```

☐ I want this plan to be visible on the history page.

☐ I want this plan to be obfuscated before saving.

Submit

- The output will show where most of the response time is spent

explain.depesz.com
PostgreSQL's explain analyze made readable

new explain history help about contact login

Result: weLs : s4

To delete this plan, you can use [this link](#).

This link will not be shown any more, so you might want to bookmark it, just in case.

Settings Add optimization

#	exclusive	inclusive	rows x	rows	loops	node
1.	0.009	5,000.435	↑ 8.3	7	1	→ Nested Loop (cost=1.14..18.32 rows=58 width=128) (actual time=5,000.398..5,000.435 rows=7 loops=1) Output: n.nspname, c.relname Buffers: shared hit=13
2.	5,000.082	5,000.082	↑ 1.0	1	1	→ Function Scan on pg_catalog.pg_sleep (cost=0.00..0.01 rows=1 width=0) (actual time=5,000.082..5,000.082 rows=1 loops=1) Output: pg_sleep.pg_sleep Function Call: pg_sleep('5'::double precision)
3.	0.041	0.344	↑ 8.3	7	1	→ Hash Join (cost=1.14..17.73 rows=58 width=128) (actual time=0.311..0.344 rows=7 loops=1) Output: c.relname, n.nspname Inner Unique: true Hash Cond: (c.relnamespace = n.oid) Buffers: shared hit=13
4.	0.279	0.279	↑ 1.0	69	1	→ Seq Scan on pg_catalog.pg_class c (cost=0.00..16.27 rows=69 width=72) (actual time=0.028..0.279 rows=69 loops=1) Output: c.relname, c.relnamespace, c.relytype, c.reloftype, c.relowner, c.relam, c.relfilenode, c.reltablespace, c.relpages, c.reltuples, c.relallvisible, c.reltoastrelid, c.relhasindex, c.relisshred, c.relpersistence, c.relkind, c.relnatts, c.relchecks, c.relhasoids, c.relhasrules, c.relhastiggers, c.relhassubclass, c.relowsecurity, c.relorowsecurity, c.relispopulated, c.relreplicant, c.relpartition, c.relrewrite, c.relfrozenxid, c.relinmxid, c.relacl, c.reloptions, c.relpartbound Filter: (c.relkind = ANY ('(f,p)::'char'[])) Rows Removed by Filter: 273 Buffers: shared hit=12
5.	0.009	0.024	↑ 1.0	5	1	→ Hash (cost=1.07..1.07 rows=5 width=68) (actual time=0.024..0.024 rows=5 loops=1) Output: n.nspname, n.oid Buckets: 1024 Batches: 1 Memory Usage: 9kB Buffers: shared hit=1
6.	0.015	0.015	↑ 1.0	5	1	→ Seq Scan on pg_catalog.pg_namespace n (cost=0.00..1.07 rows=5 width=68) (actual time=0.010..0.015 rows=5 loops=1) Output: n.nspname, n.oid Filter: (n.nspname <> 'pg_catalog'::name) Rows Removed by Filter: 1 Buffers: shared hit=1

Planning time : 0.574 ms
Execution time : 5,000.522 ms

A few notes:

\timing

- \timing , equivalent to SET TIME ON in Oracle to show the wall clock time on every SQL execution in psql

```
postgres=# \timing
```

Timing is on.

```
postgres=# explain (verbose, costs, buffers, FORMAT text) SELECT schemaname,
tablename
..output snipped..
Planning Time: 0.583 ms
Execution Time: 5000.761 ms
(27 rows)
```

Time: 5002.297 ms (00:05.002)

```
postgres=#
```

EXPLAIN ANALYZE + COST

- The ANALYZE option of EXPLAIN means to execute the SQL then print the query plan
- Below is query plan with (5 seconds) and without ANALYZE (1 second)

WITH ANALYZE

```
postgres=# \timing
Timing is on.
postgres=# explain (analyze,verbose,costs,buffers, FORMAT text) SELECT schemaname,
tablename
postgres=# FROM pg_tables, pg_sleep(5)
postgres=# WHERE schemaname <> 'pg_catalog';
```

QUERY PLAN

```
-----
Nested Loop (cost=1.14..18.32 rows=58 width=128) (actual time=5000.472..5000.519
rows=7 loops=1)
  Output: n.nspname, c.relname
  Buffers: shared hit=13
  -> Function Scan on pg_catalog.pg_sleep (cost=0.00..0.01 rows=1 width=0)
(actual time=5000.171..5000.172 rows=1 loops=1)
    Output: pg_sleep.pg_sleep
    Function Call: pg_sleep('5'::double precision)
  -> Hash Join (cost=1.14..17.73 rows=58 width=128) (actual time=0.296..0.338
rows=7 loops=1)
    Output: c.relname, n.nspname
    Inner Unique: true
    Hash Cond: (c.relnamespace = n.oid)
    Buffers: shared hit=13
    -> Seq Scan on pg_catalog.pg_class c (cost=0.00..16.27 rows=69 width=72)
(actual time=0.024..0.027 rows=69 loops=1)
      Output: c.relname, c.relnamespace, c.reltype, c.reloftype,
c.relowner, c.relam, c.relfilenode, c.reltablespace, c.relpages, c.reltuples,
c.relallvisible, c.reltoastrelid, c.relhasindex, c.relisshared, c.relpersistence,
c.relkind, c.relnatts, c.relchecks, c.relhasoids, c.relhasrules, c.relhastriggers,
c.relhassubclass, c.relrowsecurity, c.relforcerowsecurity, c.relispopulated,
c.relreplident, c.relispartition, c.relrewrite, c.relrozenxid, c.relminmxid,
c.relacl, c.reloptions, c.relpartbound
      Filter: (c.relkind = ANY ('{r,p}'::"char"[]))
      Rows Removed by Filter: 273
      Buffers: shared hit=12
    -> Hash (cost=1.07..1.07 rows=5 width=68) (actual time=0.023..0.024
rows=5 loops=1)
      Output: n.nspname, n.oid
      Buckets: 1024 Batches: 1 Memory Usage: 9kB
```



```

        Buffers: shared hit=1
        -> Seq Scan on pg_catalog.pg_namespace n (cost=0.00..1.07 rows=5
width=68) (actual time=0.009..0.015 rows=5 loops=1)
        Output: n.nspname, n.oid
        Filter: (n.nspname <> 'pg_catalog'::name)
        Rows Removed by Filter: 1
        Buffers: shared hit=1
Planning Time: 0.583 ms
Execution Time: 5000.761 ms
(27 rows)

```

Time: 5002.297 ms (00:05.002)

WITHOUT ANALYZE

```

postgres=# explain (verbose, costs, buffers, FORMAT text) SELECT schemaname,
tablename
postgres=# FROM pg_tables, pg_sleep(5)
postgres=# WHERE schemaname <> 'pg_catalog';
ERROR: EXPLAIN option BUFFERS requires ANALYZE
Time: 2.771 ms
postgres=#
postgres=#
postgres=# explain (verbose, costs, FORMAT text) SELECT schemaname, tablename
postgres=# FROM pg_tables, pg_sleep(5)
postgres=# WHERE schemaname <> 'pg_catalog';

```

QUERY PLAN

```

-----
-----
-----
-----
-----
-----
Nested Loop (cost=1.14..18.32 rows=58 width=128)
  Output: n.nspname, c.relname
  -> Function Scan on pg_catalog.pg_sleep (cost=0.00..0.01 rows=1 width=0)
    Output: pg_sleep.pg_sleep
    Function Call: pg_sleep('5'::double precision)
  -> Hash Join (cost=1.14..17.73 rows=58 width=128)
    Output: c.relname, n.nspname
    Inner Unique: true
    Hash Cond: (c.relnamespace = n.oid)
    -> Seq Scan on pg_catalog.pg_class c (cost=0.00..16.27 rows=69 width=72)
      Output: c.relname, c.relnamespace, c.reltype, c.reloftype,
c.relowner, c.relam, c.relfilenode, c.reltablespace, c.relpages, c.reltuples,
c.relallvisible, c.reltoastrelid, c.relhasindex, c.relisshared, c.relpersistence,
c.relkind, c.relnatts, c.relchecks, c.relhasoids, c.relhasrules, c.relhastriggers,
c.relhassubclass, c.relrowsecurity, c.relforcerowsecurity, c.relispopulated,
c.relreplident, c.relispartition, c.relrewrite, c.relrozenxid, c.relminmxid,
c.relacl, c.reloptions, c.relpartbound
      Filter: (c.relkind = ANY ('{r,p}'::"char"[]))
    -> Hash (cost=1.07..1.07 rows=5 width=68)
      Output: n.nspname, n.oid
      -> Seq Scan on pg_catalog.pg_namespace n (cost=0.00..1.07 rows=5
width=68)

```

```
Output: n.nspname, n.oid
Filter: (n.nspname <> 'pg_catalog'::name)
(17 rows)
Time: 1.049 ms
```

Step 3) Monitoring Postgres workload

- Download the tool at <https://github.com/dbacvetkov/PASH-Viewer/archive/master.zip>
 - Project page here <https://github.com/dbacvetkov/PASH-Viewer>
- Unzip and read the README.md
- Build the tool

```
cd /Users/kristofferson.a.arao/ash_viewer/PASH-Viewer-master
./gradlew assembleDist

cd /Users/kristofferson.a.arao/ash_viewer/PASH-Viewer-
master/build/distributions/PASH-Viewer-0.4.1/bin
```

- Create the monitoring user PGMONUSER to see query plans

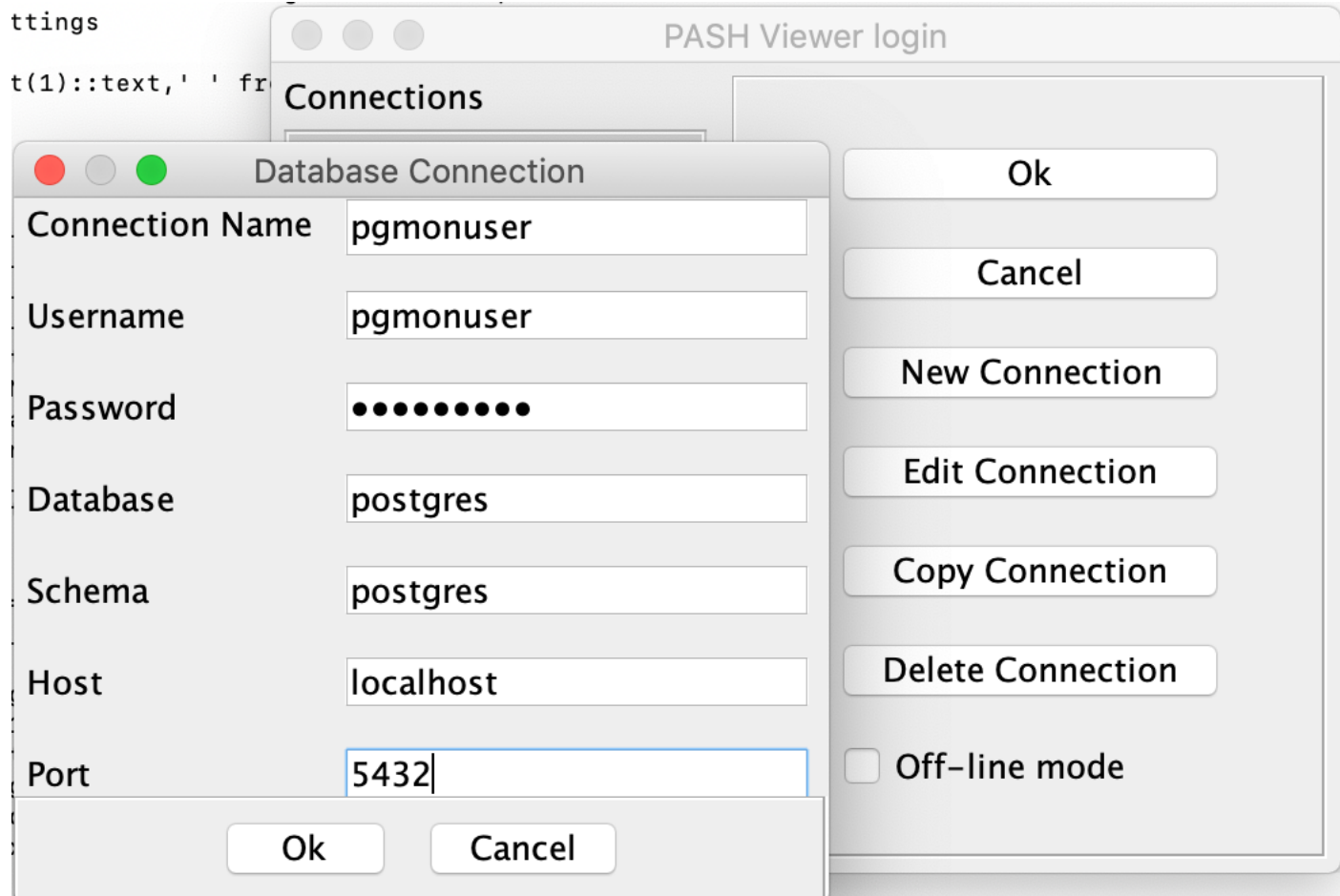
```
• from psql connect to postgres database

psql -h localhost # psql
\c postgres       # connect to postgres

• create the pgmonuser

postgres=# CREATE USER pgmonuser WITH password 'pgmonuser';
CREATE ROLE
postgres=# GRANT pg_monitor TO pgmonuser;
GRANT ROLE
postgres=#
```

- Enter connection details, use the PGMONUSER to connect



- Run PASH-Viewer

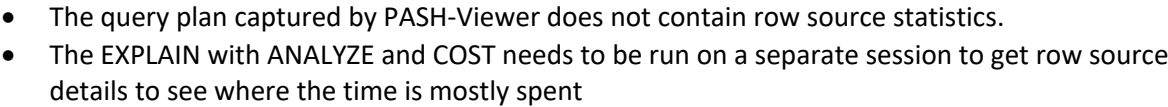
```
./PASH-Viewer
```

- The PASH-Viewer will create a directory for each connection name
- Inside the directory are two subdirectories
 - Plan directory - where SQLs and plans are stored
 - .jdb data files - for storing historical workload and raw data. This can be zipped and shared for remote troubleshooting

```
./PASH-Viewer
```

```
$ pwd
/Users/kristofferson.a.arao/ash_viewer/PASH-Viewer-master/build/distributions/PASH-Viewer-0.4.1/bin/pgmonuser
```

- Below correlates the database load with the running SQL and the underlying execution plan



explain.depesz.com

PostgreSQL's explain analyze made readable

new explain history help about contact login

Result: ddu

Settings Add optimization

#	exclusive	inclusive	rows	rows	loops	node
1.	0.007	60,001.290	18.3	7	1	Hashed Loop (cost=11.14, 18.32 rows=58 width=128) (actual time=0.001260..0.001290 rows=7 loops=1) Output: n.nspname, c.relname Buffers: shared hit=13
2.	60,001.067	60,001.067	11.0	1	1	Function Scan on pg_catalog.pg_sleep (cost=0.00..0.01 rows=1 width=0) (actual time=60.001066..60.001067 rows=1 loops=1) Output: pg_sleep, pg_sleep Function Call: pg_sleep(p)::double precision
3.	0.027	0.216	18.3	7	1	Hash Join (cost=11.14, 17.73 rows=58 width=128) (actual time=0.190..0.216 rows=7 loops=1) Output: c.relname, n.nspname Inner Unique: true Hash Cond: (c.relnamespace = n oid) Buffers: shared hit=13
4.	0.174	0.174	11.0	69	1	Seq Scan on pg_catalog.pg_class c (cost=0.00..16.27 rows=69 width=72) (actual time=0.0200..0.174 rows=69 loops=1) Output: c.relname, c.relnamespace, c.reltype, c.relowner, c.relam, c.relfisecode, c.reltabspace, c.relpages, c.relnatts, c.relchecks, c.relaohoids, c.relasrules, c.relatriggers, c.relasubstests, c.relrowsecurity, c.relforcerowsecurity, c.relispopulated, c.relkind, c.relpartition, c.relwritere, c.reflcozccid, c.relminmxid, c.relaid, c.reloptions, c.relpartbound Filter: (c.relkind = ANY ('{t,p}'::char[])) Rows Removed by Filter: 273 Buffers: shared hit=12
5.	0.005	0.015	11.0	5	1	Hash (cost=1.07..1.07 rows=5 width=68) (actual time=0.015..0.015 rows=5 loops=1) Output: n.nspname, n.oid Buckets: 1024 Batches: 1 Memory Usage: 9KB Buffers: shared hit=1
6.	0.010	0.010	11.0	5	1	Seq Scan on pg_catalog.pg_namespaces p (cost=0.00..1.07 rows=5 width=68) (actual time=0.007..0.010 rows=5 loops=1) Output: n.nspname, n.oid Filter: (n.nspname <> 'pg_catalog'::name) Rows Removed by Filter: 1 Buffers: shared hit=1

Planning time : 0.338 ms
Execution time : 60.002.038 ms

- PASH Viewer :: pgmonuser@localhost:5432/postgres (pg version = 11.0)

File Help

Settings

Top activity Detail History

Profile Size, MB

 - postgres_schema
 - pgmonuser
 - 100% 0.04 1.43 < 1
 - localhost

July 2019

Mon	Tue	Wed	Thu	Fri	Sat	Sun
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	31				

Select Delete

Preview :: pgmonuser@localhost:5432/postgres :: 08.07.2019

Active Sessions

Top activity Detail

Active Sessions

Top SQL SQL text SQL plan

PLAN 3201495233 FOR SOLID 9d4b66f73226e:

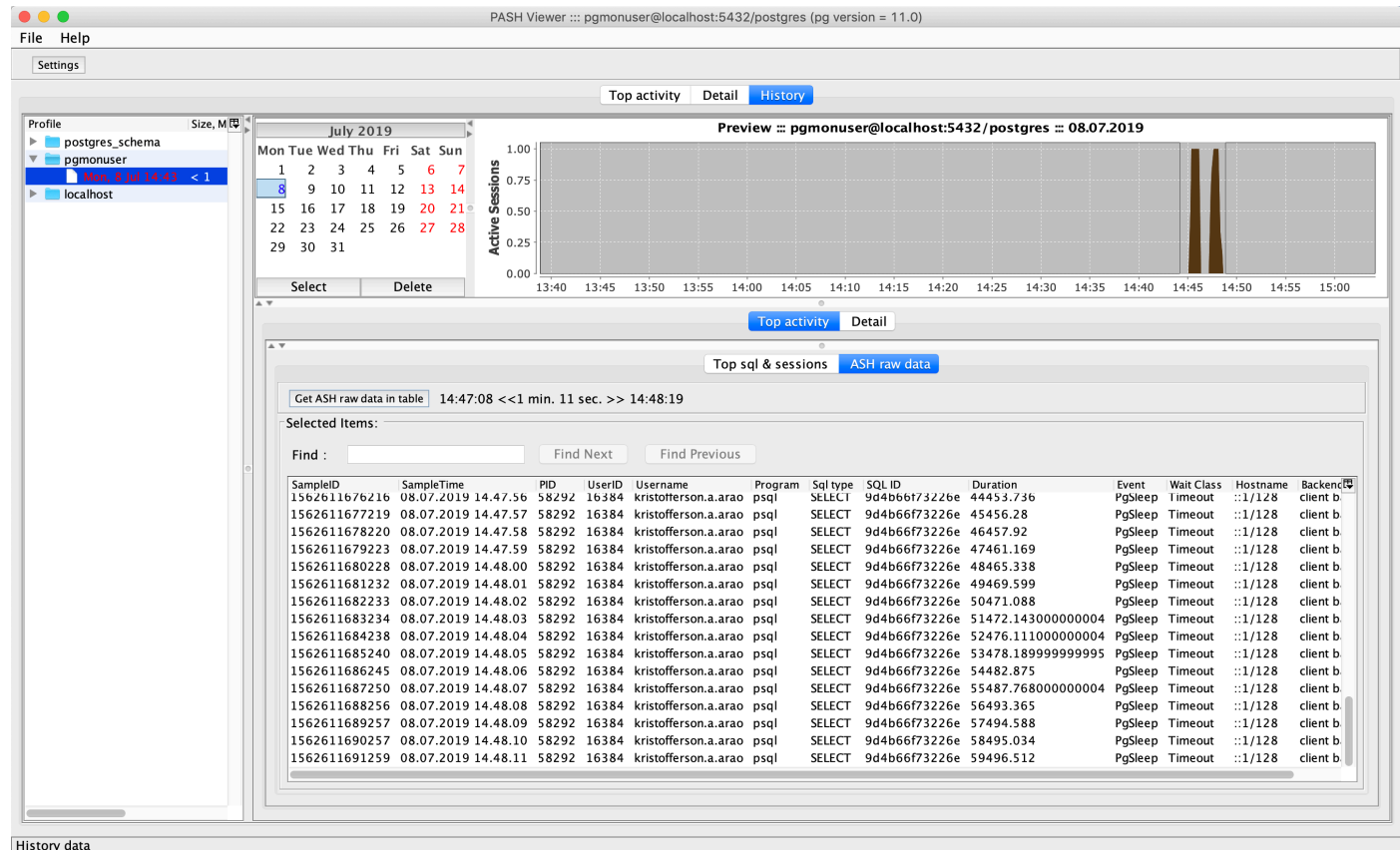
```
Nested Loop (cost=1.14..18.32 rows=58 width=128)
-> Function Scan on pg_sleep (cost=0.00..0.01 rows=1 width=0)
-> Hash Join (cost=1.14..17.73 rows=58 width=128)
    Hash Cond: (c.relnamespace = n.oid)
    -> Seq Scan on pg_class c (cost=0.00..16.27 rows=69 width=72)
    -> Seq Scan on pg_namespace n (cost=0.00..0.01 rows=1 width=4)
```

Activity % PID User Name Program Backend Type

Activity %	PID	User Name	Program	Backend Type
100.0%	58292	kristoff...	psql@::1/128	client backend

History data

- Also raw data is available for drill down



A few notes:

Zip ASH data for remote troubleshooting

- On the PASH-Viewer base directory

```
./PASH-Viewer

$ pwd
/Users/kristofferson.a.arao/ash_viewer/PASH-Viewer-master/build/distributions/PASH-Viewer-0.4.1/bin

$ ls -ltr
total 24
-rwxr-xr-x  1 kristofferson.a.arao  562225435   2442 Jul  8 01:56 PASH-Viewer.bat
-rwxr-xr-x  1 kristofferson.a.arao  562225435   5388 Jul  8 01:56 PASH-Viewer
drwxr-xr-x  5 kristofferson.a.arao  562225435    160 Jul  8 16:43 pgmonuser
```

- Zip the pgmonuser directory and share through email or Sharepoint URL

```
$ du -sm pgmonuser/
3    pgmonuser/

$ zip -r pgmonuser pgmonuser
adding: pgmonuser/ (stored 0%)
adding: pgmonuser/plan/ (stored 0%)
adding: pgmonuser/plan/9d4b66f73226e.sql (deflated 18%)
adding: pgmonuser/plan/9d4b66f73226e.plan (deflated 50%)
adding: pgmonuser/08072019164333/ (stored 0%)
adding: pgmonuser/08072019164333/00000000.jdb (deflated 92%)
adding: pgmonuser/08072019164333/je.lck (stored 0%)
adding: pgmonuser/08072019144335/ (stored 0%)
adding: pgmonuser/08072019144335/00000000.jdb (deflated 91%)
adding: pgmonuser/08072019144335/je.lck (stored 0%)

$ ls -ltr pgmonuser.zip
-rw-r--r--  1 kristofferson.a.arao  562225435  108646 Jul  8 16:45 pgmonuser.zip
```