

Анализ исторической нагрузки PostgreSQL

Расширение pg_profile

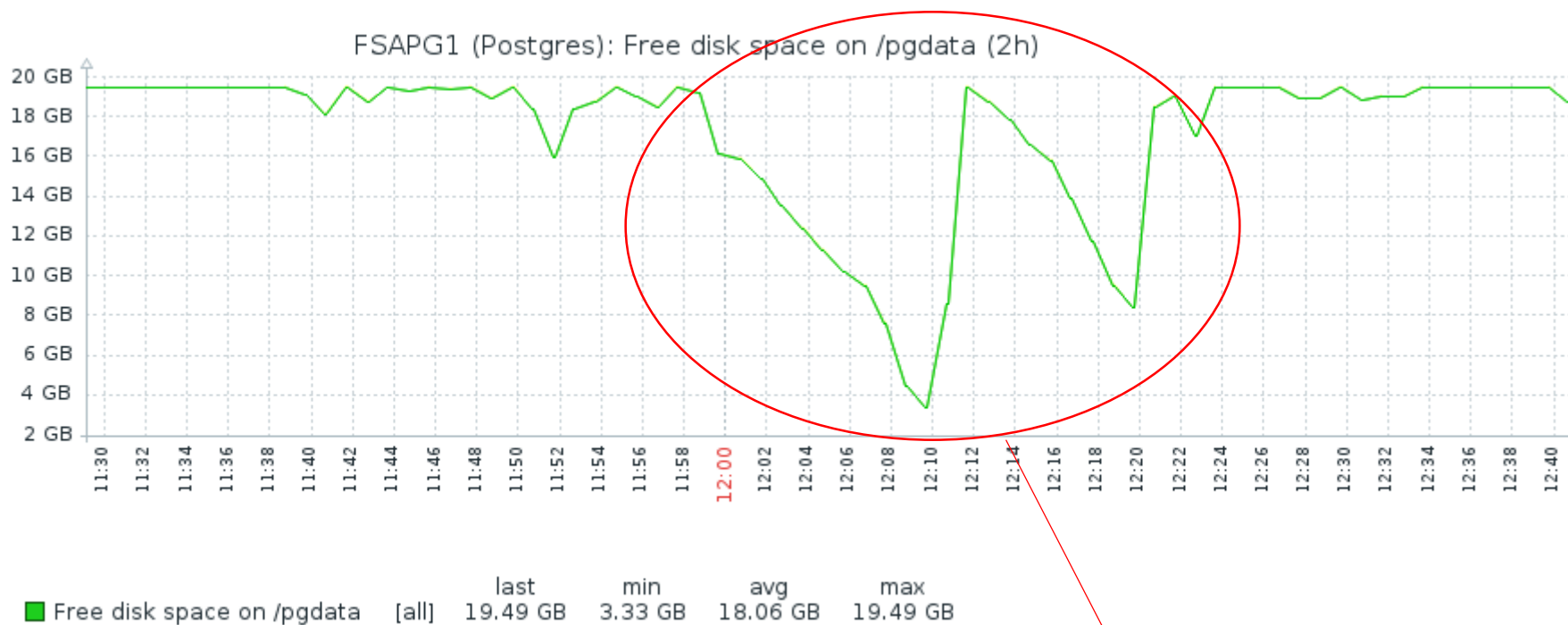


PGConf.Russia 2018

Андрей Зубков
zubkov@parma.ru

PARMA
technologies group

Постановка задачи



Что это было?

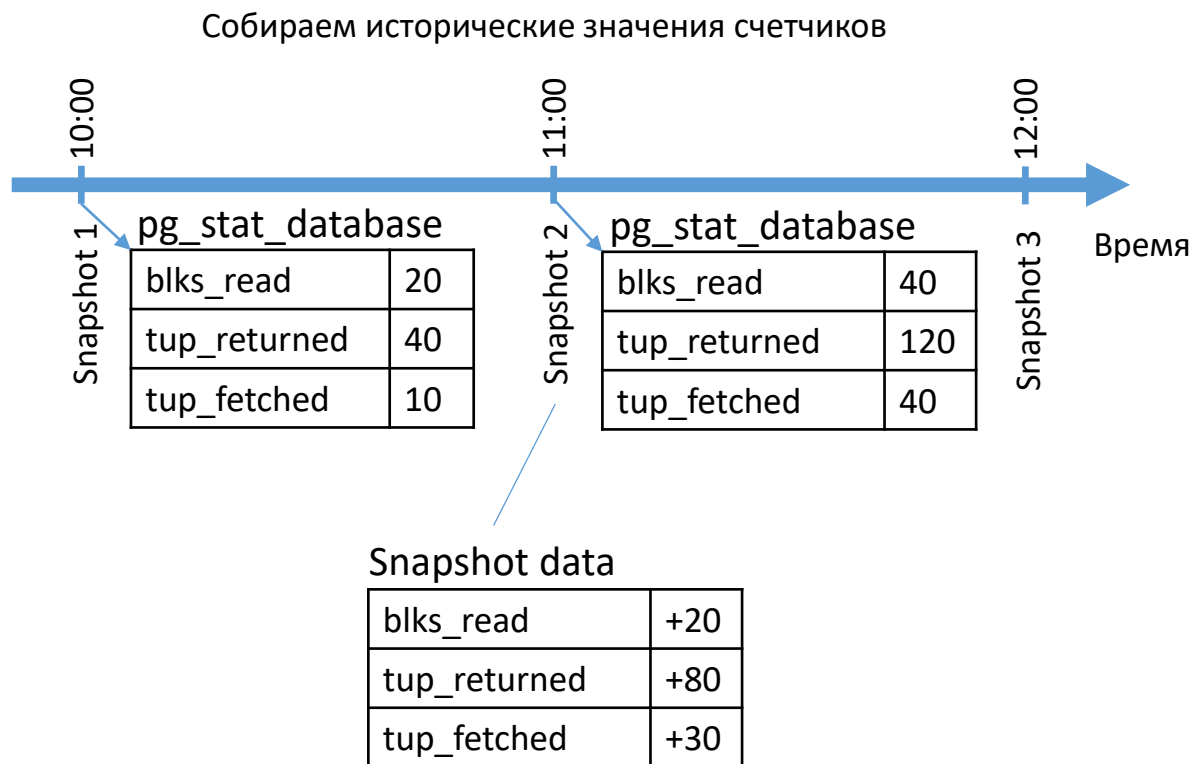
Существующие средства

- PoWA (pg_stat_statements+ на python с Web-UI)
Состоит из основного модуля и отдельного модуля для отображения UI
- pgbager
Представляет собой анализатор журналов сервера, т.е. требует детального журналирования.

Механизм

pg_stat_database

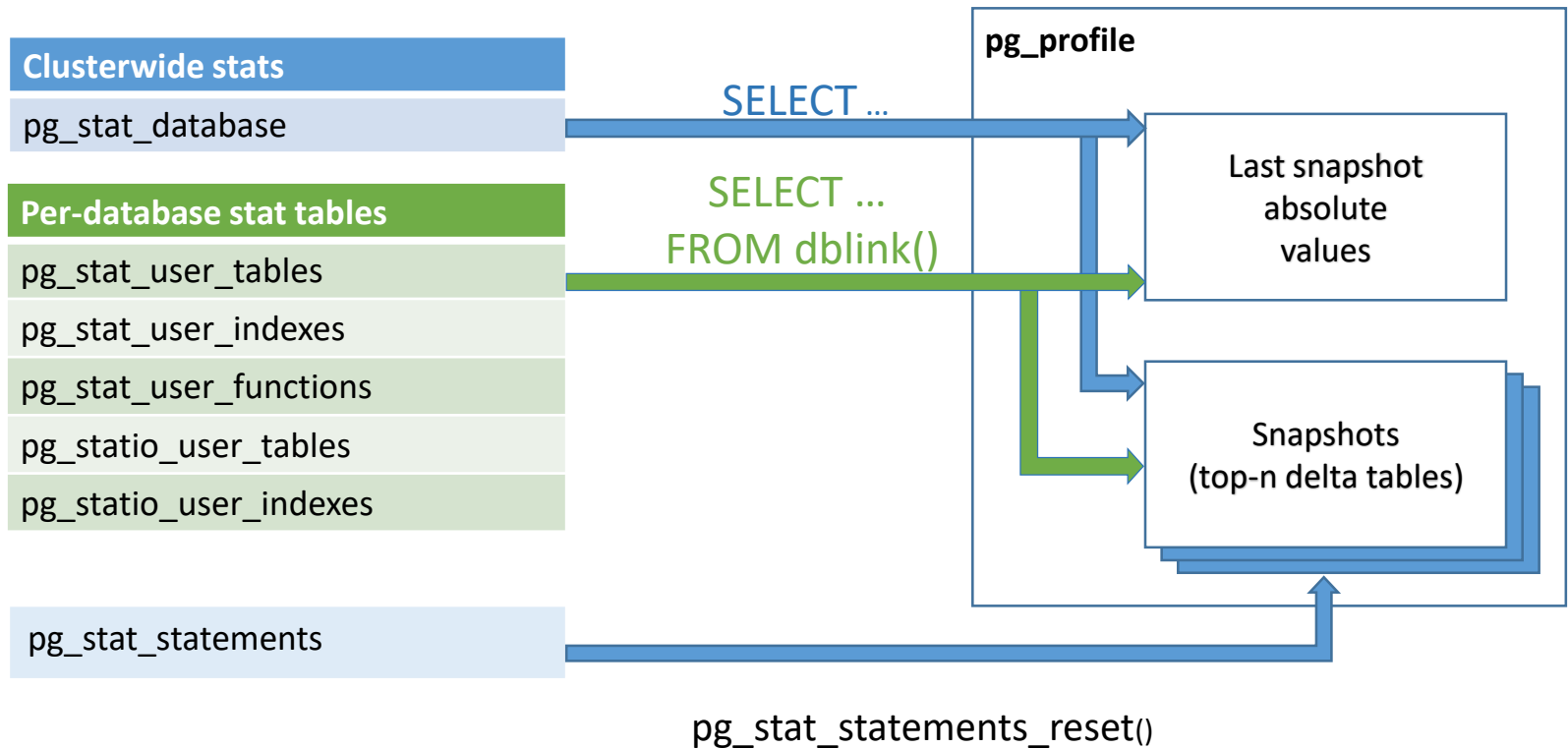
Column	Type
datid	oid
datname	name
numbackends	integer
xact_commit	bigint
xact_rollback	bigint
blks_read	bigint
blks_hit	bigint
tup_returned	bigint
tup_fetched	bigint
tup_inserted	bigint
tup_updated	bigint
tup_deleted	bigint
conflicts	bigint
temp_files	bigint
temp_bytes	bigint
deadlocks	bigint
blk_read_time	double precision
blk_write_time	double precision
stats_reset	timestamp with time zone



Наблюдаемые представления

Представление	Краткое содержание
pg_stat_statements	Статистики по выполнением выражений
pg_stat_user_tables, pg_stat_user_indexes	Статистики по таблицам и индексам: количество записей, вставок, удалений, последовательных и индексных сканирований...
pg_statio_user_tables, pg_statio_user_indexes	Статистики ввода-вывода по таблицам и индексам
pg_stat_user_functions	Статистики по вызовам и времени выполнения
pg_stat_database	Статистики по базам данных: транзакции (подтверждения и откаты), чтения блоков (с диска и из буферов), временные файлы

SELECT snapshot();



cron:

```
*/30 * * * * psql -d postgres -c 'SELECT profile.snapshot()'
```

Построение отчетов

Получение списка снимков:

```
postgres=# select * from profile.snapshot_show(1); -- за 1 сутки
 snapshot |          date_time
-----+-----
...
      480 | 2017-09-11 10:00:01+03
      481 | 2017-09-11 10:30:01+03
      482 | 2017-09-11 11:00:01+03
```

Или просто выполнить запрос к таблице *snapshots*

Построение отчета:

```
$ psql -qtc "SELECT profile.report(480,482)" -o report_480_482.html
```

Разделы отчета

Cluster statistics

Databases stats

Statements stats by database

SQL Query stats

Top SQL by elapsed time

Top SQL by executions

Top SQL by I/O wait time

Top SQL by gets

Top SQL by temp usage

Complete List of SQL Text

Schema objects stats

Most scanned tables

Top DML tables

Top growth tables

Unused indexes

I/O Schema objects stats

Top tables by I/O

Top indexes by I/O

User function stats

Top functions by total time

Top functions by executions

Vacuum related stats

Tables ordered by dead tuples ratio

Tables ordered by modified tuples ratio

Вид таблиц отчета

Databases stats

Database	Commits	Rollbacks	BlkHit%(read/hit)	Tup Ret/Fet	Tup Ins	Tup Del	Temp Bytes(Files)	Deadlocks
postgres	34588	0	99.18%(2155/261498)	1240156/37479	59353	1031	0(0)	0
register	15814	6	99.99%(530/6292902)	143564847/313535	12927	3181	0(0)	0
rssrds	3756	3	97.38%(9557/355038)	8305523/157633	447	2	0(0)	0
ral	13010	0	99.97%(81/302744)	5058306/89087	112	4	0(0)	0
gu	277	0	100.00%(0/22425)	157592/11609	0	0	0(0)	0
ral_dev	210	0	100.00%(0/35310)	190962/19680	0	0	0(0)	0

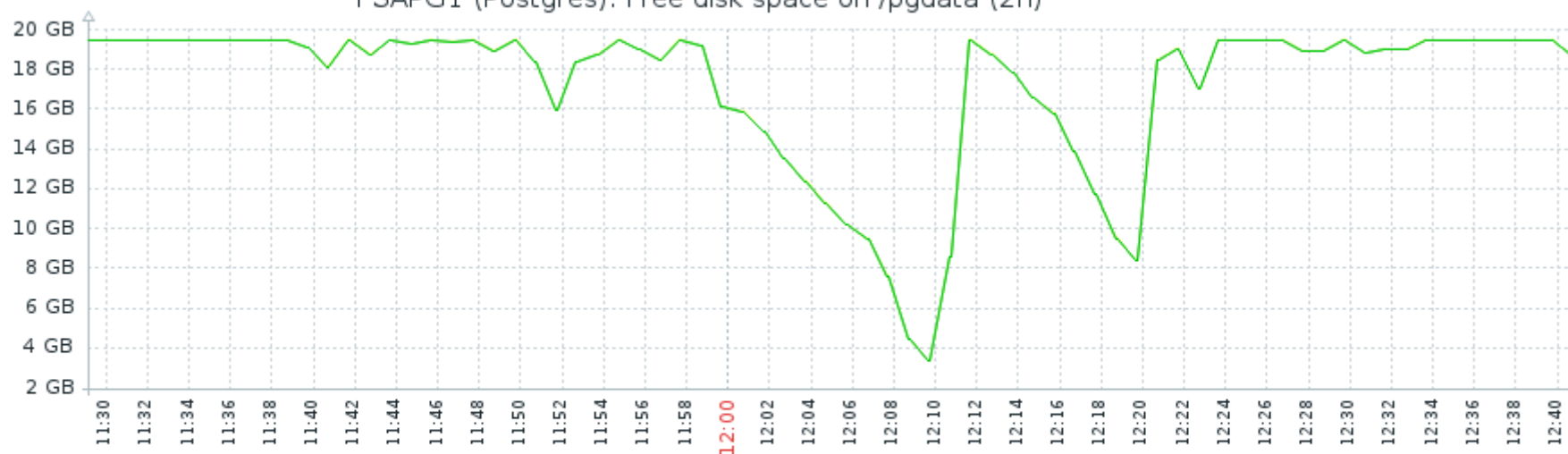
Вид таблиц отчета

Top SQL by elapsed time

Query ID	Database	Elapsed(s)	%Total	Rows	Mean(ms)	Min(ms)	Max(ms)	StdErr(ms)	Executions
d700812280	register	1603.3	44.12	3100	51720.154	32248.711	113077.066	26826.978	31
7228bbf1a1	register	1323.1	36.41	1300	101774.215	32369.266	166303.227	46378.412	13
e02453580c	register	501.8	13.81	1	501809.034	501809.034	501809.034	0.000	1
d340a3f15d	postgres	18.9	0.52	4567	4.132	0.237	32.227	5.066	4567
d97784fc9a	postgres	18.5	0.51	4537	4.080	0.177	22.322	5.199	4537
52325ac557	register	14.9	0.41	140	1062.788	651.028	1613.481	386.317	14
6cc5034a16	postgres	12.8	0.35	4476	2.857	0.166	31.543	4.500	4476

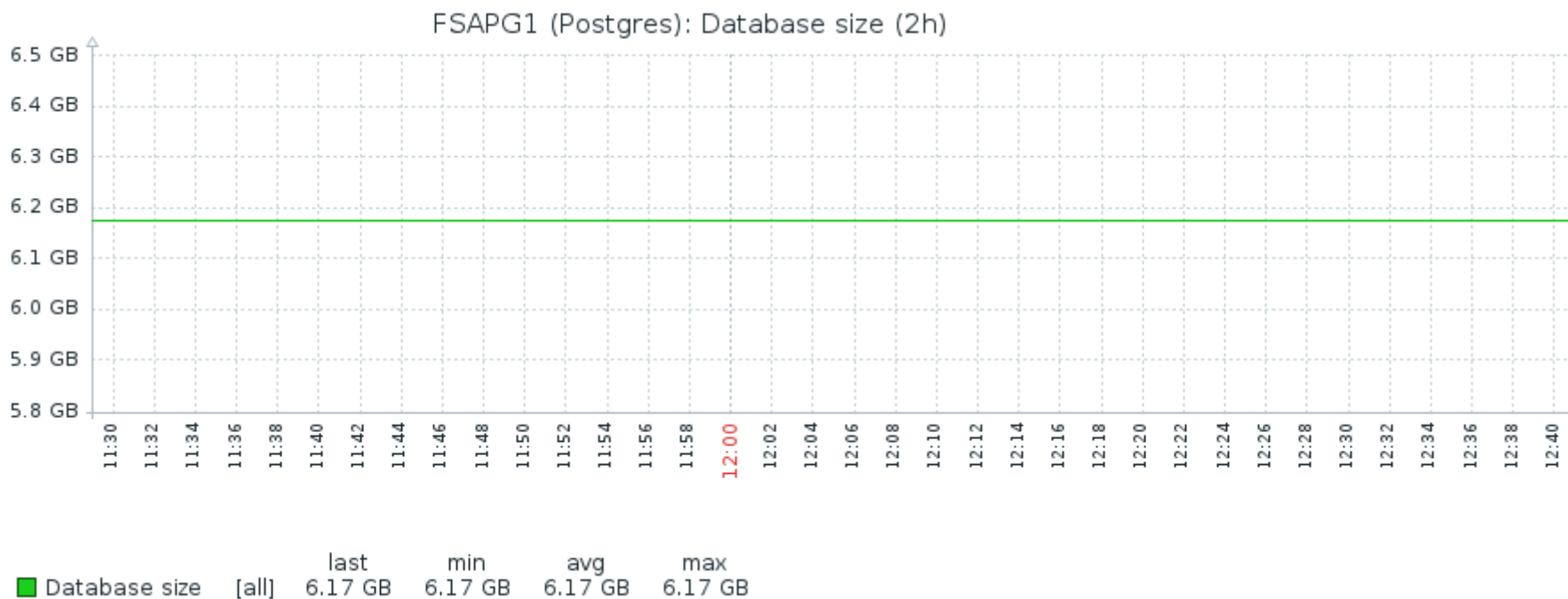
Инцидент с диском

FSAPG1 (Postgres): Free disk space on /pgdata (2h)



Free disk space on /pgdata [all] last 19.49 GB min 3.33 GB avg 18.06 GB max 19.49 GB

Размер базы не изменялся



Временные файлы... много

Databases stats

Database	Committs	Rollbacks	BlkHit%(read/hit)	Tup Ret/Fet	Tup Ins	Tup Del	Temp Bytes(Files)	Deadlocks
postgres	34588	0	99.18%(2155/261498)	1240156/37479	59353	1031	0(0)	0
register	15814	6	99.99%(530/6292902)	143564847/313535	12927	3181	73 GB(116)	0
rssrds	3756	3	97.38%(9557/355038)	8305523/157633	447	2	0(0)	0
ral	13010	0	99.97%(81/302744)	5058306/89087	112	4	0(0)	0
gu	277	0	100.00%(0/22425)	157592/11609	0	0	0(0)	0

ИЩЕМ ВИНОВНИКОВ

Top SQL by temp usage

Query ID	Database	Total(s)	Rows	Gets	Hits(%)	Temp_w (blk)	%Total	Temp_r (blk)	%Total	Executions
d700812280	register	1603.3	3100	2	100.00	4668631	57.78	4553807	57.52	31
7228bbf1a1	register	1323.1	1300	0	0.00	1957813	24.23	1909661	24.12	13
e02453580c	register	501.8	1	0	0.00	1433899	17.75	1433900	18.11	1
eeee22abea	register	1.6	600	0	0.00	9504	0.12	9506	0.12	2
a553b0d829	register	1.7	600	0	0.00	9504	0.12	9506	0.12	2

Complete List of SQL Text

QueryID	Query Text
d700812280	select dictaddres0_.aoguid as aoguid1_24_, dictaddres0_.aolevel as aolevel2_24_, dictaddres0_.offname as offname3_24_, dictaddres0_.parentguid as parentgu4_24_, dictaddres0_.shortname as shortnam5_24_ from v_fias_addrobj dictaddres0_ where dictaddres0_.aolevel=1 limit \$1

Распределение нагрузки

Top functions by total time

DB	Schema	Function	Executions	Total time	Self time	Mean time	Mean self time
STAGE	portal	gen_cube	1	15916103.78	549.54	15916103.779	549.539
STAGE	portal	gen_cube_10	1	15530417.75	256.75	15530417.748	256.754
STAGE	portal	gen_cube_02	1	6320579.52	439.33	6320579.515	439.329
STAGE	portal	gen_100_1	1	5573535.19	5573535.19	5573535.188	5573535.188
STAGE	portal	gen_100_1_10	1	4117945.13	4117945.13	4117945.133	4117945.133
STAGE	portal	gen_01_1	1	2665585.25	2665585.25	2665585.248	2665585.248

Применимость

- Проблемы производительности системы
- Аномальная нагрузка на сервере баз данных
- Нагрузочные тестирования
- Поиск активно растущих объектов
- Поиск неиспользуемых индексов
- И др.

Особенности установки

- Самый простой вариант

```
postgres=# CREATE EXTENSION pg_profile CASCADE;
```

- Аккуратный вариант

```
postgres=# CREATE EXTENSION dblink;
```

```
postgres=# CREATE EXTENSION pg_stat_statements;
```

```
postgres=# CREATE SCHEMA profile;
```

```
postgres=# CREATE EXTENSION pg_profile SCHEMA profile;
```

Настроить параметры `pg_stat_statements`

Параметры

- Количество объектов в отсортированных таблицах (и снимках):

pg_profile.topn = 20

- Время хранения снимков (в сутках):

pg_profile.retention = 7

Не без проблем...

- Сбор статистик в конце выполнения выражения
- *pg_stat_statements.track = all* собирает запросы всех уровней в одно представление , создавая проблемы с полем %Total
- *pg_relation_size()* ожидает освобождения эксклюзивной блокировки на таблице
- Хранение в снимках только топn записей может быть причиной неточности результатов

Спасибо за внимание!

https://github.com/zubkov-andrei/pg_profile



PGConf.Russia 2018

Андрей Зубков

DBA

Email: zubkov@parma.ru

PARMA Technologies Group