PGConf.Russia 2023

PostgresPro

Новые возможности наблюдения за СУБД *PostgreSQL* и *PostgresPro*

Андрей Зубков

Обсудим

- **Визуализацию наблюдений pg_profile/pgpro_pwr**
- Интерактивные отчеты pg_profile/pgpro_pwr
- Расширенные статистики вакуума 2.0
- Время архивирования
- Трассировку в СУБД PostgresPro

PostgresPro

Визуализация наблюдений pg_profile/pgpro_pwr

Сбор наблюдений `take_sample()`

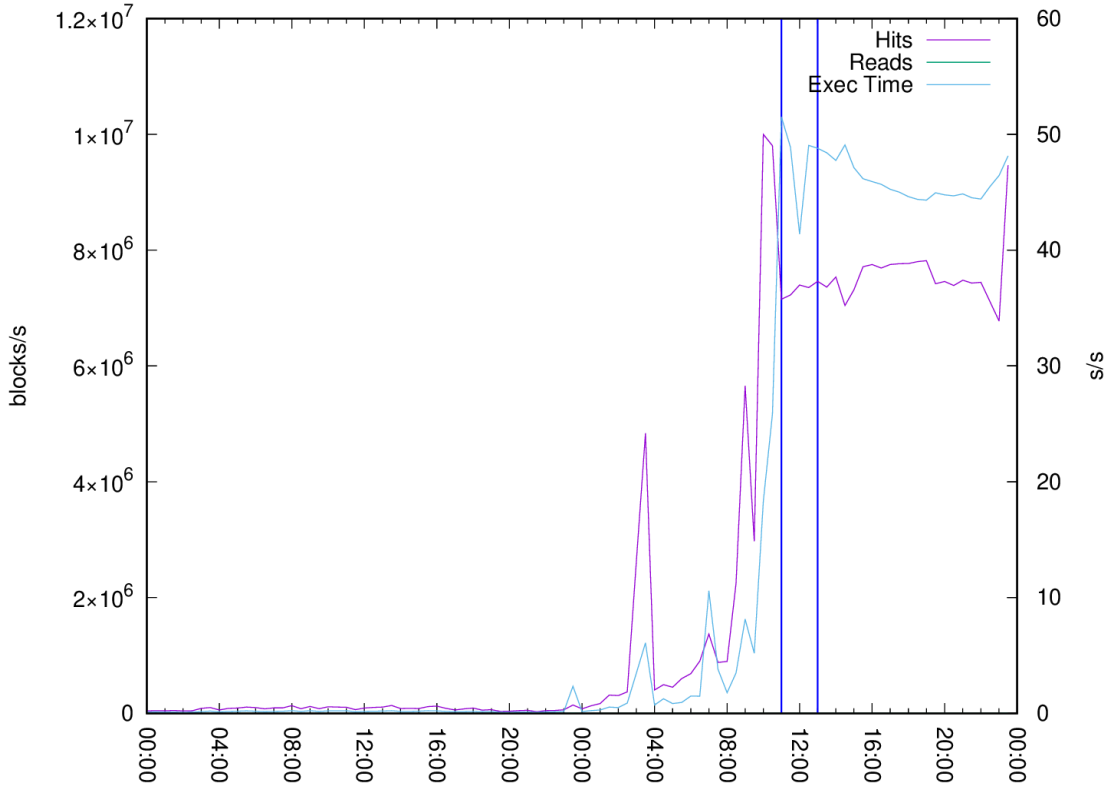
```
 sample |       sample_time
--------+--------------------------
      1 | 2021-03-26 19:00:02+00
      2 | 2021-03-26 19:08:57+00
      3 | 2021-03-26 19:30:01+00
      4 | 2021-03-26 20:00:02+00
      5 | 2021-03-26 20:30:02+00
      6 | 2021-03-26 21:00:02+00
....
    571 | 2021-04-07 15:30:02+00
    572 | 2021-04-07 16:00:02+00
    573 | 2021-04-07 16:30:02+00
(510 rows)
```

PostgresPro

Визуализация наблюдений pg_profile/pgpro_pwr

Поиск интервала для отчета

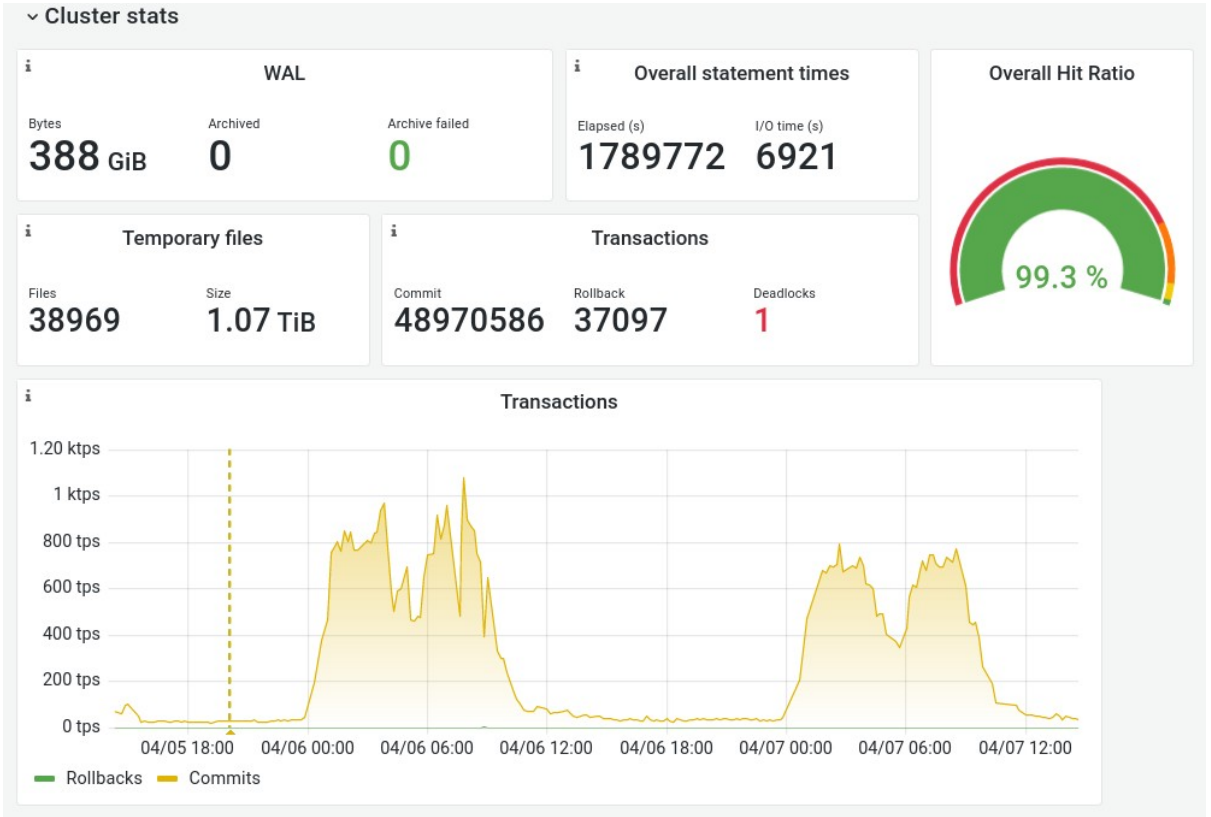
Для построения отчета требуется интервал



PostgresPro

Визуализация наблюдений

- Все данные доступны в репозитории расширения
- Grafana
- Dashboard на гитхабе



PostgresPro

Графики визуализации наблюдений в Grafana

- Cluster stats
- Database sizes
- Transactions
  - Commits by database
  - Rollbacks by database
  - Sessions/Session times
- Cluster I/O
  - Write (bgwriter, checkpointer, backends)
  - Write times by database
- Cluster I/O...
  - Reads/Read times by database
  - Hits/Hit ratio by database
- WAL
  - Size, Recs, FPIs
  - Archives/Failures
  - Write/Sync times
- Temp files and sizes by DB
- Summary statement statistics

PostgresPro

Особенности настройки Data source

Dashboard требует наличия источника данных, который

- Должен быть направлен в базу с расширением
- Пользователь в этой базе должен иметь search_path

Обсудим

- Визуализацию наблюдений pg_profile/pgpro_pwr
- **Интерактивные отчеты pg_profile/pgpro_pwr**
- Расширенные статистики вакуума 2.0
- Время архивирования
- Трассировка в СУБД PostgresPro

PostgresPro

pg_profile / pgpro_pwr 4.1 vs 4.2 (5.0?)

SELECT get_report(124,134);

- Агрегирует данные наблюдений интервала
- Формирует plain html отчет на базе шаблона

SELECT get_report(124,134);

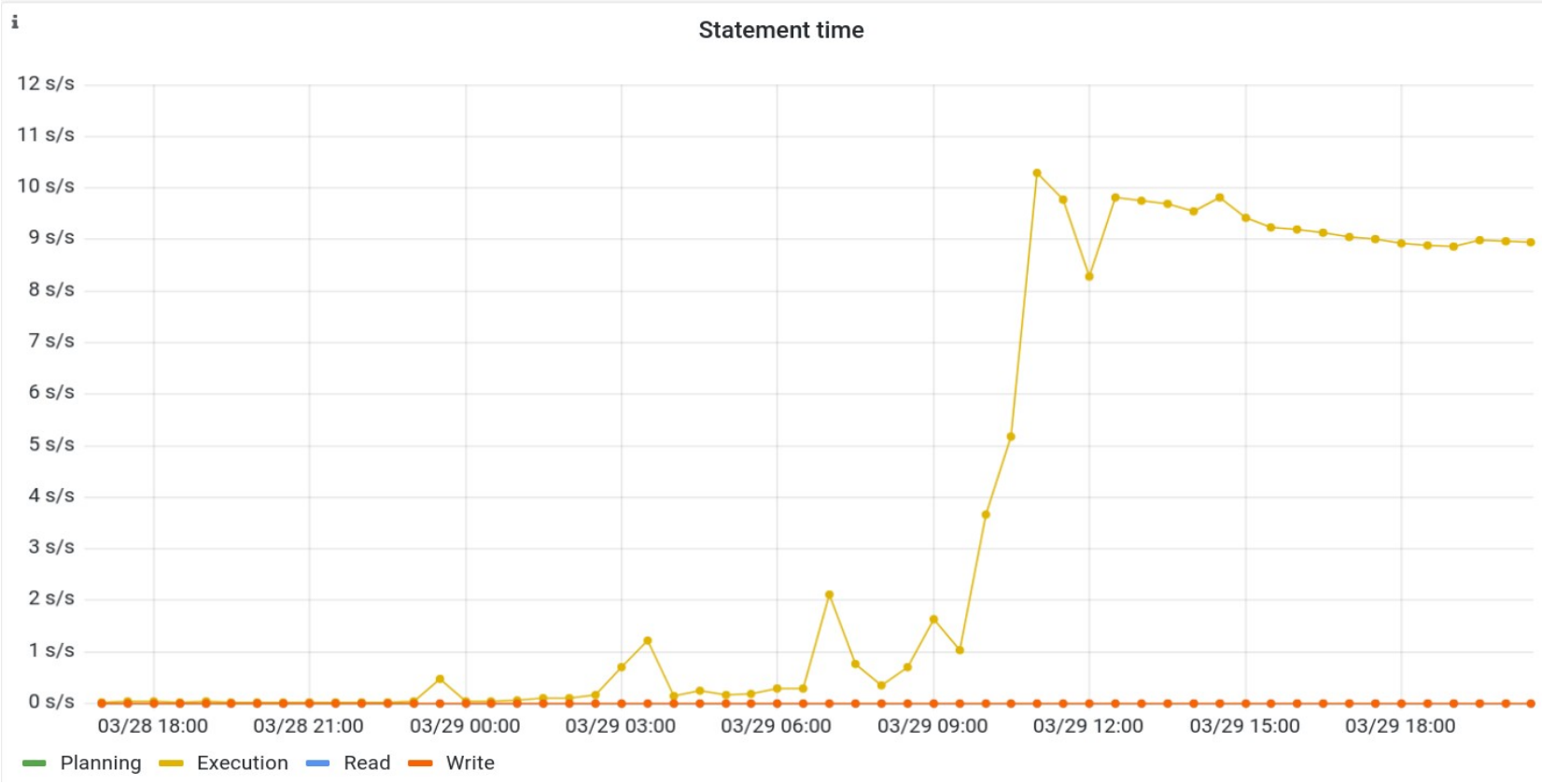
- Агрегирует данные наблюдений интервала
- Формирует JSON с данными
- Встраивает JSON в шаблон
- JS в шаблоне формирует секции

Спасибо **Евгению Шараеву**

PostgresPro

pg_profile / pgpro_pwr 4.2

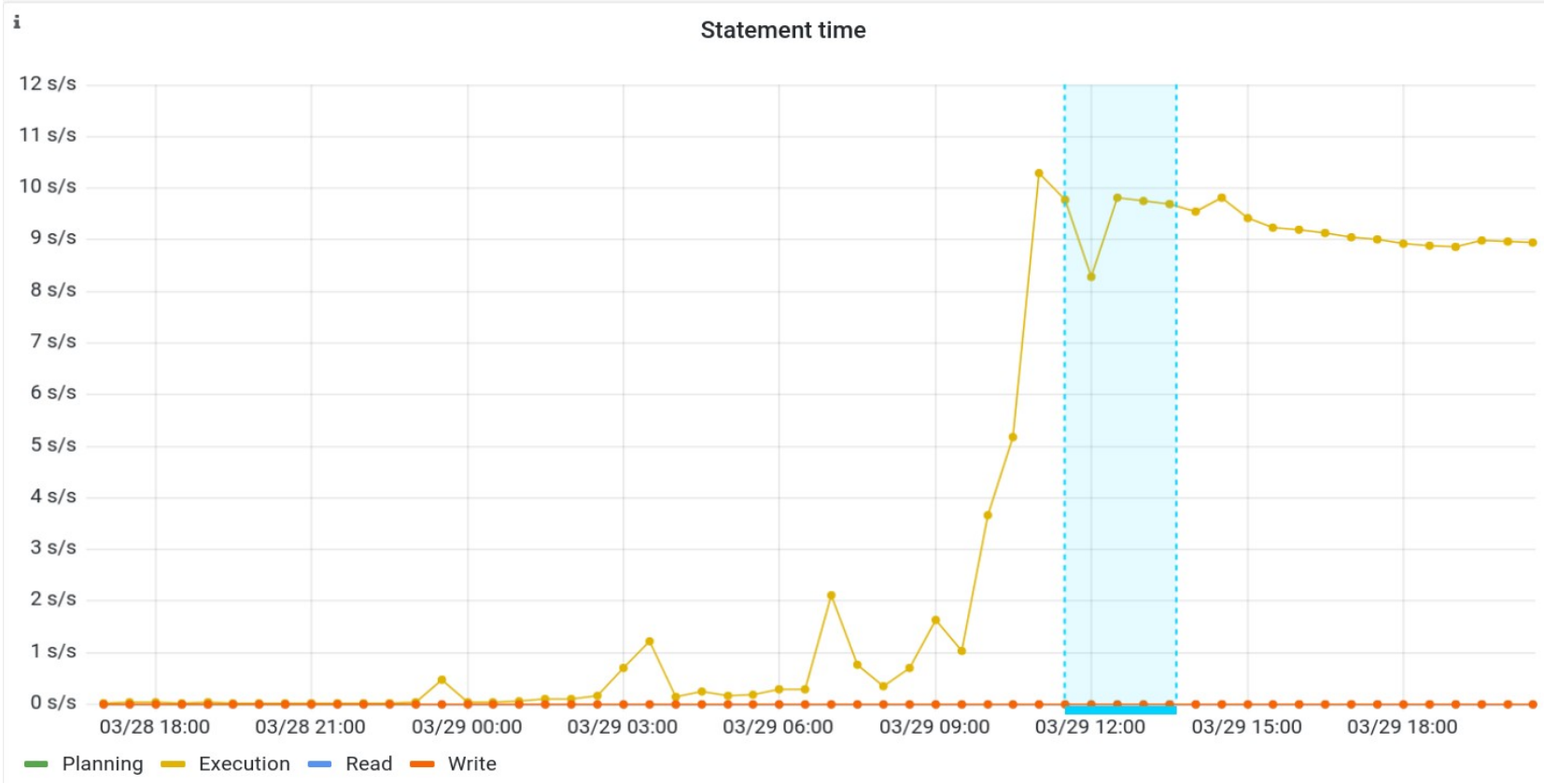
Добавим интерактивности



PostgresPro

pg_profile / pgpro_pwr 4.2

Добавим интерактивности



PostgresPro

pg_profile / pgpro_pwr 4.2

Top SQL by execution time

Query ID	Database	User	Exec (s)	%Total	Rows	Execution times (ms)				Executions
						Mean	Min	Max	StdErr	
ba551e58a1220972 [825ec9df e2]	db_1b743	user_136ac	26830. 65	26. 22	7473752	437. 744	131. 332	1268. 142	83. 563	61293
7058521854c25be5 [f c2b6ac0db]	db_1b743	user_136ac	20825. 71	20. 35	552	123. 524	33. 169	523. 835	52. 895	168597
2c3252b0a9ecf 099 [32e15bf a2b]	db_1b743	user_136ac	19553. 56	19. 11		225. 29	62. 513	725. 97	86. 111	86793
711d687f db6583af [9972b38b9c]	db_1b743	user_136ac	10303. 36	10. 07		218. 449	63. 142	628. 219	86. 247	47166
42c019f d344ccda3 [581a0cb27e]	db_1b743	user_136ac	10146. 89	9. 92		2329. 94	0. 11	4628. 985	474. 272	4355
de37a7b16ab1d9ec [476c08c031]	db_1b743	user_136ac	9594. 74	9. 38	4599	1111. 66	0. 012	4667. 248	1219. 244	8631
19858c316e39b93a [bb9daa91f 5]	db_1b743	user_136ac	1131. 97	1. 11	26980	42. 269	12. 135	261. 705	24. 96	26780

PostgresPro

pg_profile / pgpro_pwr 4.2

Top SQL by execution time

Query ID	Database	User	Exec (s)	%Total	Rows	Execution times (ms)				Executions
						Mean	Min	Max	StdErr	
ba551e58a1220972 [825ec9df e2]	db_1b743	user_136ac	26830. 65	26. 22	7473752	437. 744	131. 332	1268. 142	83. 563	61293
7058521854c25be5 [f c2b6ac0db]	db_1b743	user_136ac	20825. 71	20. 35	552	123. 524	33. 169	523. 835	52. 895	168597
2c3252b0a9ecf 099 [32e15bf a2b]	db_1b743	user_136ac	19553. 56	19. 11		225. 29	62. 513	725. 97	86. 111	86793
711d687f db6583af [9972b38b9c]	db_1b743	user_136ac	10303. 36	10. 07		218. 449	63. 142	628. 219	86. 247	47166
42c019f d344ccda3 [581a0cb27e]	db_1b743	user_136ac	10146. 89	9. 92		2329. 94	0. 11	4628. 985	474. 272	4355
de37a7b16ab1d9ec [476c08c031]	db_1b743	user_136ac	9594. 74	9. 38	4599	1111. 66	0. 012	4667. 248	1219. 244	8631
19858c316e39b93a [bb9daa91f 5]	db_1b743	user_136ac	1131. 97	1. 11	26980	42. 269	12. 135	261. 705	24. 96	26780

PostgresPro

pg_profile / pgpro_pwr 4.2

Top SQL by shared blocks fetched

Query ID	Database	User	blks fetched	%Total	Hits(%)	Elapsed(s)	Rows	Executions
42c019f d344ccda3 [581a0cb27e]	db_1b743	user_136ac	10907120944	41. 12	100	10146. 9		4355
7058521854c25be5 [f c2b6ac0db]	db_1b743	user_136ac	9443651961	35. 6	100	20825. 7	552	168597
ba551e58a1220972 [825ec9df e2]	db_1b743	user_136ac	3917979016	14. 77	100	26830. 7	7473752	61293
2c3252b0a9ecf 099 [32e15bf a2b]	db_1b743	user_136ac	1191716267	4. 49	100	19553. 6		86793
711d687f db6583af [9972b38b9c]	db_1b743	user_136ac	647187403	2. 44	100	10303. 4		47166
2e77692b5dff 5c21 [5f adf 658f 1]	db_1b743	user_136ac	45071989	0. 17	100	442. 8	1790	2012
3865b6f 15c706793 [615932b6c7]	db_1b743	user_136ac	41186701	0. 16	100	292	92797	1249

pg_profile / pgpro_pwr 4.2

Подсветка выделяет:

- Базы данных
- Табличные пространства
- Запросы
- Таблицы
- Индексы
- Функции

DB	Tablespace	Schema	Table	Size	Growth	Ins	Upd	Del	Upd(HOT)
bench	pg_default	public	grow_table	2128 kB	2128 kB	20000		10000	
			grow_table(TOAST)	79 MB	79 MB	100000		50000	
postgres	pg_default	profile	last_stat_tables_srv1	704 kB	704 kB	2014	551	1342	14
postgres	pg_default	profile	last_stat_indexes_srv1	704 kB	704 kB	2934	198	1955	4

DB	Tablespace	Schema	Table	Index	Scans	Blks	%Total
bench	pg_default	public	grow_table(TOAST)	pg_toast_551215_index	30000	261229	4. 7
bench	pg_default	public	ixbench	ixbench_pkey		220372	3. 96
postgres	pg_default	profile	last_stat_tables_srv1	pk_last_stat_tables_srv1	29390	81034	1. 46
bench	pg_default	public	grow_table	grow_table_pkey		39649	0. 71
bench	pg_default	public	grow_table	ix_grow_table	1	37677	0. 68
postgres	pg_default	pg_catalog	pg_class	pg_class_oid_index	8003	16083	0. 29

DB	Tablespace	Schema	Table	Index	Index		Table		
					Size	Growth	Ins	Upd	Del
bench	pg_default	public	grow_table(TOAST)	pg_toast_551215_index	1112 kB	1112 kB	100000		50000
bench	pg_default	public	grow_table	grow_table_pkey	240 kB	240 kB	20000		10000
postgres	pg_default	profile	last_stat_indexes_srv1	pk_last_stat_indexes_srv1	144 kB	136 kB	2934	194	1955
postgres	pg_default	profile	last_stat_tables_srv1	pk_last_stat_tables_srv1	144 kB	128 kB	2014	537	1342
bench	pg_default	public	grow_table	ix_grow_table	88 kB	88 kB	20000		10000
postgres	pg_default	profile	last_stat_statements_srv1	pk_last_stat_satements_srv1	48 kB	32 kB	242	334	157

Обсудим

- Визуализацию наблюдений pg_profile/pgpro_pwr
- Интерактивные отчеты pg_profile/pgpro_pwr
- **Расширенные статистики вакуума 2.0**
- Время архивирования
- Трассировка в СУБД PostgresPro

PostgresPro

Статистики вакуума в PostgresPro

- Накопительные ресурсные статистики
- На уровне отдельных отношений
- Доступны через pgpro_stats

PostgresPro

Статистики вакуума в PostgresPro

Особенности настройки

- Вакуум настраивается индивидуально на каждой таблице
- Вакуум пропускает all_visible и all_frozen
- Индексы сканируются целоком (не всегда, но иногда по нескольку раз)
- Агрессивный вакуум может снизить bloat
- Агрессивный вакуум увеличит нагрузку
- Агрессивный вакуум равномернее распределит нагрузку
- Нет удобных количественных критериев эффективности

Статистики вакуума в PostgresPro

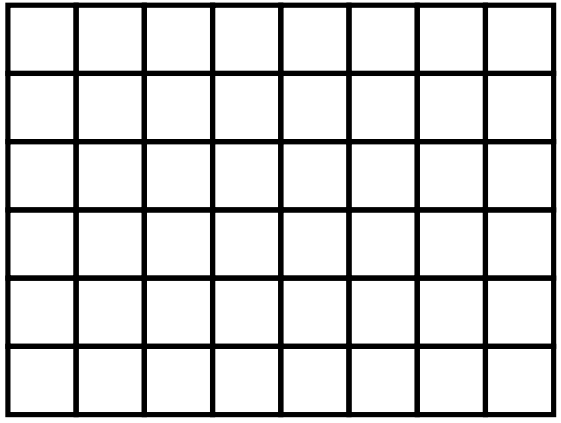
```
View "profile.pgpro_stats_vacuum_tables"
       Column          |       Type
-----------------------+-----------------
 relid                 | oid
 schema                | name
 relname               | name
 total_blks_read       | bigint
 total_blks_hit        | bigint
 total_blks_dirtied    | bigint
 total_blks_written    | bigint
 rel_blks_read         | bigint
 rel_blks_hit          | bigint
 pages_scanned         | bigint
 pages_removed         | bigint
 pages_frozen          | bigint
 pages_all_visible     | bigint
 tuples_deleted        | bigint
 tuples_frozen         | bigint
 dead_tuples           | bigint
 index_vacuum_count    | integer
 wal_records           | bigint
 wal_fpi               | bigint
 wal_bytes             | numeric
 blk_read_time         | double precision
 blk_write_time        | double precision
 delay_time            | double precision
 system_time           | double precision
 user_time             | double precision
 total_time            | double precision
 interrupts            | integer
```

```
View "profile.pgpro_stats_vacuum_indexes"
       Column          |       Type
-----------------------+-----------------
 relid                 | oid
 schema                | name
 relname               | name
 total_blks_read       | bigint
 total_blks_hit        | bigint
 total_blks_dirtied    | bigint
 total_blks_written    | bigint
 rel_blks_read         | bigint
 rel_blks_hit          | bigint
 pages_deleted         | bigint
 tuples_deleted        | bigint
 wal_records           | bigint
 wal_fpi               | bigint
 wal_bytes             | numeric
 blk_read_time         | double precision
 blk_write_time        | double precision
 delay_time            | double precision
 system_time           | double precision
 user_time             | double precision
 total_time            | double precision
 interrupts            | integer
```

Статистики вакуума в PostgresPro

- VM пуста

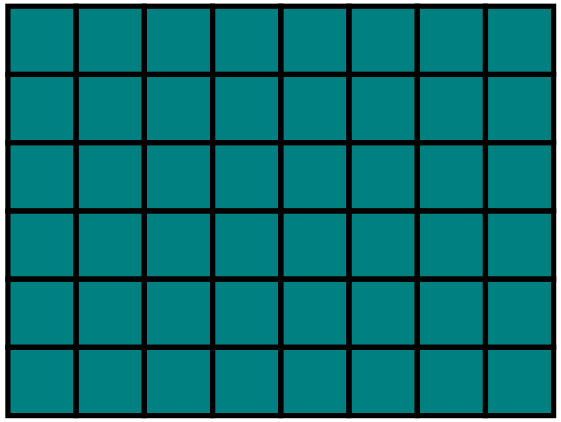
Table VM



Статистики вакуума в PostgresPro

- VM пуста
- vacuum

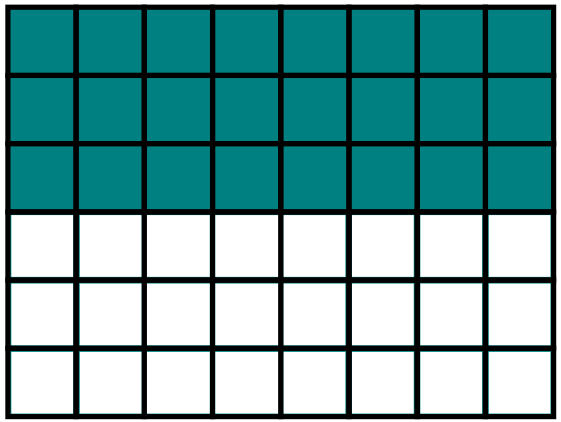
Table VM



Статистики вакуума в PostgresPro

- VM пуста
- vacuum
- DML

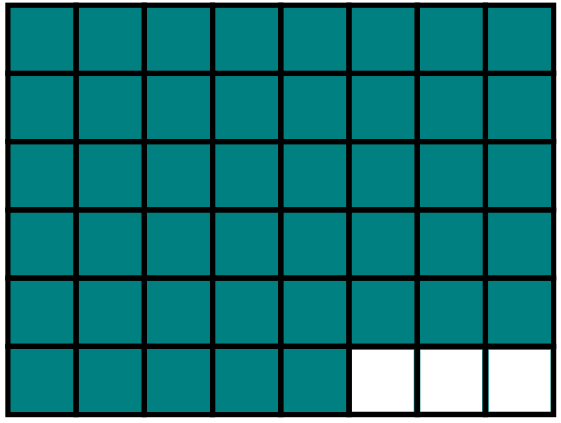
Table VM



Статистики вакуума в PostgresPro

- VM пуста
- vacuum
- DML
- vacuum

Table VM



Разморозка

```
View "profile.pgpro_stats_vacuum_tables"
        Column           |       Type
-------------------------+------------------
 relid                   | oid
 schema                  | name
 relname                 | name
 pages_frozen            | bigint
 pages_all_visible       | bigint
 rev_all_frozen_pages    | bigint
 rev_all_visible_pages   | bigint
```

PostgresPro

Статистики вакуума в pgpro_pwr (с 4.1.2)

Top tables by removed all-visible marks

DB	Tablespace	Schema	Table	All-Visible marks			Vacuum count	
				Cleared	Set	%Set	Vacuum	Autovacuum
bench	pg_default	public	ixbench	4931	5000	50. 35	11	1
postgres	pg_default	pg_catalog	pg_statistic	217	252	53. 73		2
contrib_regression	pg_default	pg_catalog	pg_statistic	20	255	92. 73		2
postgres	pg_default	profile	last_stat_statements_srv1	9	23	71. 88		2
contrib_regression	pg_default	pg_catalog	pg_class	9	22	70. 97		1
bench	pg_default	public	pgbench_tellers	8	8	50	5	1
postgres	pg_default	pg_toast	pg_toast_2619 (pg_statistic toast)	7	20	74. 07		1

PostgresPro

Статистики вакуума в pgpro_pwr (с 4.1.2)

Top tables by removed all-frozen marks

DB	Tablespace	Schema	Table	All-Frozen marks			Vacuum count	
				Cleared	Set	%Set	Vacuum	Autovacuum
postgres	pg_default	pg_catalog	pg_statistic	67	87	56. 49		2
contrib_regression	pg_default	pg_catalog	pg_statistic	10	226	95. 76		2
contrib_regression	pg_default	pg_catalog	pg_class	9	22	70. 97		1
postgres	pg_default	pg_toast	pg_toast_2619 (pg_statistic toast)	4	11	73. 33		1
contrib_regression	pg_default	pg_catalog	pg_index	4	8	66. 67		1
contrib_regression	pg_default	pg_toast	pg_toast_2619 (pg_statistic toast)	3	20	86. 96		1
postgres	pg_default	profile	last_stat_statements_srv1	1	7	87. 5		2

Обсудим

- Визуализацию наблюдений pg_profile/pgpro_pwr
- Интерактивные отчеты pg_profile/pgpro_pwr
- Расширенные статистики вакуума 2.0
- **Время архивирования**
- Трассировка в СУБД PostgresPro

PostgresPro

Время архивирования

```
postgres=# select * from pgpro_stats_archiver ;
-[ RECORD 1 ]--------+------------------------------
archived_count       | 3
last_archived_wal    | 000000010000000000000000B
last_archived_time   | 2023-04-03 19:37:13.446028+00
failed_count         | 0
last_failed_wal      |
last_failed_time     |
active_time          | 1002
archive_command_time | 655
stats_reset          | 2023-04-03 19:35:33.740743+00
```

Обсудим

- Визуализацию наблюдений pg_profile/pgpro_pwr
- Интерактивные отчеты pg_profile/pgpro_pwr
- Расширенные статистики вакуума 2.0
- Время архивирования
- **Трассировка в СУБД PostgresPro**

PostgresPro

Трассировка PostgresPro

Удобный механизм для журналирования запросов и планов

Query ID	Database	User	Exec (s)	%Total	Rows	Execution times (ms)				Executions
						Mean	Min	Max	StdErr	
ba551e58a1220972 [825ec9df e2]	db_1b743	user_136ac	96915. 54	25. 1	25998620	57. 22	0. 005	1367. 838	148. 422	1693727
7058521854c25be5 [f c2b6ac0db]	db_1b743	user_136ac	72874. 6	18. 87	5703	33. 814	0. 003	523. 835	58. 666	2155189
2c3252b0a9ecf 099 [32e15bf a2b]	db_1b743	user_136ac	69274. 21	17. 94		51. 459	0. 003	725. 97	97. 696	1346211

Например для нестабильных запросов

Или для отслеживания действий таинственного приложения

PostgresPro

Трассировка PostgresPro: Постановка задачи

Что есть сейчас?

- log_min_duration_statement
- auto_explain.log_min_duration

Что будет в pgpro_stats:

- Возможность настраивать гибкие избирательные критерии трассировки
- Запись в индивидуальные файлы

PostgresPro

Трассировка PostgresPro: Критерии трассировки

- Индивидуальное исполнение запроса будет записано при удовлетворении любого критерия трассировки
- Критерий — набор фильтров, выполняющихся вместе
- Фильтр — индивидуальный показатель в составе критерия

Трассировка PostgresPro: Фильтры

Базовые:

- `pid`
- База данных
- Хост (`client_addr`)
- Приложение (`application_name`)
- Пользователь (`username`)
- `queryid`
- `planid`

Ресурсные (по превышению):

- `duration`, `plan/exec/cpu_time`
- `shared_blks/local_blks`:
  `hit`, `read`, `fetch`, `dirt`, `writ`
- `temp_blks`:
  `read`, `written`
- `wal_bytes`
- `total_wait_time`
- `total_inval_msgs`

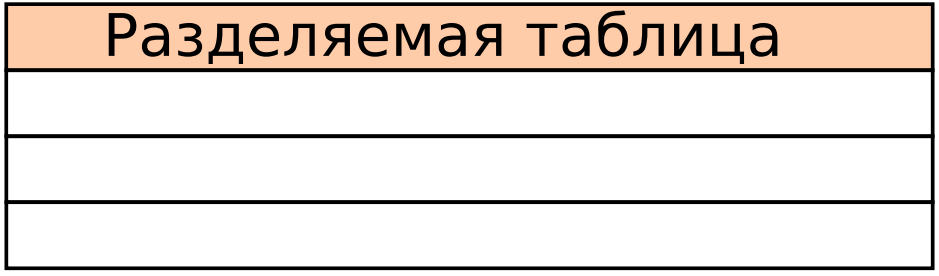
Трассировка PostgresPro: Параметры критериев

- `id` — идентификатор
- `active`
- `alias` — имя критерия
- `logfile` — имя файла в `PGDATA/pg_stats/`

Опции explain (boolean):

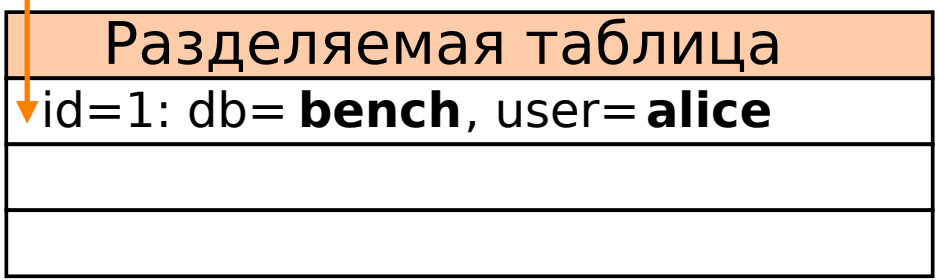
- `explain_analyze`
- `explain_verbose`
- `explain_costs`
- `explain_settings`
- `explain_buffers`
- `explain_wal`
- `explain_timing`
- `explain_format`

Трассировка PostgresPro: Архитектура



Трассировка PostgresPro: Архитектура

```
SELECT pgpro_stats_trace_insert(
  'datname', 'bench',
  'username', 'alice'
);
```

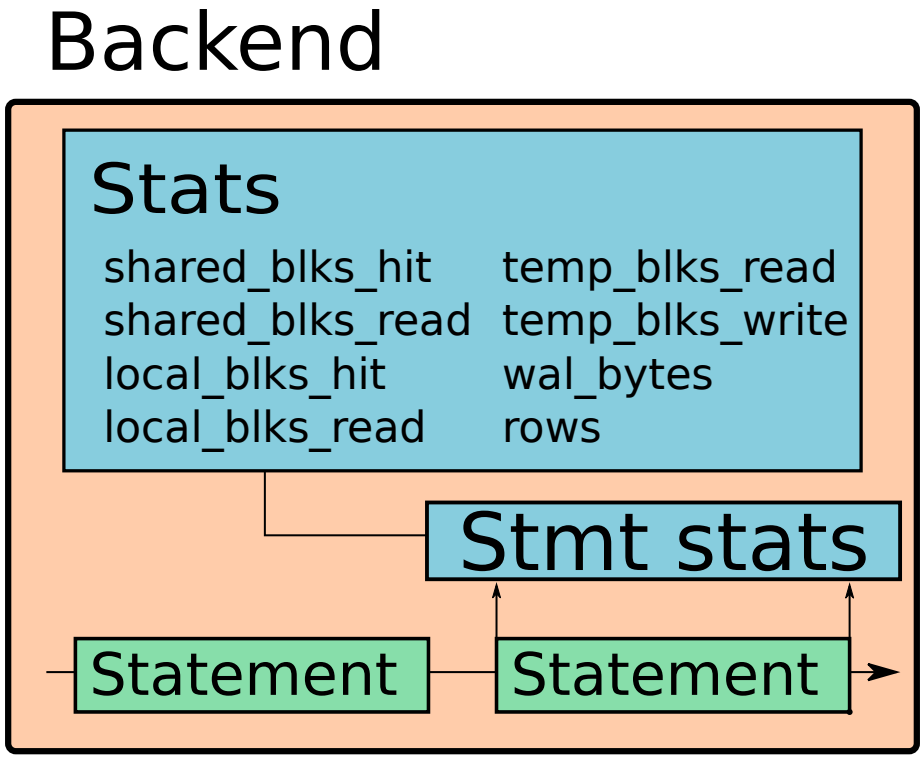


Трассировка PostgresPro: Архитектура

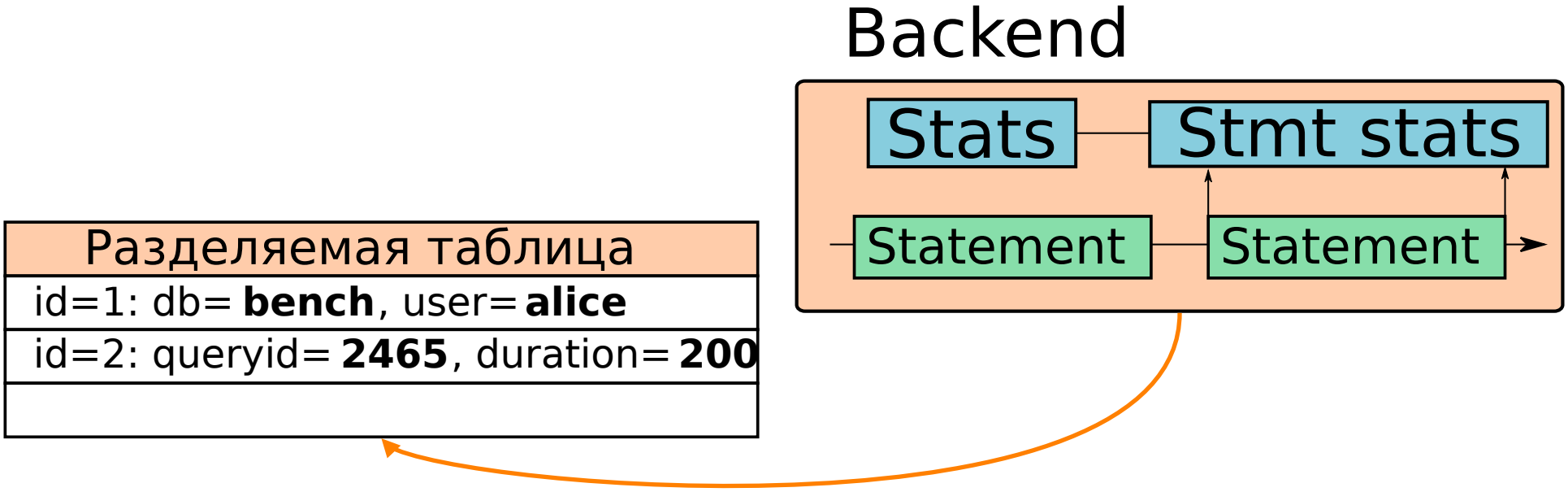
```
SELECT pgpro_stats_trace_insert(
  'queryid', '2465',
  'duration', '200'
);
```

Разделяемая таблица
id=1: db= **bench**, user= **alice**
id=2: queryid= **2465**, duration= **200**

Трассировка PostgresPro: Архитектура



Трассировка PostgresPro: Архитектура



PostgresPro

Трассировка PostgresPro

В трассировку попадает текст запроса и план:

```
Query Text: select
f.arrival_airport,
count(*)
from
flights f join ticket_flights tf USING(flight_id)
group by f.arrival_airport;
Finalize GroupAggregate  (cost=40280.47..40306.81 rows=104 width=12)
  Group Key: f.arrival_airport
  ->  Gather Merge  (cost=40280.47..40304.73 rows=208 width=12)
        Workers Planned: 2
        ->  Sort  (cost=39280.44..39280.70 rows=104 width=12)
              Sort Key: f.arrival_airport
              ->  Partial HashAggregate  (cost=39273.32..39274.36 rows=104 width=12)
                    Group Key: f.arrival_airport
                    ->  Hash Join  (cost=2269.44..34355.95 rows=983473 width=4)
                          Hash Cond: (tf.flight_id = f.flight_id)
                          ->  Parallel Seq Scan on ticket_flights tf  (cost=0.00..29504.73
rows=983473 width=4)
                          ->  Hash  (cost=1448.64..1448.64 rows=65664 width=8)
                                Buckets: 131072  Batches: 1  Memory Usage: 3589kB
                                ->  Seq Scan on flights f  (cost=0.00..1448.64 rows=65664 width=8)
```

PostgresPro

Трассировка PostgresPro: производительность

- Наибольшее влияние на короткие запросы (+ 100-200 мс.)
- Влияние на нетрассируемые запросы незначительно
- Зависит от избирательности базовых фильтров

PGConf.Russia 2023

PostgresPro

Спасибо!

Андрей Зубков
Постгрес Профессиональный