PGConf.Russia 2023

PostgresPro

Стратегический мониторинг СУБД PostgreSQL

pg_profile и pgpro_pwr

Андрей Зубков

PostgresPro

Введение

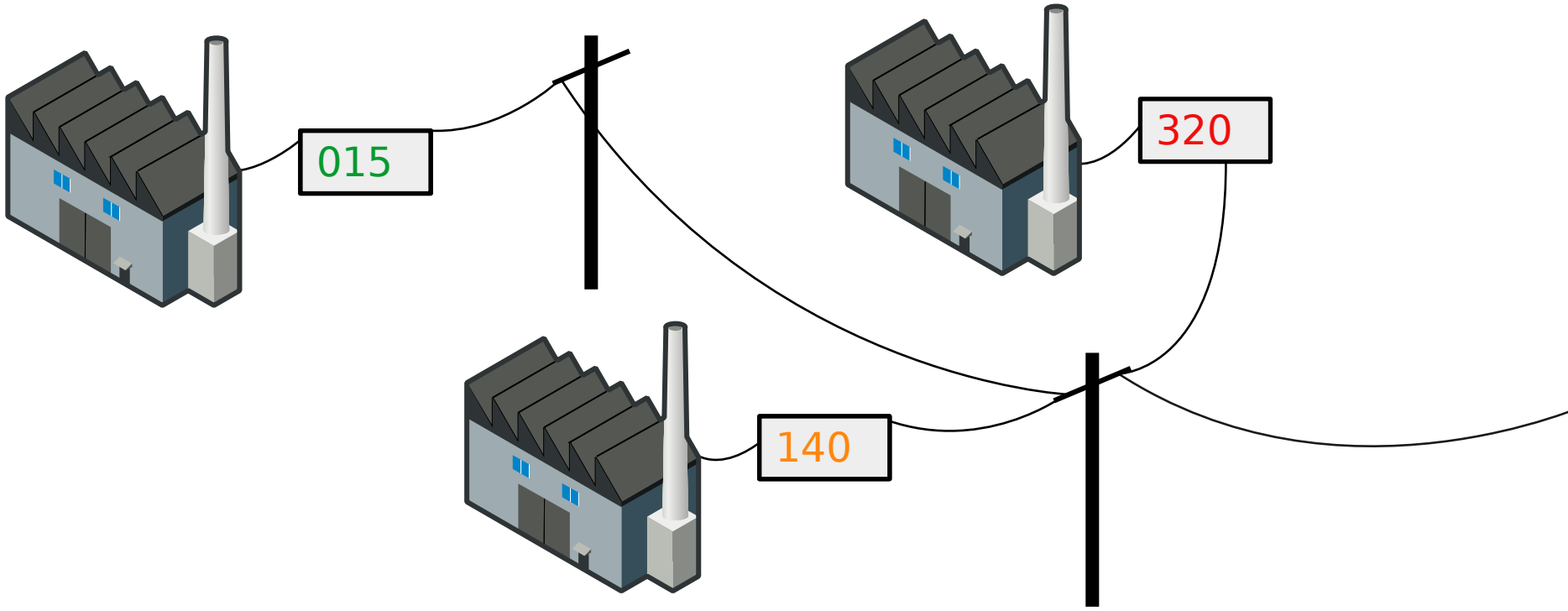
Ресурсоёмкие активности и объекты

- Локализация снижения производительности
- Расследование инцидентов
- Удовлетворение любопытства

Чинить всё будет человек, стратегический мониторинг лишь подскажет откуда начать

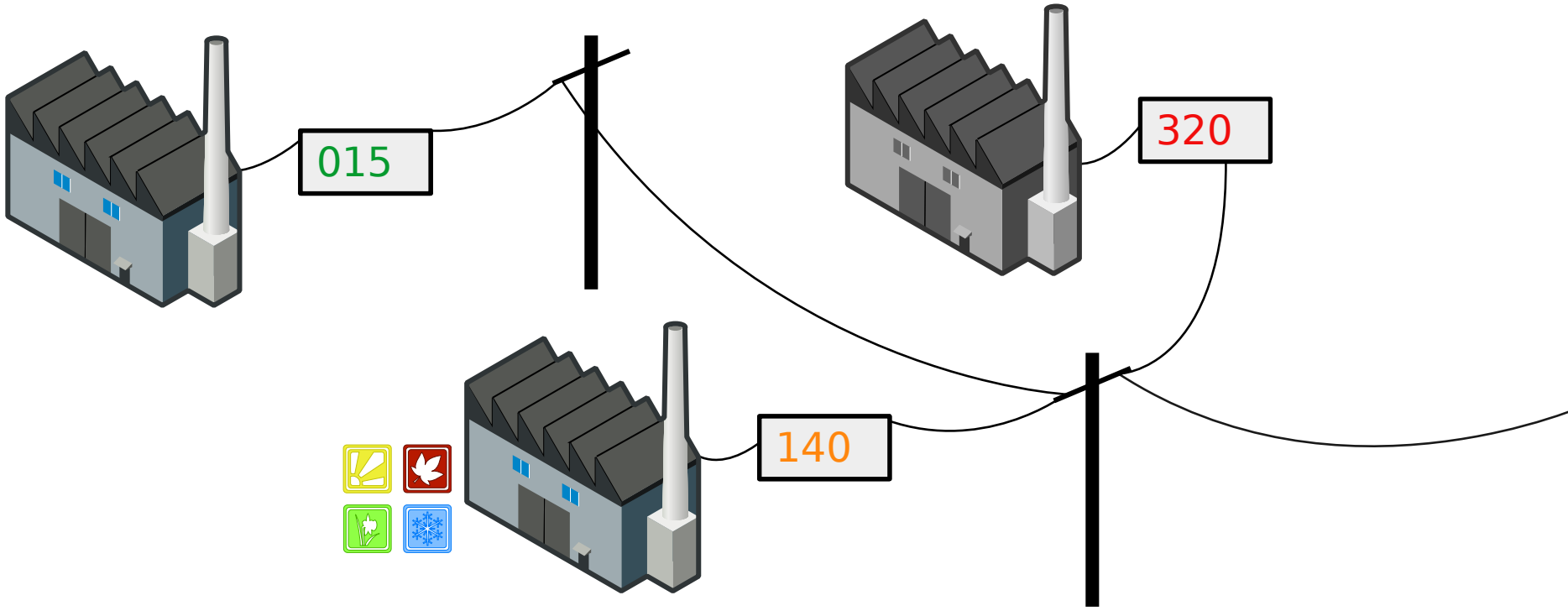
PostgresPro

Ресурсоёмкие активности



PostgresPro

Ресурсоёмкие активности



PostgresPro

The Cumulative Statistics System

- Учитывает множество ресурсов
- В отношении множества объектов
- Похоже на счетчики воды и электроэнергии

PostgresPro

The Cumulative Statistics System

```
postgres=# \d data
                  Table "profile.data"
   Column   |  Type  | Collation | Nullable | Default
------------+--------+-----------+----------+---------
 section_id | bigint |           |          |
 row_data   | json   |           |          |
```

PostgresPro

The Cumulative Statistics System

```
postgres=# SELECT
  relname,seq_scan,seq_tup_read,heap_blks_read,
  heap_blks_hit,toast_blks_read,toast_blks_hit
FROM pg_stat_all_tables join pg_statio_all_tables
  using (relid,schemaname,relname)
WHERE relname='data';

-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 87
seq_tup_read    | 292494
heap_blks_read  | 384
heap_blks_hit   | 16704
toast_blks_read | 1
toast_blks_hit  | 6
```

The Cumulative Statistics System

```
-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 87
seq_tup_read    | 292494
heap_blks_read  | 384
heap_blks_hit   | 16704
toast_blks_read | 1
toast_blks_hit  | 6
```

PostgresPro

The Cumulative Statistics System

```
postgres=# SELECT count(*) FROM data;
-[ RECORD 1 ]
count | 3362
```

```
-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 88      (+1)
seq_tup_read    | 295856  (+3362)
heap_blks_read  | 384
heap_blks_hit   | 16896   (+192)
toast_blks_read | 1
toast_blks_hit  | 6
```

PostgresPro

The Cumulative Statistics System

```
postgres=# SELECT count(*) FROM data;
-[ RECORD 1 ]
count | 3362

postgres=# SELECT sum(length(row_data::text)) FROM data;
-[ RECORD 1 ]
sum | 1408526
```

```
-[ RECORD 1 ]---+-------
relname         | data
seq_scan        | 88      (+1)
seq_tup_read    | 295856  (+3362)
heap_blks_read  | 384
heap_blks_hit   | 16896   (+192)
toast_blks_read | 2       (+1)
toast_blks_hit  | 6
```

The Cumulative Statistics System

- pg_stat_database
- pg_stat_tablespaces
- pg_stat_bgwriter
- pg_stat_archiever
- pg_stat_wal
- *pg_stat_statements*
- *pg_stat_kcache*
- pg_stat_all_tables
- pg_stat_all_indexes
- pg_statio_all_tables
- pg_statio_all_indexes
- pg_stat_user_functions
- pg_settings
- pgpro_stats

PostgresPro

Исходные требования

- Периодичность наблюдений
- Простое получение результатов в удобном виде
- Минимум дополнительных средств
- Хранить данные только ресурсозатратных объектов

PostgresPro

Архитектура pg_profile/pgpro_pwr

Расширение на *pl/pgsql*, состоящее из:

- Таблиц репозитория для хранения данных снимков
- Функций сбора данных для снимков
- Функций построения отчетов
- Служебных таблиц и функций

PostgresPro

Процедура установки

pg_profile:

- Скачать релизный архив с github
- Распаковать:

```
tar xzf pg_profile-X.X.tar.gz \
  --directory $(pg_config --sharedir)/extension
```

- CREATE EXTENSION pg_profile SCHEMA profile;

pgpro_pwr устанавливается пакетным менеджером

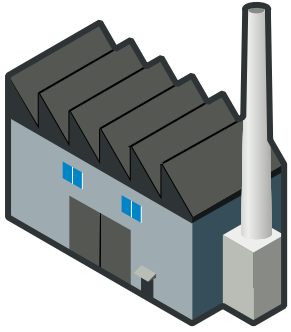
PostgresPro

Первый снимок

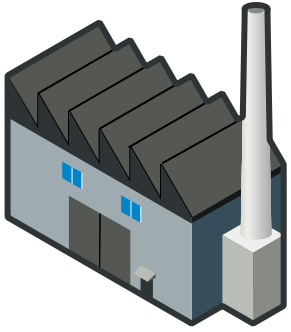
```
postgres=# SELECT * FROM profile.take_sample();
 server | result |   elapsed
--------+--------+-------------
 local  | OK     | 00:00:06.01
(1 row)

postgres=# select show_samples();
           show_samples
-------------------------------------
 (1,"2023-02-21 08:34:03+00",t,,,)
(1 row)
```

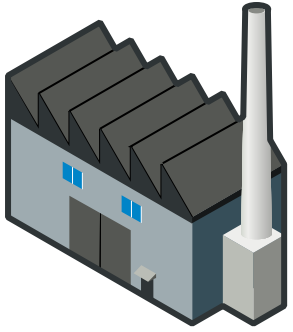
Что в снимке?



320



332



345

Sample 1

320

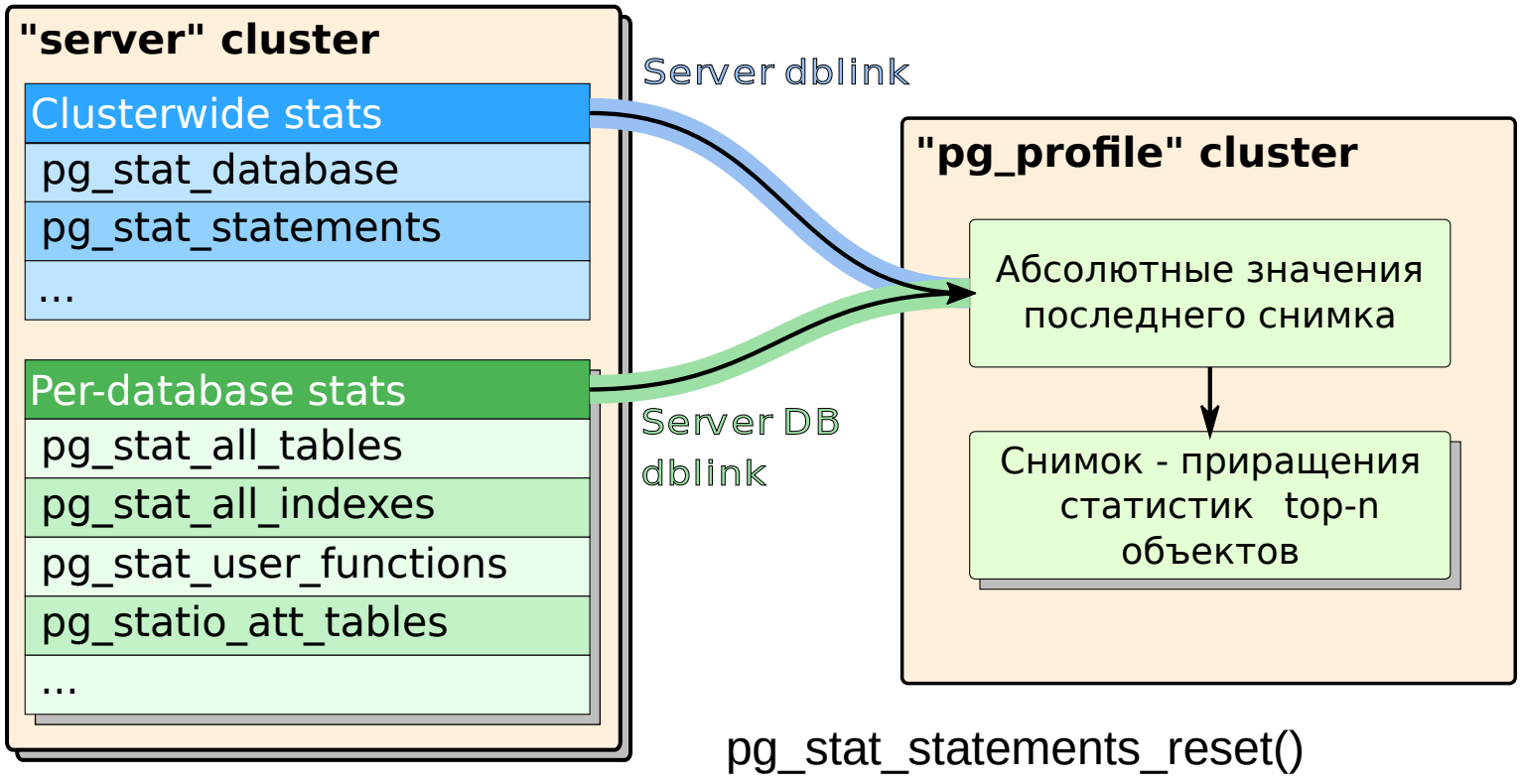
Sample 2

12

Sample 3

13

Снимок


"server" cluster
Clusterwide stats
pg_stat_database
pg_stat_statements
...
Per-database stats
pg_stat_all_tables
pg_stat_all_indexes
pg_stat_user_functions
pg_statio_att_tables
...
Server dblink
Server DB dblink
"pg_profile" cluster
Абсолютные значения последнего снимка
Снимок - приращения статистик top-n объектов
pg_stat_statements_reset()

PostgresPro

Создание сервера

```
create_server(
    server name,
    server_connstr text,
    server_enabled boolean = TRUE,
    max_sample_age integer = NULL,
    description text = NULL)
```

- *server_name* — имя сервера
- *server_connstr* — строка подключения (база с pg_stat_statements)
- *server_enabled* — включение в общий take_sample()
- *max_sample_age* — политика сохранения
- *description* — описание (включается в отчет)

PostgresPro

Изменение сервера

- `drop_server(server)`
  - удаление сервера
- `rename_server(server, new_name)`
  - переименование сервера
- `set_server_description(server, description)`
  - описание сервера (будет включено в отчёт)

PostgresPro

Изменение сервера

- `set_server_max_sample_age(server name, max_sample_age integer)`
- `set_server_connstr(server name, new_connstr text)`
- `show_servers()`
- `enable_server(server)`
- `disable_server(server)`

PostgresPro

Сбор снимков с серверов

- `take_sample()`

  - выполняет снимки на всех enabled серверах

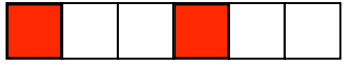
- `take_sample(server)`

  - явное выполнение снимка на сервере

- `take_sample_subset(sets_cnt, current_set)`

  - подмножество enabled серверов

  `take_sample_subset(3,0)`



PostgresPro

Сбор снимков с серверов

- `take_sample()`

  - выполняет снимки на всех enabled серверах

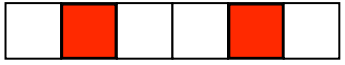
- `take_sample(server)`

  - явное выполнение снимка на сервере

- `take_sample_subset(sets_cnt, current_set)`

  - подмножество enabled серверов

  `take_sample_subset(3,1)`



PostgresPro

Сбор снимков с серверов

- `take_sample()`

  - выполняет снимки на всех enabled серверах

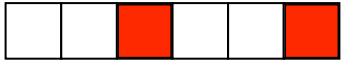
- `take_sample(server)`

  - явное выполнение снимка на сервере

- `take_sample_subset(sets_cnt, current_set)`

  - подмножество enabled серверов

  `take_sample_subset(3,2)`



PostgresPro

Политика хранения

В порядке возрастания приоритета:

- Общий параметр `pg_profile.max_sample_age`
- `max_sample_age` для сервера
- *Baseline* — параметр `days`

Устаревшие снимки удаляются при выполнении нового снимка

PostgresPro

Сбор размеров отношений

Есть два способа измерений:

- pg_class.relpages — собирается всегда
- pg_relation_size() — собирается по умолчанию

Точные размеры можно собирать редко (на уровне сервера):

- *window_start* — время начала суточного окна измерения размеров
- *window_duration* — протяженность окна во времени
- *sample_interval* — минимальный интервал между снимками с размерами

```
take_sample(server name, skip_sizes boolean)
```

PostgresPro

Baseline

Baseline — непрерывная именованная последовательность снимков

- Обладает собственной политикой хранения
- Может использоваться как интервал для отчета
- Служит для сохранения эталонной нагрузки

PostgresPro

Baseline

Создание:

- `create_baseline([server,] baseline_name, start_id, end_id[, days integer])`
- `create_baseline([server,] baseline_name, time_range tstzrange [, days integer])`

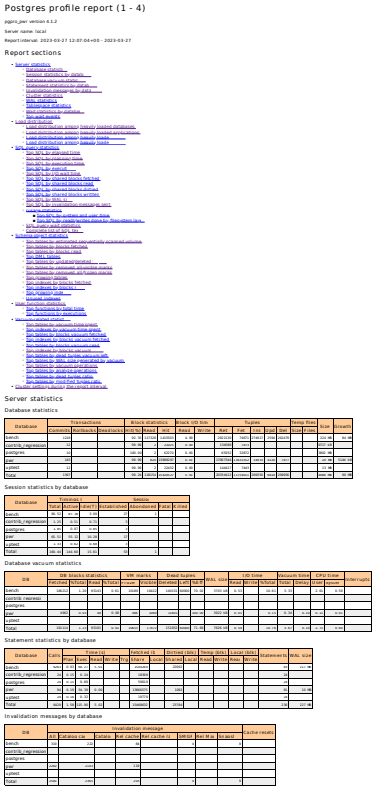
Удаление:

- `drop_baseline([server,] baseline_name)`

Отчеты

Отчет — html документ, содержащий суммарную статистику за требуемый* интервал

* с точностью до одного снимка



Postgres profile report (1 - 4)

pgpro_pwr version 4.1.2

Server name: local

Report interval: 2023-03-27 12:07:04+00 - 2023-03-27

Report sections

Server statistics

Database statistics

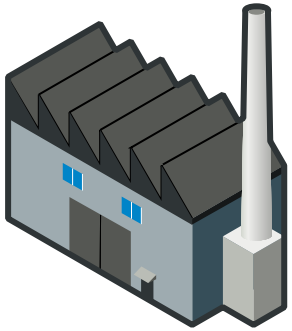
Session statistics by database

Database vacuum statistics

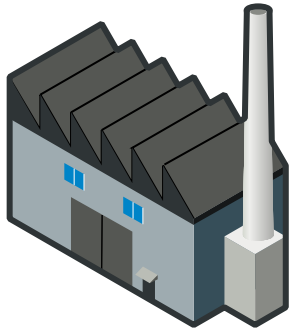
Statement statistics by database

Invalidation messages by database

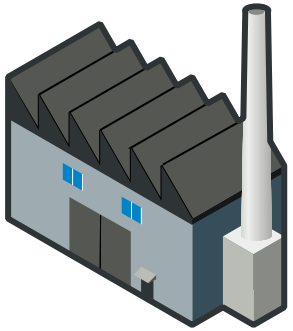
Отчеты



320



332



345

Снимок 1

320

Снимок 2

12

Снимок 3

13

Отчет с 1 по 3 снимки покажет 25 (12 + 13)

PostgresPro

Построение отчета

- get_report([server,] start_id, end_id [,description] [, with_growth])
- get_report([server,] time_range *tstzrange* [,description] [, with_growth])
- get_report([server,] baseline_name [,description] [, with_growth])

```
[postgres@pg15 run]$ psql -Aqtc "select profile.get_report(1,2)" \
 -o report_1_2.html
```

Содержание отчета

Содержание отчета

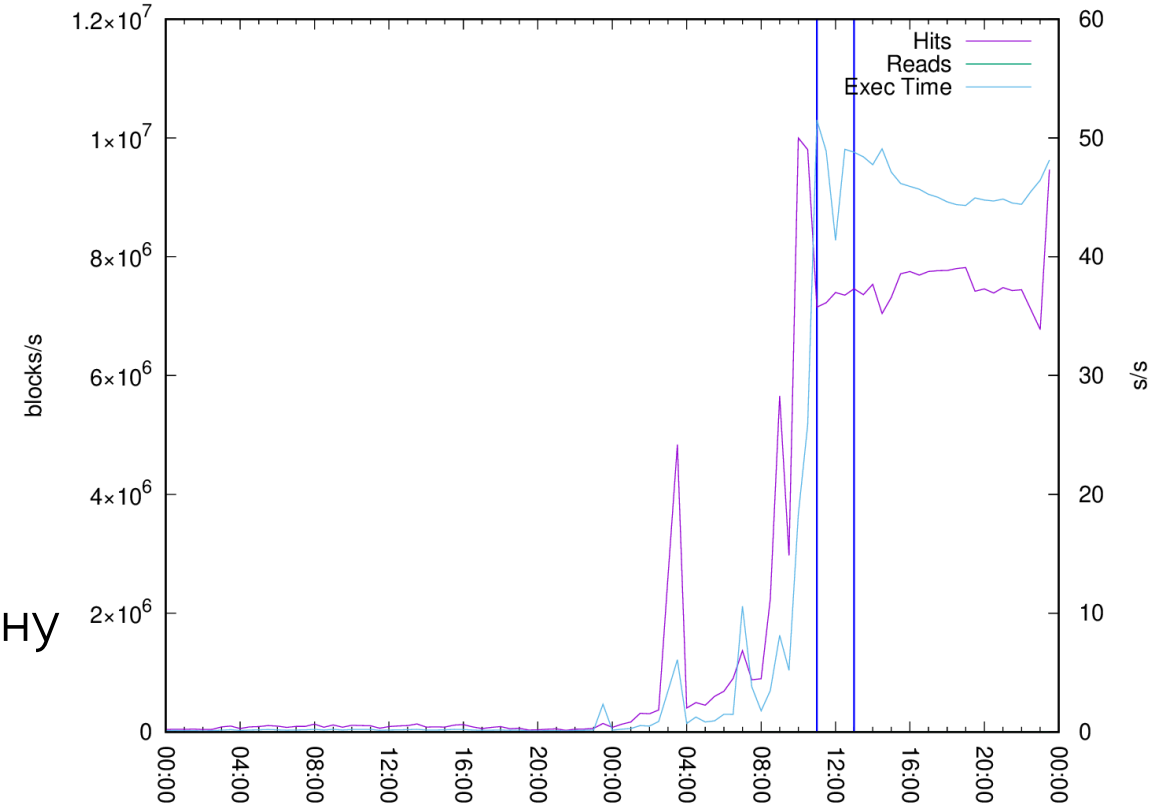
Содержание отчета

PostgresPro

Пример инцидента

Объективные данные наблюдений:

- Существенная деградация производительности
- Выраженное изменение нагрузки



Мы наблюдаем признаки проблемы и хотим понять причину

Искать их надо в отчете по интервалу с наблюдаемыми признаками

PostgresPro

Top SQL by execution time

Query ID	Database	Exec (s)	%Total	Rows	Execution times (ms)				Executions
					Mean	Min	Max	StdErr	
825ec9df e2 [ba551e58a1220972]	demodb	17677. 69	26. 10	4976275	436. 777	131. 332	1268. 142	87. 207	40473
f c2b6ac0db [7058521854c25be5]	demodb	13814. 61	20. 40	435	124. 600	33. 169	523. 835	53. 414	110872
32e15bf a2b [2c3252b0a9ecf 099]	demodb	12796. 92	18. 90	65%	224. 436	62. 513	725. 970	85. 991	57018
9972b38b9c [711d687f db6583af]	demodb	6819. 08	10. 07	75%	216. 334	63. 142	628. 219	86. 583	31521
581a0cb27e [42c019f d344ccda3]	demodb	6763. 17	9. 99		2318. 536	0. 110	4348. 005	515. 938	2917
476c08c031 [de37a7b16ab1d9ec]	demodb	6257. 31	9. 24	3100	1098. 931	0. 012	4284. 943	1233. 238	5694
bb9daa91f 5 [19858c316e39b93a]	demodb	820. 60	1. 21	19459	42. 600	12. 135	261. 705	25. 354	19263

PostgresPro

Top SQL by shared blocks fetched

Query ID	Database	blks fetched	%Total	Hits(%)	Elapsed(s)	Rows	Executions
581a0cb27e [42c019fd344ccda3]	demodb	7294930558	41.42	100.00	6763.2		2917
fc2b6ac0db [7058521854c25be5]	demodb	6232122854	35.38	100.00	13814.6	435	110872
825ec9dfe2 [ba551e58a1220972]	demodb	2583010863	14.66	100.00	17677.7	4976275	40473
32e15bfa2b [2c3252b0a9ecf099]	demodb	774440330	4.40	100.00	12796.9		57018
9972b38b9c [711d687fdb6583af]	demodb	427932206	2.43	100.00	6819.1		31521
615932b6c7 [3865b6f15c706793]	demodb	33263701	0.19	100.00	225.8	77099	962
5fadf658f1 [2e77692b5dff5c21]	demodb	33045697	0.19	100.00	304.9	1268	1381

91%

PostgresPro

Top tables by blocks fetched

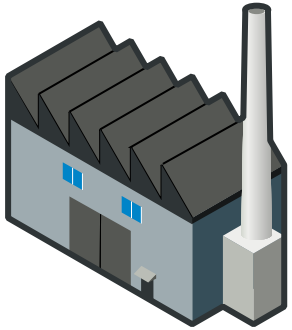
DB	Tablespace	Schema	Table	Heap		Ix		TOAST		TOAST-Ix	
				Blks	%Total	Blks	%Total	Blks	%Total	Blks	%Total
demodb	pg_default	i6c	i6_n_m	9481392382	53.64	357341693	2.02				
demodb	pg_default	i6c	i6_d_t_d	1656550069	9.37	4975941917	28.15				
demodb	pg_default	i6c	i6_d_t_m	696556409	3.94	31248149	0.18	4	0.00	66	0.00
demodb	pg_default	i6c	i6_n_as	91667256	0.52	4863695	0.03				

PostgresPro

Top indexes by blocks fetched

DB	Tablespace	Schema	Table	Index	Scans	Blks	%Total
demodb	pg_default	i6c	i6_d_t_d	i6_d_t_d_pk	1654718992	4975805807	28. 15
demodb	pg_default	i6c	i6_n_m	i6_n_m_ix1	91626609	322193350	1. 82
demodb	pg_default	i6c	i6_n_m	i6_n_m_ix2	88555	34801154	0. 20
demodb	pg_default	i6c	i6_d_u	i6_d_u_uk	8529766	25684839	0. 15

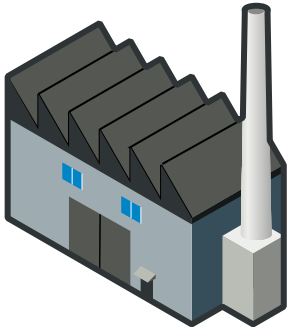
Сравнительный отчет



320

Снимок 1

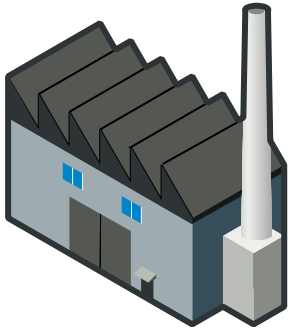
320



332

Снимок 2

12



345

Снимок 3

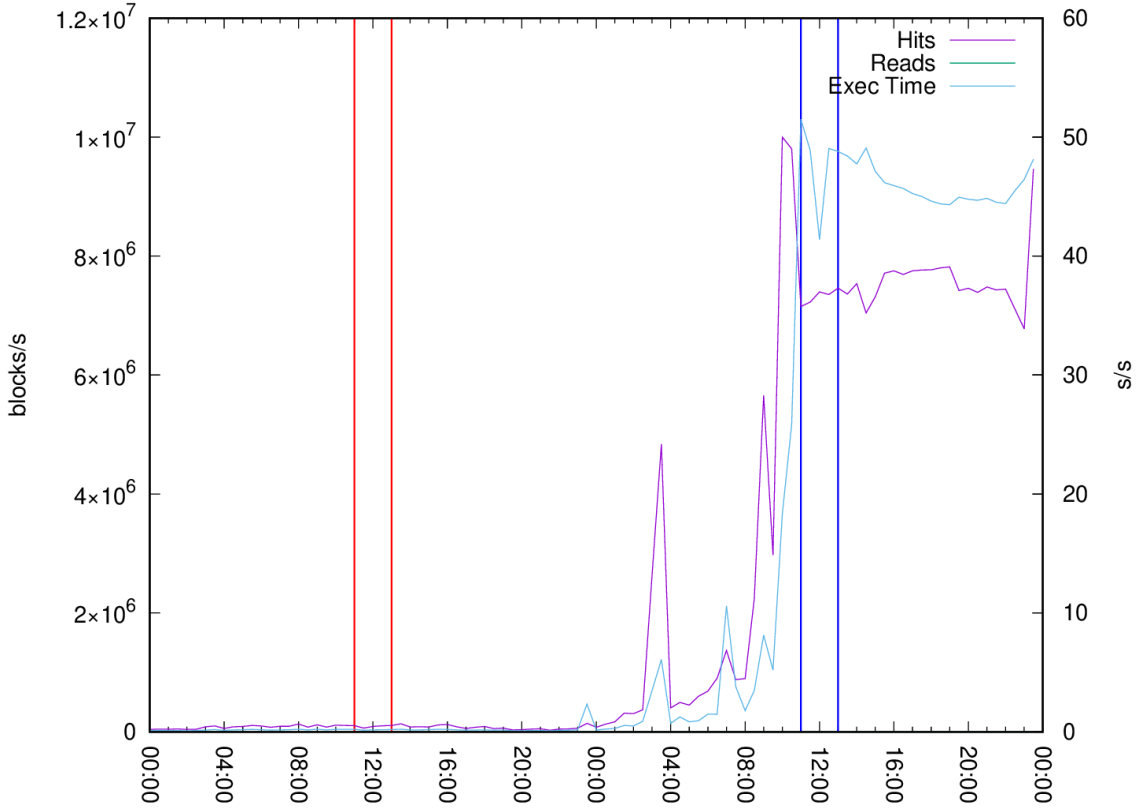
13

Сравнительный отчет для интервалов 1 (1-2) и 2 (2-3)

Object	I	Watt
Fact 1	1	12
	2	13

PostgresPro

«Вчера всё работало!»



PostgresPro

Построение сравнительного отчета

- `get_diffreport([server,] start1_id, end1_id, start2_id, end2_id [,description] [, with_growth])`
- `get_diffreport([server,] time_range1 tstzrange, time_range2 tstzrange, [,description] [, with_growth])`
- `get_diffreport([server,] baseline_name1, baseline_name2, [,description] [, with_growth])`
- \+ Различные комбинации

Database statistics

Database	I	Transactions			Block statistics			Tuples					Temp files		Size	Growth
		Commits	Rollbacks	Deadlocks	Hit(%)	Read	Hit	Ret	Fet	Ins	Upd	Del	Size	Files		
demodb	1	1554854	1804		99.79	461906	218000570	701175479	90047284	27420	88226	9603			29 GB	7512 kB
	2	3439169	7307		99.99	1615331	17662792017	61934924526	25927080482	225683	357037	48921			29 GB	53 MB
mdb	1	284			100.00		25876	162460	8712						7901 kB	
	2	284			100.00		24812	161980	8232						7901 kB	
postgres	1	2963			99.90	475	495350	1563478	89189	33269	736	32028			22 MB	296 kB
	2	2970			99.95	658	1459023	5186650	437424	33524	747	32063			27 MB	368 kB
Total	1	1558101	1804		99.79	462381	218521796	702901417	90145185	60689	88962	41631			29 GB	7808 kB
	2	3442423	7307		99.99	1615989	17664275852	61940273156	25927526138	259207	357784	80984			29 GB	53 MB

PostgresPro

Top SQL by execution time

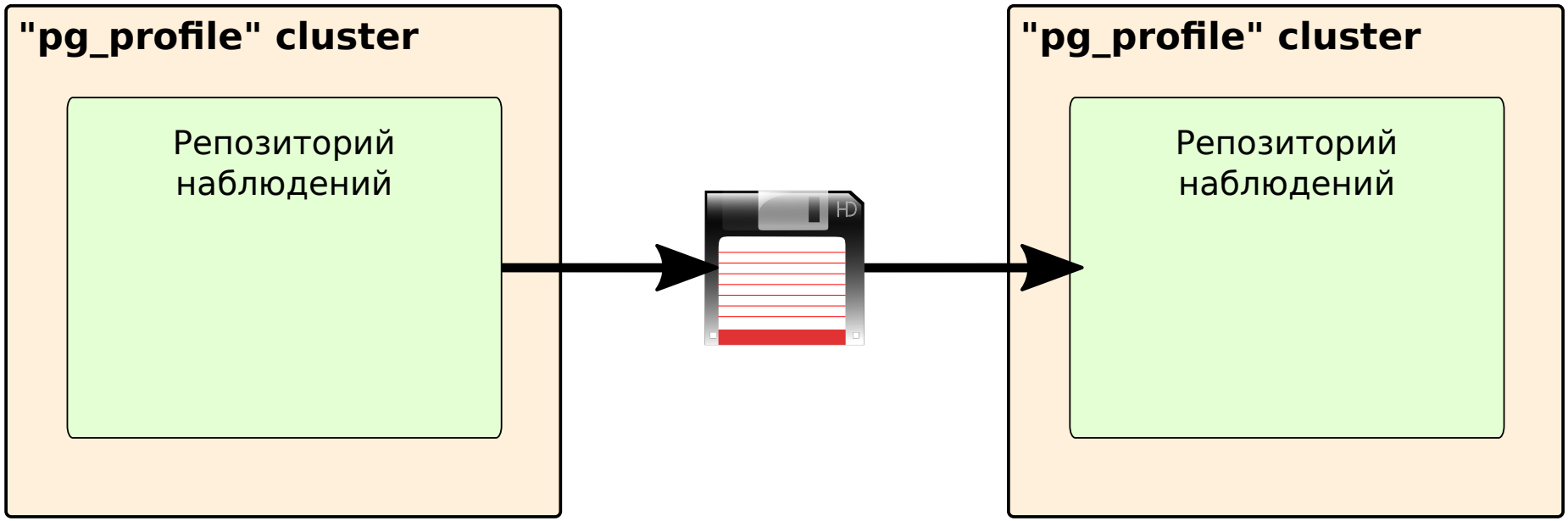
Query ID	Database	I	Exec (s)	%Total	Rows	Execution times (ms)				Executions
						Mean	Min	Max	StdErr	
825ec9df e2 [ba551e58a1220972]	demodb	1	1. 85	0. 76	347	0. 011	0. 005	9. 809	0. 046	171201
		2	17677. 69	26. 10	4976275	436. 777	131. 332	1268. 142	87. 207	40473
f c2b6ac0db [7058521854c25be5]	demodb	1	1. 74	0. 71	135	0. 010	0. 003	11. 883	0. 067	172062
		2	13814. 61	20. 40	435	124. 600	33. 169	523. 835	53. 414	110872
32e15bf a2b [2c3252b0a9ecf 099]	demodb	1	1. 08	0. 44		0. 009	0. 003	9. 954	0. 062	114724
		2	12796. 92	18. 90		224. 436	62. 513	725. 970	85. 991	57018
581a0cb27e [42c019f d344ccda3]	demodb	1	62. 76	25. 76		965. 550	870. 881	1085. 988	58. 479	65
		2	6763. 17	9. 99		2318. 536	0. 110	4348. 005	515. 938	2917
9972b38b9c [711d687f db6583af]	demodb	1	0. 61	0. 25		0. 010	0. 003	7. 502	0. 057	58624
		2	6819. 08	10. 07		216. 334	63. 142	628. 219	86. 583	31521
476c08c031 [de37a7b16ab1d9ec]	demodb	1								
		2	6257. 31	9. 24	3100	1098. 931	0. 012	4284. 943	1233. 238	5694

PostgresPro

Top tables by blocks fetched

DB	Tablespace	Schema	Table	I	Heap		Ix		TOAST		TOAST-Ix	
					Blks	%Total	Blks	%Total	Blks	%Total	Blks	%Total
demodb	pg_default	i6c	i6_n_m	1	536961	0. 25	2418551	1. 10				
				2	9481392382	53. 64	357341693	2. 02				
demodb	pg_default	i6c	i6_d_t_d	1	36214123	16. 53	108835089	49. 68				
				2	1656550069	9. 37	4975941917	28. 15				
demodb	pg_default	i6c	i6_d_t_m	1	17166613	7. 84	1239130	0. 57				
				2	696556409	3. 94	31248149	0. 18	4	0. 00	66	0. 00
demodb	pg_default	i6c	i6_n_as	1	574752	0. 26	692931	0. 32				
				2	91667256	0. 52	4863695	0. 03				

Перенос данных



PostgresPro

Экспорт

Табличная функция:

- `export_data([server, min_sample_id, max_sample_id, obfuscate_queries])`

PostgresPro

Экспорт

```
postgres=# \copy (select * from export_data()) to 'export.csv'
COPY 64136
```

PostgresPro

Экспорт

```
postgres=# \copy (select * from export_data()) to 'export.csv'
COPY 64136

postgres=# \copy (SELECT * FROM export_data()) to program
   'gzip > export.csv.gz'
COPY 64136
```

PostgresPro

Импорт

```
postgres=# CREATE TABLE import (section_id bigint, row_data json);
CREATE TABLE
```

PostgresPro

Импорт

```
postgres=# CREATE TABLE import (section_id bigint, row_data json);
CREATE TABLE

postgres=# \copy import from 'export.csv'
COPY 64136

postgres=# \copy import from program 'zcat export.csv.gz'
COPY 64136
```

PostgresPro

Импорт

```
postgres=# CREATE TABLE import (section_id bigint, row_data json);
CREATE TABLE

postgres=# \copy import from 'export.csv'
COPY 64136

postgres=# \copy import from program 'zcat export.csv.gz'
COPY 64136

[vacuum, analyze, freeze]

postgres=# SELECT * FROM import_data('import');
```

PostgresPro

Импорт

- Сервера импортируются в *disabled*
- Возможны конфликты (`'local' = 'local'`)

```
import_data(
    data regclass,
    server_name_prefix text)
```

- Поддерживается «догрузка»

PostgresPro

Дополнительные возможности pgpro_pwr

- Статистики выражений на уровне планов
- Статистики ожиданий
- Расширенные статистики очистки
- Распределение нагрузки
- Статистики инвалидаций

PostgresPro

Статистики выражений на уровне планов

Query ID	Plan ID	Database	User	Executions	%Total	Rows	Mean(ms)	Min(ms)	Max(ms)	StdErr(ms)	Elapsed(s)
3f8e23e0397789a4 [aea97c96d0]	9a87b6430c6a983f	bench	postgres	857	13.33	857	5.496	0.167	57.934	7.423	4.7
57772371ebfee860 [166ff d72e0]	ee59889426be44bd	bench	postgres	857	13.33	857	0.138	0.069	0.415	0.033	0.1
83bdfc341819c5c1 [8814020b04]	5a3629823d809a70	bench	postgres	857	13.33	857	0.115	0.054	0.304	0.031	0.1
bbeaf99b17933e69 [ab99808a96]	f2a17e1b5e7b85d5	bench	postgres	857	13.33	857	0.084	0.043	0.294	0.027	0.1
6a65499fac19262e [e6ca685e5a]	696114fbe759f402	bench	postgres	857	13.33	857	0.057	0.028	0.952	0.052	0.0
939c2f56e1f6a174 [a10a30e56f]		bench	postgres	857	13.33		0.014	0.007	0.114	0.009	0.0

Статистики ожиданий

IO wait event type						
Query ID	Plan ID	Database	User	Waited (s)	%Total	Details
3f8e23e0397789a4 [aea97c96d0]	9a87b6430c6a983f	bench	postgres	4.52	61.41	DataFileRead: 4.52
53906b4fab355166 [6c5463efce]	60e283bc47225a89	bench	postgres	0.68	9.24	DataFileExtend: 0.68
8410930711dd1989 [5fe5a1952e]		bench	postgres	0.39	5.30	DataFileRead: 0.39
f6939cc6b6a1b5dc [f220160d00]		bench	postgres	0.23	3.13	DataFileImmediateSync: 0.23
48664d4e5310f566 [dc312b21f4]		bench	postgres	0.21	2.85	WALSync: 0.18 DataFileTruncate: 0.03

PostgresPro

Распределение нагрузки

Resource	Load distribution
Total time (sec.)	psql: 77.05 | load app: 21.68 | pgbench: 13. | pg | t | t
Executed count	pgbench: 3637 | benchapp2: 1440 | benchapp1: 1132 | pg
I/O time (sec.)	pgbench: 3.03 | benchapp1: 1.07 | benchapp2: 0.89
Blocks fetched	psql: 14259951 | pgben | l | p
Shared blocks read	pgbench: 95751
Shared blocks dirtied	psql: 11495 | load app: 10193 | pgbench | t
Shared blocks written	psql: 11001 | load app: 10189 | pg
WAL generated	psql: 101 MB | load app: 85 MB | pgbench: 38 MB
Temp and Local blocks written	
Temp and Local blocks read	
Invalidation messages sent	psql: 2440 | pgber
Cache resets	

PostgresPro

Расширенные статистики очистки

Database vacuum statistics

DB	DB blocks statistics				VM marks		Dead tuples			WAL size	I/O time			Vacuum time		CPU time		Interrupts
	Fetched	%Total	Read	%Total	Frozen	Visible	Deleted	Left	%Eff		Read	Write	%Total	Total	Delay	User	System	
bench	186212	1. 20	95143	0. 61	10189	16412	140151	60000	70. 02	3703 kB	0. 53		10. 61	3. 33		2. 61	0. 59	
contrib_regression																		
postgres																		
pwr	4902	0. 03	48	0. 00	466	1060	11801		100. 00	3922 kB	0. 01		0. 15	0. 34	0. 10	0. 11	0. 01	
uptest																		
Total	191114	1. 23	95191	0. 61	10655	17472	151952	60000	71. 69	7626 kB	0. 54		10. 76	3. 67	0. 10	2. 71	0. 60	

PostgresPro

Расширенные статистики очистки

Top tables by vacuum time spent

DB	Tablespace	Schema	Table	Vacuum time		I/O time		CPU time		Vacuum count		Fetched		Scanned
				Total	Delay	Read	Write	User	System	Vacuum	Autovacuum	Total	Heap	
bench	pg_default	public	ixbench	2.84		0.53		2.31	0.54	11		122821	19183	7098
bench	pg_default	pg_toast	pg_toast_624036 (grow_table toast)	0.39		0.00		0.25	0.05	9		61726	60466	20150
bench	pg_default	public	grow_table	0.09		0.00		0.04	0.00	9		1417	696	228
pwr	pg_default	pg_catalog	pg_statistic	0.06	0.01	0.00		0.01	0.00		2	857	808	284
pwr	pg_default	pg_catalog	pg_trigger	0.05				0.00	0.00		1	141	56	15
pwr	pg_default	pg_catalog	pg_attribute	0.05	0.03			0.02	0.00		1	592	323	125
pwr	pg_default	pg_catalog	pg_proc	0.05	0.04	0.00		0.01	0.00		1	638	560	223

PostgresPro

Расширенные статистики очистки

Top indexes by vacuum time spent

DB	Tablespace	Schema	Table	Index	Vacuum time		I/O time		CPU time		Vacuum count (table)		Fetched	
					Total	Delay	Read	Write	User	System	Vacuum	Autovacuum	Total	Index
bench	pg_default	public	ixbench	ixbench_pkey	2. 38		0. 53		1. 85	0. 53	11		122821	103587
bench	pg_default	pg_toast	pg_toast_624036 (grow_table toast)	pg_toast_624036_index	0. 02		0. 00		0. 02	0. 00	9		61726	1107
pwr	pg_default	pg_catalog	pg_depend	pg_depend_depender_index	0. 01	0. 00	0. 00		0. 00	0. 00		1	452	152
pwr	pg_default	pg_catalog	pg_attribute	pg_attribute_relid_attnam_index	0. 01	0. 00			0. 01	0. 00		1	592	155
bench	pg_default	public	grow_table	grow_table_pkey	0. 01		0. 00		0. 01	0. 00	9		1417	235
pwr	pg_default	pg_catalog	pg_attribute	pg_attribute_relid_attnum_index	0. 01	0. 00			0. 00	0. 00		1	592	103
pwr	pg_default	pg_catalog	pg_depend	pg_depend_reference_index	0. 01	0. 00	0. 00		0. 00	0. 00		1	452	110
bench	pg_default	public	grow_table	ix_grow_table	0. 01		0. 00		0. 01	0. 00	9		1417	83
pwr	pg_default	pg_catalog	pg_statistic	pg_statistic_relid_att_inh_index	0. 01	0. 00	0. 00		0. 00	0. 00		2	857	39

PostgresPro

Расширенные статистики очистки

Что там еще с очисткой

- Top tables by vacuum time spent
- Top indexes by vacuum time spent
- Top tables by blocks vacuum fetched
- Top indexes by blocks vacuum fetched
- Top tables by blocks vacuum read
- Top indexes by blocks vacuum read
- Top tables by dead tuples vacuum left
- Top tables by WAL size generated by vacuum

PostgresPro

Invalidation messages by database

DB	Invalidation messages sent								Cache resets
	All	Catalog cache	Catalog	Rel cache	Rel cache (all)	SMGR	Rel Map	Snapshot	
bench	319	222		84		4		9	
contrib_regression									
postgres									
pwr	2262	2143		119					
uptest									
Total	2581	2365		203		4		9	

PostgresPro

Top SQL by invalidation messages sent

Query ID	Plan ID	Database	User	Invalidation messages sent								
				All	%Total	Catalog cache	Catalog	Rel cache	Rel cache (all)	SMGR	Rel Map	Snapshot
f04c4ba125945b63 [03eb8793ed]	bbe8973af504983e	pwr	postgres	2262	87.64	2143		119				
9dce1ab20005a8c [ca83efb885]		bench	postgres	111	4.30	91		13				7
8410930711dd1989 [5fe5a1952e]		bench	postgres	96	3.72	64		32				
48664d4e5310f566 [dc312b21f4]		bench	postgres	23	0.89	14		7		2		
e27327334db76a96 [f9f5f11404]		bench	postgres	21	0.81	10		11				

PGConf.Russia 2023

PostgresPro

Спасибо!

Андрей Зубков
Постгрес Профессиональный