Утилита стратегического мониторинга PostgreSQL — **pg_profile**

Андрей Зубков
Postgres Professional

Задачи стратегического мониторинга

Поиск ресурсоёмких активностей и объектов

- Локализация снижения производительности
- Поиск направлений оптимизации
- Расследование инцидентов
- Анализ нагрузочного тестирования
- Удовлетворение любопытства

Доступные средства

Инкрементальные статистики:

- The Cumulative Statistics System
- pg_stat_statements + pg_stat_kcache
- pg_wait_sampling
- pgpro_stats (PostgresPro)

The Cumulative Statistics System

Система инкрементальных статистик

- Учитывает множество ресурсов
- В отношении множества объектов

The Cumulative Statistics System

- pg_stat_database
- pg_stat_tablespaces
- pg_stat_bgwriter
- pg_stat_archiever
- pg_stat_wal
- pg_stat_io
- *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

Требования

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

Архитектура решения

Расширение [*pl/pgsql*], состоящее из:

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

Процедура установки

1. Скачать релизный архив с github
2. Распаковать:

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

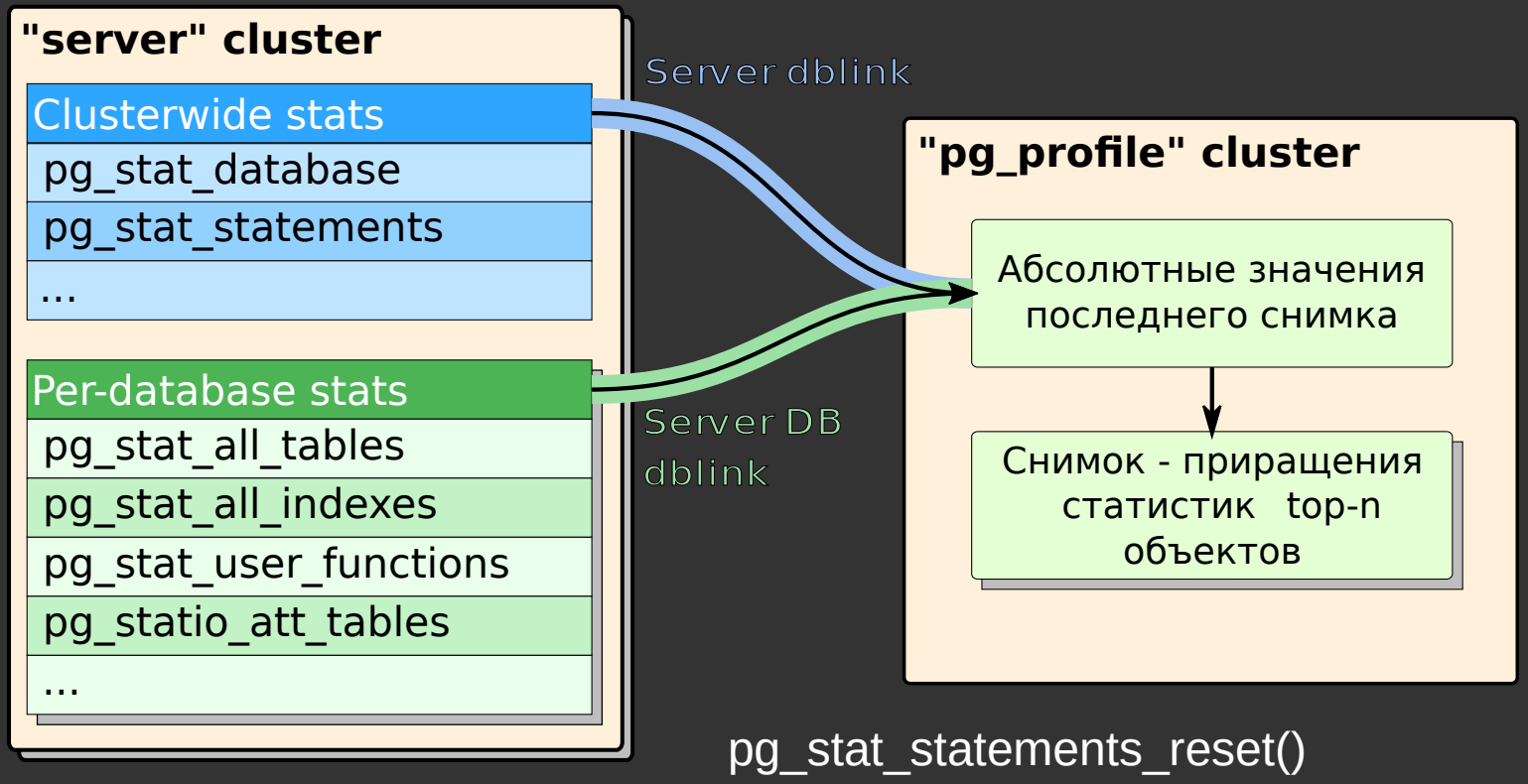
3. CREATE EXTENSION pg_profile SCHEMA profile;

Первый снимок

```
postgres=# SELECT * FROM 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)
```

Выполнение снимка



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)`

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)`

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)`

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

Baseline

Непрерывная именованная последовательность снимков

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

Baseline

Создание:

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

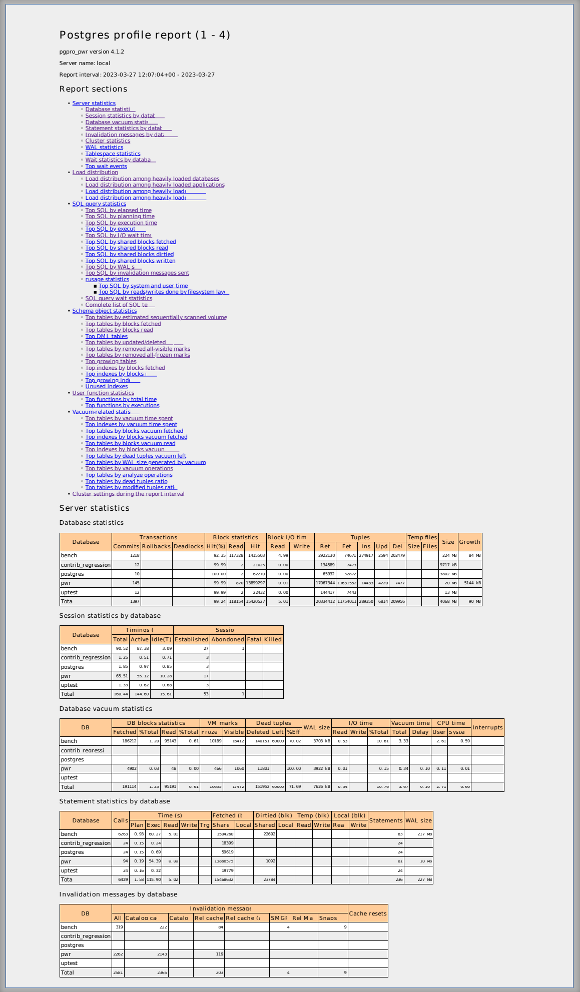
Удаление:

- `drop_baseline([server,] baseline_name)`

Отчёты

Отчет – html файл, 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

Построение отчёта

- 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 отчёта

Содержание отчёта

Содержание отчёта

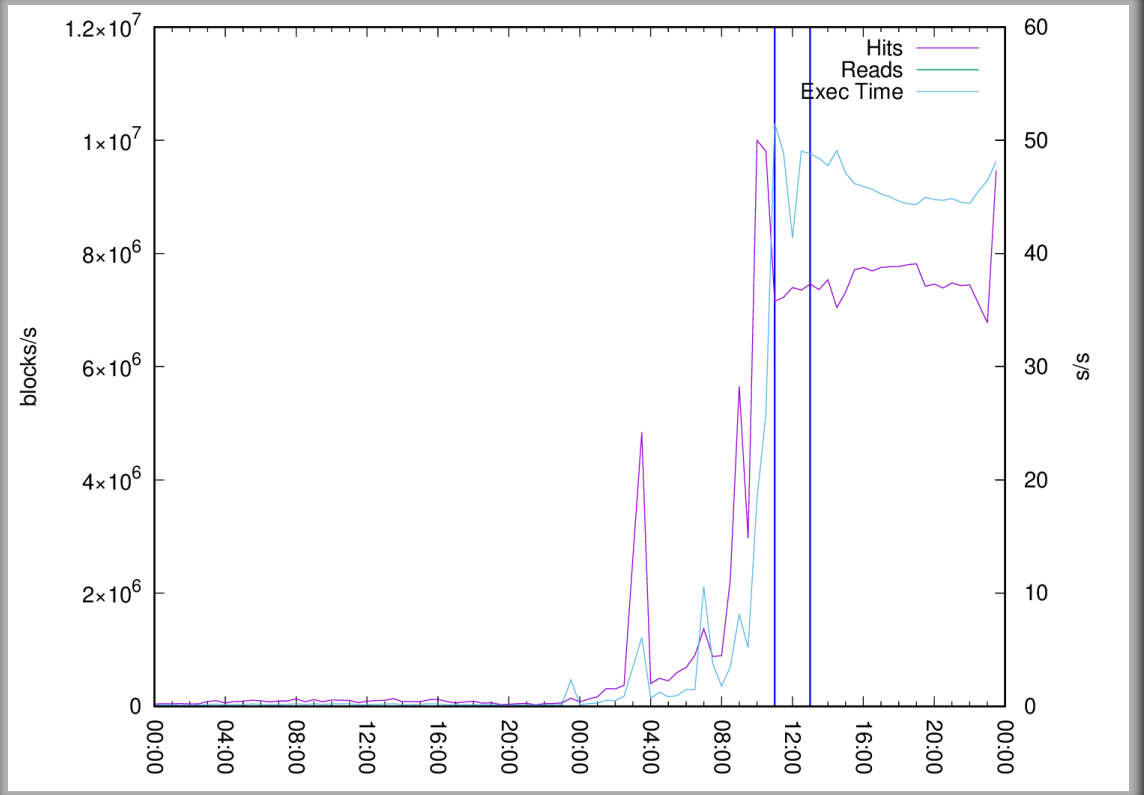
Пример инцидента

Объективные данные наблюдений:

- Существенная деградация производительности
- Выраженное изменение нагрузки

Мы наблюдаем признаки проблемы и хотим локализовать причину

Поиски удобно начинать с отчета



Top SQL by execution time

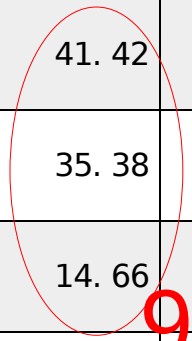
Query ID	Database	Exec (s)	%Total	Rows	Execution times (ms)				Executions
					Mean	Min	Max	StdErr	
825ec9df e2 [ba551e58a1220972]	demodb	17677. 69	26. 10	4976275	436. 777	131. 332	1268. 142	87. 207	40473
f c2b6ac0db [7058521854c25be5]	demodb	13814. 61	20. 40	435	124. 600	33. 169	523. 835	53. 414	110872
32e15bf a2b [2c3252b0a9ecf 099]	demodb	12796. 92	18. 90		224. 436	62. 513	725. 970	85. 991	57018
9972b38b9c [711d687f db6583af]	demodb	6819. 08	10. 07		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

65%

75%

Top SQL by shared blocks fetched

Query ID	Database	blks fetched	%Total	Hits(%)	Elapsed(s)	Rows	Executions
581a0cb27e [42c019f d344ccda3]	demodb	7294930558	41. 42	100. 00	6763. 2		2917
f c2b6ac0db [7058521854c25be5]	demodb	6232122854	35. 38	100. 00	13814. 6	435	110872
825ec9df e2 [ba551e58a1220972]	demodb	2583010863	14. 66	100. 00	17677. 7	4976275	40473
32e15bf a2b [2c3252b0a9ecf 099]	demodb	774440330	4. 40	100. 00	12796. 9		57018
9972b38b9c [711d687f db6583af]	demodb	427932206	2. 43	100. 00	6819. 1		31521
615932b6c7 [3865b6f 15c706793]	demodb	33263701	0. 19	100. 00	225. 8	77099	962
5f adf 658f 1 [2e77692b5dff 5c21]	demodb	33045697	0. 19	100. 00	304. 9	1268	1381



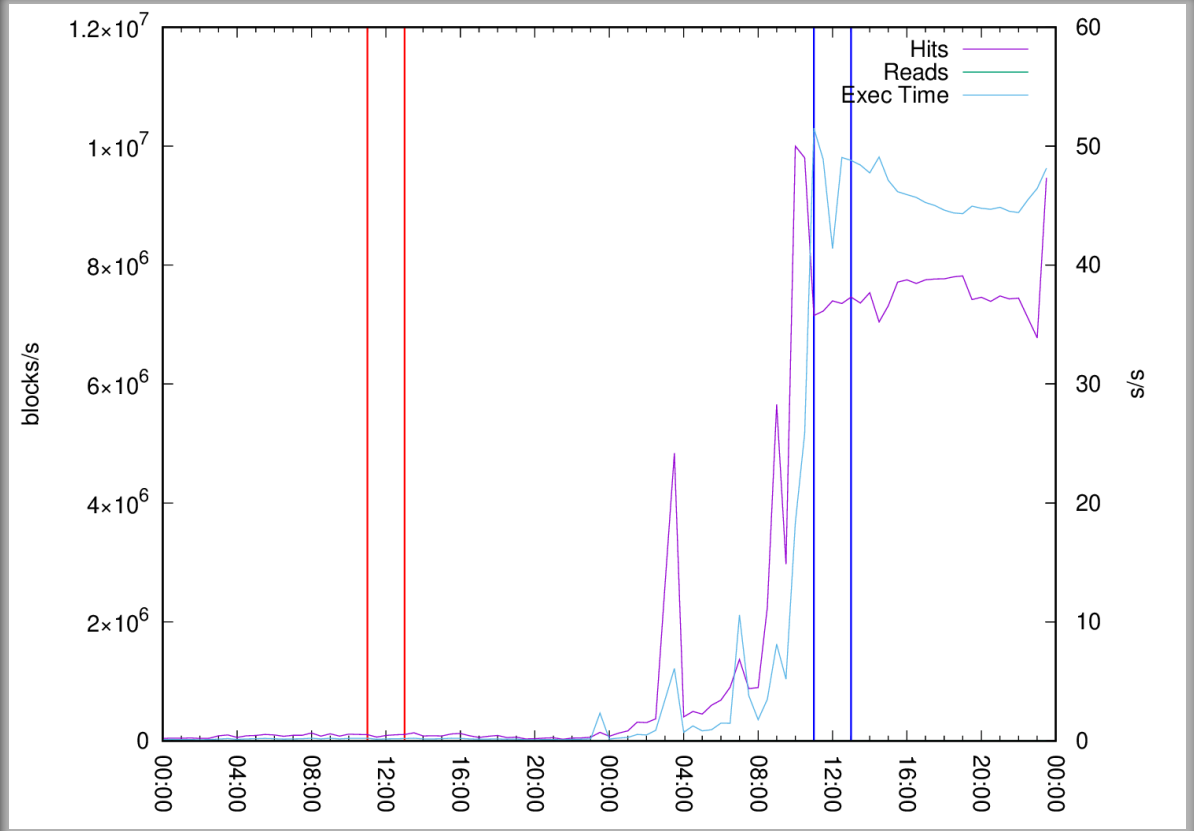
Top tables by blocks fetched

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				

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

Вчера всё работало!



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

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

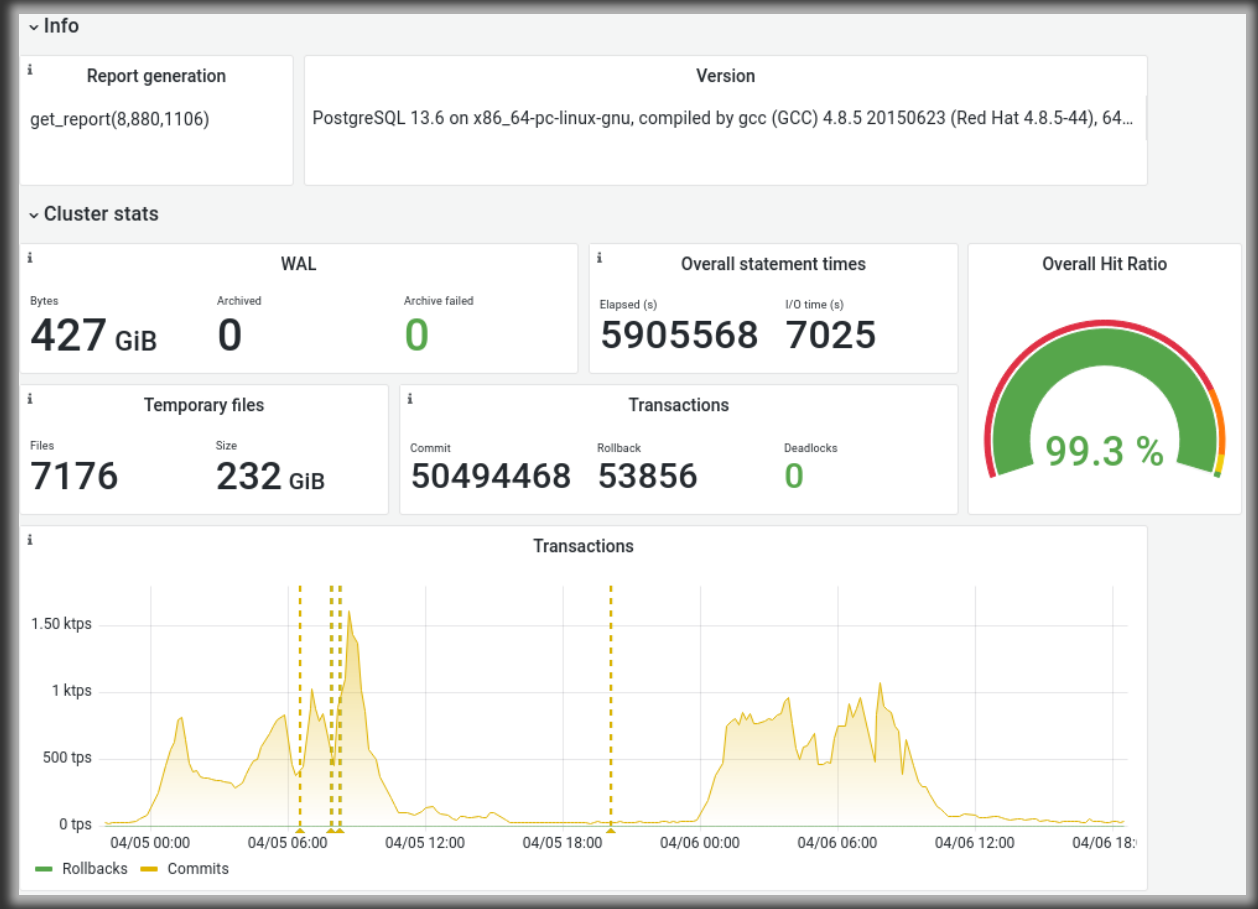
Top tables by blocks fetched

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				

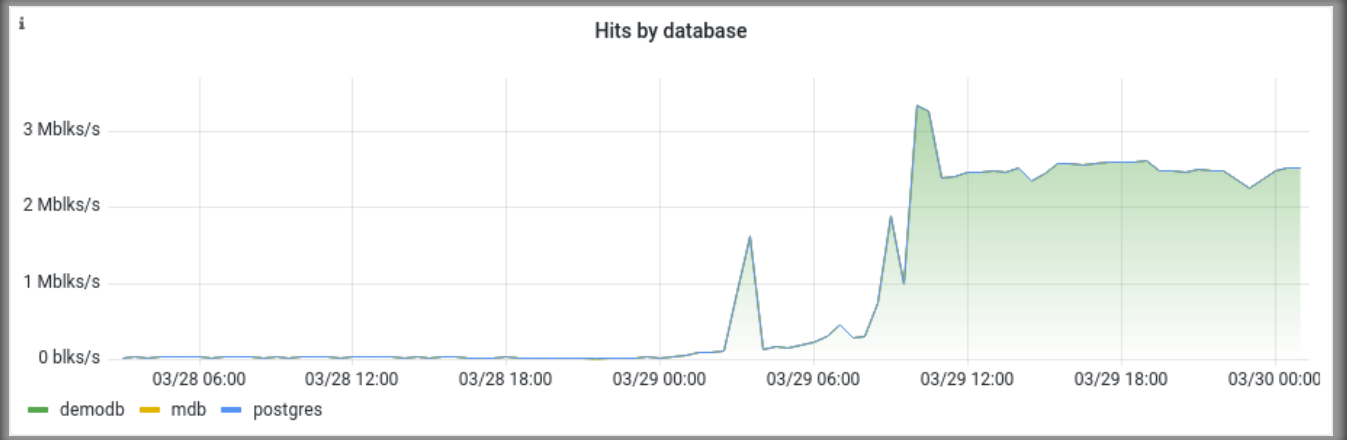
Визуализация наблюдений (Grafana)

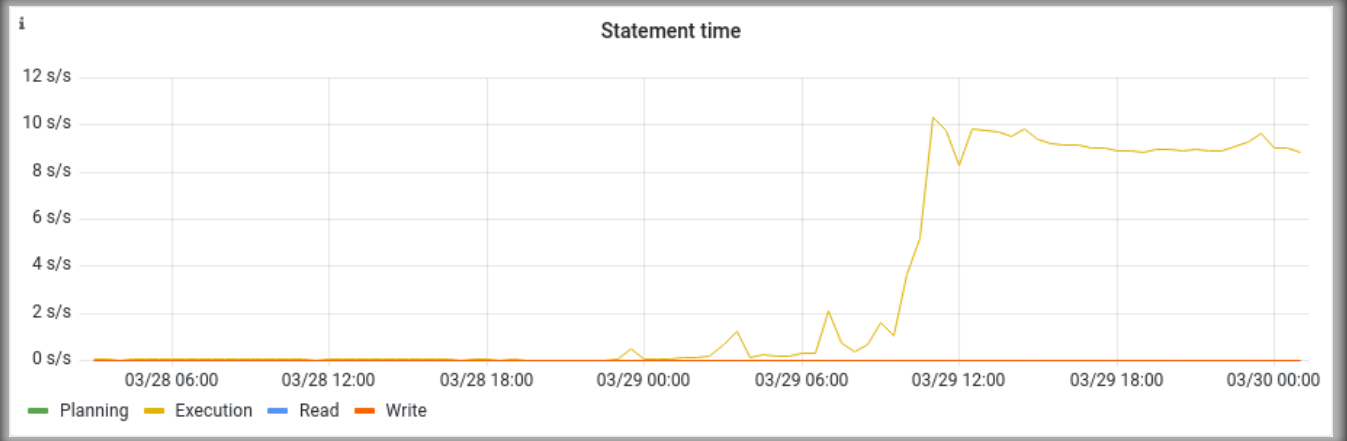
Grafana

PostgreSQL
pg_profile repository



Визуализация наблюдений (Grafana)





Расширенные возможности pgpro_pwr

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

Распределение нагрузки

Resource	Load distribution
Total time (sec.)	psql: 77.05 · load app: 21.68 · pgbench: 13.
Executed count	pgbench: 3637 · benchapp2: 1440 · benchapp1: 1132
I/O time (sec.)	pgbench: 3.03 · benchapp1: 1.07 · benchapp2: 0.89
Blocks fetched	psql: 14259951 · pgben
Shared blocks read	pgbench: 95751
Shared blocks dirtied	psql: 11495 · load app: 10193 · pgbench
Shared blocks written	psql: 11001 · load app: 10189
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	

Database vacuum statistics

Database	DB blocks statistics				VM marks		Dead tuples			WAL	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	208879	2. 89	121549	1. 68	10190	15946	93080	60000	60. 8	4633 kB	0. 66		12. 81	3. 99		3. 06	0. 7	
contrib_regression																		
postgres	5357	0. 07	605	0. 01	325	760	10321	3	99. 97	1787 kB	0. 01		0. 18	4. 47	0. 12	0. 08	0. 03	
profile																		
uptest																		
Total	214236	2. 96	122154	1. 69	10515	16706	103401	60003	63. 28	6420 kB	0. 67		12. 99	8. 47	0. 12	3. 15	0. 72	

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

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	6288	5750	47.77	16	
postgres	pg_default	pg_catalog	pg_statistic	292	256	46.72		2
postgres	pg_default	pg_toast	pg_toast_2619	20	18	47.37		1
postgres	pg_default	profile	last_stat_statements_srv1	11	27	71.05		2
postgres	pg_default	profile	tables_list	8	8	50		
postgres	pg_default	profile	last_stat_vacuum_indexes_srv1	6	6	50		
bench	pg_default	public	pgbench_tellers	4	4	50	5	1
bench	pg_default	public	pgbench_branches	4	3	42.86	5	

Расширенные статистики очистки

- 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

Применяйте на практике

PostgresPro

Андрей Зубков
Postgres Professional

zubkov-andrei / **pg_profile**