

Мониторинг состояния баз данных

Сергей Злобин

Системный администратор в ООО «Контент Онлайн»



Сергей Злобин

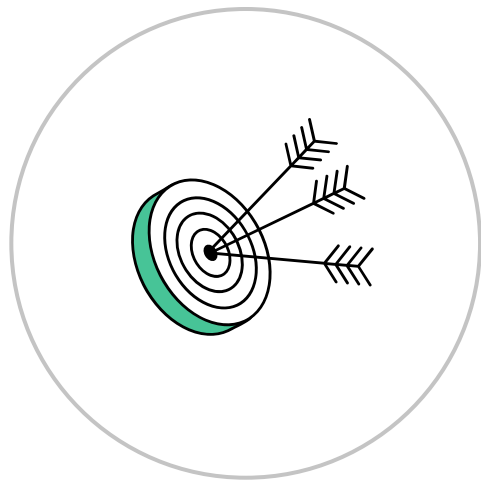
О спикере:

- инженер-программист, системный администратор
- опыт работы с базами данных в роли прикладного разработчика и администратора БД
- более 20 лет в IT



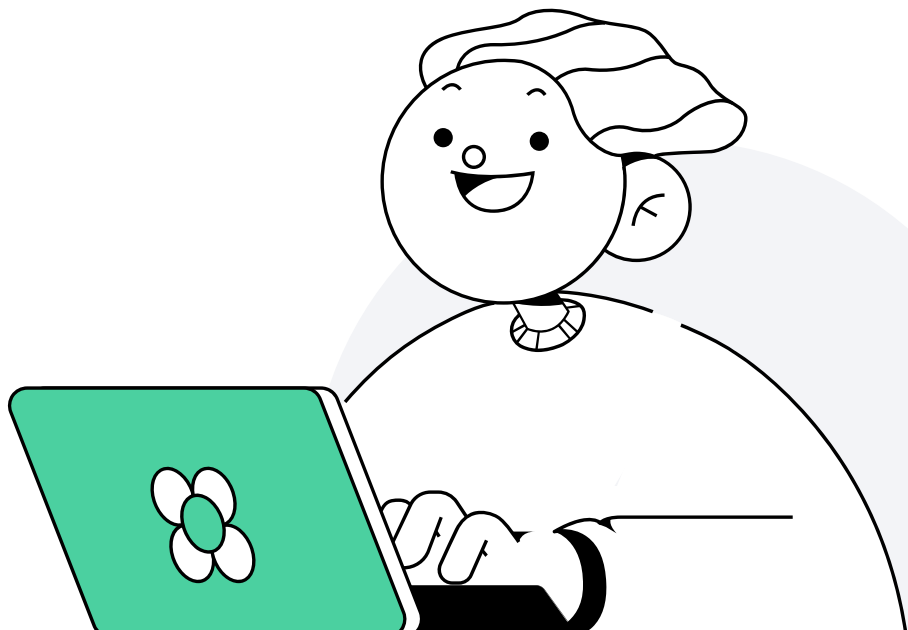
Цели занятия

- Разобрать цели и задачи мониторинга
- Понять, как БД помогает контролировать проактивную реакцию
- Узнать, как управлять поведением БД
- Познакомиться с планировщиком и статистикой



План занятия

- 1 Мониторинг
- 2 Инструментарий
- 3 Метрики производительности
- 4 Метрики скорости и состояния — компоненты БД
- 5 Планировщик/оптимизатор
- 6 Мониторинг сессий и процессов
- 7 Журналирование
- 8 Итоги
- 9 Домашнее задание



Мониторинг



1



Мониторинг

Периодический сбор метрик и событий работы системы и наблюдение за ними с целью анализа данных для своевременного выявления негативных влияний



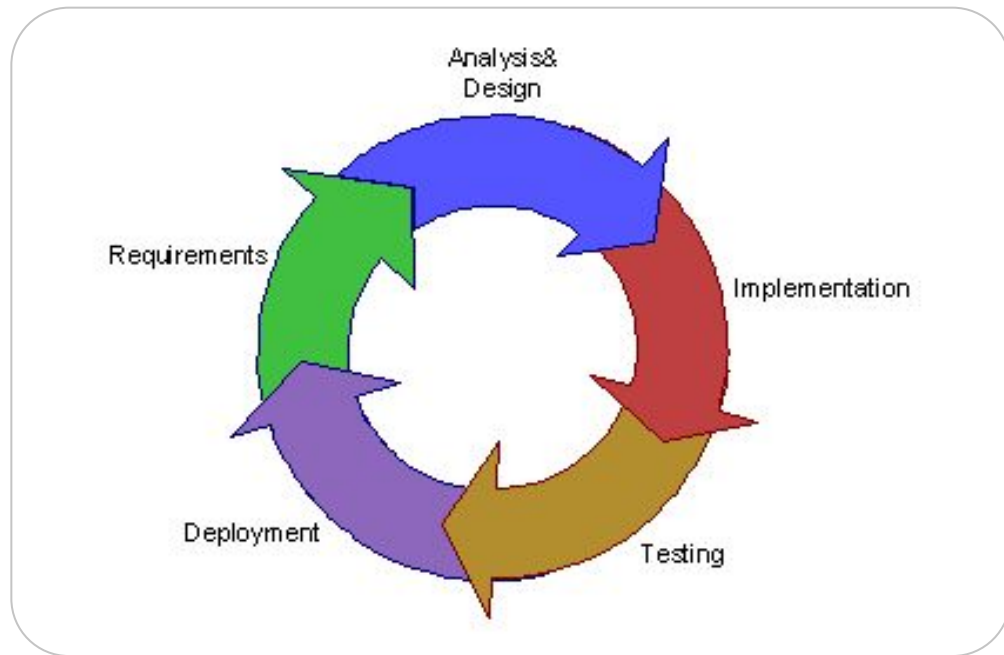
**Для чего нужен
мониторинг?**

Мониторинг

Мониторинг бывает:

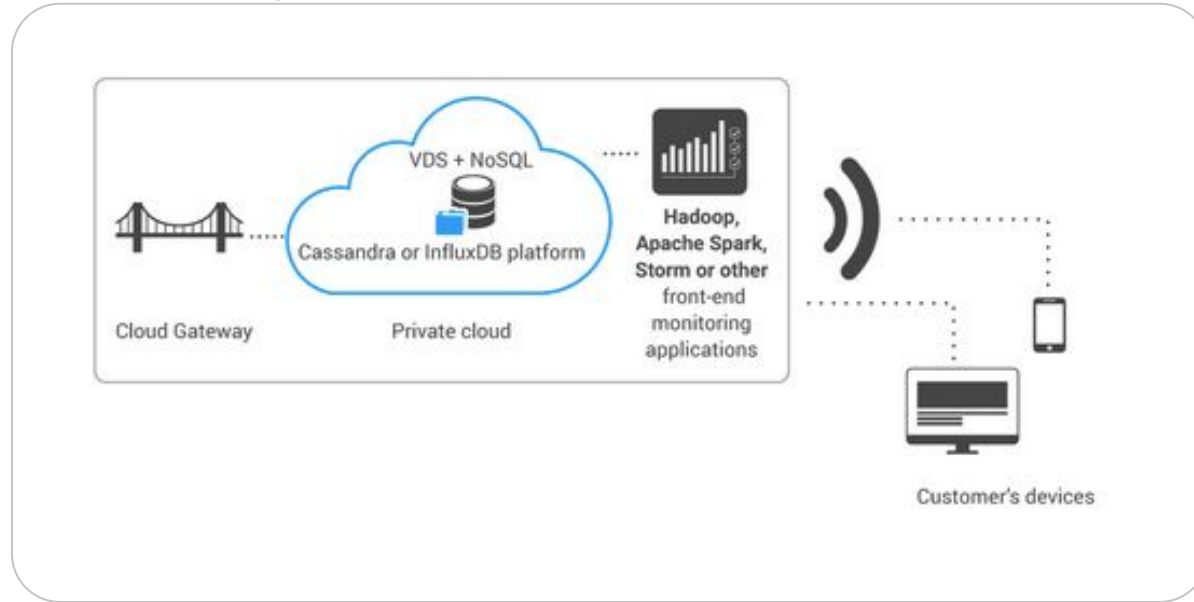
- проактивным
- реактивным

Мониторинг — это не действие или задача, а **циклический процесс**



Таргетированный мониторинг

Считывается один или целый ряд параметров из одного источника. Их проверяют на превышение установленных минимальных или максимальных порогов допустимости



Комплексный мониторинг

Считывается ряд параметров с различных источников комплексной системы. Его анализируют и в случае необходимости по прописанным алгоритмам оповещают пользователя или присылают ему указание к действию



Метрики производительности

- Счётчики загрузки в процентах
- Счётчики загруженности ресурса
- Длина очередей
- Время выполнения операции
- Наличие задержек или неудач операции
- Счётчики объёма ресурса

Статистика

- Количество обращений к ресурсу
- Объём ресурса
- Операции по типу и по количеству
- Количество обращений по методам

Журнал событий — сборщик сообщений

- Лог-файл, журнал событий и сообщений
- Публикация информации по сети через стандартные протоколы
- Стандартный Unix-формат — демон syslog

Блокировки и события операций

- Частично блокирующие
- Полностью блокирующие
- События ожидания

```
postgres@acer:/var/log/postgresql$ tail -f postgresql-15-main.log | egrep --color  
'Lock'  
2023-02-25 19:38:35.457 MSK [5920] postgres@test LOG:  process 5920 still waiting  
for ShareLock on transaction 897 after 1000.203 ms
```

Инструментарий



2

Инструментарий

Какие инструменты существуют:

- 1 Утилиты, входящие в состав ОС или БД «из коробки»
- 2 Утилиты open source, которые не входят в стандартный дистрибутив ОС или БД
- 3 Коммерческие или свободные утилиты (системы управления)
- 4 Специализированные системы мониторинга открытого или коммерческого типа

Инструментарий

① Утилиты, входящие в состав ОС или БД «из коробки»:

- команды операционной системы
- запрос статистической или диагностической информации в БД

```
user@acer:~$ vmstat 3
```

```
procs  -----memory-----  ---swap--  -----io----  -system--  -----cpu-----  
r  b    swpd   free   buff  cache   si   so    bi   bo    in   cs us sy id wa st  
1  0      0 1298132 159848 2661576    0    0   346   104   415 1424 12  4 78  6  0  
0  0      0 1303716 159848 2658032    0    0    0    0  1348 3158  5  3 92  0  0
```

Инструментарий

② Утилиты open source, которые не входят в стандартный дистрибутив ОС или БД:

- утилиты, которые являются расширением к ОС и написаны сторонними командами разработчиков
- ПО, распространяемое в рамках условно свободного лицензирования

```
ooo
1 [|||||] 8.5% Tasks: 150, 727 thr; 1 running
2 [|||||] 7.1% Load average: 0.52 0.85 0.99
3 [|||||] 14.5% Uptime: 00:25:54
4 [|||||] 7.8%
Mem[|||||] 2.34G/5.73G
Swp[|||||] 0K/2.00G

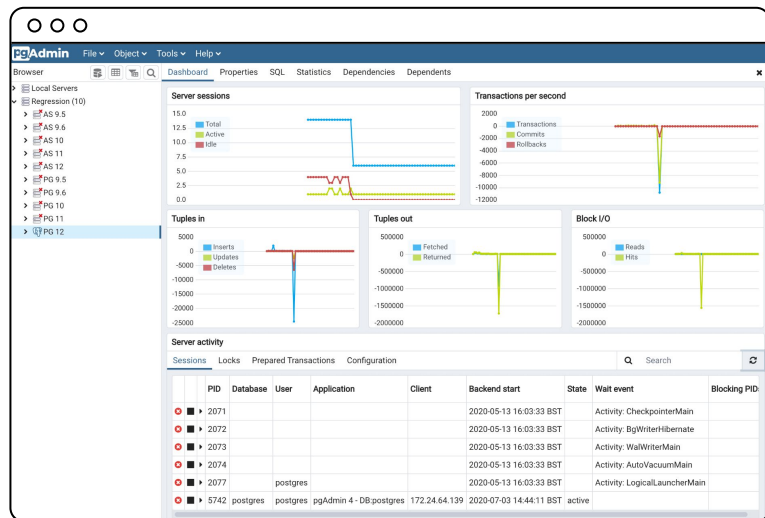
PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
2705 nsvntsov   20    0 3860M 398M 193M S 15.1 6.8  7:07.04 /usr/lib/firefox/firefox -new-window
2097 nsvntsov   -6    0 3752M 20476 15316 S 8.5 0.3  0:53.13 /usr/bin/pulseaudio --daemonize=no --log-target=jour
2006 nsvntsov    9   -11 3752M 20476 15316 S 7.9 0.3  0:53.62 /usr/bin/pulseaudio --daemonize=no --log-target=jour
2835 nsvntsov   20    0 3860M 398M 193M S 6.5 6.8  1:52.16 /usr/lib/firefox/firefox -new-window
2090 nsvntsov   20    0 318M 70704 43612 S 3.9 1.2  1:29.28 /usr/lib/xorg/Xorg vt2 -displayfd 3 -auth /run/user/
3006 nsvntsov   20    0 3860M 398M 193M S 3.9 6.8  0:43.55 /usr/lib/firefox/firefox -new-window
5377 nsvntsov   20    0 11300 4420 3348 R 2.6 0.1  0:00.51 htop
2227 nsvntsov   20    0 4067M 229M 86100 S 2.0 3.9  1:02.86 /usr/bin/gnome-shell
2811 nsvntsov   20    0 3860M 398M 193M S 2.0 6.8  0:34.75 /usr/lib/firefox/firefox -new-window
2845 nsvntsov   20    0 3860M 398M 193M S 1.3 6.8  0:27.76 /usr/lib/firefox/firefox -new-window
2829 nsvntsov   20    0 3860M 398M 193M S 1.3 6.8  0:14.60 /usr/lib/firefox/firefox -new-window
3098 nsvntsov   20    0 2902M 493M 133M S 1.3 8.4  1:38.92 /usr/lib/firefox/firefox -contentproc -childID 4 -is
5346 nsvntsov   20    0 10356 4176 3292 S 0.7 0.1  0:00.07 tmux
3133 nsvntsov   20    0 2902M 493M 133M S 0.7 8.4  0:01.27 /usr/lib/firefox/firefox -contentproc -childID 4 -is
3096 nsvntsov   20    0 2552M 194M 99344 S 0.0 3.3  0:12.27 /usr/lib/firefox/firefox -contentproc -childID 3 -is
4880 nsvntsov   20    0 439M 8192 6632 S 0.0 0.1  0:00.13 /usr/lib/speech-dispatcher-modules/sd_espeak-ng /etc

F1Help F2Setup F3SearchF4Filter F5Tree F6SortB,F7Vice -F8Nice -F9Kill F10Quit
```

Инструментарий

3 Коммерческие или свободные утилиты (системы управления):

- специализированное ПО, которое направлено на функционал управления и/или обслуживания и содержит некоторые опции мониторинга
- специализированное ПО, которое является инструментом разработки и содержит некоторые опции мониторинга



Что является целью мониторинга

- **Сервер** — аппаратное обеспечение (сервера, системы хранения данных — диски, сеть), виртуальная платформа
- **ОС** — утилиты считывания параметров работы ОС, системных метрик
- **Производительность БД** — представления производительности, метрики систем мониторинга
- **Состояние работы БД** — журнал работы

Метрики производительности

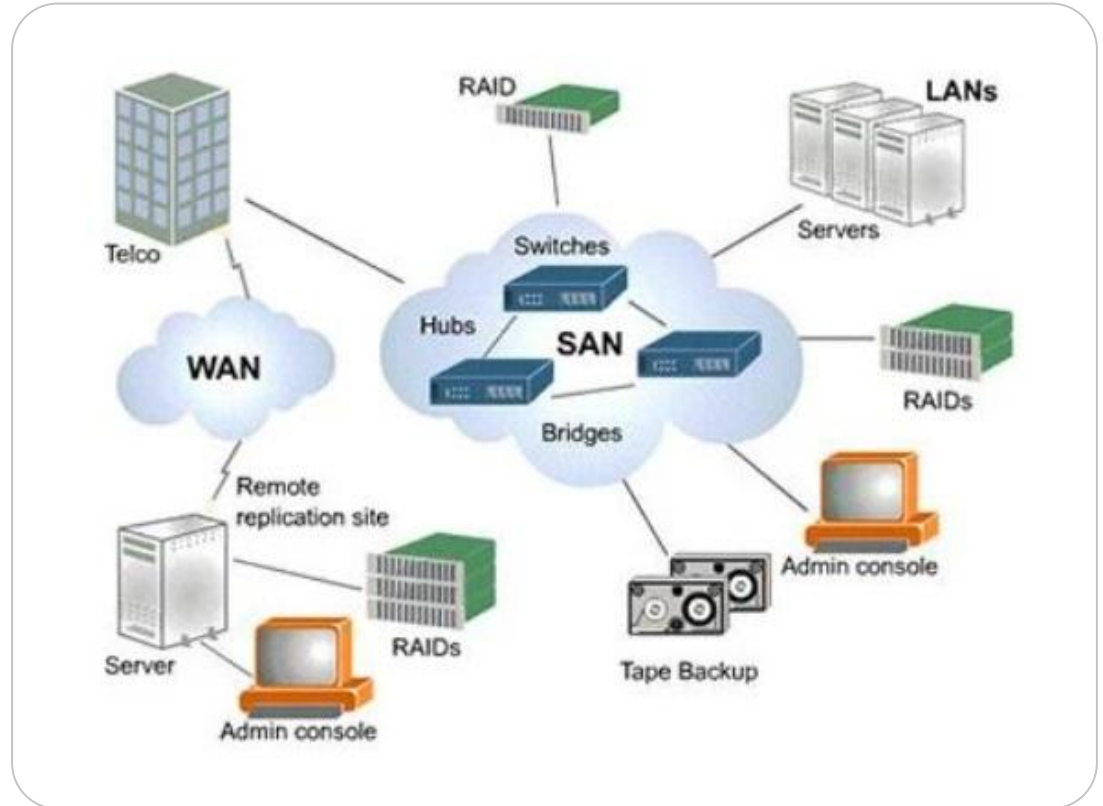
Поле битвы — сервер



3

Метрики производительности

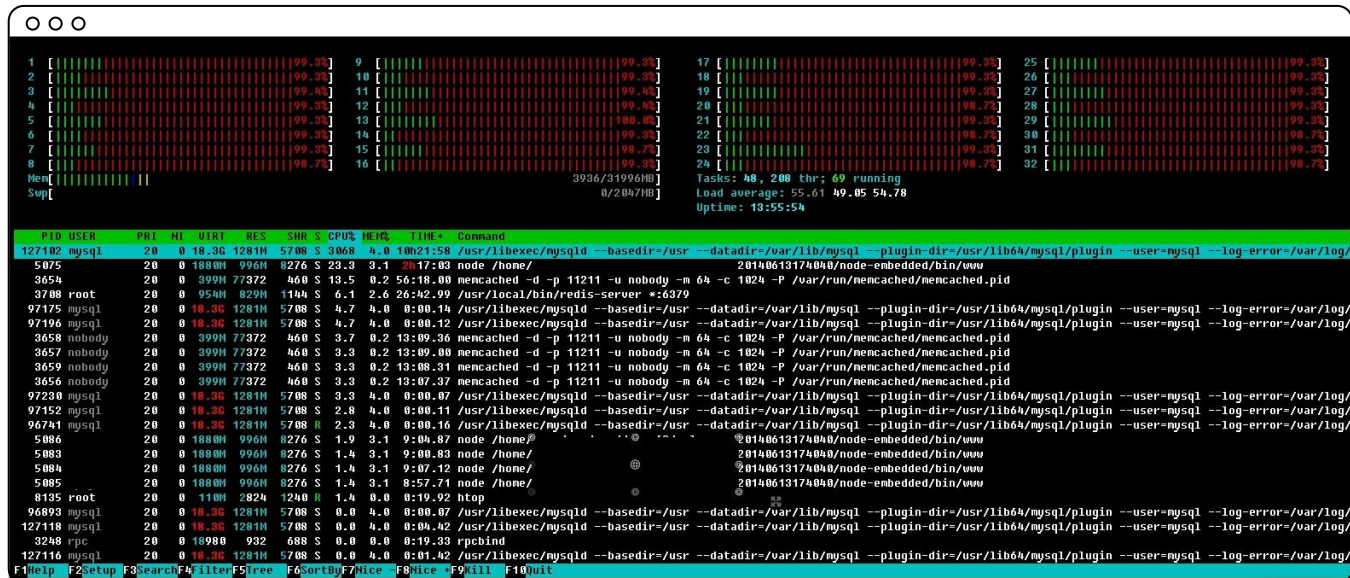
Аппаратная конфигурация
периферийного окружения
хоста БД



Метрики производительности

Процессор/память сервера

Сервер всегда работает в многопользовательском режиме. Каждый процесс конкурирует за процессорное время. На сервере должно быть столько ядер, или нитей (виртуальных процессоров), сколько в среднем активных пользовательских сессий



Источник

Метрики производительности

Вымещение процессов БД в SWAP приводит к значительной деградации производительности

```
user@acer:~$ vmstat -a 3
```

```
procs -----memory----- ---swap-- -----io----- -system-- -----cpu-----
 r  b   swpd   free  inact active   si   so   bi   bo   in   cs us  sy id  wa st
 7   0 788060 113136 1257092 4053844     0   83 1906 2382 5022 16854 78 18  0  5  0  0
 5   0 788060 223972 1163064 4013908     1   11  496  864 6098 13628 79 19  0  2  0  0
 3   0 788672 405108 1179796 3820952     0   48   96  345 3802 12137 80 19  0  1  0  0
```


Метрики производительности

Метрики производительности

- процессор: ps, sar, perf
- ОЗУ, SWAP: vmstat, top
- диск: iostat, blktrace, iotop
- сеть: netstat, iostat, iptraf, netstat
- дополнительные утилиты: top, htop, swapon, nicstat, numstat, mpstat, ss, pcstat

Метрики скорости и состояния — компоненты БД

Поле битвы — база данных



4

Производительность БД

состоит из достаточности ресурсов системы
и сбалансированного взаимодействия её
компонентов между собой при работе с ОС



**Сборщик статистики — это подсистема,
которая собирает и отображает информацию
о работе сервера**

Метрики скорости и состояния

Параметры настройки сбора статистики:

- track_activities
- track_counts
- track_functions
- track_io_timing
- track_wal_io_timing

Метрики скорости и состояния

Представления статистики основных процессов БД:

- pg_stat_replication
- pg_stat_progress_analyze
- pg_stat_progress_create_index
- pg_stat_progress_vacuum
- pg_stat_progress_cluster

Метрики скорости и состояния

Мониторинг работы сессий пользователей и процессов:

- Представление `pg_stat_activity`. В нём для каждого серверного процесса будет присутствовать по одной строке с информацией
- Экстеншен — представление `pg_stat_statements`, которое содержит отдельные строки для каждой комбинации идентификатора БД, расширенную статистическую информацию

Планировщик/ оптимизатор



5



Планировщик

Механизм, реализующий анализ, оптимизацию и выполнение запросов пользователей.

PostgreSQL 15 не поддерживает возможность использования хинтов — подсказок планировщику

Планировщик/оптимизатор

Методы доступа к данным:

- соединение с вложенным циклом (Nested loop)
- соединение слиянием (Merge join)
- соединение по хешу (Hash join)

Планировщик/оптимизатор

Статистика объектов может считываться посредством представлений словаря БД:

- pg_class
- pg_stat_database_conflicts
- pg_index
- pg_stat_all_tables
- pg_stat_database
- pg_stat_all_indexes

```
test=# select relname, reltype, relpages, reltuples from pg_class where relkind='r' limit 3;
```

relname	reltype	relpages	reltuples
trans		16421	63
pg_statistic		12029	32
pg_type		71	15

(3 rows)

Планировщик/оптимизатор

Сбалансированная настройка конфигурации БД.

Настройка экземпляра БД определена в файле

`/etc/postgresql/'version'/'cluster_name'/postgresql.conf`,

который разбит по темам и содержит несколько разделов параметров

Планировщик/оптимизатор

Определены следующие виды контекста:

- internal
- postmaster
- sighup
- superuser-backend
- backend

Планировщик/оптимизатор

Определены следующие виды контекста:

- superuser
- user

```
test=# select name, setting, context from pg_settings where name like '%memory%';
```

name	setting	context
dynamic_shared_memory_type	posix	postmaster
min_dynamic_shared_memory	0	postmaster
shared_memory_type	mmap	postmaster

(3 rows)

Перерыв



Мониторинг сессий и процессов



6

Мониторинг сессий и процессов

Целый набор представлений, начинающихся с `pg_stat_*`, позволяет получить необходимую информацию для анализа.

Терминал 1 — апдейт строки:

```
test=# select pid, state, client_addr from pg_stat_activity where state='active';
 pid   | state | client_addr
-----+-----+-----
 130160 | active |
 130902 | active |
(2 rows)
```

Мониторинг сессий и процессов

Обнаружение блокировок, считывание состояния запроса пользователя

- pg_locks
- pg_backend_pid()
- pg_blocking_oids(pid)

Терминал 1 — апдейт строки

```
test=# begin transaction;
BEGIN
test=# update tm set name='EEE' where id=5;
UPDATE 1
test=# select pg_backend_pid();
 pg_backend_pid
-----
          130824
(1 row)
```

Мониторинг сессий и процессов

Терминал 2 — апдейт той же строки, что и терминал 1

```
test=# select pg_backend_pid();
 pg_backend_pid
-----
          130160
(1 row)
test=# begin transaction;
BEGIN
test=# update tm set name='EE' where id=5;
```

Мониторинг сессий и процессов

Терминал 3 — проверяем, чтобы в представлении pg_locks не было блокировок

```
test=# select locktype, transactionid, pid, mode, granted from pg_locks where  
transactionid in (select transactionid from pg_locks where not granted);
```

locktype	transactionid	pid	mode	granted
transactionid	879	130160	ShareLock	f
transactionid	879	130824	ExclusiveLock	t

(2 rows)

Мониторинг сессий и процессов

Команда прерывания выполнения команды в сессию:

```
test=# select pg_cancel_backend('130248');
pg_cancel_backend
-----
t
(1 row)
test=#
```

Команда прерывания работы сессии:

```
test=# select pg_terminate_backend('130248');

pg_terminate_backend
-----
t
(1 row)
test=#
```

Журналирование



7

Основная задача журналирования:

регистрация ошибок и протоко-
лирование работы сервера

Журналирование

Методы протоколирования сообщений сервера: stderr, csvlog и syslog.

На Windows также поддерживается eventlog.

- log_destination (string)
- logging_collector (boolean)
- log_directory (string)
- log_filename (string)

Журналирование

Параметры настройки журналирования:

- `log_rotation_age` (integer)
- `log_rotation_size` (integer)
- `log_truncate_on_rotation` (boolean)
- `syslog_facility` (enum)

Журналирование

Настройки для этих порогов определяются следующими параметрами:

- `log_min_messages` (enum)
- `log_min_error_statement` (enum)
- `log_min_duration_statement` (integer)

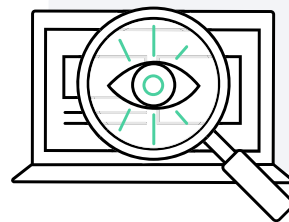
Журналирование

Параметры логирования конкретных событий:

- `log_autovacuum_min_duration` (integer)
- `log_checkpoints` (boolean)
- `log_error_verbosity` (enum), допустимые значения: TERSE, DEFAULT и VERBOSE)
- `log_lock_waits` (boolean)
- `log_statement` (enum), допустимые значения: none (отключено), ddl, mod и all (все команды)

Демонстрация работы

Мониторинг работы сессии и процесса



Демонстрация работы

Сначала наполним таблицу данными и создадим индекс, а затем проконтролируем процесс на всех этапах.

Создадим таблицу и наполним её данными:

```
test=# create table t1 (id bigint, transaction text);  
CREATE TABLE  
test=# insert into t1 select generate_series(1,30000000) as id, md5(random()::text)  
as transaction;  
INSERT 0 30000000
```

Демонстрация работы

Изучим статистику по сессиям, чтобы понять, почему запрос замер и что происходит:

```
pid | username | application_name | wait_event_type | wait_event | state |  
backend_xid | backend_type | query  
-----+-----+-----+-----+-----+-----+-----  
--+-----+-----+-----+-----+-----+-----  
-----+-----+-----+-----+-----+-----+-----  
131393 | postgres | psql | | | active | | client  
backend | select pid, username, application_name, wait_event_type, wait_event, state,  
backend_xid, backend_type, query from pg_stat_activity where state='active';  
135040 | postgres | psql | | | active | | 894 |  
client backend | insert into t1 select generate_series(1,30000000) as id,  
md5(random()::text) as transaction;  
(2 rows)
```

Демонстрация работы

Теперь создадим индекс и проконтролируем его выполнение:

```
test=# create index on t1 (transaction);
```

Во 2-ом терминале мониторим состояние (прогресс процесса) создания индекса:

```
test=# select * from pg_stat_progress_create_index ;
```

pid	datid	datname	relid	index_relid	command	phase
		lockers_total	lockers_done	current_locker_pid	blocks_total	blocks_done
		tuples_total	tuples_done	partitions_total	partitions_done	
135040	16402	test	16432	0	CREATE INDEX	building index: scanning
table		0	0	0	0	280374 158254
	0		0	0	0	

(1 row)

Демонстрация работы

Теперь создадим индекс и проконтролируем его выполнение:

[illegible]

Итоги

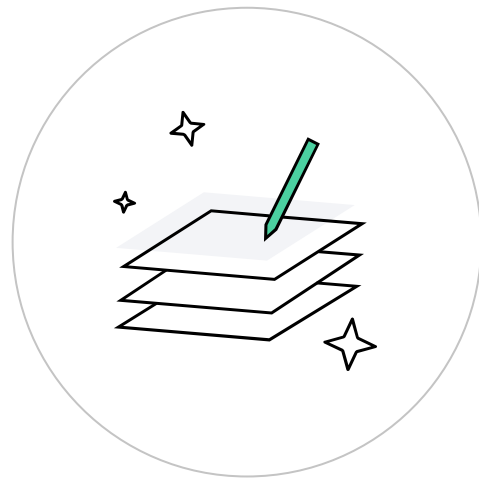
- Рассмотрели, что такое мониторинг, его цели и виды
- Обсудили, как реализовать мониторинг в PostgreSQL
- Узнали о возможностях настройки мониторинга, детализации и методах передачи метрик и событий
- Затронули тему событий и статистики БД, влияющих на работу и состояние как экземпляра БД в целом, так и запросов



Домашнее задание

Давайте посмотрим ваше домашнее задание

- 1 Вопросы по домашней работе задавайте в чате группы
- 2 Задачи можно сдавать по частям
- 3 Зачёт по домашней работе ставят после того, как приняты все задачи



Дополнительные материалы

- [Мониторинг работы БД](#) — статья об умении читать вывод трассировки
- [Мониторинг использования диска](#)
- [Регистрация ошибок и протоколирование работы сервера](#) — настройка журнала работы БД
- [Функции передачи сигнала серверу](#)



Задавайте вопросы и пишите отзыв о лекции

Сергей Злобин

Системный администратор в ООО «Контент Онлайн»

