Диагностика postgresql с точки зрения системного администратора

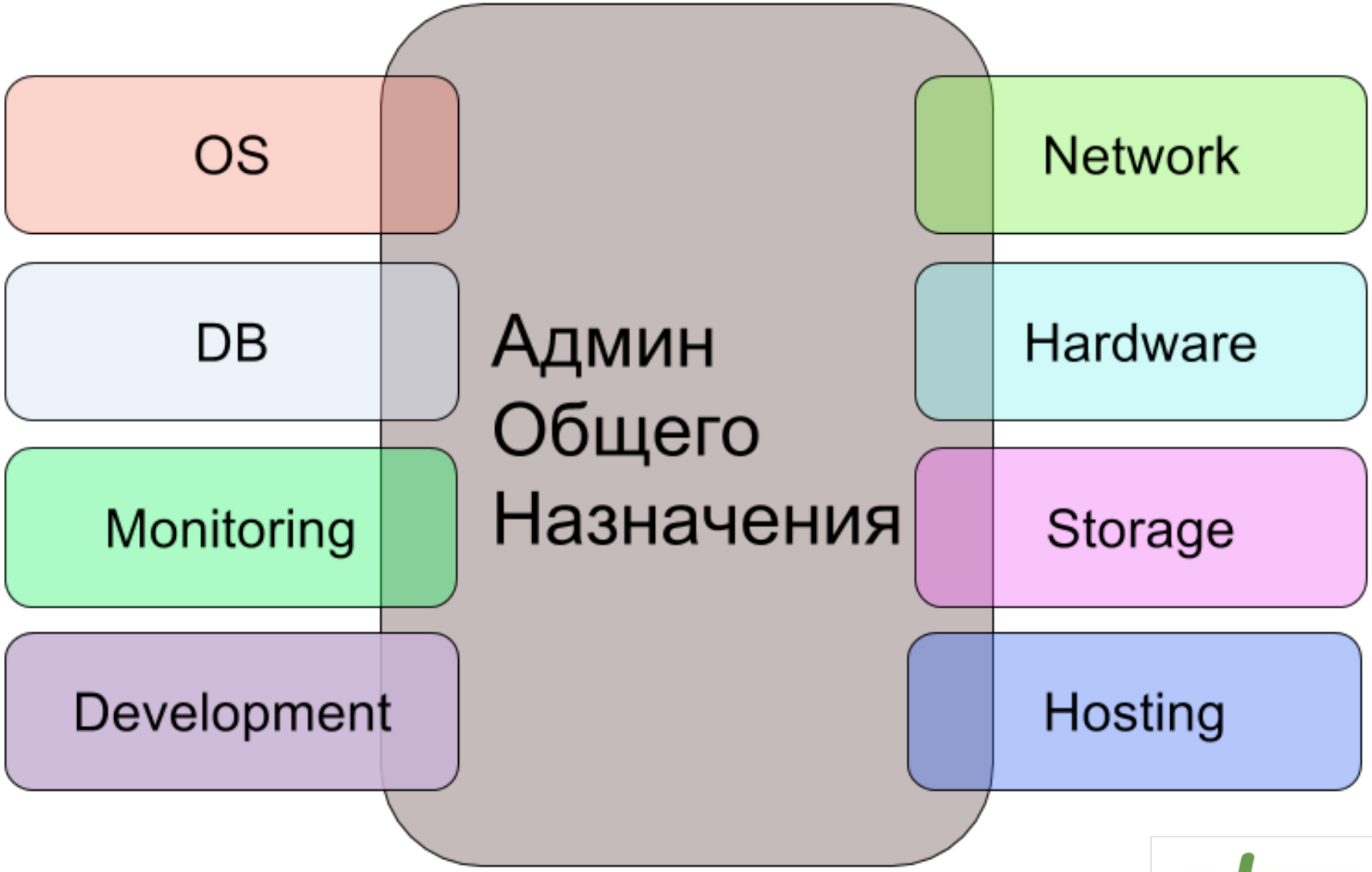
Николай Сивко

okmeter.io

Кто говорит

- Очень долго работал админом
- Руководил эксплуатацией в hh.ru
- Основал и работаю в okmeter.io − сервис мониторинга

Проблема



Проблема

- Для хорошего uptime нужно понимать, как работает **каждый** компонент вашей инфраструктуры
- БД обычно самый сложный компонент, непросто масштабируется, непросто добиться отказоустойчивости
- Лучше если БД управляют профессиональные DBA, но это доступно не всем проектам
- По верхам понимать, как работает БД нужно любому админу

Что хочет знать админ

- БД для остального проекта – ресурс: работает ли он штатно или есть проблемы?

- БД потребляет ресурсы сервера:
  - на что тратятся ресурсы?
  - пора добавлять или можно как-то снизить?
  - какой именно запрос просить ускорить?

Не затронем:

- Backup/restore
- Репликацию
- Оптимизацию запросов

Как сейчас работает БД

- Работа БД – выполнять запросы

- Время выполнения очень хороший показатель:
  - если уперлись в ресурсы: время растет
  - появились тяжелые запросы: уперлись в ресурсы ->время растет
  - ждем на блокировках: время растет
  - пропали запросы: время падает (запросы от мониторинга быстрые)

Время выполнения запросов

- **pg_stat_activity** – мгновенный снимок всех исполняющихся запросов

  extract(epoch from (now()-xact_start)) – время в текущем состоянии (active, idle in transaction)

- **pg_stat_statements** – счетчики по завершенным запросам

  total_time – кумулятивная сумма времен выполнения запроса

- Сниффинг сети между клиентами и pg

Смотрим по pg_stat_statements

- Время выполнения запроса обычно ~1ms, получить полную картину периодическими запросами в pg_stat_activity нереально

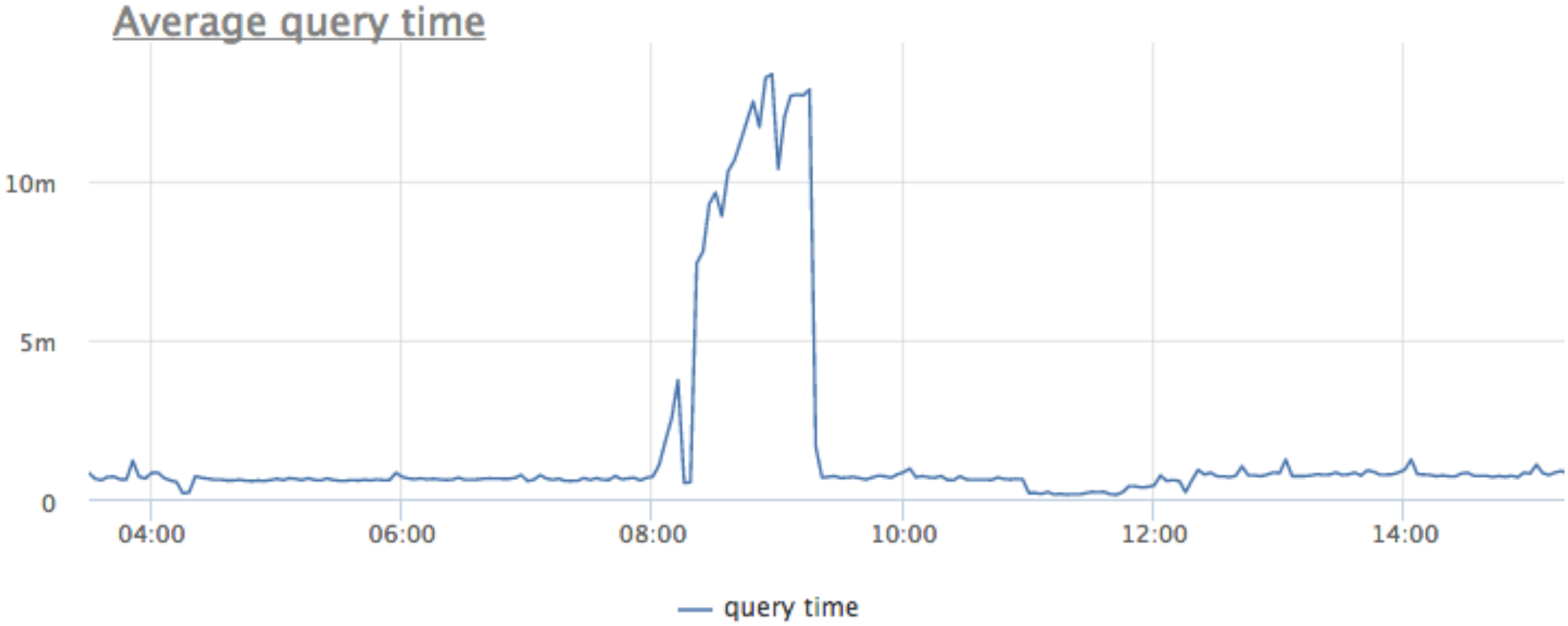
- По pg_stat_statements мы к сожалению не можем посчитать гистограмму, придется смотреть на среднее

  [rate(total_time)/rate(calls)]

- Для общей картины смотрим на суммарное среднее:

  [sum(rate(total_time))/sum(rate(calls))]

Среднее плохо. Но видно же аномалию?



okmeter

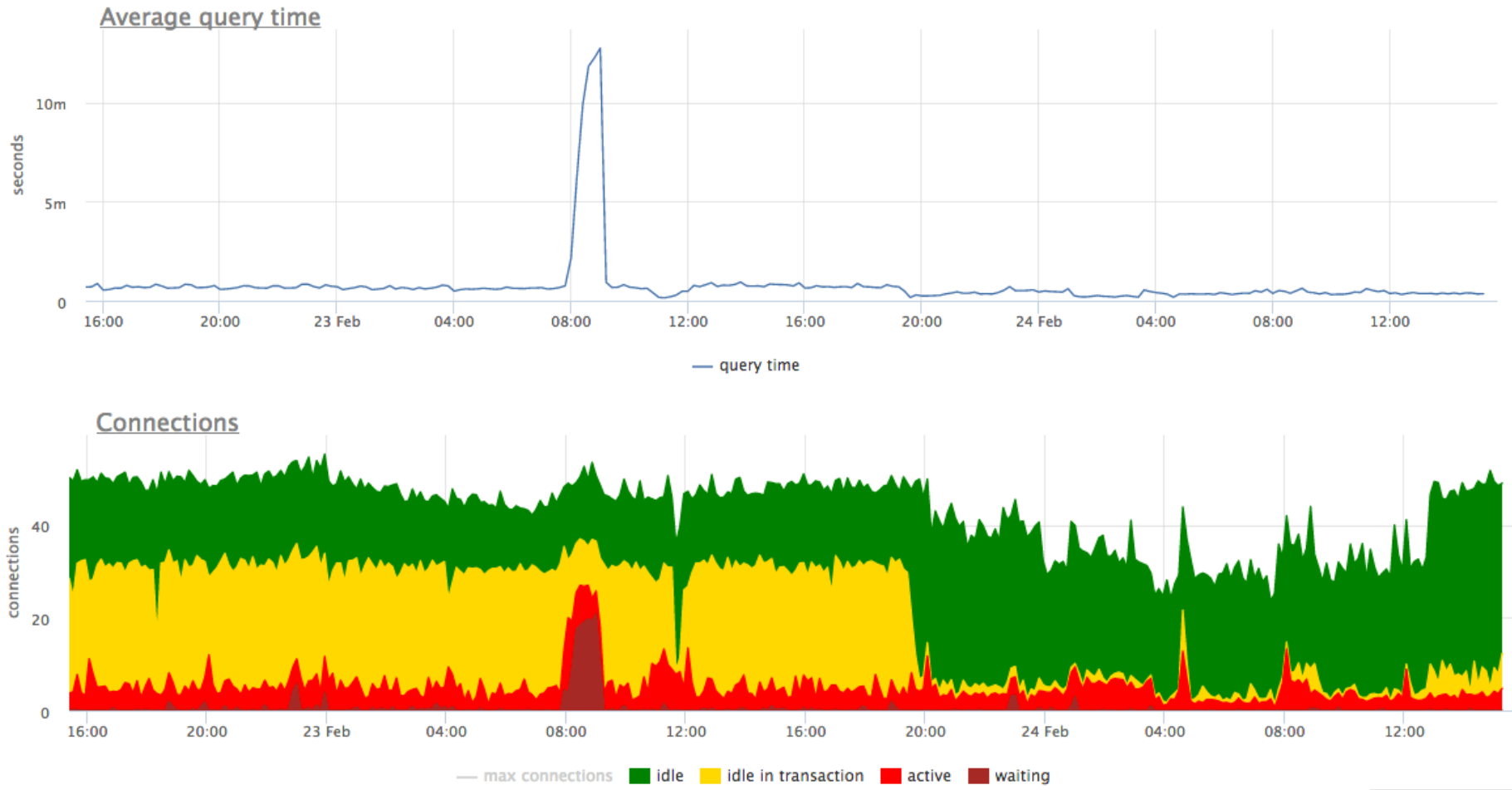
Среднее по-прежнему плохо

- Любая аномалия (выброс/провал) на среднем не обязательно проблема

НО

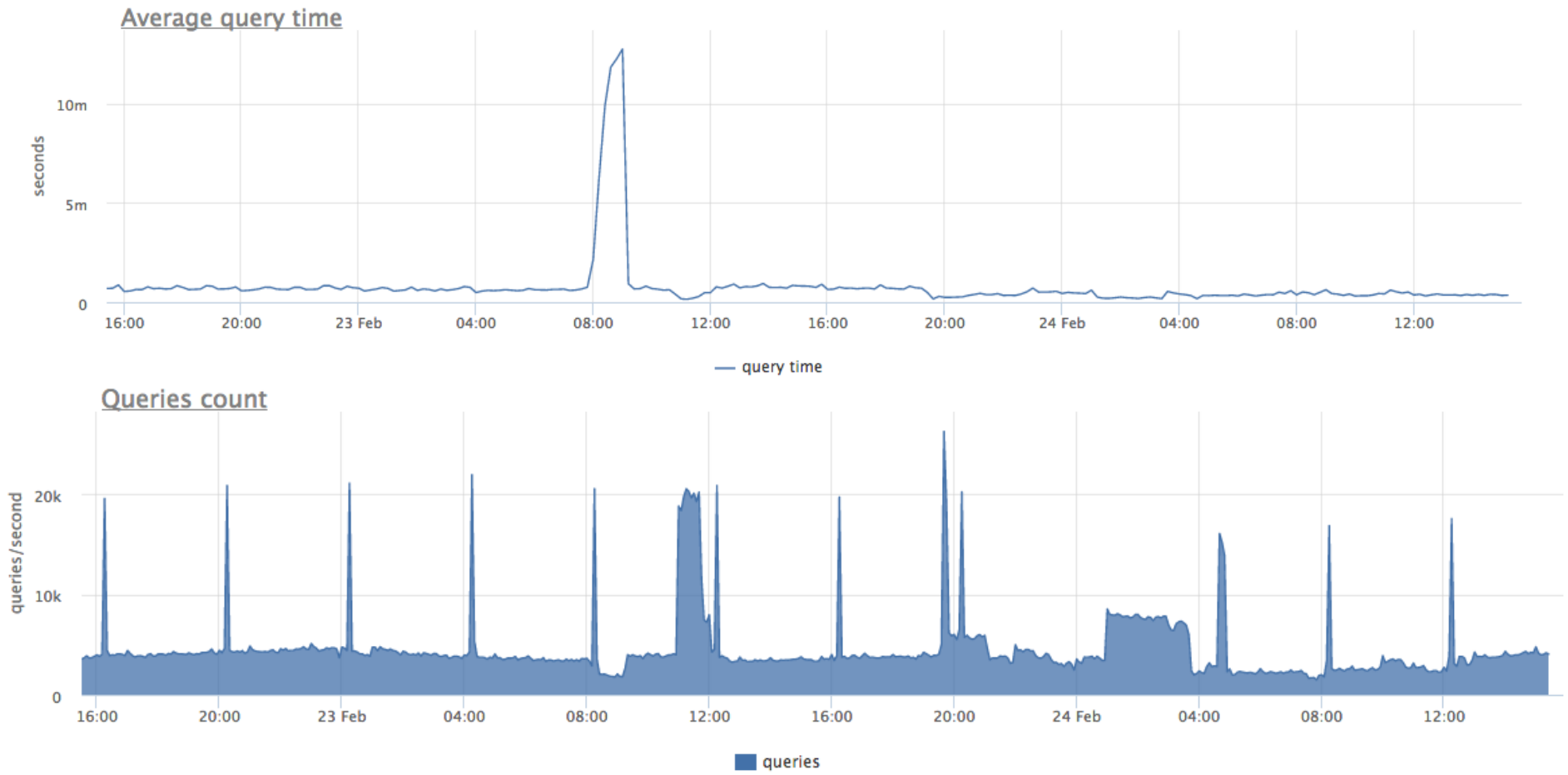
- Скорее всего любую проблему будет видно на среднем!

Не пропали соединения?



okmeter

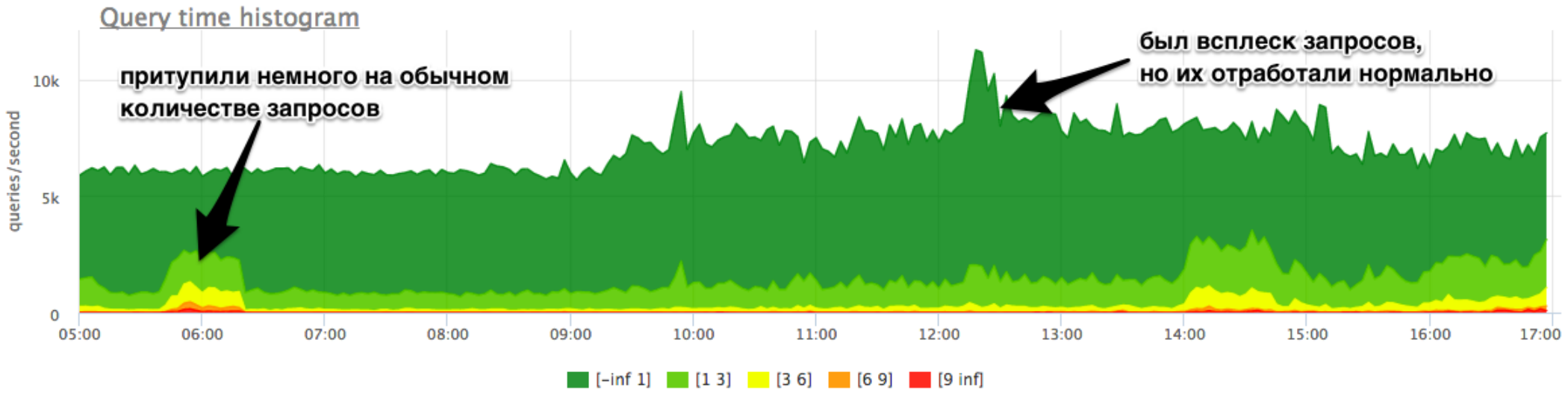
Сколько запросов?



okmeter

Как работает БД?

- Однозначно сложно сказать
- По 3-4 графикам можно понять
- **Хорошо бы** иметь гистограмму, тогда было бы проще:

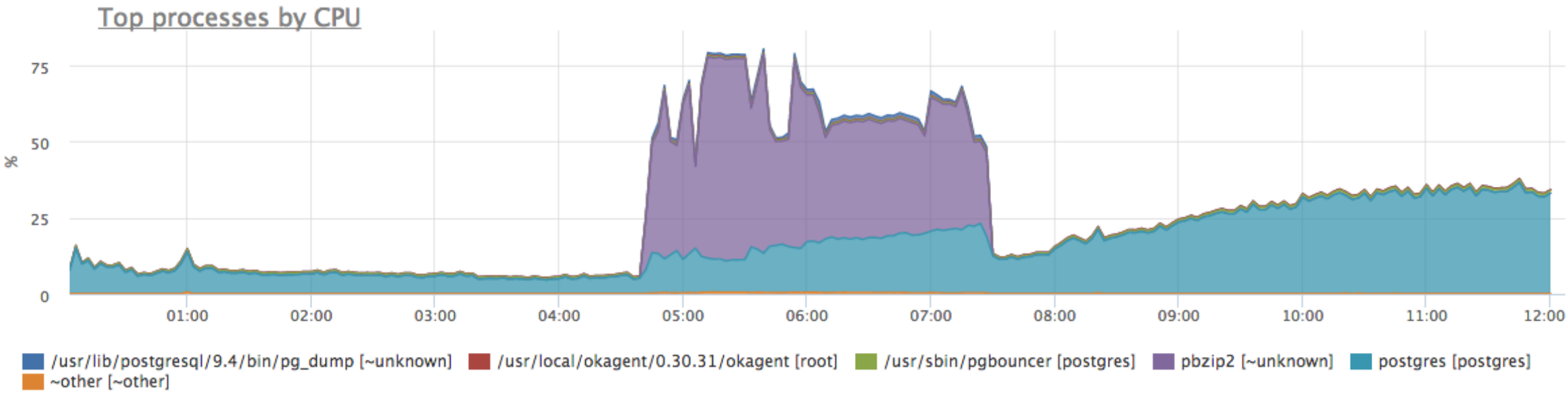


okmeter

Если проблема есть

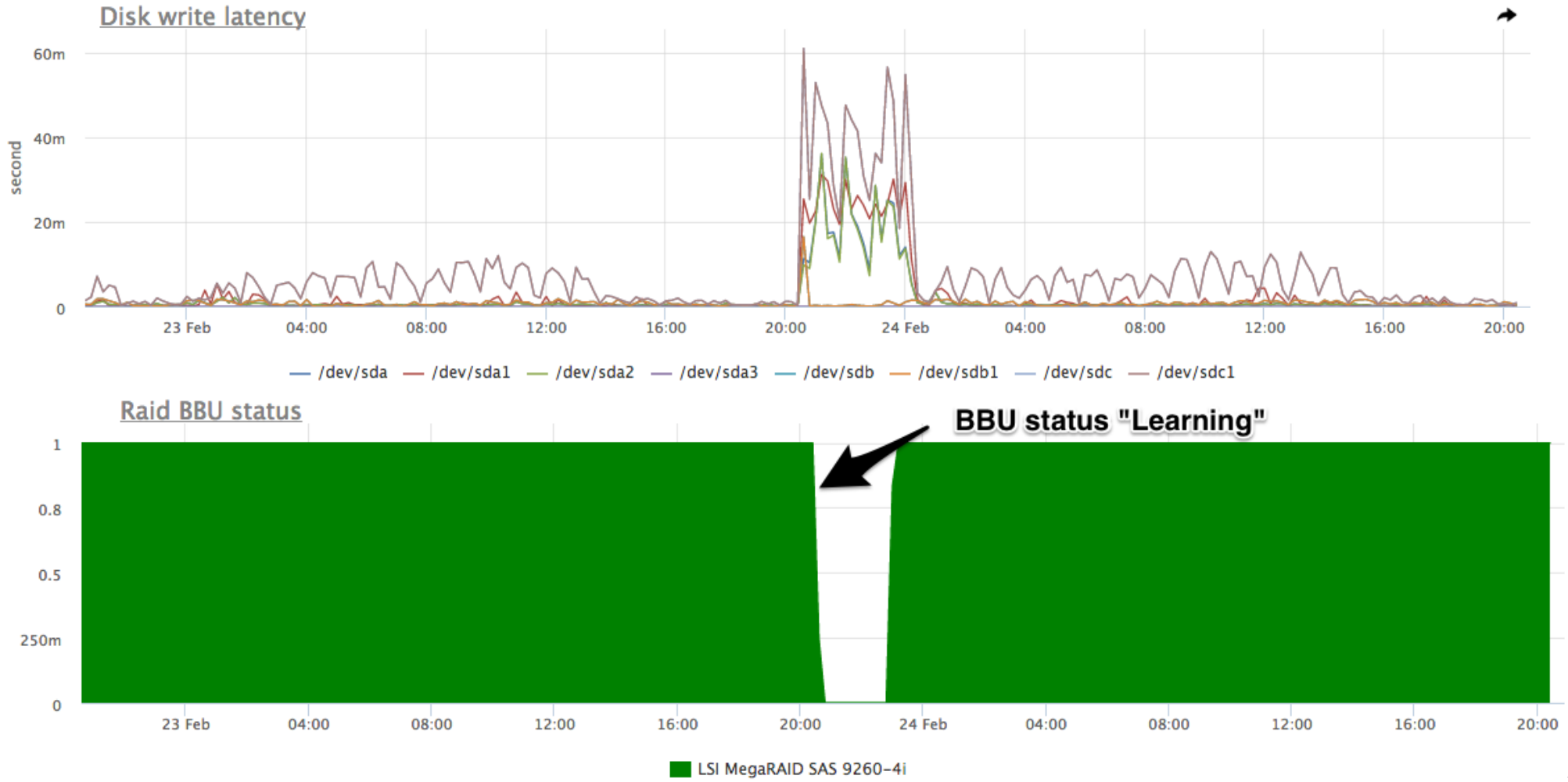
- Что с ресурсами: CPU, disk, net?
- Может их съел какой-то другой процесс?
- Ресурсов стало меньше (деградация сервера)

Другой процесс?



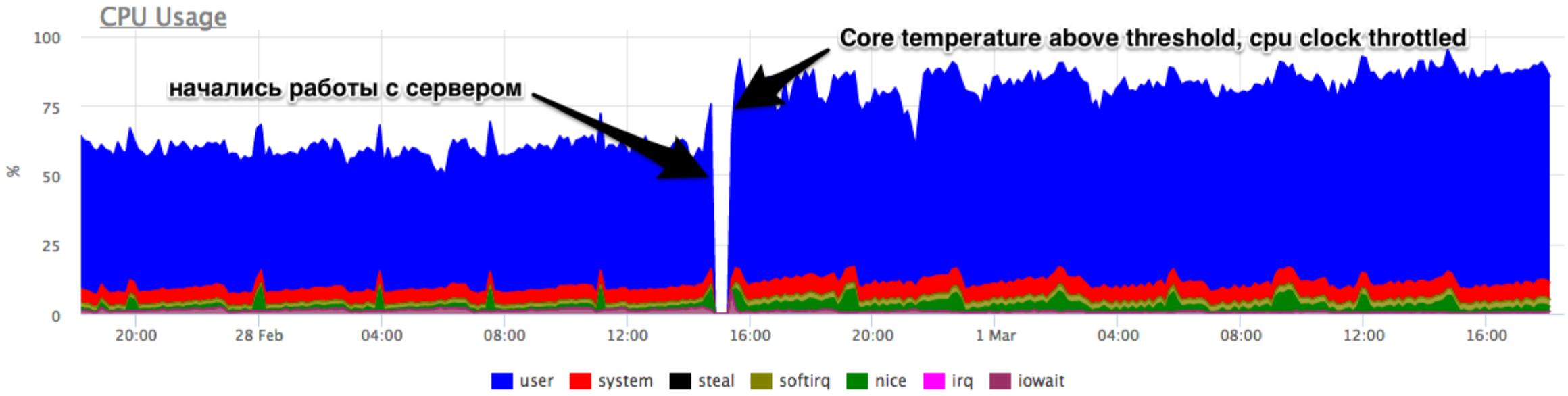
okmeter

Деградация сервера



okmeter

Деградация сервера



okmeter

Ресурсы изнутри postgresql

- Убедились, что потребляет сам PG
- Хотим больше конкретики: запросы/таблицы/индексы

Ресурсы: cpu

- Будем считать, что

total_query_time = disk_wait_time + cpu_time

- Это **вранье**, если

- существенное время ожидаем блокировок − потребляем меньше CPU

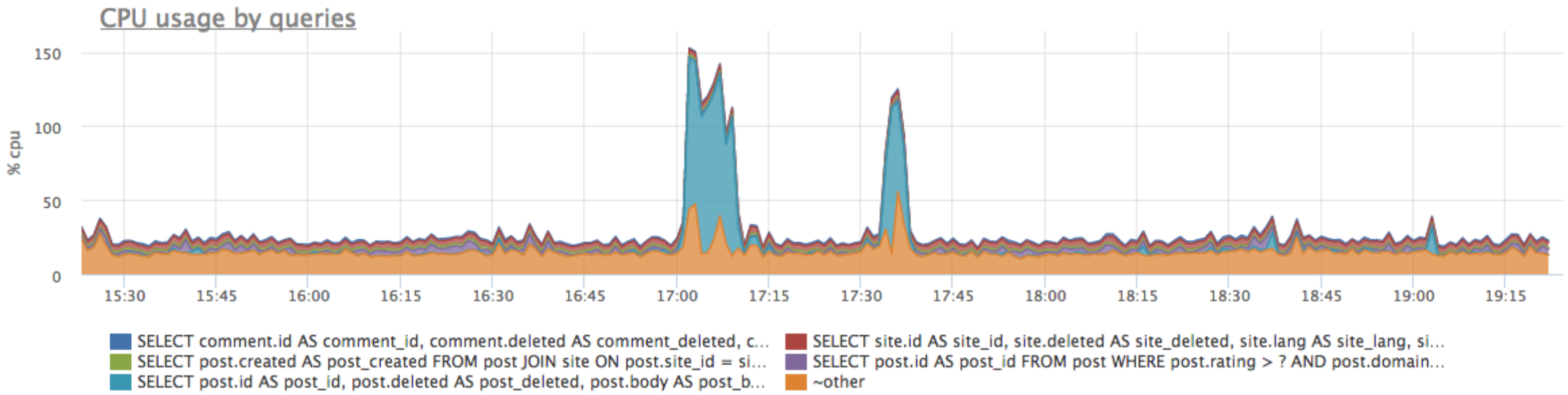
- parallel query (9.6+) − потребляем больше CPU

Ресурсы: cpu

- Не нужно смотреть на “медленные” запросы, когда говорим про ресурсы

- Правильно смотреть на **сумму времени**

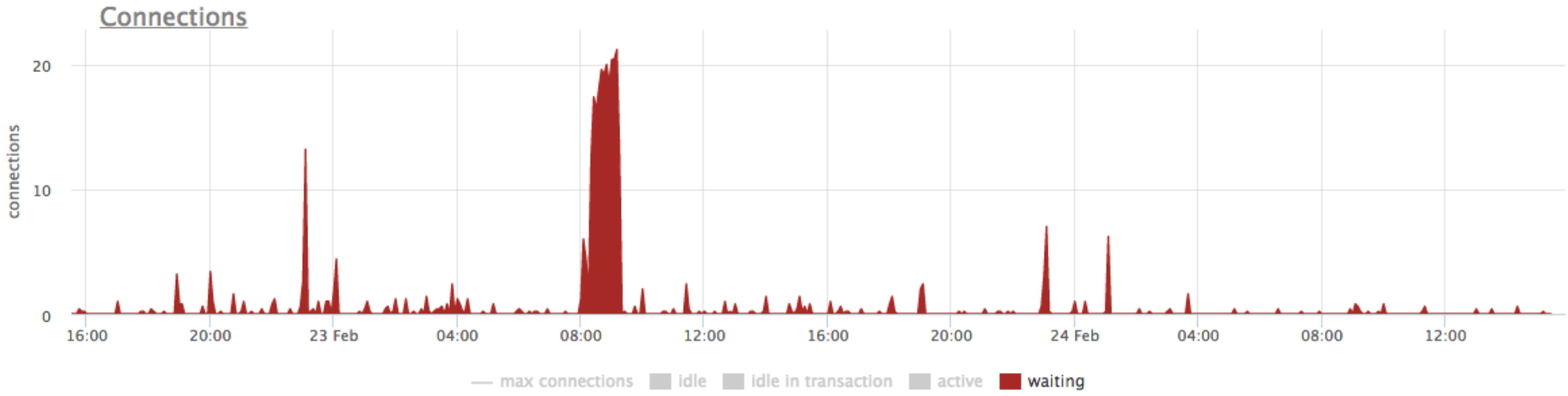
CPU: запросы



okmeter

CPU: нужно исключить блокировки

- Выбросы на cpu_time могут быть из-за локов
- Перепроверяем по pg_stat_activity
- pg_stat_activity + pg_locks: можем узнать, кто залочил и кто ждет



okmeter

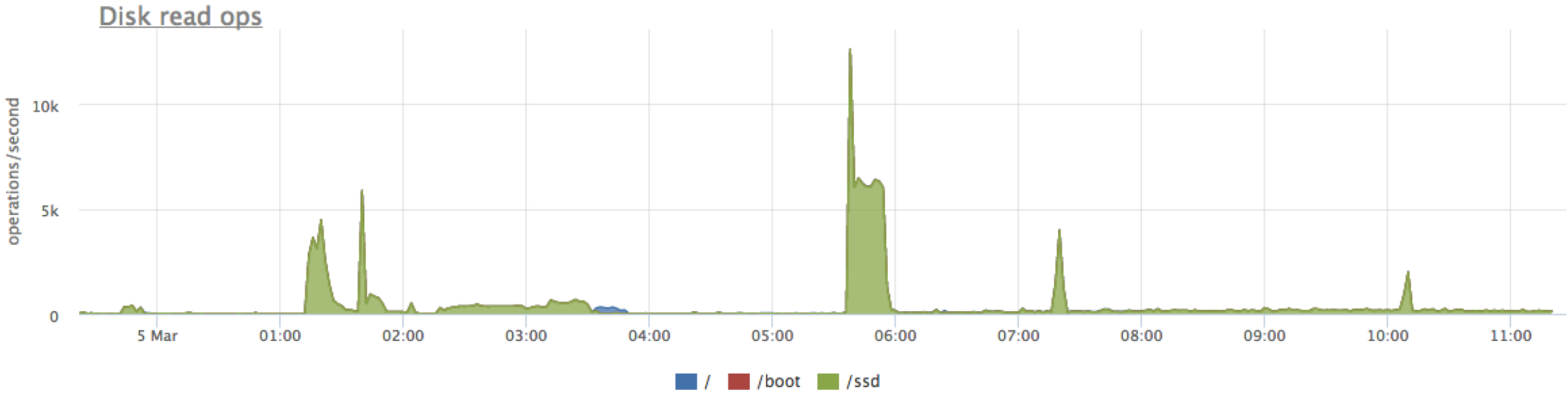
Ресурсы: диск.чтение

- **shared_blks_read**: чтение страниц с диска в случае buffer cache miss (до диска дойдет, если будет еще page cache miss)
- **local_blks_read**: чтение данных временных таблиц
- **temp_blks_read**: чтение временных файлов (когда для запроса нужно больше памяти, чем **work_mem**)

Сложности мониторинга диска

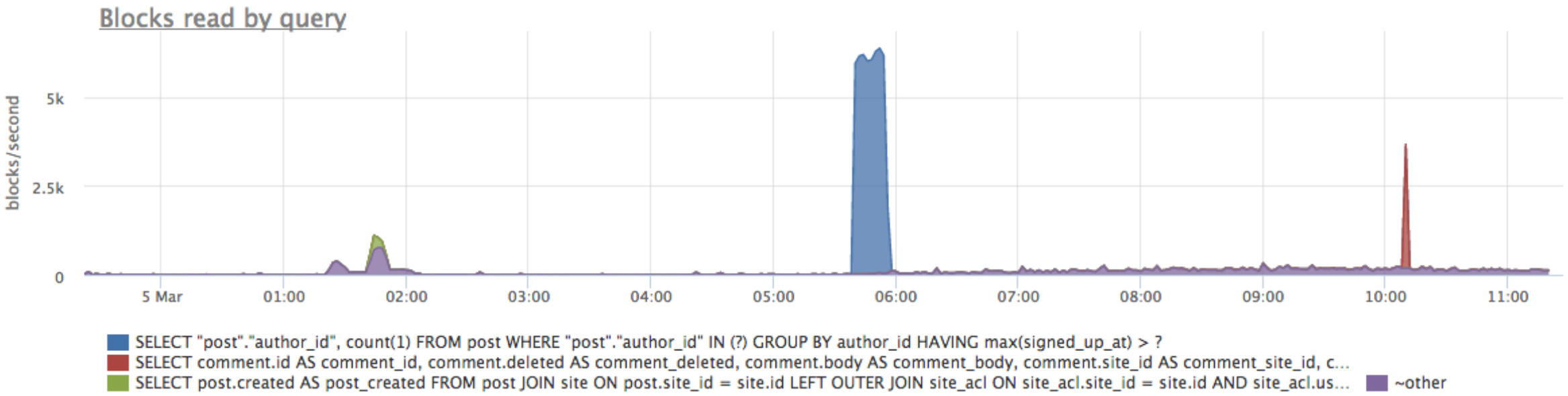
- Процесс делает системный вызов read()
- Он отражается в /proc/<pid>/io
- Но до диска может не дойти, если прочитался из page cache
- С другой стороны в /proc/<pid>/io попадает например чтение из stdin

Диск: кто грузит по чтению?



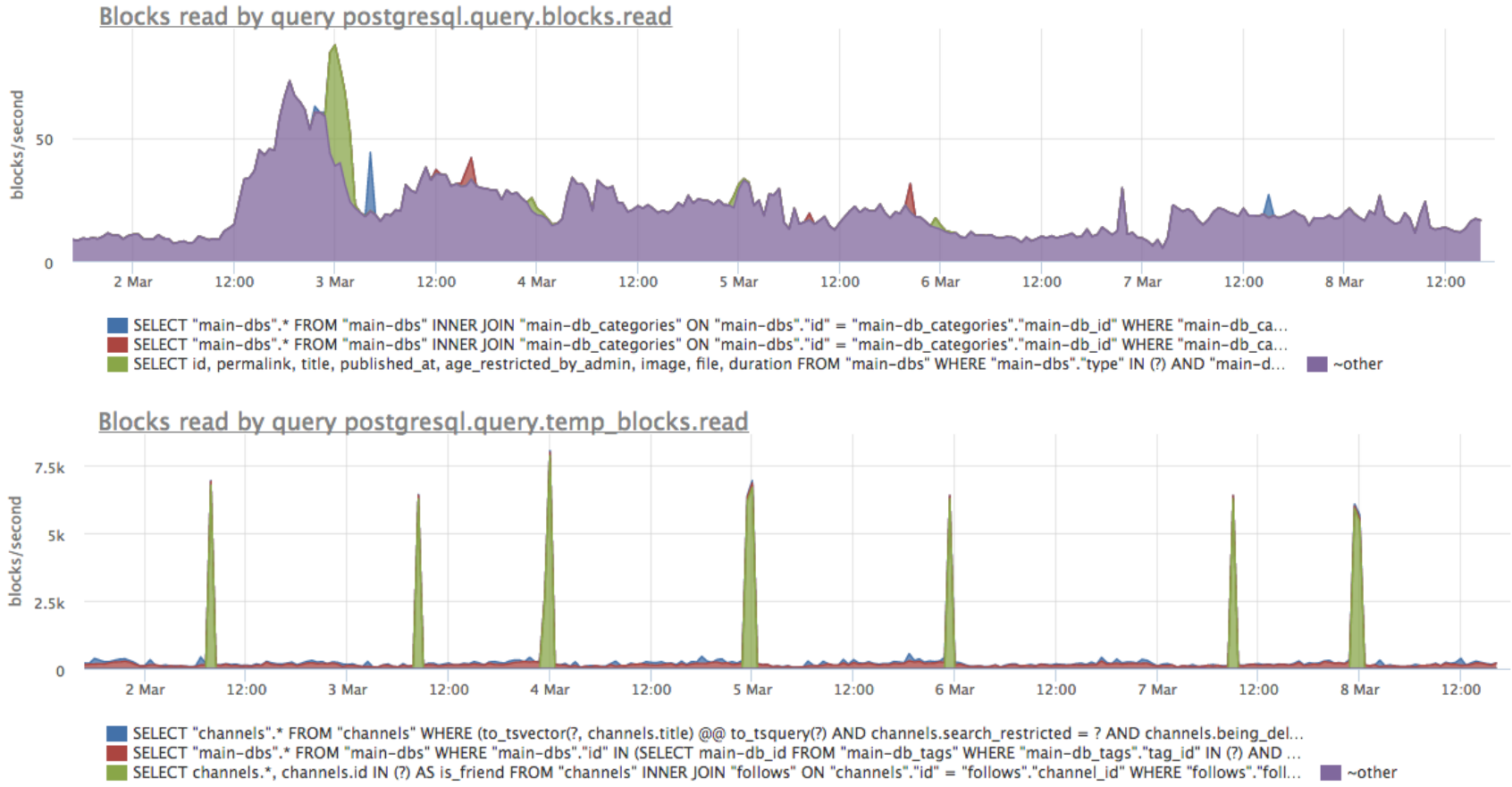
okmeter

Диск: кто грузит по чтению?



okmeter

Диск: blks_read **VS** templ_blks_read



okmeter

Диск: blks_read VS templ_blks_read

- Разные запросы: нужно видеть и одни и другие

- Чиним по-разному:

  - heap: добавляем памяти/меняем запрос/перекладываем данные по-другому/ускоряем дисковую подсистему

  - temp: увеличиваем work_mem/меняем запрос/переносим на аналитическую реплику

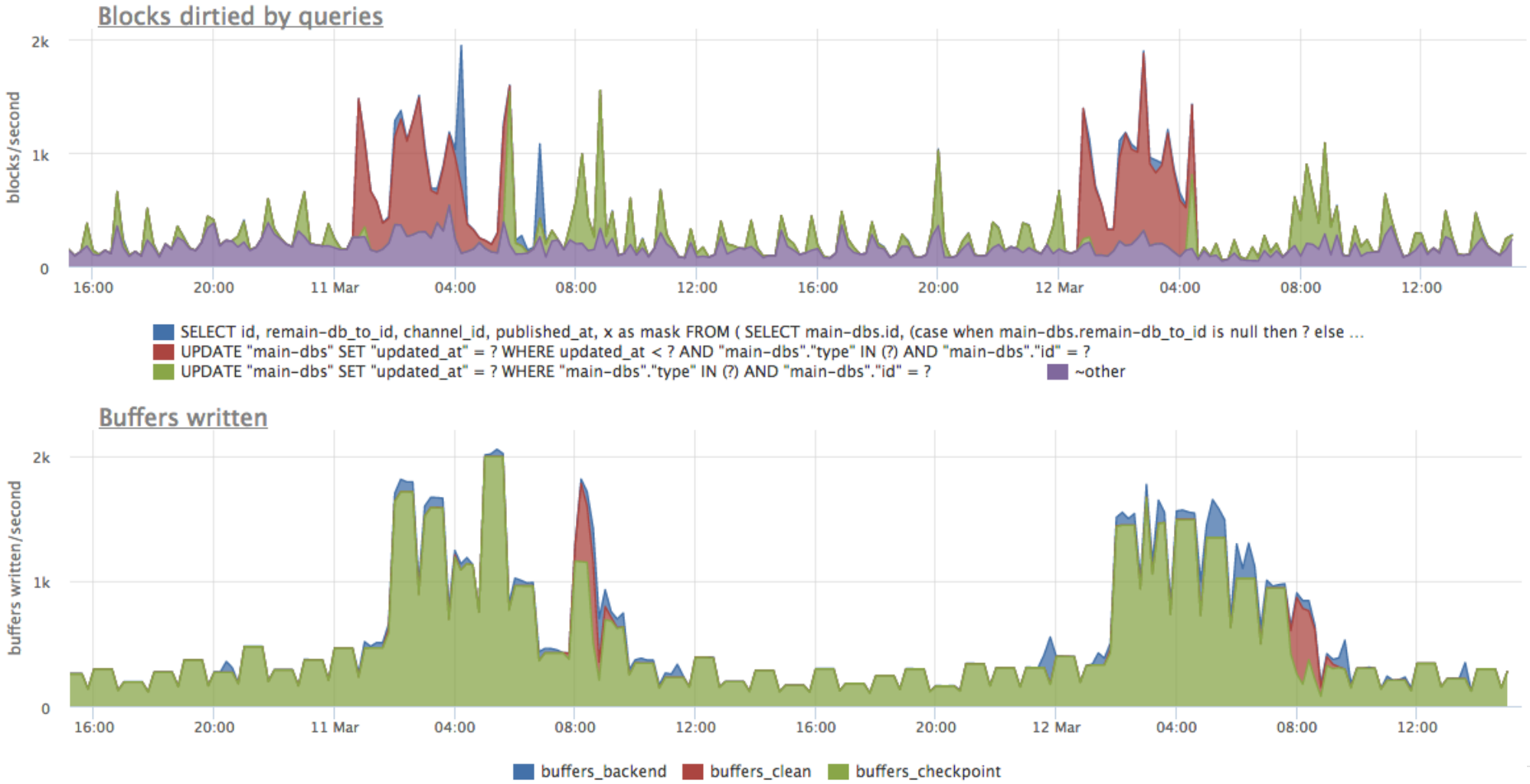
Ресурсы: диск.запись

- WAL
- Синхронизация “грязных” страниц
- Временные файлов
- Запись временных таблиц

Ресурсы: диск.запись

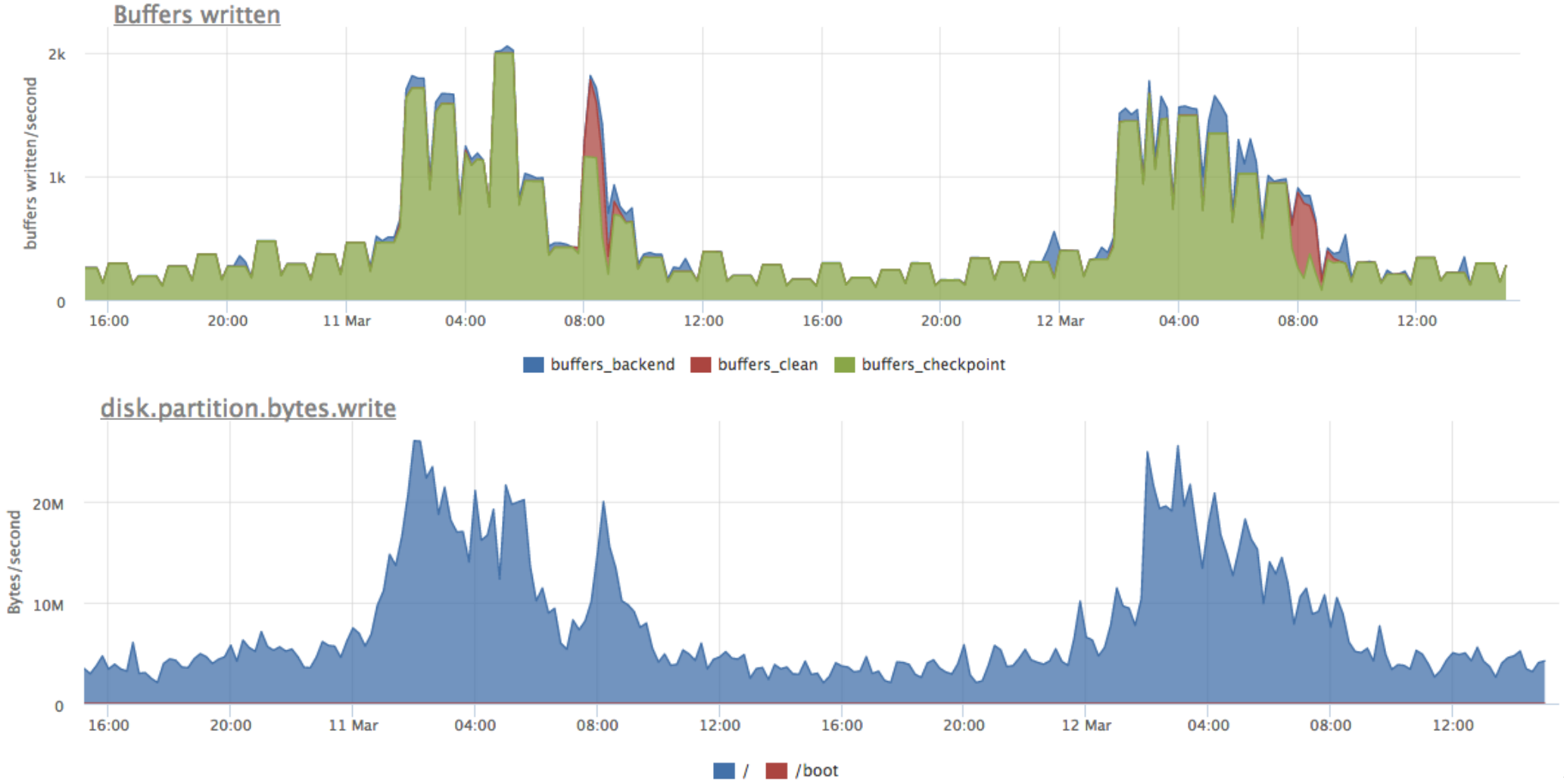
- Любые изменения “пачкают” страницы и записываются в WAL
- Можем ориентироваться только по dirty pages
- Запись “грязных” страниц на диск **не синхронна**
- Пишет или **backend** или **checkpointer** или **bgwriter**

Диск: dirty pages - checkpointer



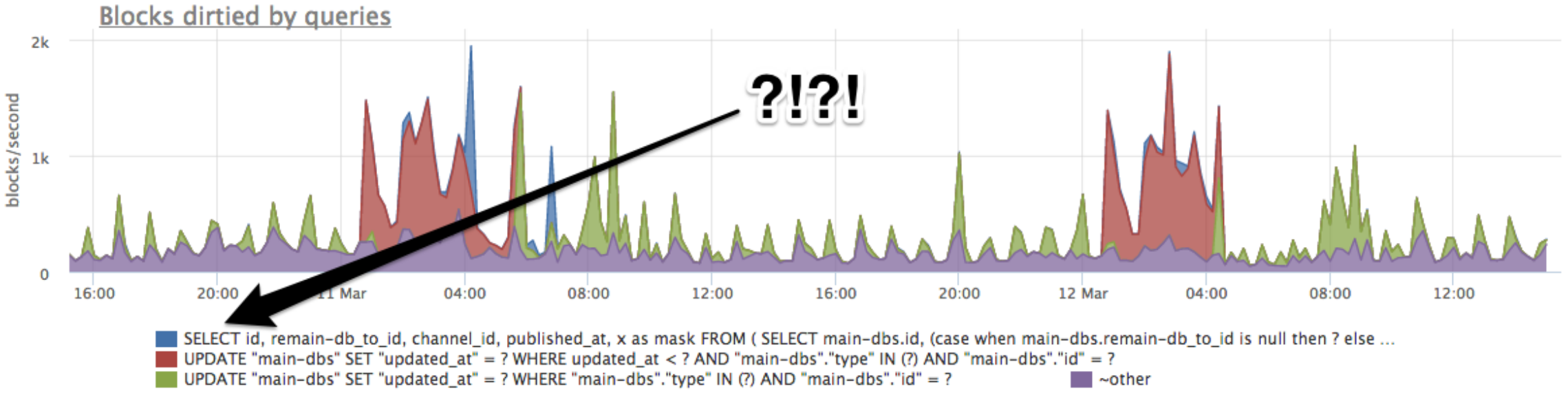
okmeter

Диск: checkpointer – disk traffic



okmeter

WTF: SELECT пачкает страницы?

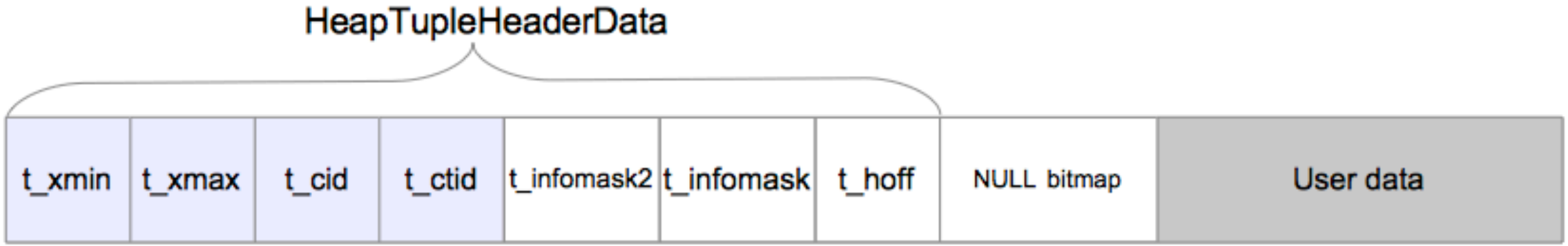


okmeter

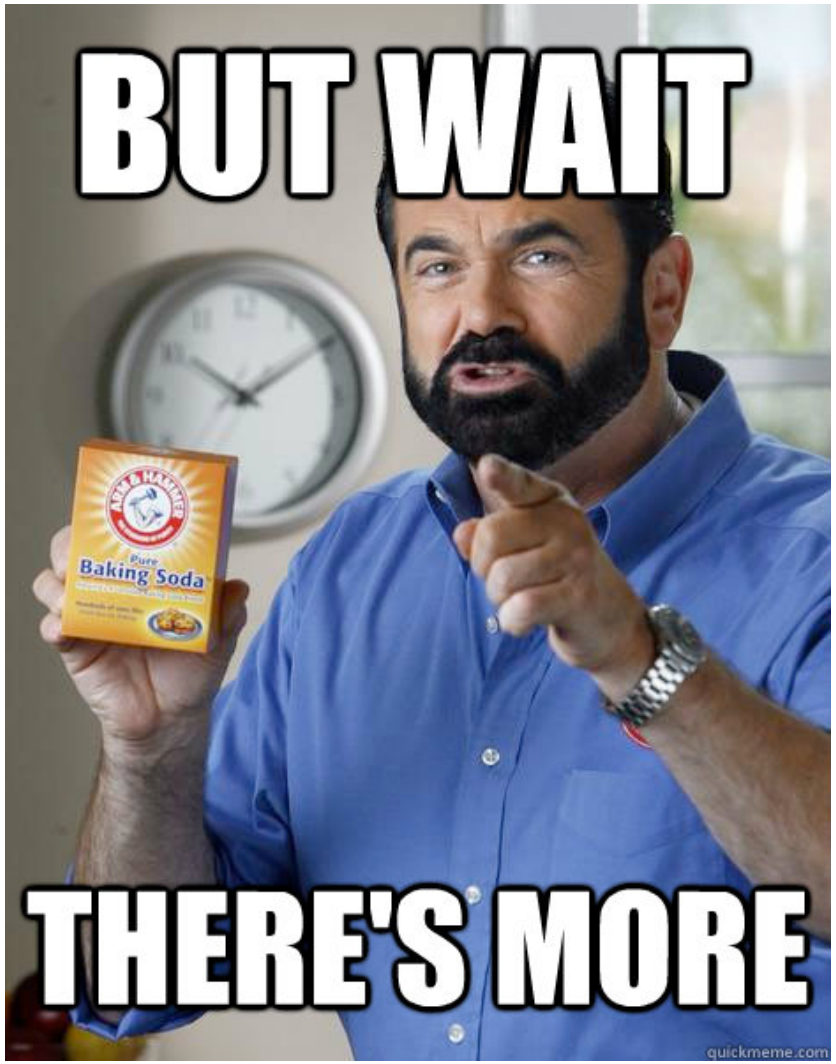
WTF: SELECT пачкает страницы?

okmeter

WTF: SELECT пачкает страницы?!?!



- INSERT в транзакции txid1 -> tuple (t_xmin=txid1)
- SELECT не знает завершилась ли txid1 (commited/aborted)
  - Идем в CLOG узнавать статус txid1 (это достаточно дорого)
  - Если txid1 завершена, сохраняем **hint bits** в tuple (t_infomask)
  - Помечаем **страницу** как “грязную”



А ещё SELECT может вызывать синхронную запись

SELECT может вызывать запись

- Если при чтении нам нужно прочитать страницу, которой нет в buffer cache, мы читаем ее с диска
- Если в buffer cache нет свободного места, нужно что-то вытеснить
- Если кандидат на вытеснение помечен как dirty, **мы его пишем на диск**

Ресурсы: сеть

- PG не предоставляет информацию по количество результирующих байт по запросу

- Можно **очень примерно** посчитать по pg_stat_statements.**rows**

Итого:

- Нужно хорошо понимать, как PG работает с ресурсами сервера
- Осознавать физический смысл метрик, на которые вы смотрите
- Обязательно нужно объяснять для себя найденные аномалии, это позволит полностью разобраться с деталями реализации той или иной подсистемы

Спасибо за внимание!

Вопросы?

Николай Сивко

nsv@okmeter.io