OZON tech

Российский опенсорс-пулер PgDoorman

Дмитрий Васильев

Эксперт по разработке информационных систем,
группа PostgreSQL DBA, Ozon

AGENDA 🖍️

01 Пулер в Ozon

02 PgBouncer

02 Odyssey

04 Что было на рынке: PgCat

05 PgDoorman

06 Получилось здорово, хотим поделиться с сообществом

01

Пулер в Ozon

PostgreSQL в Ozon сегодня

45K

Инстансов баз данных

10K/32K

В проде

~14 mil

Всего TPS

~10 PTB

Данных

~62K

Ядер CPU

А нужен ли вам DB-пулер?

threads/clients	TPS				
	direct connection	default_pool_size			
		14	56	112	600
56	4605,54	1498,70	1818,56	1821,72	1811,76
150	983,39	1562,12	1786,84	1783,84	806,72
300	337,87	1452,15	1791,35	1763,56	323,11
600	223,33	1535,95	1844,16	1768,83	253,99

02

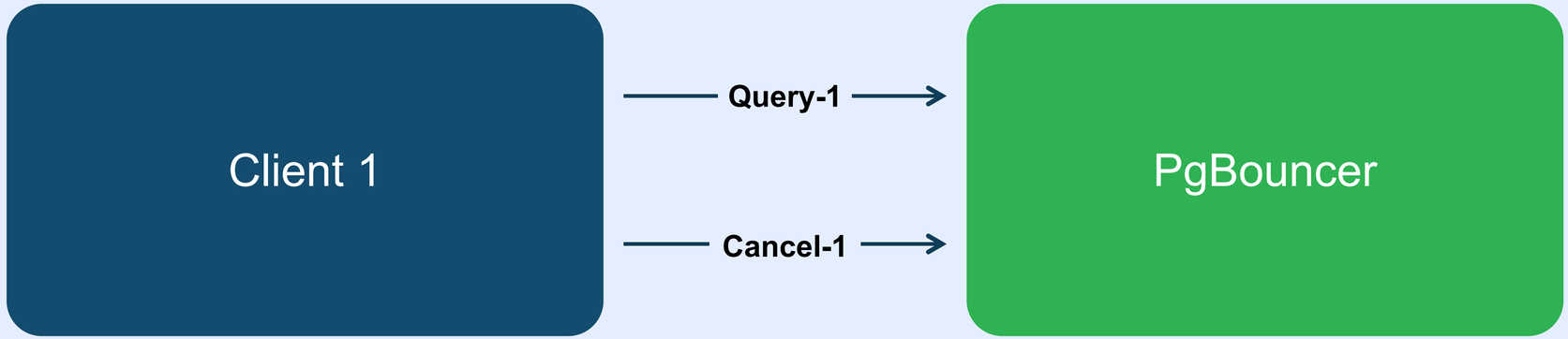
PgBouncer

Задача: получить **15К tps** с приемлемым **p99**

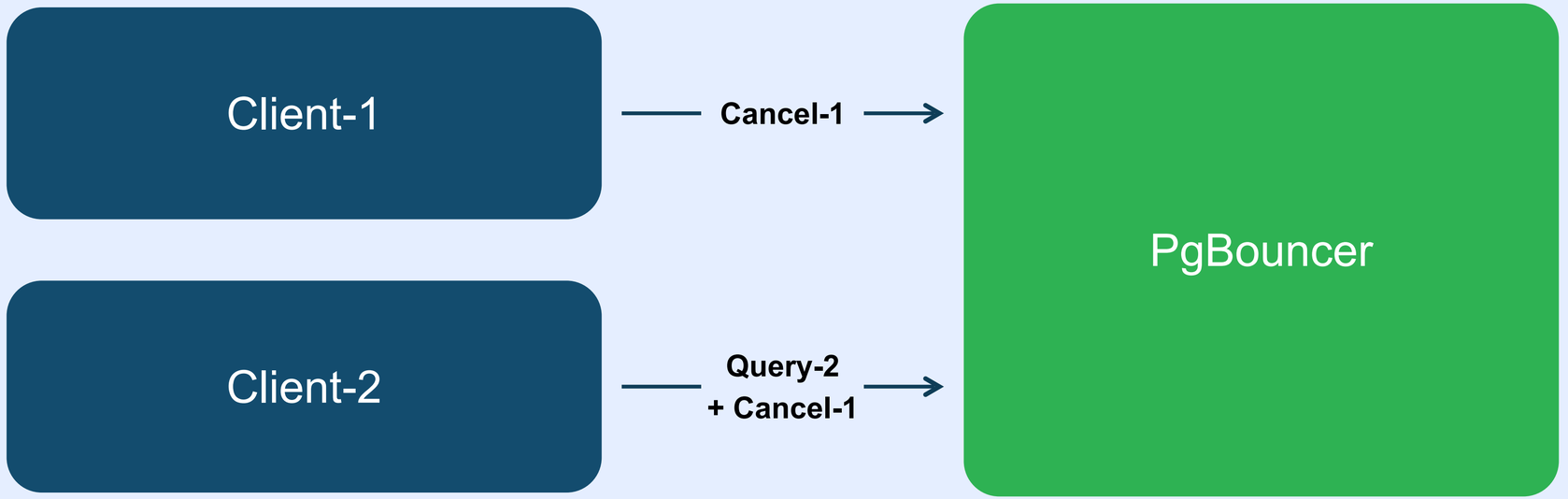
Мы жили с PgBouncer, пока на одном из проектов с 128 шардов не упёрлись в производительность:

- некоторые шарды отлично держали нагрузку
- некоторые резко деградировали в логах переподключения клиентов
- залить железом не получалось

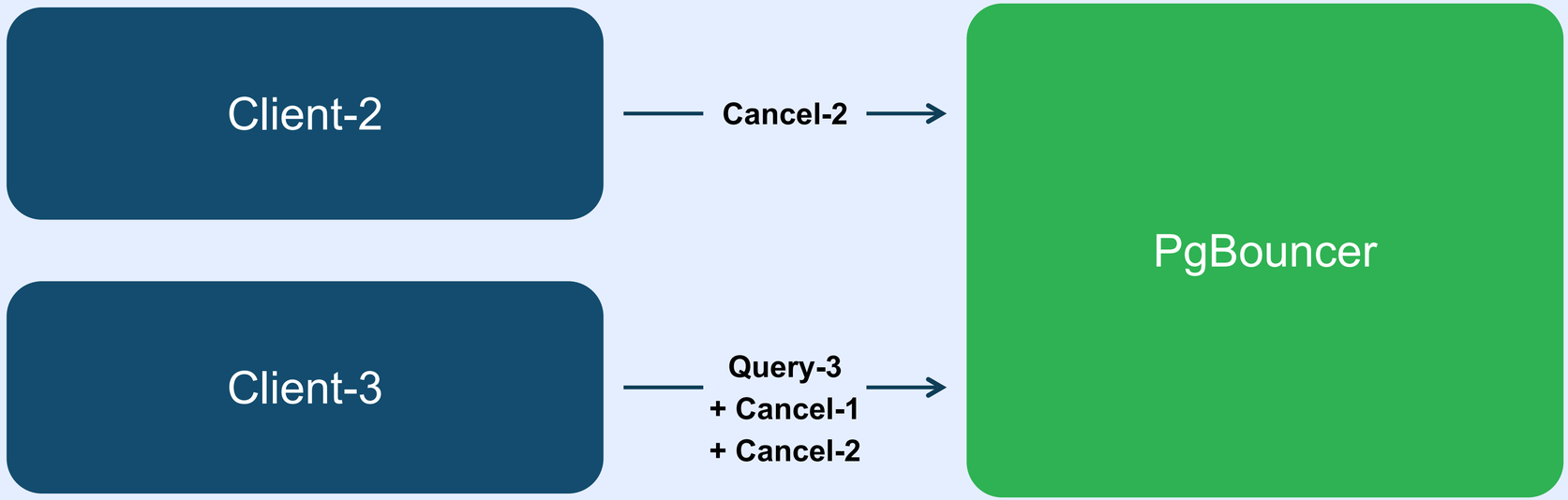
PgBouncer: cancelation storm



PgBouncer: cancelation storm



PgBouncer: cancelation storm



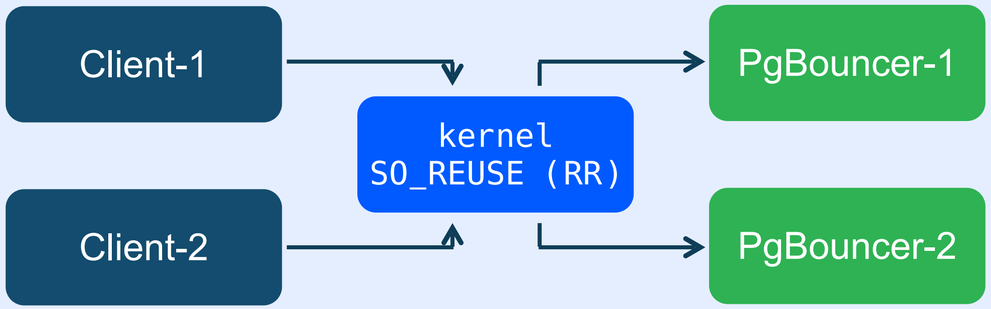
Окей, добавляем multiple PgBouncers

Поднимаем несколько инстансов PgBouncer на одном порту: 2-4-8 штук

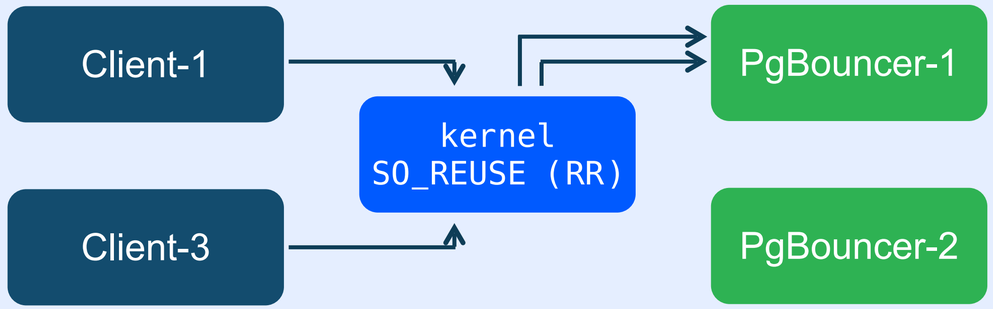
- Изменили **1 x pool_size = 80** → **8 x pool_size = 10**
- Стресс-тесты начали проходить, но только после рестарта пулеров :)
- Заметили, что некоторые серверные бэкенды по нулям, а у некоторых PgBouncer’ов заметно меньше клиентов, чем у других (разница до 2-4 раз)

SO_REUSE распределяет R-R, а клиент отключается, когда захочет

t1



t2



PgBouncer — прощай!

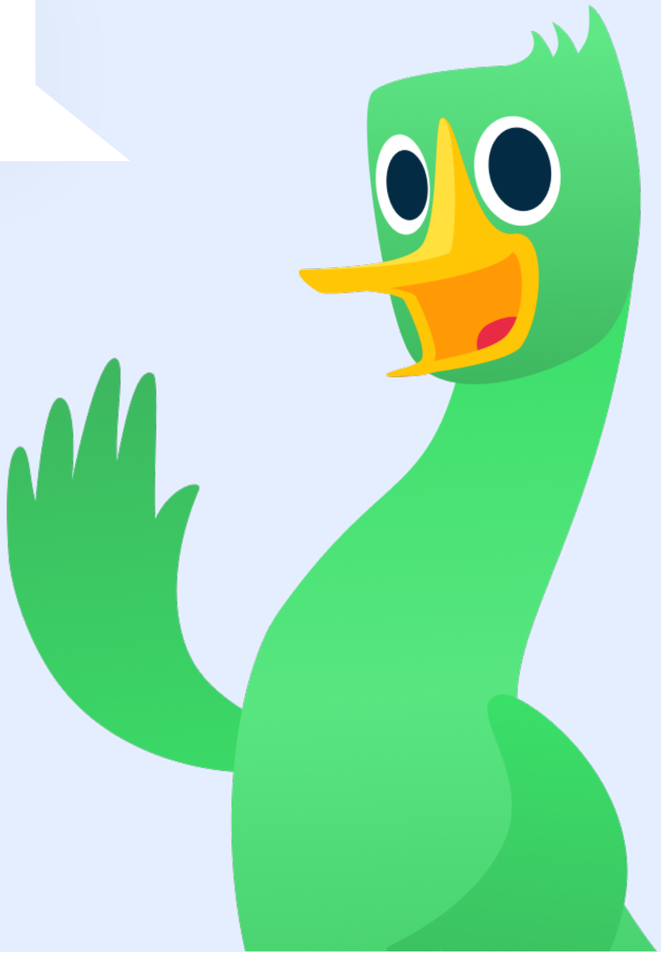
К качеству исполнения и поддержки клиентов вопросов нет. Возможно, потому что это очень старый продукт: к нему все привыкли и знают все его баги.

- Долгое время практически не развивался до появления Odyssey
- Достаточно простой и имеет ограниченный функционал (v1.16)
- Однопоточный event-loop
- По ощущениям хватает до **5-10K tp/s**
- Не вывозит много TLS-соединений

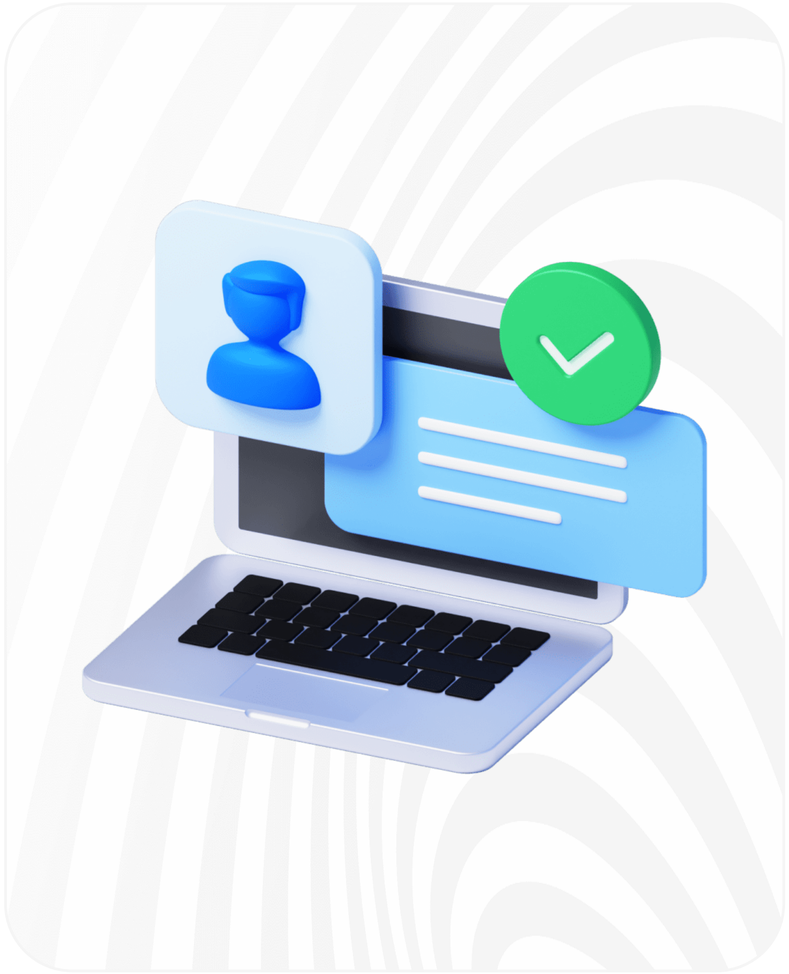
Здравствуй, Odyssey!

Глоток свежего воздуха!

- Запросто держит **20-30K tp/s**
  - Запрос «;» — **120K tp/s**
- Переживает cancelation storm
- Базовая поддержка prepared statements

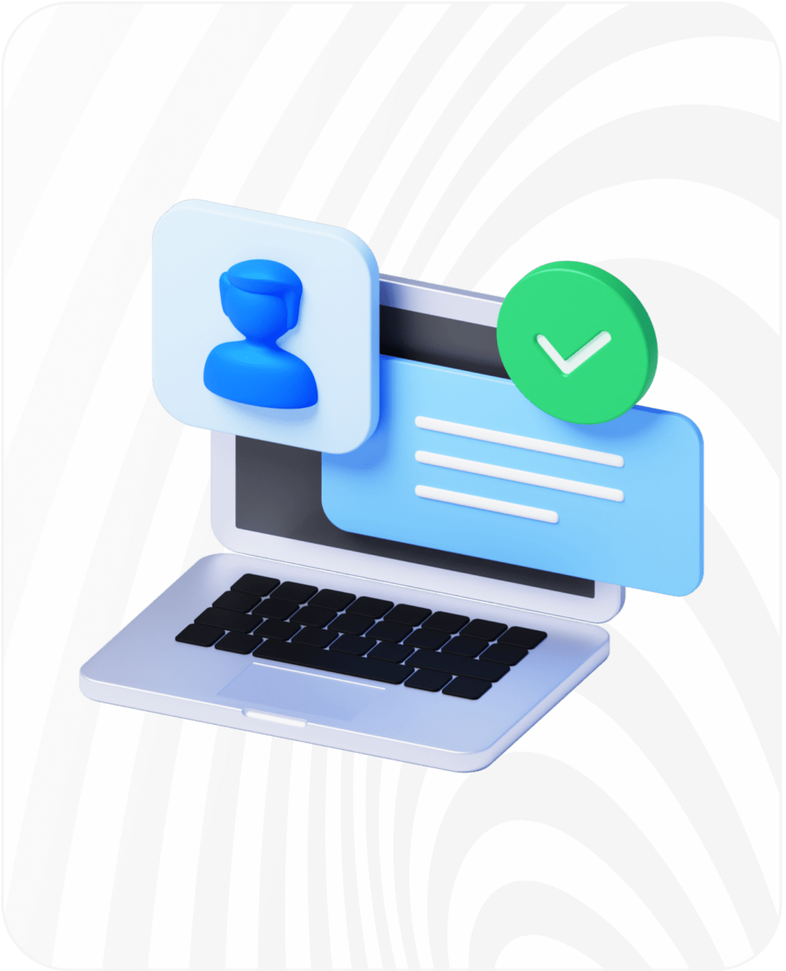


Окей, prepared_statements + pgx нам помогли



Окей, prepared_statements + pgx нам помогли

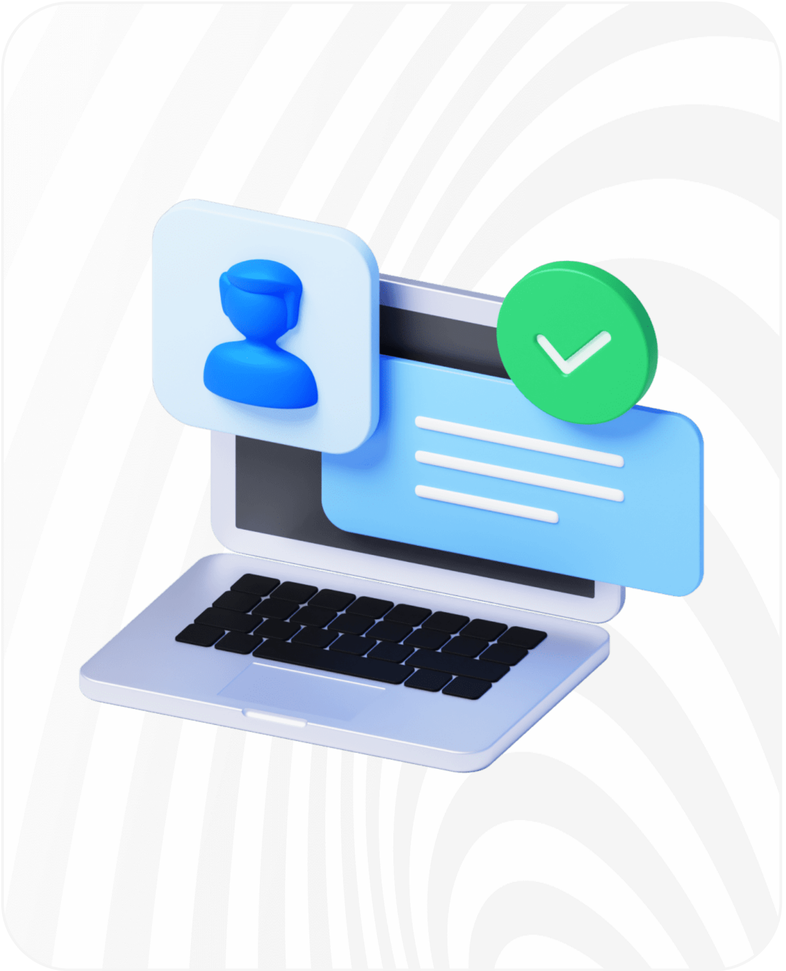
Мы достигаем целевых значений на Odyssey после включения prepared



Окей, prepared_statements + pgx нам помогли

Мы достигаем целевых значений на Odyssey после включения prepared

Руководство довольно: теперь внедряем это на всех



03

Odyssey

Odyssey: проблемы при внедрении

Не работает SCRAM с dotnet Npgsql (channel binding)

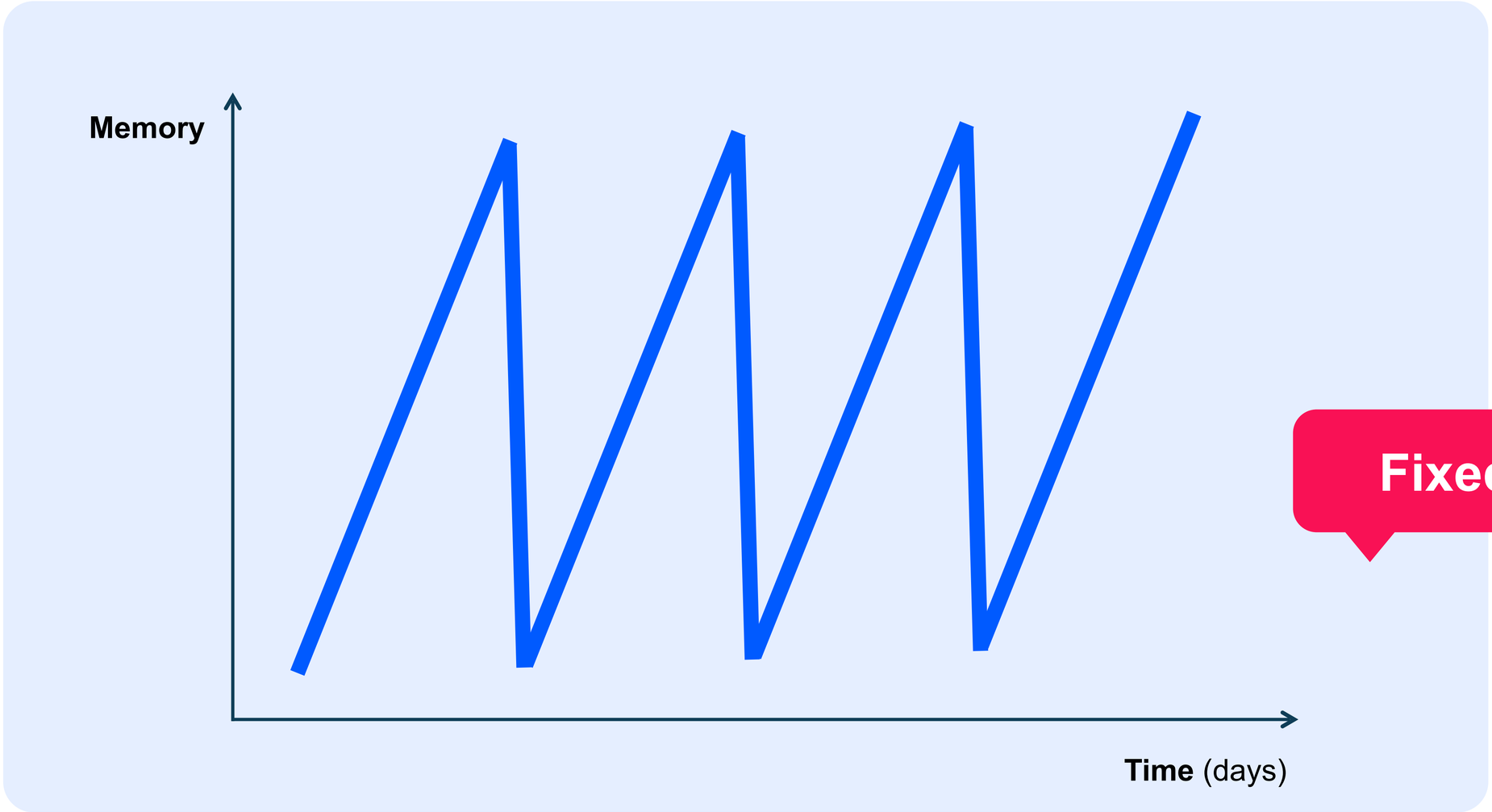
- Маленький грязный патч помог

Отключаем prepared statements по дефолту, чтобы не навредить dotnet Npgsql :(

Не умеет HBA

- Достали старый патч и дотолкали в master

Odyssey: частые OOM



Odyssey: OOM

Odyssey: OOM

После каждого OOM у клиента в пуле остаются коннекты с оборванным TCP-соединением, запись в него — 500’ка в сервисе

Odyssey: OOM

- После каждого OOM у клиента в пуле остаются коннекты с оборванным TCP-соединением, запись в него — 500’ка в сервисе

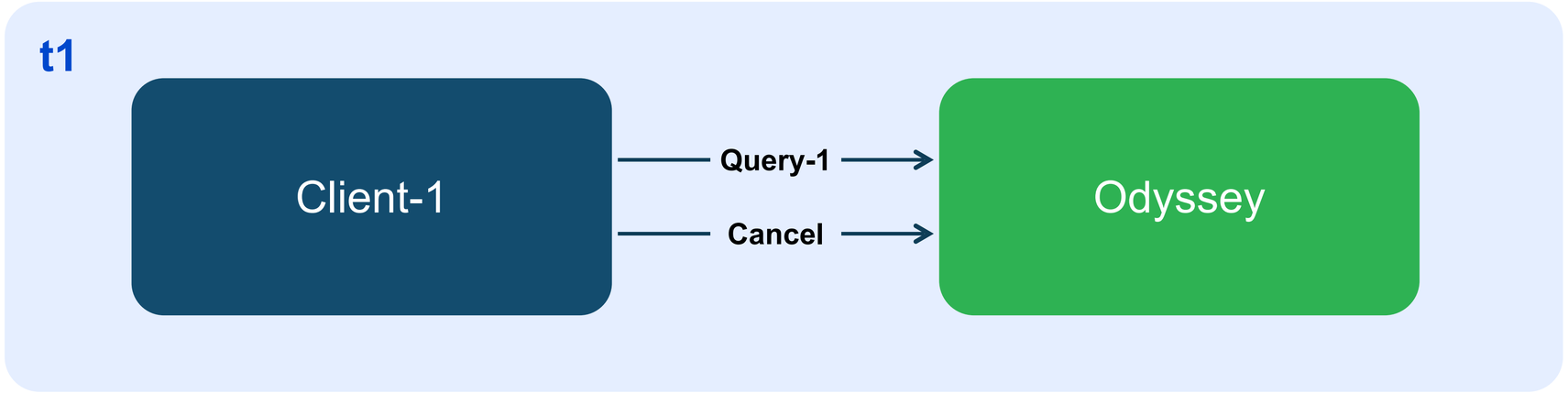
- Из-за LRU использования доступ к сломанному TCP-соединению происходит отложено, фон ошибок может продолжаться LifeTime/IdleTimeout

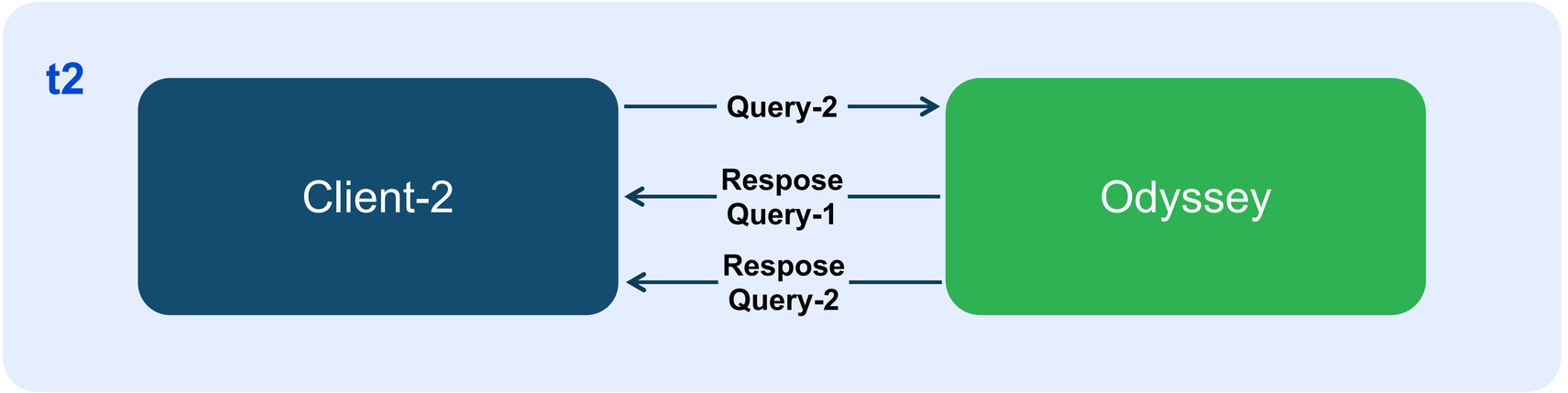
Odyssey: OOM

- После каждого OOM у клиента в пуле остаются коннекты с оборванным tcp-соединением, запись в него — 500’ка в сервисе
- Из-за LRU использования доступ к сломанному tcp-соединению происходит отложенно, фон ошибок может продолжаться LifeTime/IdleTimeout
- Победили на стороне клиента — каждые N-секунд фоново пингуем открытый коннект

Odyssey: проблемы, выявленные в процессе эксплуатации

Баг **очень редкий**, ещё реже повторяется при TLS

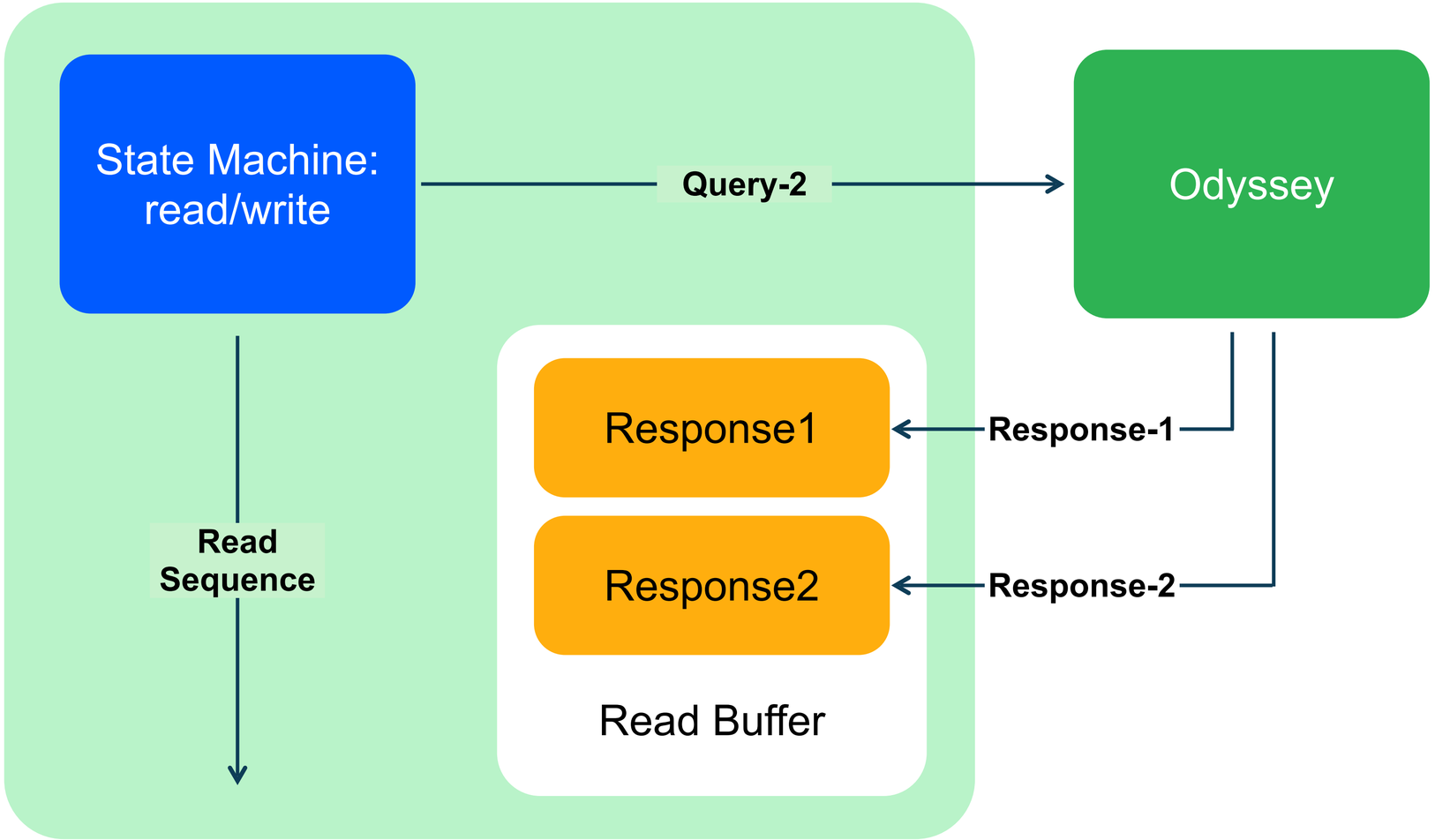




Fixed

Odyssey: проблемы, выявленные в процессе эксплуатации

Баг **очень редкий**, ещё реже повторяется при TLS



- Клиент асинхронно скачивает ответы в Read Buffer.

  В итоге вся очередь ответов в этом буфере отравлена неправильными ответами не на свой запрос.

Fixed

Включение TLS — воркэраунд, фоново ищем решение

Включение TLS — воркэраунд, фоново ищем решение

Учим разработчиков включать sslmode в приложении. Это тяжело (вкомпилено)!

Включение TLS — воркэраунд, фоново ищем решение

Учим разработчиков включать sslmode в приложении. Это тяжело (вкомпилено)!

Даём ручки для запрета подключения не по TLS

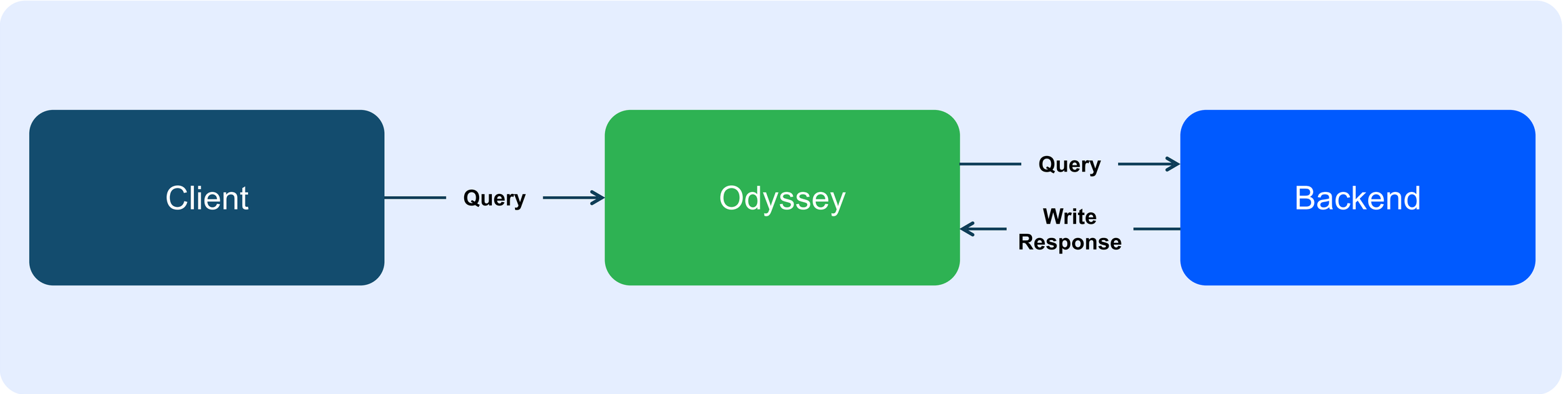
Odyssey: CPU usage при «толстом» ответе

```
time PGPORT=odyssey PGSSLMODE=require psql -Atc "select repeat( ‘1’, 40000000)» > /dev/null

real 0m20.812s

time PGPORT=pgbouncer PGSSLMODE=require psql -Atc "select repeat( ‘1’, 40000000)» > /dev/null

real 0m0.714s
```

Fixed

Возвращаем пользователей с «толстым ответом» на PgBouncer

Odyssey: обновления приносили сюрпризы



Мы поняли, что…

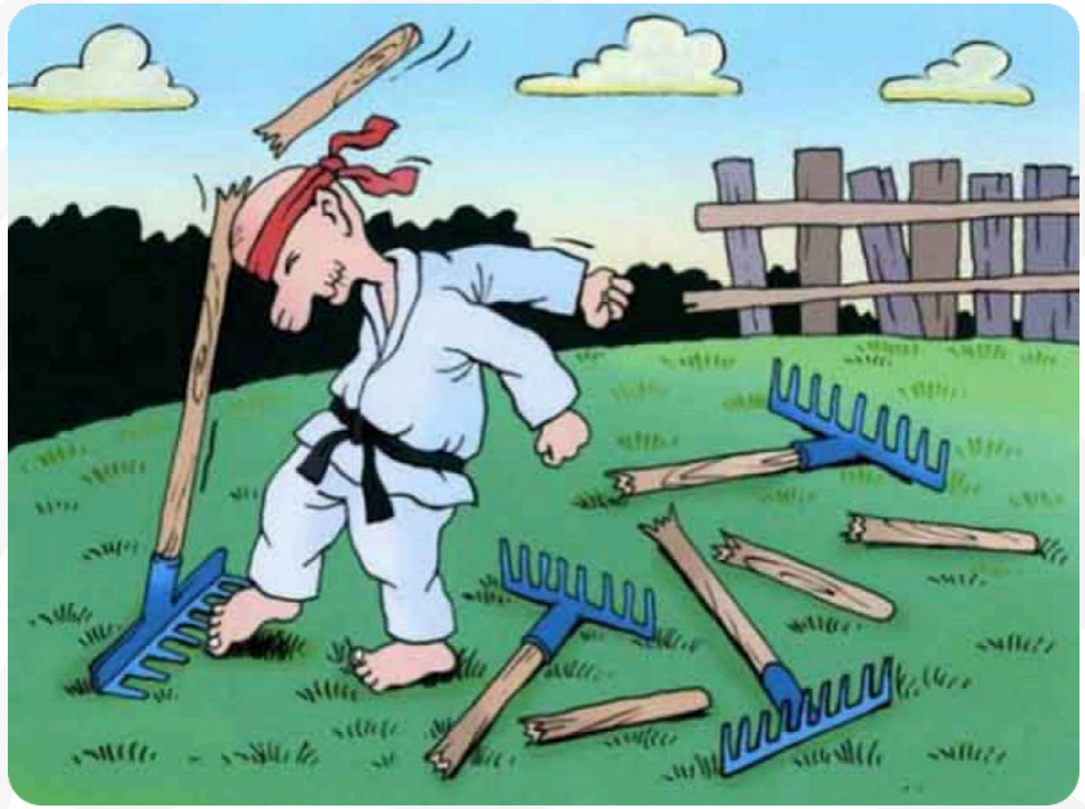
04

Что было на рынке: PgCat

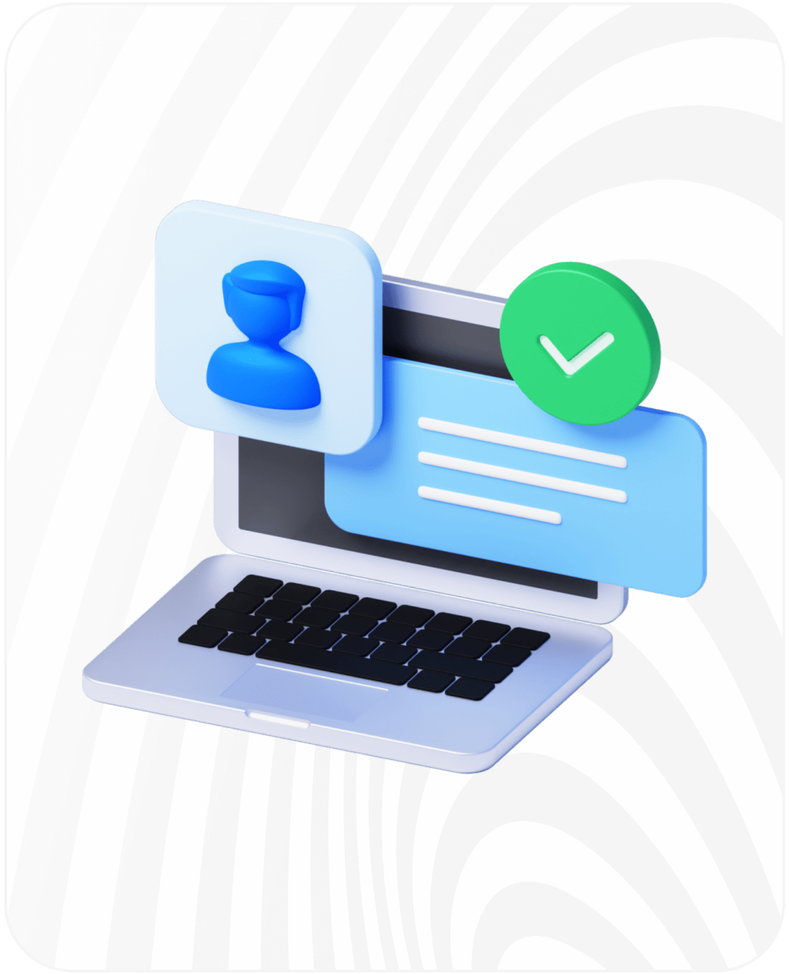
Реализации: PgCat

- Базовая поддержка **prepared statements**
- **«Многопоточный»** как Odyssey
- Поддержка **шардинга**
- Поддержка **балансировки**
- **Auth passthrough**
- **Mirroring**

PgCat внедряем?

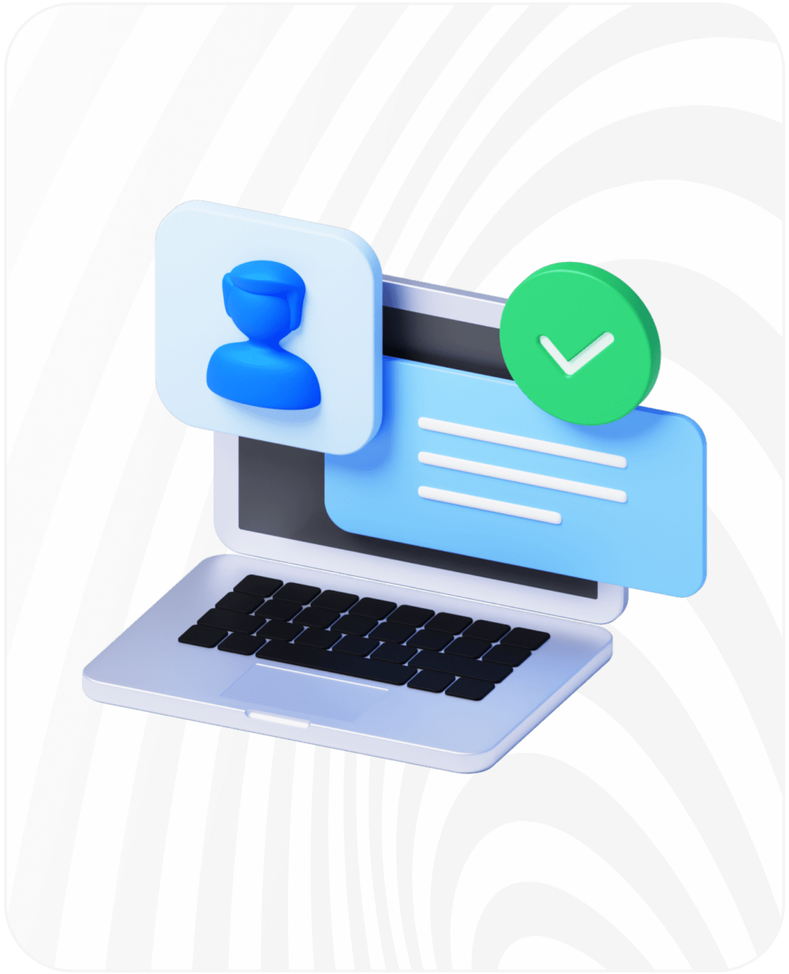


Реализации: PgCat



Реализации: PgCat

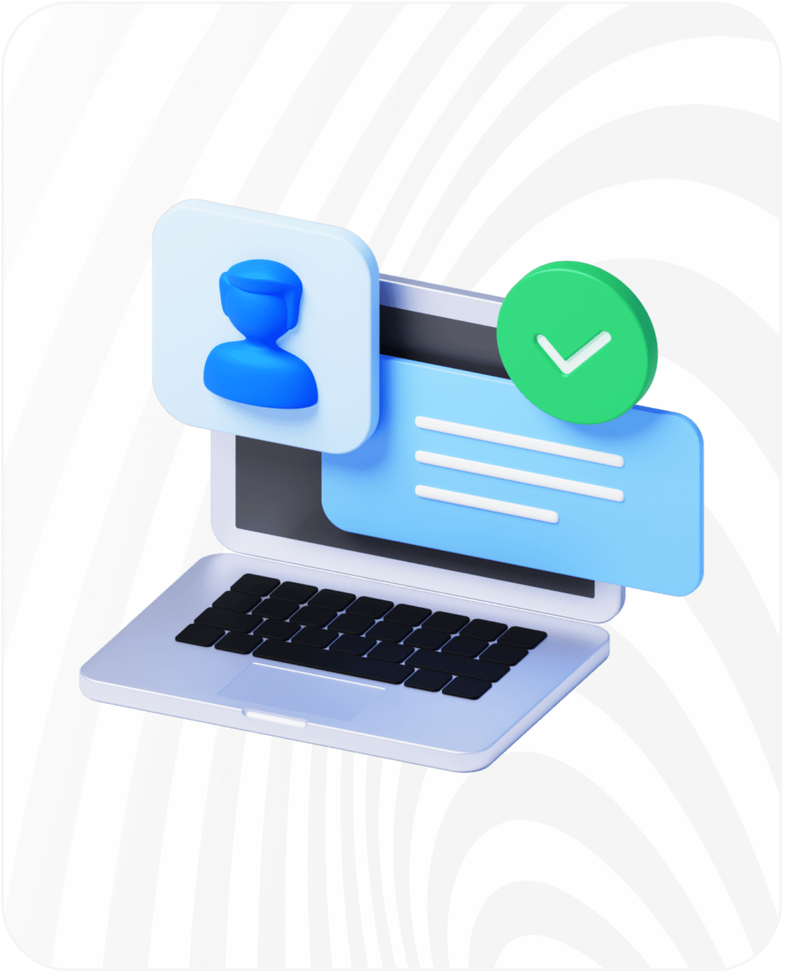
Для достижения уровня производительности Odyssey базового перформанса не хватало



Реализации: PgCat

Для достижения уровня производительности Odyssey базового перформанса не хватало

Показался слишком жирным по функционалу



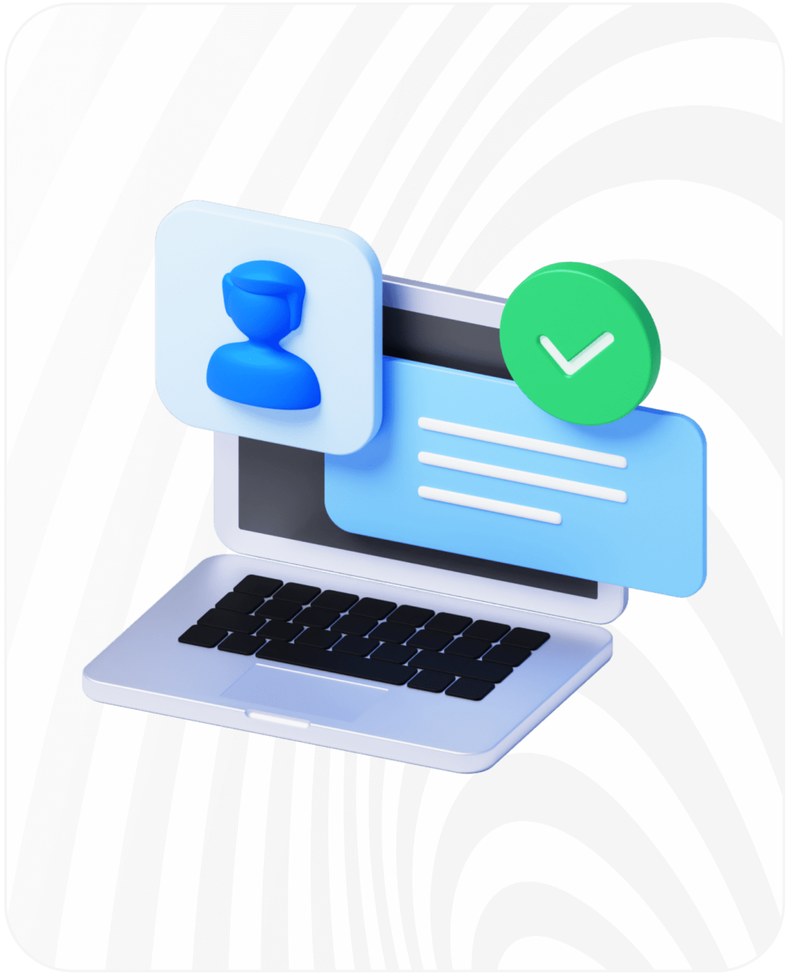
Реализации: PgCat

Для достижения уровня производительности Odyssey базового перформанса не хватало

Показался слишком жирным по функционалу

Ядро достаточно простое (3 файла)

- Асинхронность на Tokio



Реализации: PgCat

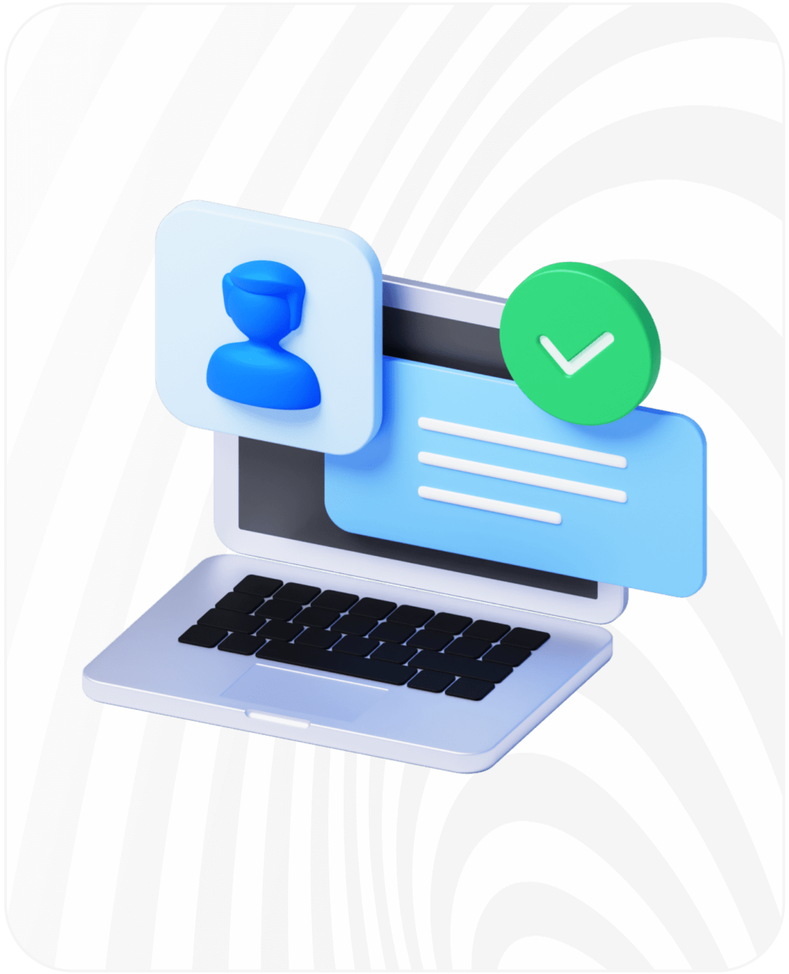
Базового перформанса не хватало, чтобы достичь уровня производительности Odyssey

Показался слишком жирным по функционалу

Ядро достаточно простое (3 файла)

- Асинхронность на Tokio

Хорошо покрыт тестами



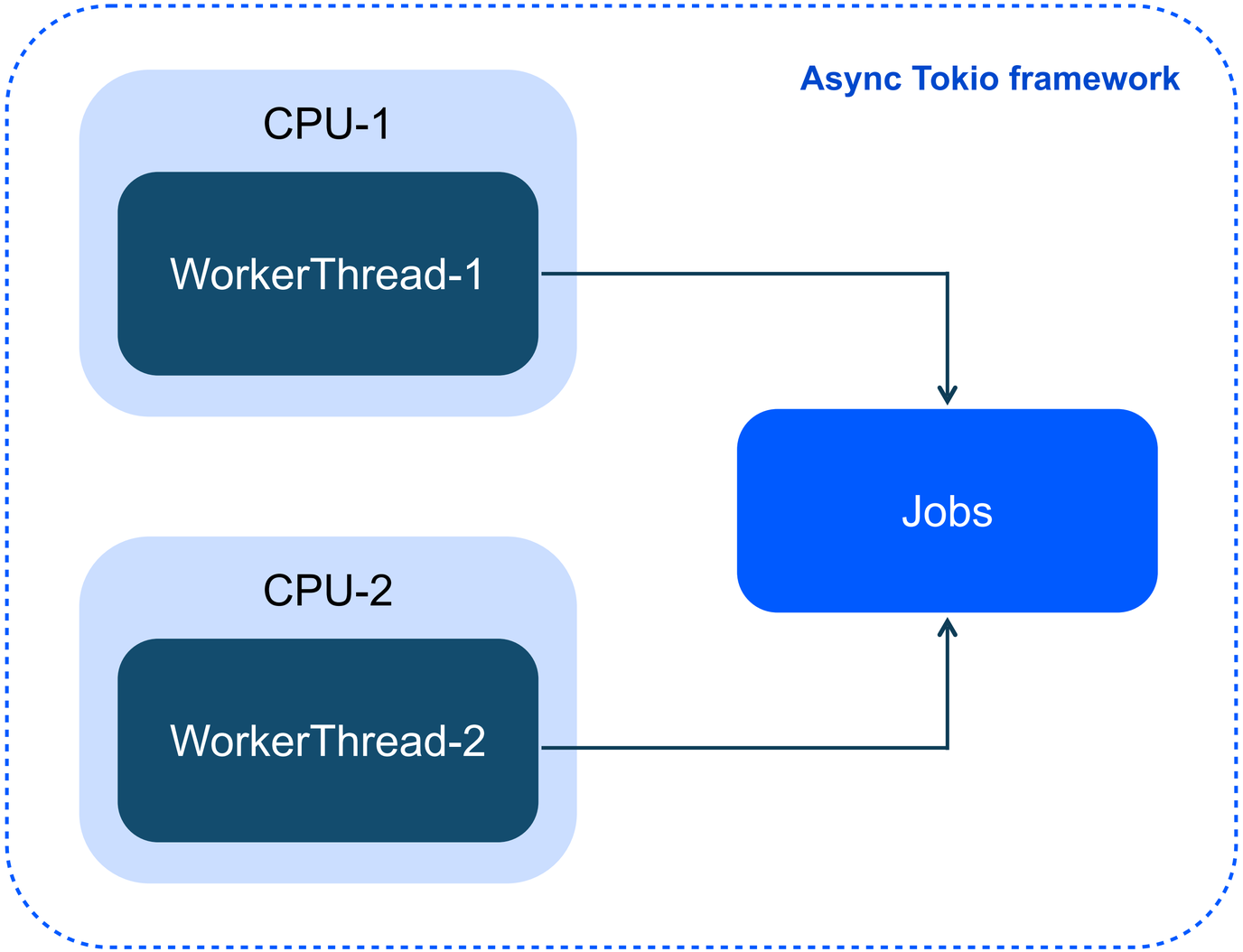
05

PgDoorman

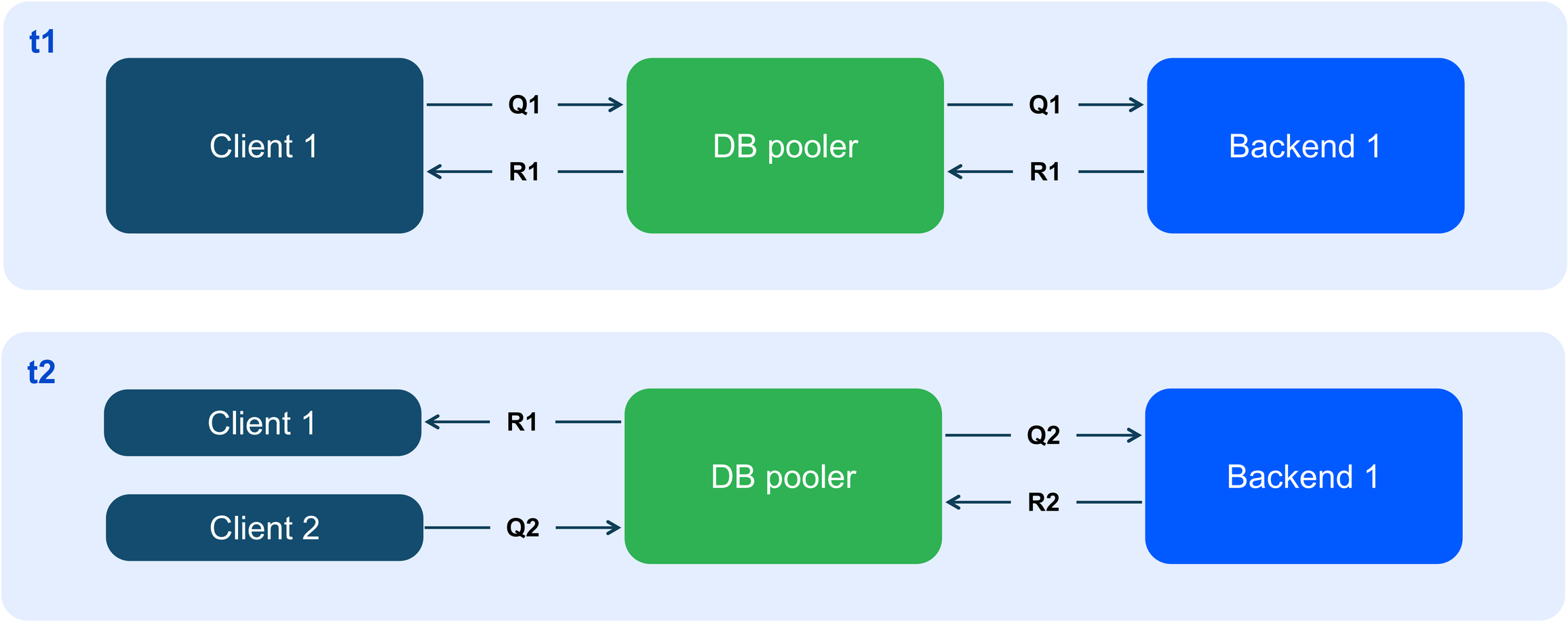
PgDoorman: что удалили из PgCat

- ~~Поддержка **шардинга**~~
- ~~Поддержка **балансировки**~~
- ~~**Auth passthrough**~~
- ~~**Mirroring**~~

Пиннинг тредов, настройка таймингов по дефолту на OLTP

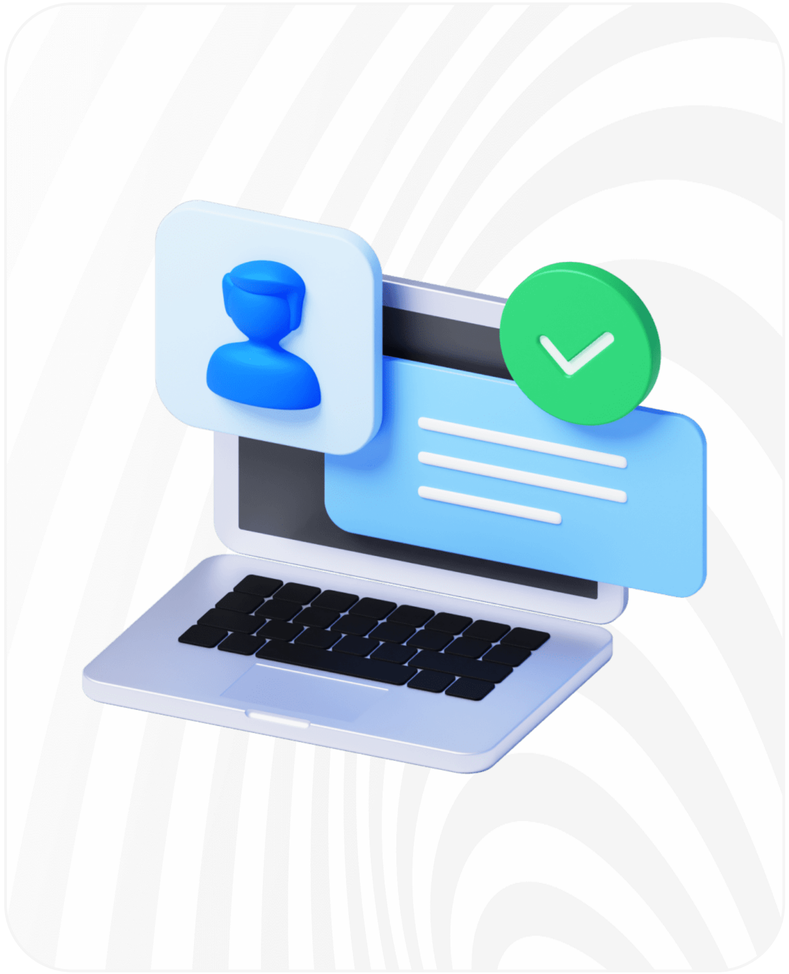


Если мы понимаем, что можем освободить серверный бэкенд, — освобождаем его, не дожидаясь записи в клиента



PgDoorman: что ещё добавили

Упор на производительность

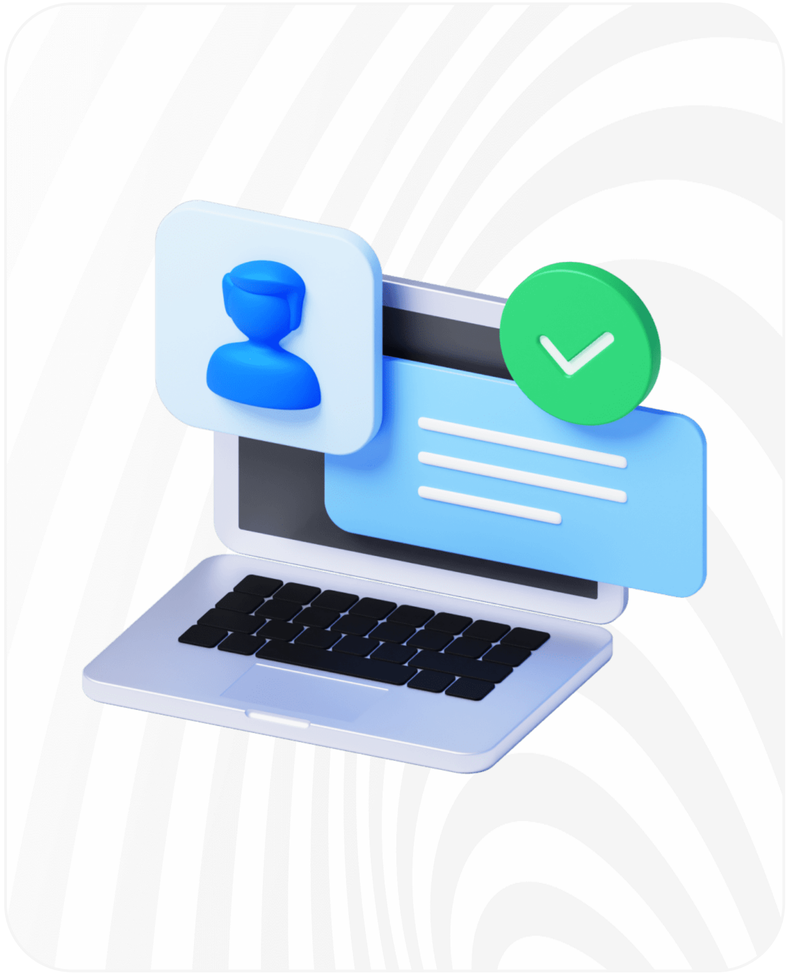


PgDoorman: что ещё добавили

Упор на производительность

Заменили пул bb2 на упрощенный форк deadpool

- Сделали пул не жадным (из-за connectivity check открывались все коннекты)



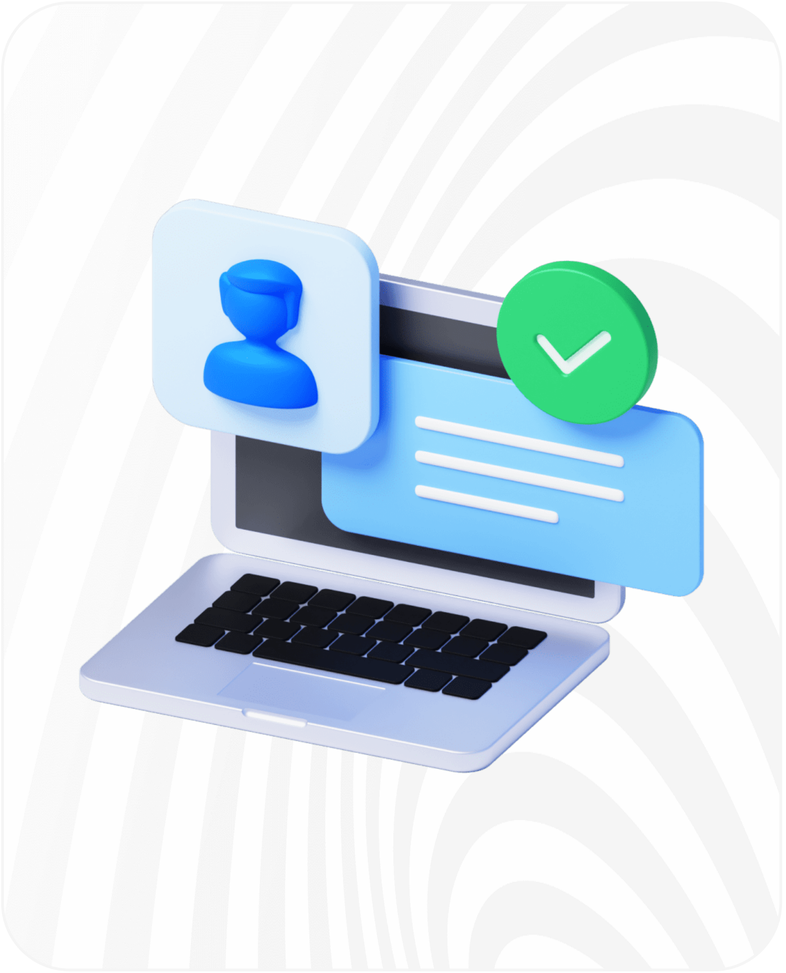
PgDoorman: что ещё добавили

Упор на производительность

Заменили пул bb2 на упрощенный форк deadpool

- Сделали пул не жадным (из-за connectivity check открывались все коннекты)

Backend — LRU policy (достаётся максимально прогретый)



PgDoorman: что ещё добавили

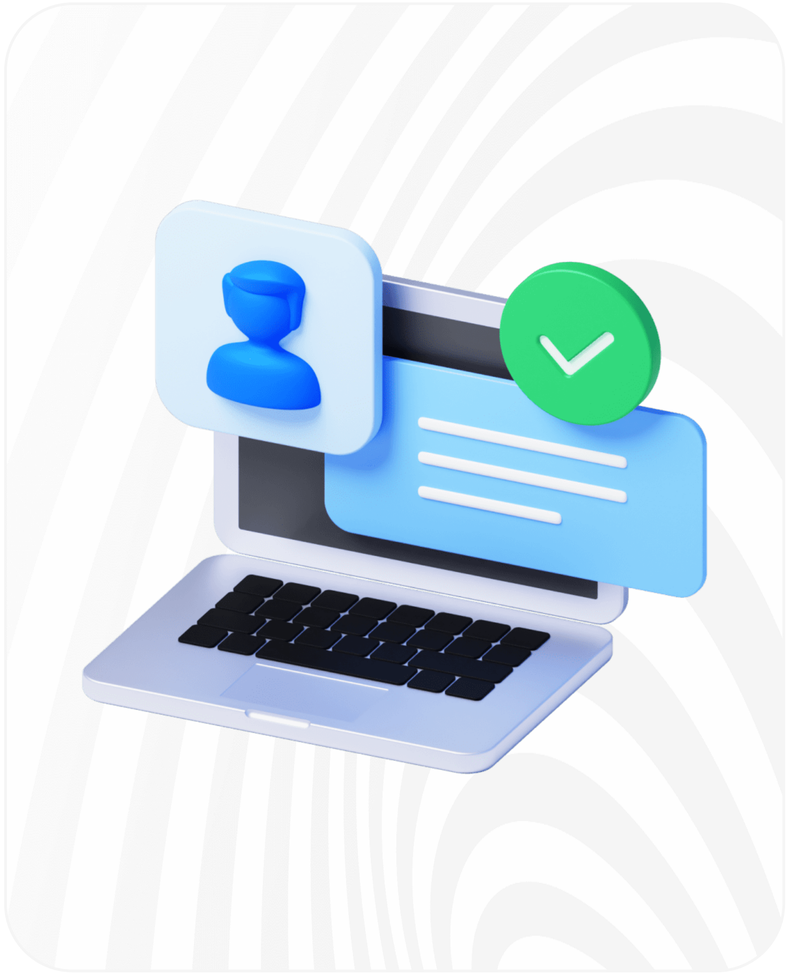
Упор на производительность

Заменили пул bb2 на упрощенный форк deadpool

- Сделали пул не жадным (из-за connectivity check открывались все коннекты)

Backend — LRU policy (достаётся максимально прогретый)

Добавили поддержку unix-socket (с регулировкой буферов)



PgDoorman: что ещё добавили

Упор на производительность

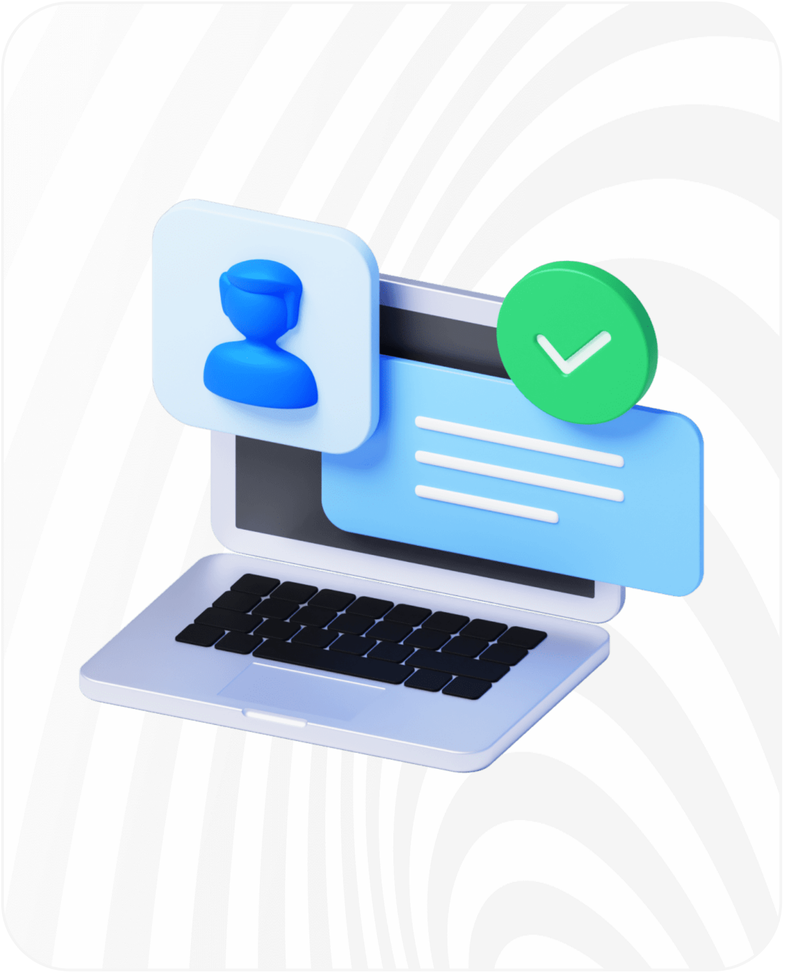
Заменили пул bb2 на упрощенный форк deadpool

- Сделали пул не жадным (из-за connectivity check открывались все коннекты)

Backend — LRU policy (достаётся максимально прогретый)

Добавили поддержку unix-socket (с регулировкой буферов)

Добавили несколько виртуальных пулов для уменьшения lock contention



PgDoorman: окей, достигли перформанса Odyssey

- Тестировали на старом проекте из 128 шардов
- Общие тесты показали стабильность решения
- Позволило сэкономить много CPU-ядер на машинах
- Начальство дало добро — внедряем на всех!

PgDoorman: внедрение в Ozon происходило около года

- Это очень длинная история, заслуживающая отдельного доклада
- За это время мы сделали много улучшений, направленных на стабильность
- Добавили поддержку почти всех популярных языков

06

Получилось здорово, хотим поделиться с сообществом

Как мы позиционируем PgDoorman?

PgDoorman — это:

- Многопоточный PgBouncer
- Толерантный ко множеству драйверов: включил и работай
- Мы стремимся к тому, что у вас есть пулер и он просто работает, вы не знаете его имени :)

Чем PgDoorman не является:

- Балансировщиком нагрузки
- Инструментом для шардирования вашей базы

Почему PgDoorman?

	PgDoorman	PgBouncer	Odyssey
Высокая скорость выполнения запроса при большом кол-ве tps на один инстанс	✓	–	✓
Эффективная работа в облаке	✓	–	✓
Поддержка фичей драйвера NPGSQL	✓	✓	–
Корректная работа со SCRAM	✓	✓	–
Показатели p99	**В 2 раза** по сравнению с Odyssey*	–	–

*по версии «МедСофт»

Попробуйте!

https://github.com/ozontech/pg_doorman

https://t.me/pg_doorman

На моём проекте нагрузка ниже 10К RPS.
Зачем мне PgDoorman?

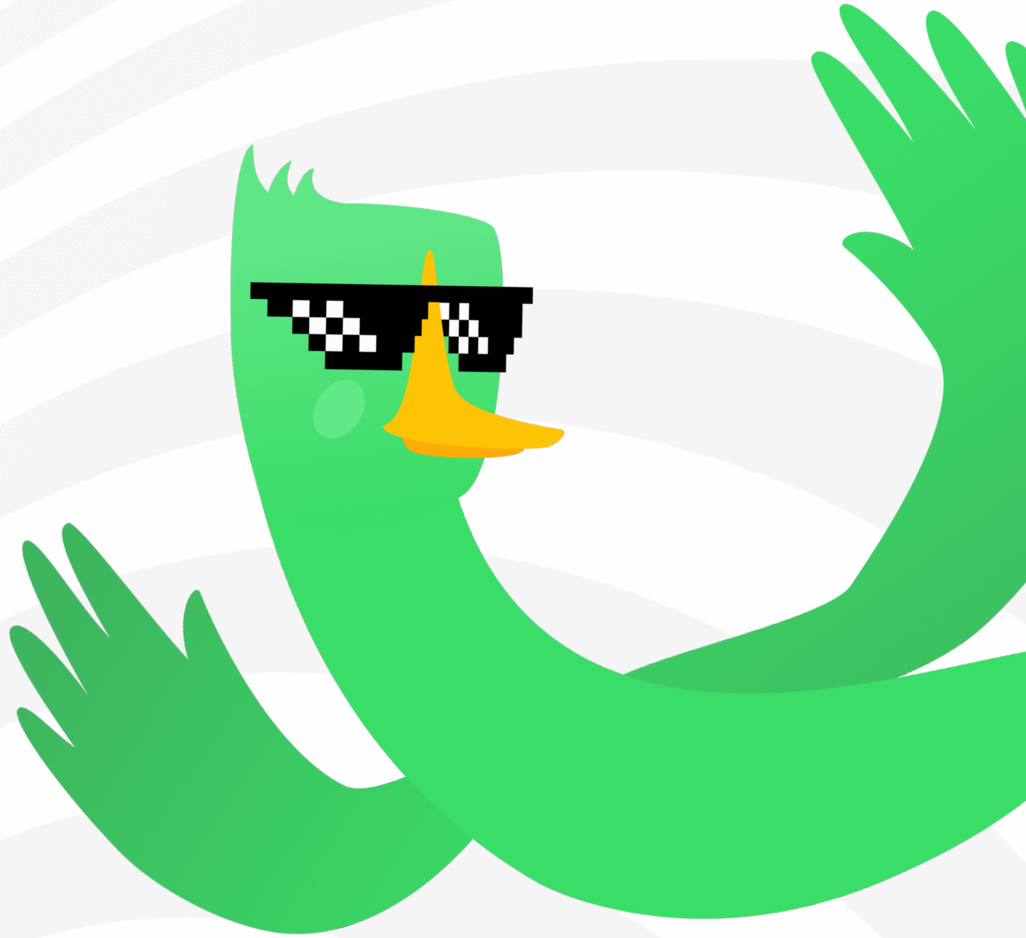
На проекте с нагрузкой ниже 10К RPS можно работать с PgBouncer, но, установив PgDoorman, **вы получите лучший 99-й процентиль**

Как поставить PgDoorman?

14 строчек — минимальный пример
для запуска, несложная конфигурация

OZON tech

Спасибо за внимание!
Вопросы?



Дмитрий Васильев

Эксперт по разработке информационных систем,
группа PostgreSQL DBA, Ozon

dmitrivasilyev@ozon.ru