



Миграция с СУБД Oracle/MS SQL Server на PostgreSQL

Игорь Мельников
i.melnikov@postgrespro.ru

A vertical decorative bar with a blue-to-pink gradient.

Инструменты для миграции

A vertical decorative bar with a blue-to-pink gradient.

Поддержка миграции в СУБД Postgres Pro

A vertical decorative bar with a blue-to-pink gradient.

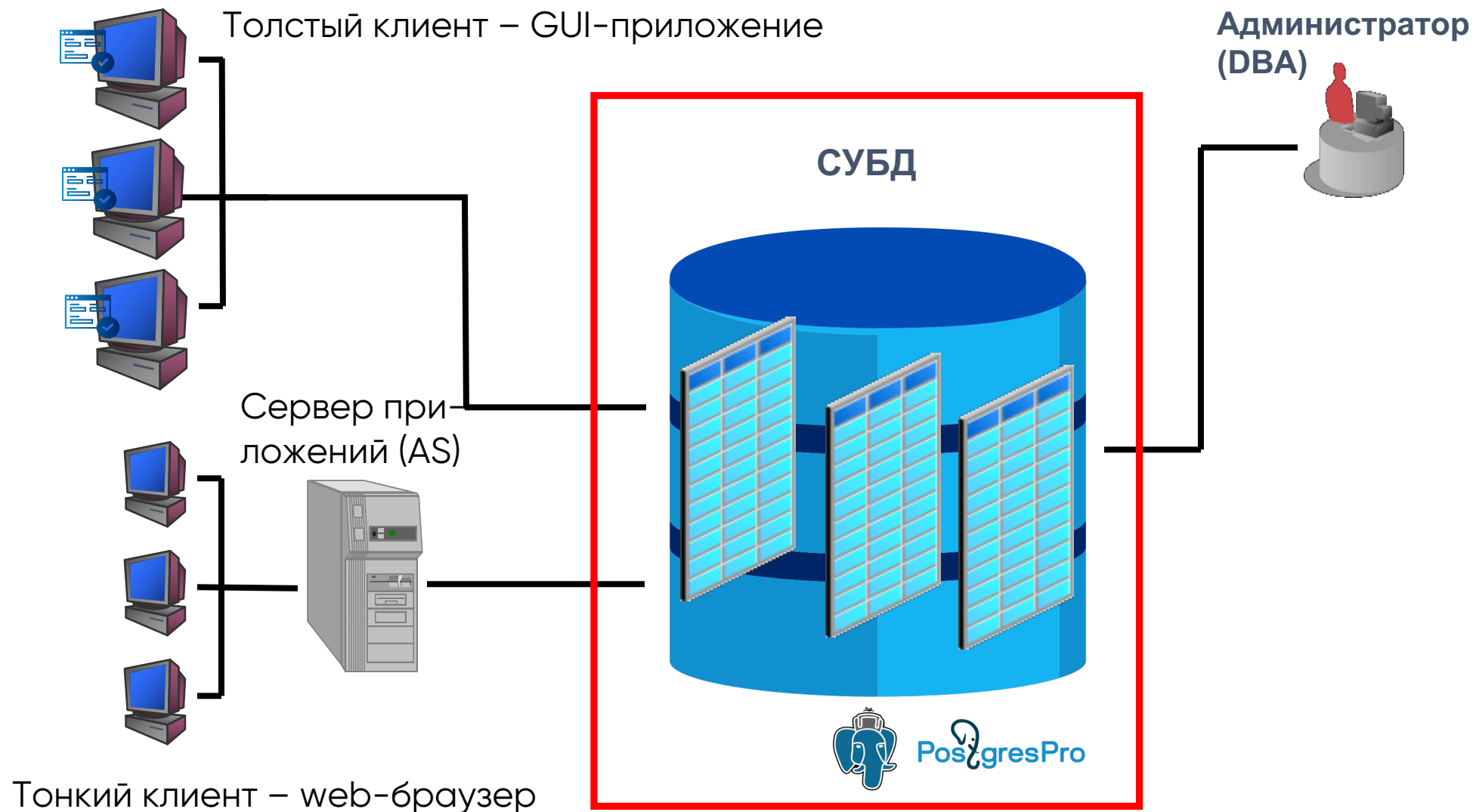
Примеры проектов по миграции на СУБД Postgres Pro

A vertical decorative bar with a blue-to-pink gradient.

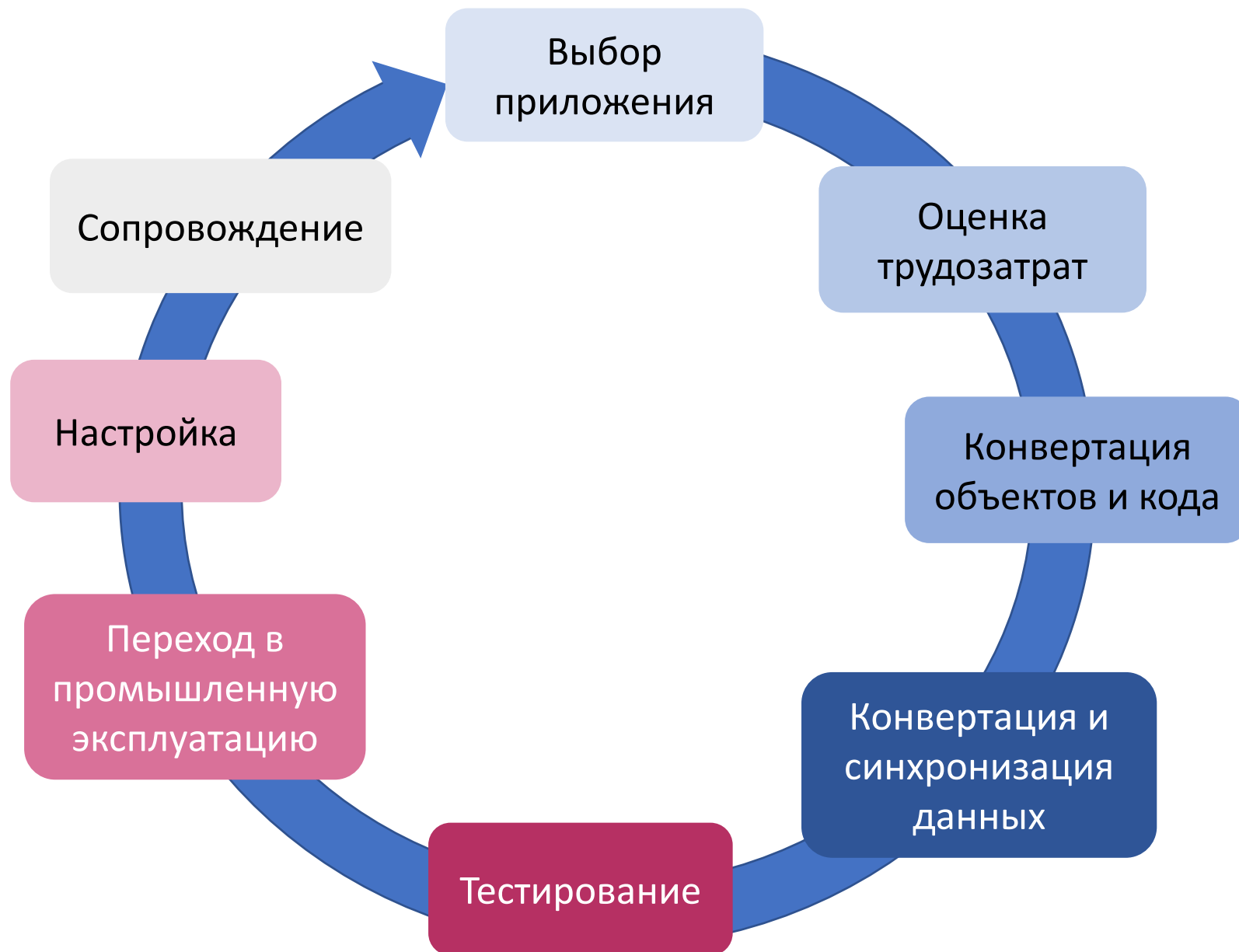
Дополнительная информация

Инструменты для миграции на PostgreSQL

Миграция компонентов приложения



Этапы проекта по миграции – жизненный цикл



- Выбор приложения
- Оценка трудозатрат
- Конвертация объектов и кода
- Миграция данных
- Тестирование
- Переход в production
- Оптимизация и тюнинг
- Сопровождение

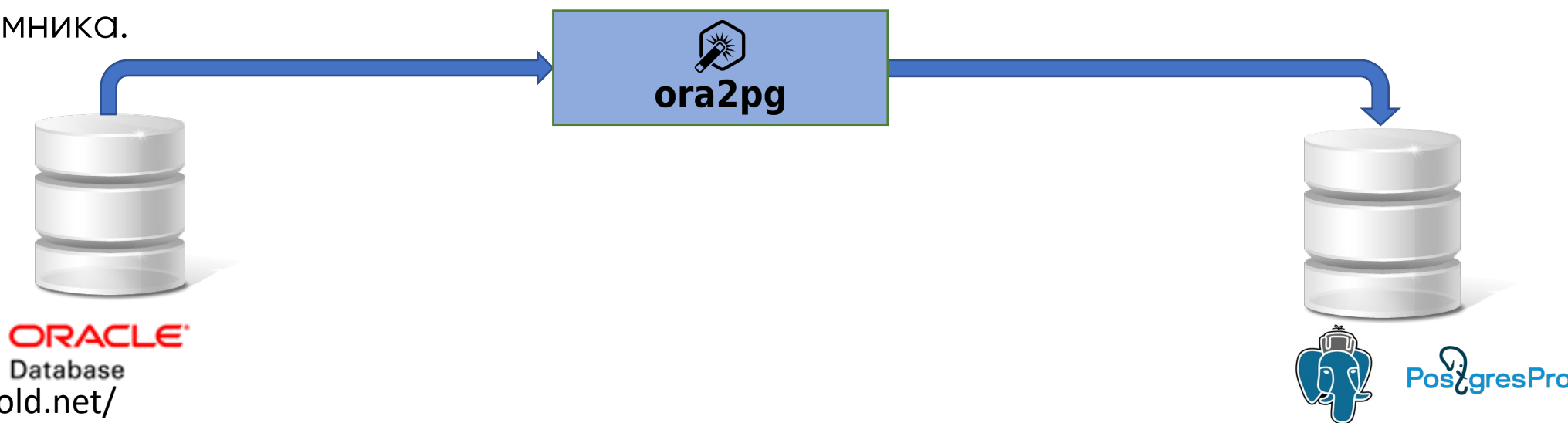
Инструменты миграции на PosgreSQL

- На рынке присутствует большое количество продуктов, как бесплатных (open source) так и платных:
 - ORA2Pg, slqserver2pg, Ispirer, Pentaho kettle, Diasoft, Конвертум и т.д.
 - Расширение ORAFCE – библиотека, эмулирующих поведение системных функций Oracle в PostgreSQL
- Миграция структуры и данных – автоматическая
- Миграция кода – полуавтоматическая
- У Ora2PG – оценка трудоемкости и продолжительности миграции



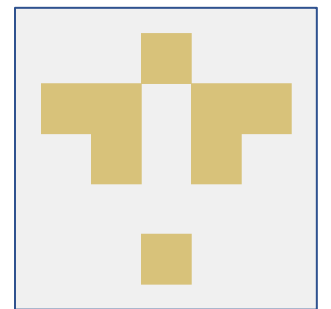
Утилита ORA2PG – миграция с Oracle на PostgreSQL

- Анализирует исходный код PL/SQL в Oracle и формирует DDL-скрипты в синтаксисе PostgreSQL :
 - Транслятор кода PL/SQL Oracle в синтаксис Postgres PL/pgSQL
 - Возможность настройки списка объектов для переноса
 - Возможность исключения из переноса нескольких схем
 - Возможность настройки параметров через конфигурационный файл
 - Возможность настройки сопоставления типов данных СУБД-источника и СУБД-приемника.



ORAFCE – Oracle's compatibility functions and packages

- Orafce – это популярное open-source расширение, которое позволяет применять с PostgreSQL дополнительные функции и операторы как в Oracle.
- Решает проблему использования встроенных функций Oracle, для которых нет аналога в PostgreSQL.
 - Таблица DUAL
 - Concat, nvl, nvl2, decode
 - Dbms_output
 - Utl_file, dbms_pipe, dbms_alert, dbms_sql
 - Работа с датой – add_month, last_date, next_day
- Автоматически включен в дистрибутив Postgres Pro
- Чтобы узнать больше о функциях Orafce и о возможностях расширения:
<https://github.com/orafce/oracle>



Миграция данных в PostgreSQL

ORA2PG:

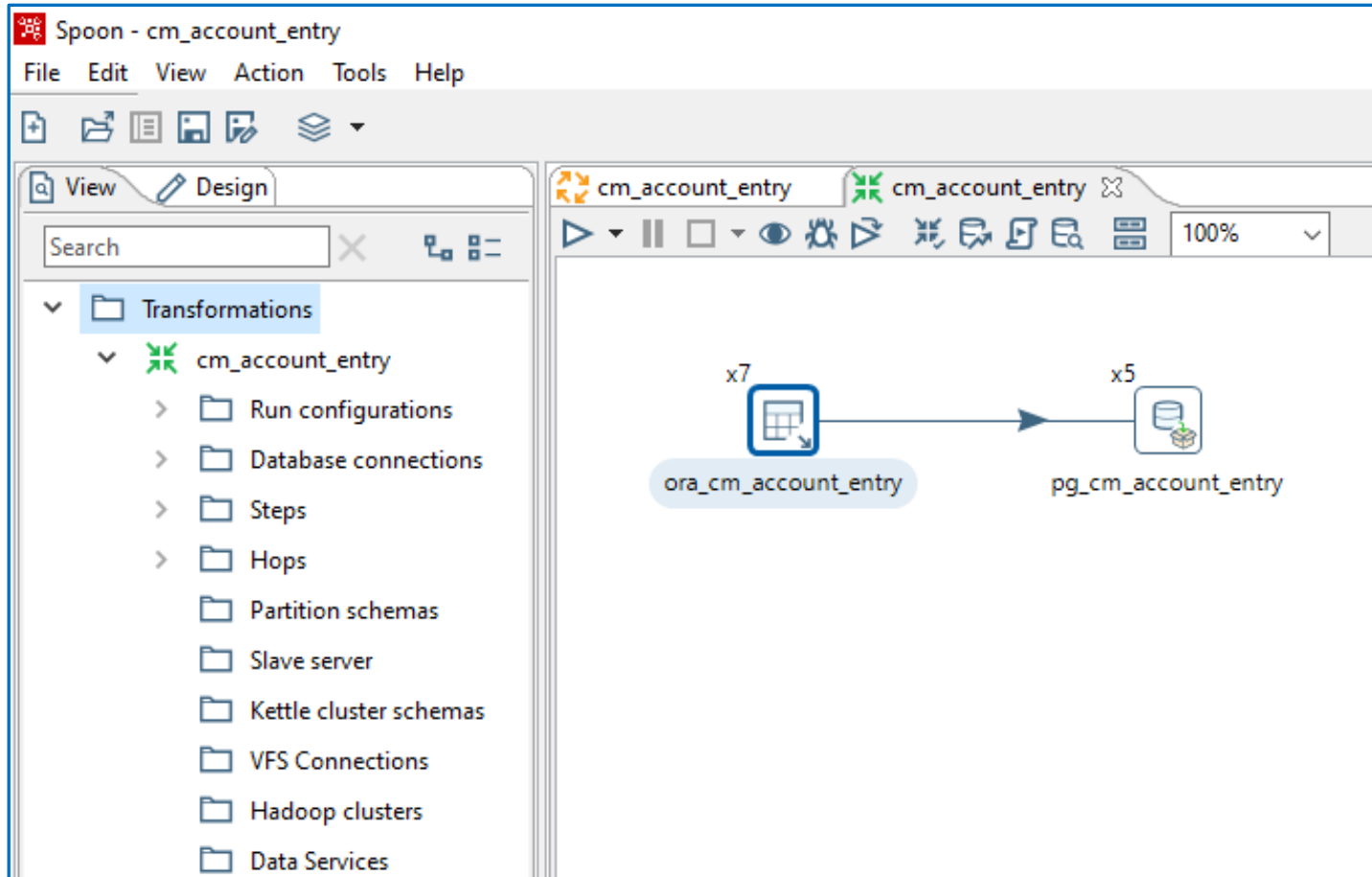
- Перенос данных из Oracle в PostgreSQL
- Возможен с помощью выгрузки в текстовые файлы или напрямую
- Настройка параметров конвертации данных

pentaho kettle:

ETL-инструмент, позволяющий

1. Разрабатывать различные сценарии извлечения, обработки и сохранения данных
2. Внедрять логику, реализованную на языке Java, в ETL-процесс
3. Запускать сценарии по расписанию
4. Взаимодействовать с различными СУБД
5. Взаимодействовать с Apache Kafka

Пример преобразования Pentaho Kettle



- ora_cm_account_entry читает данные из СУБД Oracle указанным запросом и количеством потоков для одновременного чтения.
- pg_cm_account_entry записывает данные в СУБД PostgreSQL командой COPY, что значительно уменьшает время переноса.

Время простоя при миграции продуктива

Допустимое время простоя составляет десятки часов:

- Допустим простой на выходные и праздники
- Объем БД не превышает 1 ТБ
- Рекомендуется для миграции кода и данных использовать ORA2PG



Допустимое время простоя составляет несколько часов или размер БД превышает 1ТБ:

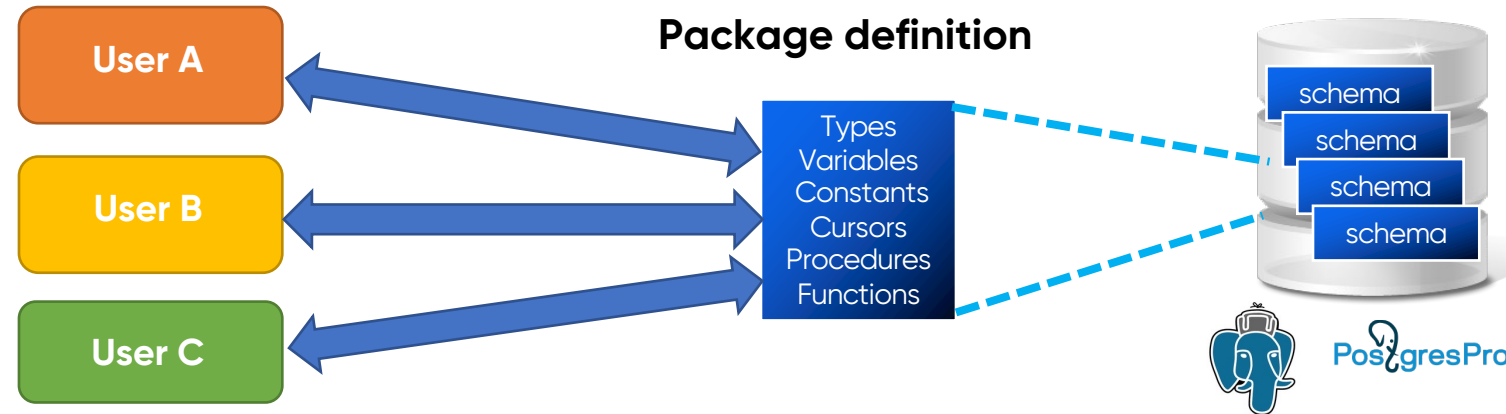
- Время простоя жестко регламентировано и ограничено бизнесом
- Рекомендуется для миграции кода использовать ORA2PG
- Для миграции данных использовать средства репликации, например: Pentaho Kettle



Поддержка миграции в СУБД Postgres Pro

Поддержка пакетов PL/pgSQL в Postgres Pro 15 Enterprise:

- Пакет является отдельной схемой
- Поддерживается функции инициализации – функция с фиксированным именем “***__init__***” – вызывается в момент первого обращения к пакету
- Поддерживаются глобальные переменные – все переменные объявленные в процедуре инициализации “***__init__***”
- Поддерживаются обращение к элементам других пакетов (схем), в том числе к глобальным переменным!



Пример пакета в СУБД Postgres Pro Enterprise 15

```
CREATE OR REPLACE PACKAGE sample_pkg
```

```
CREATE TYPE THostInfo AS  
(  
    Name    VARCHAR(32),  
    Description VARCHAR(256)  
)
```

```
CREATE FUNCTION __init__() RETURNS VOID AS $$  
DECLARE  
    v_g_host_name VARCHAR(50);  
BEGIN  
    v_g_host_name := get_default_host_name();  
END;  
$$
```

- Функция инициализации пакета, все объявленные в ней переменные являются глобальными переменными пакета

Использование пакета в Postgres Pro Ent 15

```
DO $$  
#import sample_pkg  
BEGIN  
  RAISE NOTICE 'Hostname => %', sample_pkg.v_g_host_name;  
END;  
$$;
```

• Прагма для импорта пакета

```
NOTICE: Hostname => localhost  
DO
```

```
SELECT sample_pkg.get_default_host_name();  
get_default_host_name
```

• Доступ к глобальной переменной пакета

```
-----  
localhost  
(1 строка)
```

• Вызов функции пакета в запросе

Пакеты 2.0 – дальнейшее развитие

- Несколько пакетов в одной схеме
- Поддержка приватности элементов пакета
-

Утилита ORA2PGPRO:

- Форк от ORA2PG, с дополнительными новыми возможностями
- Транслятор кода пакетов Oracle в синтаксис Postgres Pro 15 Ent
- Генерация скриптов выдачи привилегий на пакет (grant)
- Конвертация синтаксиса автономных транзакций Oracle PL/SQL в синтаксис Postgres Pro
- Исправление ошибок ORA2PG
- И т.д.



- Новое расширение PGPRO_UTL для Postgres Pro Enterprise
- Реализация системных пакетов из Oracle, которые отсутствуют в ORAFCE:
 - UTL_MAIL
 - UTL_SMTP
 - UTL_HTTP
 - DBMS_LOB
 - DBMS_APPLICATION_INFO
- В дальнейшем – реализация обработки очередей в БД (аналог Oracle Advanced Queuing)

Примеры проектов по миграции

Пример проекта по миграции

Распределенное технологическое приложение; работает на всех железных дорогах России:

- Порядка 250 таблиц, 50 Гб данных
- Приложение выполняет крупные (сотни Мб данных) и редкие транзакции (период в десятки секунд и минуты) плюс постоянно выполняются короткие мелкие транзакции (десятки раз в секунду)
- Более 200 хранимых процедур на PL/SQL (объем 60000 строк) в нескольких пакетах
- Более 40 серверов БД
- Эксплуатация 24/7; регулярные обновления серверных приложений 3-6 раз в год, включая БД

Windows	Толстые клиенты	C++, MQ
Linux	Web-клиенты	JavaScript, GWT, Angular
Linux	Сервера приложений	Java, Tomcat, ActiveMQ
Linux	Сервера баз данных	Oracle 11g SE, хранимые процедуры PL/SQL

<https://pgconf.ru/2020/274661>

Пример проекта по миграции – РЖД

Используемые инструменты:

- Утилита ORA2PG для миграции метаданных и кода
- Потребовалась ручная корректировка кода после транслятора ORA2PG

Трудозатраты:

- Состав команды проекта – двое на PostgreSQL и один – на адаптацию сервера приложений
- Продолжительность проекта – 1 год
- Проект включал в себя тестирование производительности, оптимизацию и развертывание в production – переключение на PostgreSQL существующих баз данных Oracle
- Половину объема работ заняла модификация кода приложения – полгода

Пример проекта по миграции – сбытовая компания

Исходная БД:

- Платформа – Redhat Linux 7 x86-64
- Oracle Database 11.2
- Объем БД – 600ГБ
- Приложение – система управления документами Documentum
- Бизнес-логика реализована на стороне сервера приложений (нет хранимого PL/SQL-кода внутри БД)
- Приложение работает в режиме 8x5 – максимально допустимое время простоя составляет 2 суток (выходные)



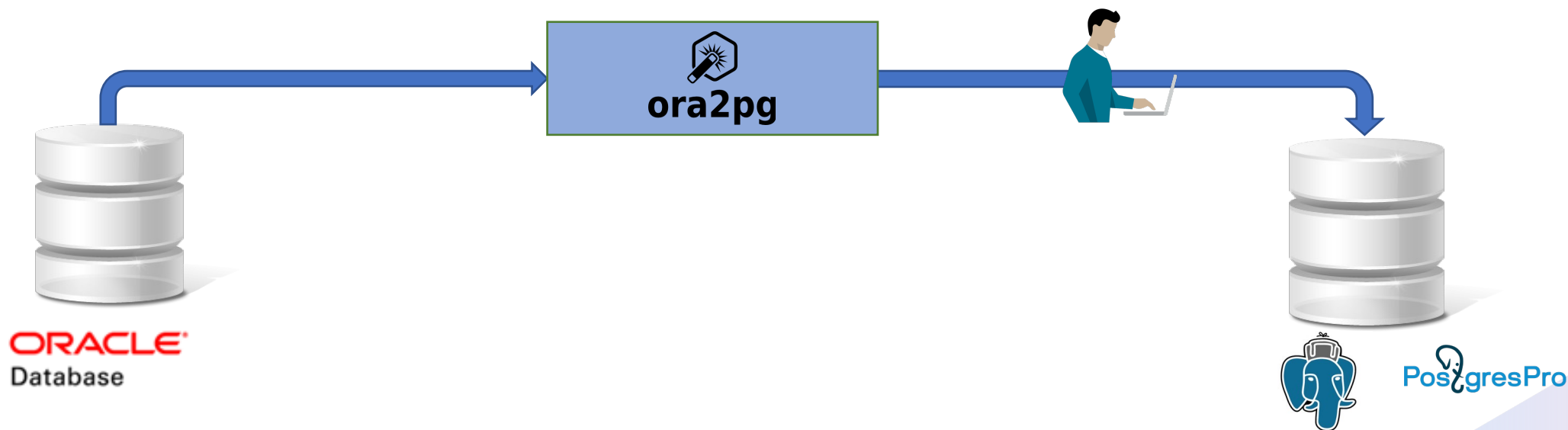
Целевая БД:

- Платформа – Redhat Linux 7 x86-64
- Postgres Pro 10 Standard



Миграция метаданных и данных в СУБД Oracle

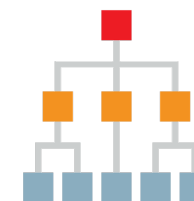
- Перенос метаданных (определений таблиц, индексов и т.д.) переносилось с помощью open-source утилиты ORA2PG:
 - Выгружались DDL-скрипты создания объектов в синтаксисе Postgres Pro
 - Затем эти скрипты дорабатывались вручную
 - Скрипты выполнялись в СУБД Postgres Pro



- Для миграции данных применялся Pentaho Kettle

Этапы проекта

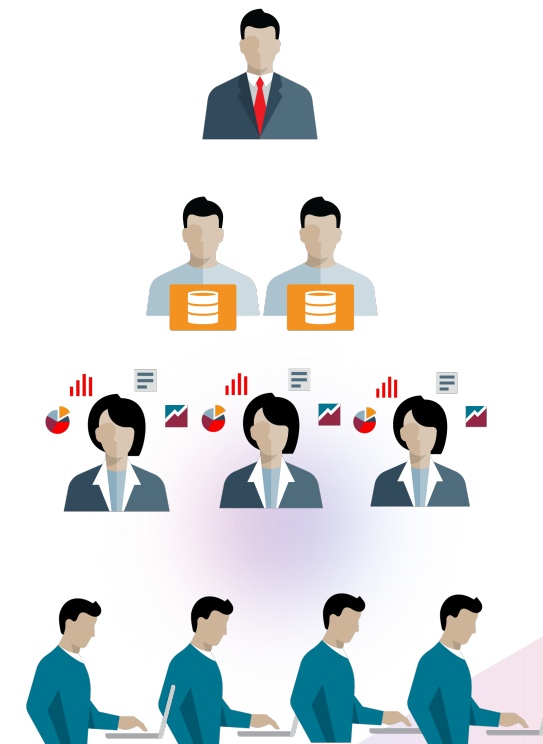
- Перенос данных и метаданных
- Тестирование и оптимизация SQL-запросов (продолжительность 3 месяца)
 - Замена синтаксиса внешних соединений Oracle (оператор "+") в синтаксис ANSI JOIN (оператор "OUTER JOIN")
 - Выключение параллелизма запросов (parallel query) в PG – вызывал повышенную загрузку ЦПУ
 - Создание индексов для устранения полного сканирования таблиц (full scan)
- Ввод в промышленную эксплуатацию
 - Занял 2 дня (выходные дни), включая тестирование перед переключением



Продолжительность и состав команды проекта

Проект занял 6 месяцев, состав команды проекта

- Руководитель проекта – 1 человек
- Администраторы БД – 2 человека
- Тестировщики – 3 человека
- Программисты – 4 человека



Пример проекта по миграции – государственное учреждение

Исходная БД:

- Платформа – Centos 7 x86-64
- Oracle Database 12c
- Объем БД – 4 ТБ
- Приложение для управления документооборотом
- Основной объем БД – документы (двоичные объекты – BLOB)
- Особенность: большой сервер (8 CPU и 1 ТБ RAM)



Целевая БД:

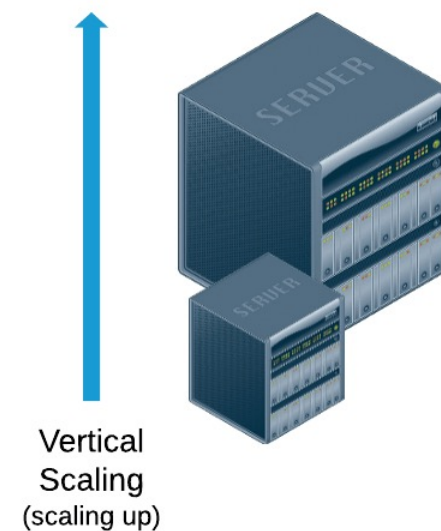
- Платформа – Centos 7 x86-64
- Postgres Pro 10 Enterprise



Масштабирование Postgres Pro на большом сервере

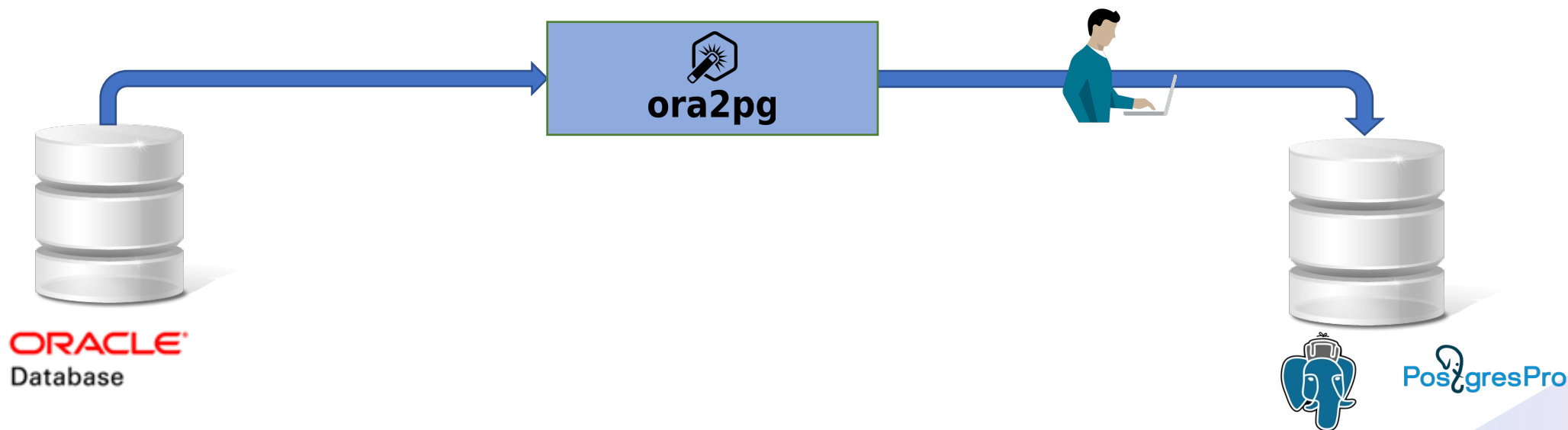
Была решена проблема низкой производительности СУБД при работе на большом числе ядер

- Были произведены доработки в Postgres Pro – проведена оптимизация буфер-менеджера и блокировок (LWLock)
- Внесены доработки в карту видимости и сделаны оптимизации
- Патч был предложен в “ванильный” Postgres



Перенос кода и данных из Oracle в Postgres Pro

- Перенос метаданных (определений таблиц, индексов и т.д.) переносилось с помощью open-source утилиты ORA2PG:
 - Выгружались DDL-скрипты создания объектов в синтаксисе Postgres Pro
 - Затем эти скрипты дорабатывались вручную
 - Скрипты выполнялись в СУБД Postgres Pro



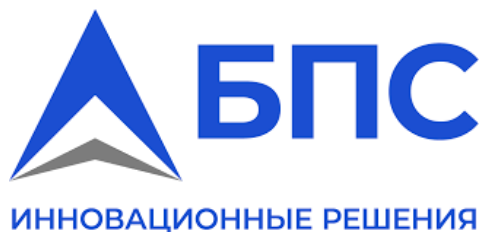
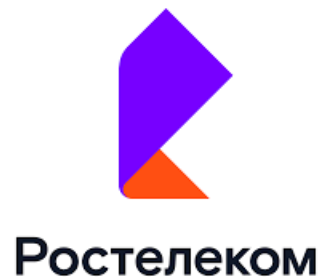
- Для синхронизации данных применялся Pentaho Kettle

Продолжительность и состав команды проекта

Проект занял 12 месяцев, состав команды проекта включал 6 человек

Переход был осуществлен в праздничные дни (4 дня)

Примеры мигрировавших заказчиков



Правительство Московской области

Межведомственная система электронного документооборота переведена на СУБД PostgreSQL за 1,5 месяца.

[подробнее ►](#)



Банк «Открытие»

Объем данных OTRS занимал 800 Гб и размещался на сервере с 24-ядерным процессором. Хранимые процедуры содержали около 5 000 строк кода, а отчеты – 10 000 строк.

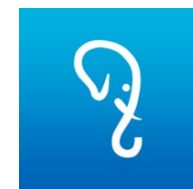
В результате перевода системы на PostgreSQL производительность возросла при сохранении объема оборудования.

[подробнее ►](#)

Заключение

Выводы:

- Много инструментов для автоматизации миграции Ora2Pg, sqlserver2pgsql, Ispirer, Pentaho kettle, Diasoft
- Расширение OraFCE
- Множество проектов сделали партнеры
- Трудоемкость миграции зависит от сложности и объемности приложения
- Это не "ROCKET SCIENCE"
- Консультации, тренинг, HandBook, White Paper
- При миграции обязательны тестирование (функциональное и нагрузочное), построение архитектуры надежности, безопасности и т.д.



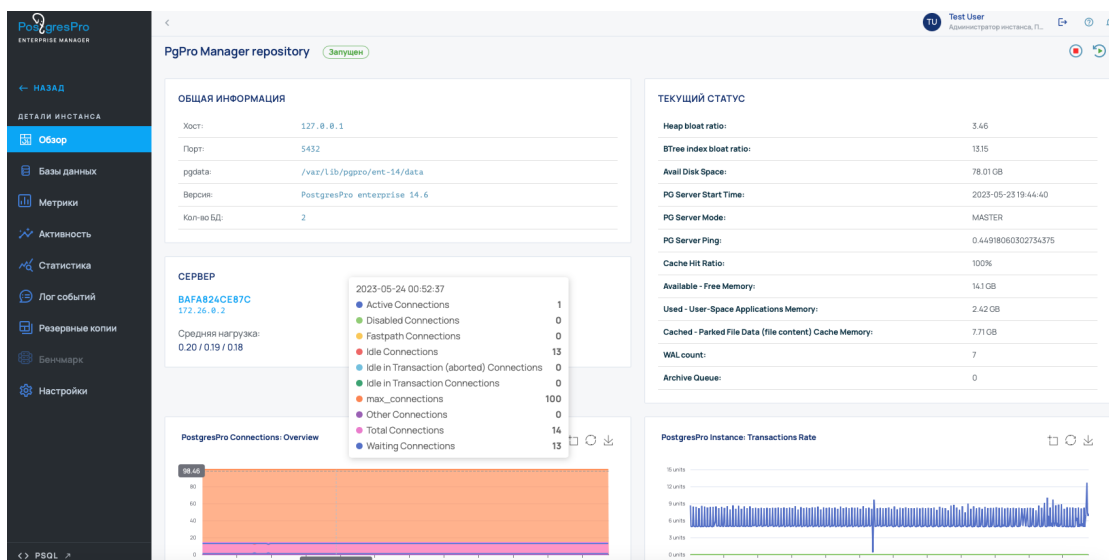
Полезные ссылки

Инструменты для миграции:

- ora2pg. <https://github.com/darold/ora2pg>
- sqlserver2pgsql. <https://github.com/dalibo/sqlserver2pgsql>
- pentaho kettle. <https://help.pentaho.com/Documentation/9.1>
- debezium connector for Oracle.
<https://debezium.io/documentation/reference/stable/connectors/oracle.html>
- SymmetricDS. <https://www.symmetricds.org/>
- oracle_fdw – модуль работы с СУБД Oracle из СУБД PostgreSQL.
https://github.com/laurenz/oracle_fdw
- Модуль pg_variables. https://github.com/postgrespro/pg_variables
- Oracle GoldenGate. <https://www.oracle.com/integration/goldengate>

Postgres Pro Enterprise Manager – PPEM

- Графическая система мониторинга и управления БД
Postgres – все БД в одной консоли
- Вывод сводной информации на дашборде
- Визуализация основных метрик производительности
– графики с выбором интервала наблюдения
- Статистика по SQL-запросам
- Можно посмотреть активность и ожидания по сессиям, лог событий
- Запуск/останов/перезапуск инстанса
- Выполнение backup/restore Базы Данных
- Изменение параметров настройки



Протестировать



Протестировать
СУБД Postgres Pro:



117036, Москва, ул. Дмитрия Ульянова, 7А



8 (495) 150-06-91

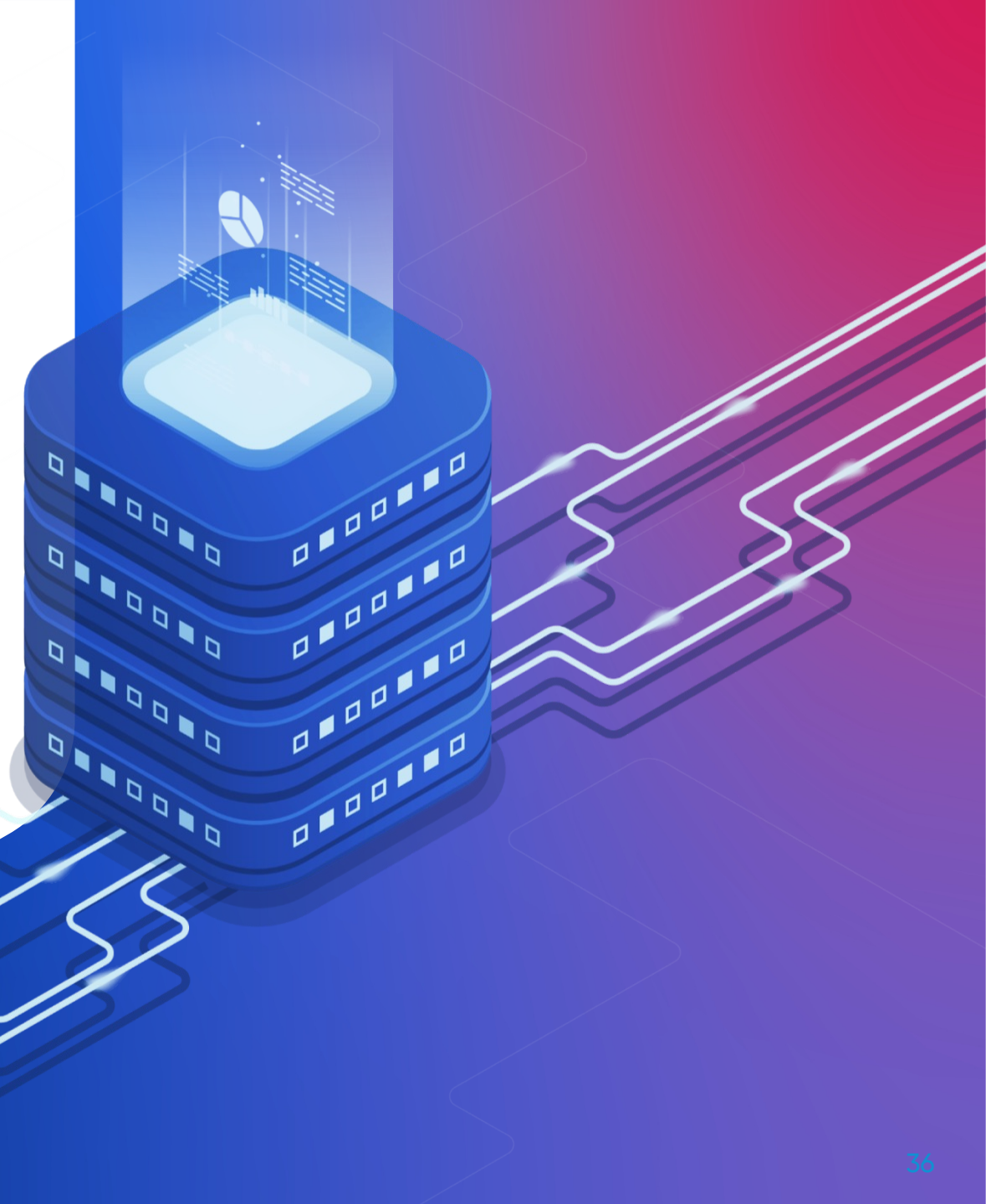


sales@postgrespro.ru

postgrespro.ru

PosgresPro

Спасибо
за внимание



PostgresPro

Q&A

