Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Уйти с Oracle и MS SQL на PostgreSQL: что это значит для бизнеса

Наша методика: оценка ora2pg, узел Postgres Pro для 1С, параллельный прогон и план отката

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

О ЧЁМ ЭТОТ РАЗБОР

Суть проблемы

Заказчик остаётся с СУБД, которую нельзя легально продлить и некому патчить: новые ядра MS SQL не купить, обновления Oracle недоступны, а прикладная система растёт. Мы решаем задачу управляемого перевода на PostgreSQL: считаем объём переделки по объектам схемы, переносим данные и код, держим два контура параллельно и переключаем по регламенту с проверенным откатом.

Почему это важно бизнесу

- Без патчей СУБД растёт технический долг: через 2-3 года прикладное ПО перестанет ставиться на старую платформу
- Оценка миграции по объектам схемы даёт бюджет заранее, а не «плюс два миллиона» на середине проекта
- Параллельный прогон двух контуров убирает риск многодневного простоя учёта и склада при переключении
- Регламент бэкапа переписывается целиком: .bak и задания SQL Agent базу PostgreSQL не восстановят
- Лицензии прикладных систем: часть вендоров не поддерживает PostgreSQL официально — это выясняем до старта

ЧТО МЫ ЗАКЛАДЫВАЕМ

Ключевые параметры реализации

18.4

Текущий минорный релиз PostgreSQL, на него ставим новые узлы; ветка 14 EOL 12.11.2026

PostgreSQL, versioning policy

RAM/4

shared_buffers на узле 1С; effective_cache_size = RAM минус shared_buffers

Postgres Pro 18, прил. K

256

max_locks_per_transaction для 1С: на штатных 64 срывается удаление временных таблиц

Postgres Pro 18, прил. K

20

from_collapse_limit и join_collapse_limit вместо штатных 8: запросы 1С дают десятки соединений

Postgres Pro 18, прил. K

5 мин

Вес единицы миграции в ora2pg (COST_UNIT_VALUE): из них складывается смета в чел.-днях

Ora2Pg 25, конфигурация

3

io_workers по умолчанию при io_method=worker в PostgreSQL 18: поднимаем под NVMe

PostgreSQL 18, resource config

РЕАЛИЗАЦИЯ 1 ИЗ 3

Этап 1. Инвентаризация схемы и смета переделки

Что настраиваем

Исходный узел Oracle или MS SQL, только чтение; отчёт собираем на отдельной машине аудита

Как мы это делаем

1. Ora2Pg: в ora2pg.conf задаём TYPE SHOW_REPORT и ESTIMATE_COST 1, запускаем ora2pg -t SHOW_REPORT --estimate_cost --dump_as_html > report.html
2. Разбираем отчёт по классам объектов: таблицы, представления, триггеры, пакеты PL/SQL, автономные транзакции, вызовы DBMS_* — они и дают основной вес сметы
3. MS SQL Ora2Pg читает с версии 24 (через Microsoft ODBC Driver), но T-SQL идёт частично: курсоры, CLR, linked servers и задания SQL Agent считаем ручной работой
4. Фиксируем список прикладного ПО и версий драйверов, проверяем заявленную вендором поддержку PostgreSQL
5. На выходе — смета в человеко-днях и перечень объектов, которые дешевле переписать, чем конвертировать

РЕЗУЛЬТАТ

Заказчик получает бюджет и срок до начала работ, а мы — список рисковых объектов. Отчёт ora2pg переснимаем перед стартом миграции: за время согласования схема успевает измениться, и повторный прогон ловит новые пакеты и триггеры.

КЛЮЧЕВОЙ НЮАНС

Единица миграции ora2pg — не факт, а калибруемая оценка: COST_UNIT_VALUE (по умолчанию 5 минут) подгоняем под свою команду по первым конвертированным пакетам.

РЕАЛИЗАЦИЯ 2 ИЗ 3

Этап 2. Узел СУБД под 1С: сборка, локаль, параметры

Что настраиваем

Новый сервер Postgres Pro для 1С рядом с действующим MS SQL, без остановки прод-контура

Как мы это делаем

1. Ставим сертифицированную сборку Postgres Pro (ветки 18.x/17.x/16.x), сверяя пару «версия платформы 8.3.x — версия СУБД» по системным требованиям 1С
2. Кластер инициализируем явно: initdb --locale-provider=icu --icu-locale=ru-RU; контрольные суммы страниц в PostgreSQL 18 включены по умолчанию
3. Базовый тюнинг: pgpro_tune -P 1c.tune -D $PGDATA, далее shared_buffers=RAM/4, effective_cache_size=RAM-shared_buffers, work_mem 32-256MB, maintenance_work_mem 1GB
4. Обязательны для 1С: standard_conforming_strings=off, escape_string_warning=off, row_security=off, max_locks_per_transaction=256, from/join_collapse_limit=20, jit=off
5. Включаем pg_stat_statements и auto_explain.log_min_duration=3000ms, чтобы после переключения видеть деградировавшие планы, а не слушать жалобы

РЕЗУЛЬТАТ

Узел готов к нагрузочной проверке до переключения. Тестовый контур держим не менее недели: прогоняем закрытие месяца, регламентные операции, обмены и печатные формы на реальной копии базы.

КЛЮЧЕВОЙ НЮАНС

Провайдер сортировки задаётся на initdb и меняется только пересозданием кластера: ошибка в локали всплывает на сортировке ФИО и на уникальных индексах.

РЕАЛИЗАЦИЯ 3 ИЗ 3

Этап 3. Переключение, бэкап и окно отката

Что настраиваем

Оба контура одновременно; новый регламент резервного копирования и точка возврата на MS SQL

Как мы это делаем

1. Регламент бэкапа пишем заново: pgBackRest (repo1-retention-full, archive-async=y при archive_command=pgbackrest archive-push) или pg_probackup FULL/PAGE/DELTA/PTRACK
2. Проверяем не бэкап, а восстановление: полный restore на отдельный стенд плюс PITR на произвольный момент, с замером времени до готовности базы
3. Переключение в нерабочее окно: старый экземпляр не удаляем, а останавливаем и оставляем как точку отката минимум на месяц
4. Первые 2-3 недели смотрим pg_stat_statements по total_exec_time: тяжёлые отчёты чиним индексами, статистикой и правкой запроса, а не откатом проекта
5. После массовой загрузки гоняем VACUUM ANALYZE по всей базе и ставим autovacuum_naptime=20s: на пустой статистике планировщик выбирает плохие планы

РЕЗУЛЬТАТ

Переключение перестаёт быть точкой невозврата: есть остановленный, но живой старый экземпляр и проверенное восстановление нового. Деградацию отчётов ловим по статистике запросов в первые дни, а не через месяц по жалобам пользователей.

КЛЮЧЕВОЙ НЮАНС

WAL-архив живёт по правилам retention бэкапов: при repo1-retention-full=2 держим запас места под retention+1 копий, иначе архив выберет весь раздел.

ЧТО МЫ ВЫУЧИЛИ НА ПРАКТИКЕ

Подводные камни

- ✕ **Локаль и сортировка заданы по умолчанию**
  В C-локали сортировка побайтовая: «Ёлкин» встаёт раньше «Абрамова». Локаль и ICU задаём на initdb — потом только пересоздание кластера.

- ✕ **Ожидание, что планы будут как в MS SQL**
  Планировщики разные: отчёт на 30 с превращается в минуты. Лечим индексами, ANALYZE и переписыванием запроса, а не увеличением памяти сервера.

- ✕ **Автоконвертер считается достаточным**
  ora2pg переводит синтаксис, но пакеты, автономные транзакции и вызовы DBMS_* доводим руками в PL/pgSQL. Эти объекты закладываем в смету отдельно.

- ✕ **Старые скрипты бэкапа оставлены как есть**
  Задания SQL Agent и файлы .bak в PostgreSQL не работают. Ставим pgBackRest или pg_probackup и проверяем именно восстановление, включая PITR.

- ✕ **Нет прогона закрытия месяца до переключения**
  Регламентные операции 1С грузят СУБД иначе, чем повседневная работа. Прогоняем закрытие и обмены на тестовом контуре минимум за неделю.

- ✕ **Игнор max_locks_per_transaction**
  На штатных 64 транзакция очистки временных таблиц срывается и оставляет мусор. Ставим 256 сразу: параметр применяется только с перезапуском сервера.

- ✕ **Старый сервер погашен сразу после переезда**
  Пропадает точка отката и источник для сверки цифр. Держим остановленный экземпляр и его бэкапы минимум месяц после переключения.

- ✕ **Поддержка вендора ПО не проверена**
  Часть прикладных систем технически работает на PostgreSQL, но официально не поддерживается. Запрашиваем позицию вендора письменно до старта.

УРОВНИ РЕАЛИЗАЦИИ

Как правильно

МИНИМУМ

- Отчёт ora2pg SHOW_REPORT с ESTIMATE_COST до начала любых работ
- Кластер инициализирован с явной локалью и провайдером сортировки
- Бэкап pgBackRest или pg_probackup с проверенным тестовым восстановлением
- Старый экземпляр СУБД остановлен, но сохранён как точка отката

НОРМАЛЬНО

- Параметры по прил. К: max_locks_per_transaction=256, collapse_limit=20, RAM/4
- Тестовый контур с прогоном закрытия месяца и обменов не менее недели
- pg_stat_statements и auto_explain включены до переключения, а не после
- PITR отработан на стенде с замером времени восстановления

ХОРОШО

- Сертифицированная сборка Postgres Pro под конкретную версию платформы 1С
- Параллельный прогон двух контуров с помодульным переключением и сверкой данных
- Алерты на age(datfrozenxid) и на idle in transaction в pg_stat_activity
- VACUUM ANALYZE и переиндексация в расписании, а не по факту жалоб

ПРОВЕРЬТЕ У СЕБЯ

Чек-лист самопроверки

- [ ] Есть отчёт ora2pg или инвентаризация T-SQL с оценкой объёма ручной переделки?
- [ ] Проверена пара «версия прикладной платформы — поддерживаемая версия PostgreSQL»?
- [ ] Кластер создан с нужной локалью и провайдером сортировки, сортировка ФИО проверена?
- [ ] Задан max_locks_per_transaction=256 с перезапуском и from/join_collapse_limit=20?
- [ ] Прогнано закрытие месяца и обмены на тестовой копии реальной базы?
- [ ] Новый регламент бэкапа развёрнут и восстановление проверено на отдельном стенде?
- [ ] Отработан PITR на произвольный момент времени с замером длительности?
- [ ] Включены pg_stat_statements и auto_explain для контроля планов после переезда?
- [ ] Старый экземпляр СУБД сохранён как точка отката минимум на месяц?
- [ ] Получена позиция вендоров прикладного ПО о поддержке PostgreSQL?

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.

ЧТО МЫ ДЕЛАЕМ

Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Аудит схемы и смета миграции: отчёт ora2pg, разбор T-SQL, список объектов на переписывание
- Разворачиваем узел PostgreSQL или Postgres Pro: локаль, параметры под 1С, расширения мониторинга
- Переносим базы, поднимаем тестовый контур и прогоняем закрытие месяца вместе с бухгалтерией
- Ставим бэкап pgBackRest или pg_probackup с проверкой восстановления и PITR
- Ведём первый месяц после переключения: разбор тяжёлых запросов, индексы, автовакуум

15+
лет в ИТ-поддержке

50
рабочих мест — наш профиль

МТС
дата-центр, Москва

КОНТАКТЫ

Обсудить вашу задачу

Сайт	**itfresh.ru**
Телефон	**+7 903 729-62-41**
Telegram	**@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.

itfresh.ru

ДОКУМЕНТАЦИЯ И НАШ СТЕК

Техническая база

01 PostgreSQL: Resource Consumption, io_method и io_workers (postgresql.org — 18.4)

02 PostgreSQL: Versioning Policy, сроки поддержки веток (postgresql.org — 2026)

03 PostgreSQL: initdb, --locale-provider и --icu-locale (postgresql.org — 18.4)

04 Postgres Pro: Приложение K. Настройка для решений 1С (postgrespro.ru — 18)

05 Postgres Pro: pgpro_tune и pg_probackup (FULL/PAGE/DELTA/PTRACK) (postgrespro.ru — 18)

06 Ora2Pg: Migration Cost Assessment, SHOW_REPORT, MSSQL (ora2pg.darold.net — 25)

07 pgBackRest: Configuration Reference, retention и archive-async (pgbackrest.org — 2026)

08 1С:Предприятие 8 — системные требования, СУБД (v8.1c.ru — 2026)

09 Наш чек-лист переключения СУБД и окна отката (itfresh.ru — 2026)

Основано на официальной документации продуктов и нашей практике внедрения.