

# Миграция высоконагруженных решений 1С на Linux/PostgreSQL

Без потерь  
в производительности

**Елена Скворцова,**  
руководитель направления  
технологической экспертизы  
ООО «ИТ-Экспертиза»



# Технология перехода

Долго. Дорого. Безопасно



# Этапы миграции, или о чём будем говорить



## Подготовка

- Аудит кода на предмет наличия платформеннозависимых механизмов
- Проектирование целевой технической архитектуры
- Нагрузочное тестирование и рефакторинг
- Настройка мониторинга технологического качества

## Переход

- Подготовка продуктивной среды
- Подготовка плана миграции и плана отката
- Перенос данных
- Переключение пользователей

## Поддержка и стабилизация

- Мониторинг работы системы
- Устранение проблем стабильности
- Устранение проблем производительности

# Подготовка к миграции

## Аудит кода



### 1. **Linux на сервере и на клиенте или только на сервере?**

- **Только сервер:** режим сложности средний
- **Клиент и сервер:** режим сложности беспощадный
- Если первом варианте есть место для маневра, то во втором его нет и возможно, что от некоторого ПО/Оборудования придется отказаться

### 3. **Провести автоматический анализ кода**

- Найти платформеннозависимый код: для этого мы доработали 1С:АПК (автоматизированная проверка конфигураций)
- Не обращать внимания на неиспользуемые механизмы (см. п. 2)
- Провести рефакторинг

## ЗАПОВЕДИ:

COM – зло

Внешние компоненты, созданные по технологии Native API, должны быть реализованы для разных ОС и архитектур

Сразу использовать открытый формат офисных документов

Проводить тестирование функционала конфигурации на разных ОС



# Подготовка к миграции

## Целевая техническая архитектура

Определяем нефункциональные требования в зависимости от уровня критичности бизнес-системы:



### Требования к **доступности**

Размер технологических окон, максимальное время простоя по причине сбоя и плановых работ



### Требования к **надежности**

RTO, RPO



### Требования к **безопасности**

Механизмы аутентификации, использование защищенных протоколов



### Требования к **производительности**

Для целей миграции – «не хуже, чем было»

# Подготовка к миграции

## Целевая техническая архитектура

Описываем технические решения

### Параметры оборудования серверного ландшафта:

- продуктивный
- предпродуктивный
- тестовый
- разработки

Отталкиваемся от параметров исходной продуктивной системы, но помним о будущем нагрузочном тестировании

### Используемые компоненты:




- Версия и конфигурация кластера сервера приложений 1С и веб-сервера
- Файловое хранилище
- Версия и конфигурация кластера PostgreSQL

Нужен автоматический failover? Выбираем и проверяем:  
Patroni + ETCD + HA-Proxy, Corosync/Pacemaker и наверняка что-то еще...

# Подготовка к миграции

## Целевая техническая архитектура

### Стратегия и средства резервного копирования и восстановления

-  `pg_dump + pg_restore`: логический бэкап, консистентный на момент начала
-  `pg_basebackup`: бэкап файловой системы всего кластера => 1 база в кластере для продуктива
-  `pg_probackup`: тоже бэкап файловой системы, но с инкрементами (и блэkdжеком)

### План обслуживания

1. `REINDEX INDEX (CONCURRENTLY)`
2. `vacuumdb --all --analyze --freeze`
3. `pg_repack: REINDEX + CLUSTER + VACUUM FULL` онлайн



# Подготовка к миграции

## Мониторинг технологического качества

**ДО**  
миграции

- Список ключевых операций для контроля производительности
- Встраивание замеров, расчет APDEX
- Сбор метрик загрузки оборудования
- Сбор технологического журнала

**ПОСЛЕ**  
миграции

- ✓ **Контроль APDEX:** не должно стать хуже (лучше не обязательно 😊 )
- ✓ **Контроль загрузки оборудования:** ключевые метрики не должны выходить за установленные пределы
- ✓ **Технологический журнал:** следим за проблемами стабильности и ошибками прикладного кода



# Подготовка к миграции

## Нагрузочное тестирование

### Задачи:

- Проверить корректность сайзинга
- Подобрать оптимальные настройки 1С и PostgreSQL для конкретной системы
- Выявить узкие места и сделать рефакторинг
- Научиться расследовать проблемы производительности и стабильности в новой среде

### РЕШЕНИЕ:

**нагрузочный тест** – имитация в целевой среде пользовательской нагрузки, максимально приближенной к продуктивной.



# Примеры из жизни

Почему важность  
нагрузочного  
тестирования трудно  
переоценить?



# Фатальное замедление времени расчета себестоимости в 1С:ERP

со счастливым концом

Время выполнения этапа «Распределение затрат и расчет себестоимости»

В продуктивной системе на **MS SQL Server**: **3,5 часа**

- ключевой запрос в MS SQL выполнялся **3 минуты**,  
в Postgres – **≈17 часов** до момента отмены

```
-> Index Scan using tmpind_3556 on tt786 t4 (cost=0.17..42.41 rows=1 width=156) (actual time=1181.692..1181.692 rows=0 loops=7840)
   Index Cond: (_q_001_f_003rref = t2._q_001_f_036rref)
   Filter: (('\\x08'::bytea = _q_001_f_007_type) AND ('\\x000000f8'::bytea = _q_001_f_007_rtref) AND ('\\x08'::bytea = _q_001_f_01
   Rows Removed by Filter: 2530447
```

В нагрузочном тесте на **PostgresPro Enterprise**: **> 18 часов**

- после оптимизации **одного** запроса: **6 часов**

# Деградация производительности PostgreSQL

## в интенсивном тесте 1С:Розница с монотонными операциями

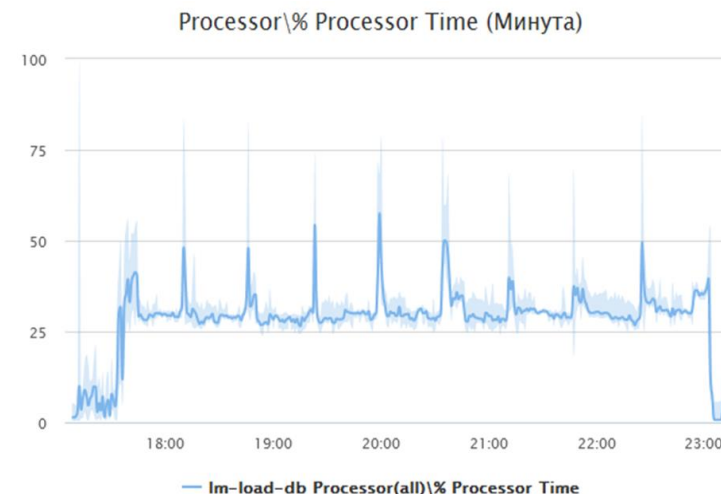
### Особенности:

- 2000+ пользователей
- высокая интенсивность
- одинаковые операции
- активное использование временных таблиц

Итерация	Приоритет	Целевое время	Количество замеров	APDEX
Ключевая операция				
Открытие кассовой смены	1	20,00	1 896	0,930
Закрытие кассовой смены	2	60,00	6	0,000
Документ перемещение проведение	3	5,00	86	0,442
Приемка товара	4	5,00	43	0,814
Отчет "Продажи" (формирование)	5	30,00	500	1,000
Документ "Чек" (Проведение/Запись)	6	10,00	3 602	0,707
Документ Чек ККМ /(Проведение)	7	10,00	1 107	0,407
Итого		0,10	34 433	0,243

### Последствия:

- 2000+ бэкендов
- 80+ дампов зависших рабочих процессов кластера серверов 1С. Анализ показал, что рабочие процессы зависали в ожидании ответа от PostgreSQL
- Неудовлетворительная производительность ключевых операций
- ...все это на фоне низкой утилизации ресурсов сервера СУБД



# Деградация производительности PostgreSQL



## в интенсивном тесте 1С:Розница с монотонными операциями

### Анализ:

- pg\_stat\_activity и pgpro/pg\_wait\_sampling
- perf record -F 99 -a -g --call-graph=dwarf sleep
- ...обращаемся за помощью в Postgres Pro

После оптимизации на новой сборке **APDEX 0.979!**

```
=> SELECT event_type, event, SUM (count) sum_cnt
FROM pg_wait_sampling_profile
GROUP BY event_type, event
ORDER BY sum cnt desc;
```

event_type	event	sum_cnt
LWLock	LockManager	36341376
Client	ClientRead	16135394
Activity	LogicalLauncherMain	47336

**NB**

Каждый серверный процесс отправляет в общую циклическую очередь сообщения об аннулировании кэша в результате изменения данных или их схемы. При высокой нагрузке, где большое количество серверных процессов интенсивно меняет данные, это может приводить к чрезмерно частому сбросу кэша и последующему поднятию в него данных.

<https://postgrespro.ru/docs/enterprise/14/release-proee-14-7-1>

Улучшение производительности, ранее сниженной из-за многочисленных сбросов кэша. Данное улучшение было достигнуто благодаря уменьшению количества отправляемых серверными процессами друг другу сообщений о событиях аннулирования, касавшихся временных таблиц.

# Миграция

## План миграции

1

Описывает процесс миграции по шагам в хронологической последовательности

Например:

1. Запрет подключения к системе на исходном сервере
2. Создание бэкапа средствами СУБД
3. Выгрузка базы в формат .dt

2

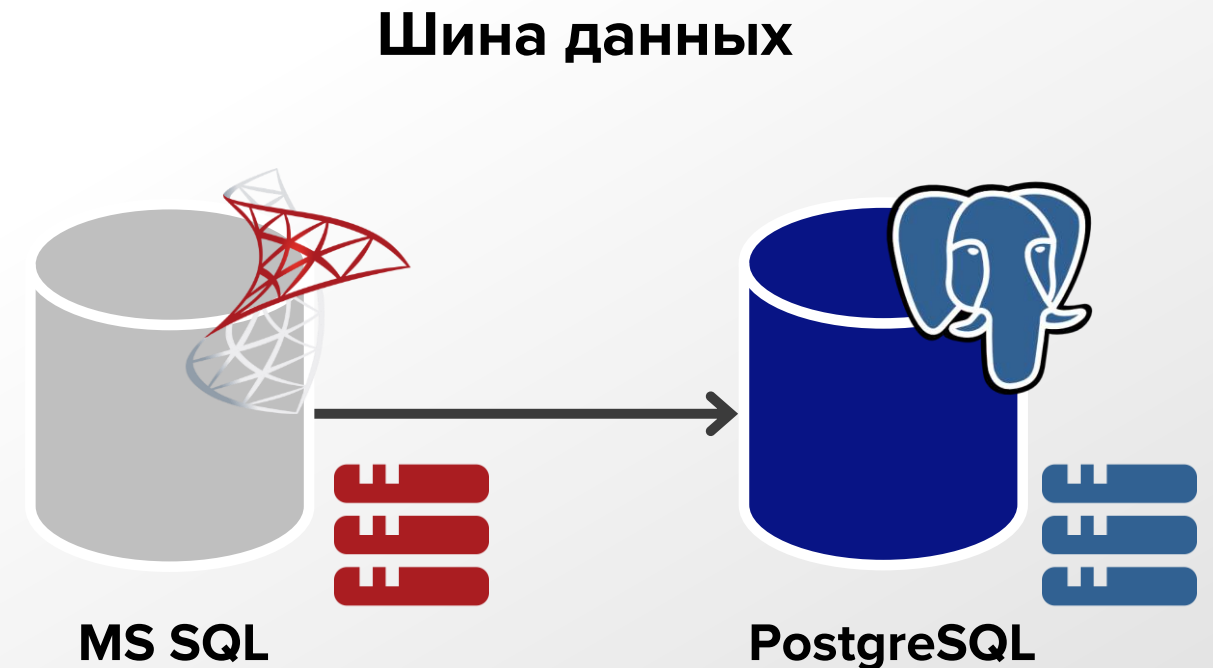
Каждый шаг плана миграции имеет ответственного, плановую продолжительность и критерии проверки успешности

3

Для каждого критерия проверки при отрицательном результате указан шаг плана отката или другие корректирующие действия

# Миграция Перенос данных

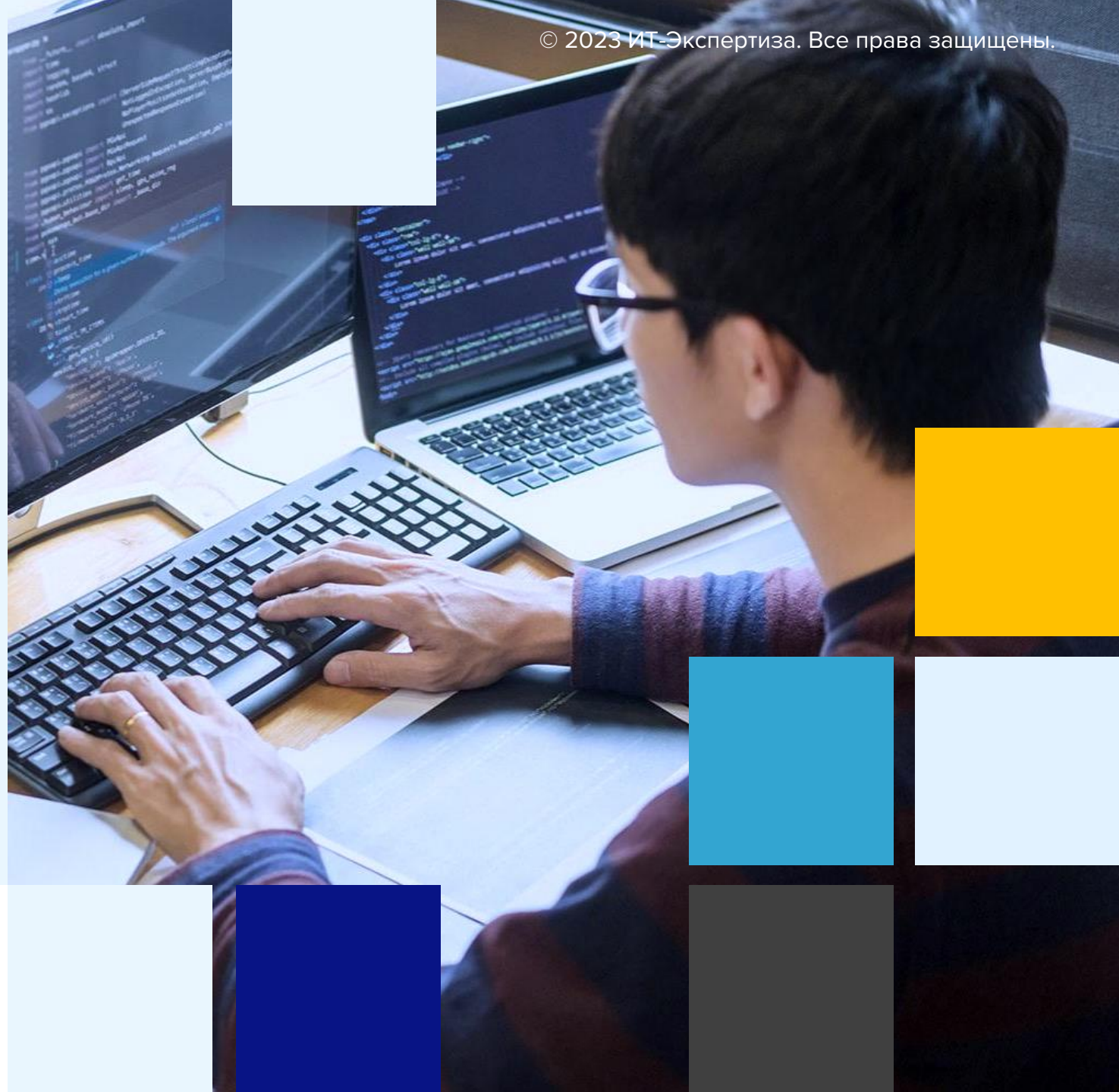
- Обычно данные переносятся через промежуточную выгрузку в формат 1С:Предприятие .dt
- Для базы в 10 Тб вам потребуется... пара недель?
- Многопоточная выгрузка позволит превратить пару недель в пару дней. Если закончится успешно 😊





# Подводя итоги

Ключи к успешной  
миграции



# Подготовка к миграции: квест длиною в ...?

- Нагрузочный тест и оптимизация действительно требуют времени
- Это время можно и нужно с пользой потратить на обучение ваших DBA и системных администраторов
- Часто достаточно исправить несколько запросов, и все «взлетит»



**KEEP  
CALM  
AND  
SWITCH TO  
PostgreSQL**

Ваши вопросы?



# Елена Скворцова

Руководитель направления  
технологической экспертизы  
ООО «ИТ-Экспертиза»

## САЙТ

<https://it-expertise.ru>

## ТЕЛЕФОН

+7 499 450 28 86

## EMAIL

[info@it-expertise.ru](mailto:info@it-expertise.ru)

## АДРЕС

119435 г. Москва,  
ул. Малая Пироговская, 16

