



Миграция с Oracle на PostgreSQL

Игнатов А.В.

www.postgrespro.ru

Цели миграции на PostgreSQL

Снижение стоимости решения:

- Снижение стоимости лицензий и обслуживания СУБД
- Гибкость при развертывании тестовых сред

Цели миграции на PostgreSQL (продолжение)

Использование СУБД с открытым исходным кодом:

- Системы с открытым исходным кодом безопаснее
- Независимость от производителя системы, возможность смены поставщика услуг, связанных с техподдержкой системы

Цели миграции на PostgreSQL (продолжение)

Преимущества использования:

- Расширяемость, комбинирование структурированных и слабоструктурированных данных, надежность

Миграция с Oracle на PostgreSQL

Что помогает при миграции с Oracle на PostgreSQL:

- Похожесть языка PL/SQL и pgplsql
- Расширяемость PostgreSQL
- Поддержка PostgreSQL многих языков программирования — pl/perl, pl/python, pl/tcl и других

Этапы проекта по миграции

- Аудит системы Заказчика
- Принятие решения о миграции по итогам аудита
- Развертывание стенда для миграции
- Конвертация данных
- Конвертация кода
- Функциональное тестирование
- Разработка средств и сценариев отката на исходную систему (опционально).

Этапы проекта по миграции (продолжение)

- Оптимизация структур хранения и кода
- Нагрузочное тестирование
- Оптимизация производительности
- Принятие решения о миграции
- Миграция
- Техническая поддержка системы после миграции

Аудит системы Заказчика

Факторы, влияющие на возможность миграции:

Фактор	На что влияет
Размер БД	Время простоя при миграции
Наличие в коде приложения Заказчика функционала, не имеющего аналога в PostgreSQL	Трудоемкость миграции, необходимость изменения логики работы приложения, стабильность системы после миграции
Объем и доступность кода приложения	Трудоемкость миграции по сравнению с разработкой нового приложения

Аудит системы Заказчика (продолжение)

Фактор	На что влияет
Поддержка со стороны разработчика приложений СУБД PostgreSQL	Трудоемкость миграции и тестирования
Наличие связанных систем, взаимодействующих с мигрируемой системой	Трудоемкость миграции и тестирования

Принятие решения о миграции по итогам аудита

- Postgres - не волшебник и он не есть стопроцентный аналог Oracle
- Не всегда получается перевести систему один в один просто используя ctrl+c, ctrl+v
- Трудозатраты на миграцию могут превысить трудозатраты на создание системы с нуля

Типичные проблемы при конвертации данных и кода

- Медленная работа с LOBs при использовании oci драйвера
- Отсутствие необходимого функционала в Postgres (некоторые модули вида `dbms_*`, `utl_*`)
- Отсутствие некоторых типов данных (глобальные пакетные переменные, `VARRAY` or `Nested Table`)

Типичные проблемы при конвертации данных и кода (продолжение)

- Отсутствие хинтов для запросов
- Немного странная реализация партиционирования в Postgres(PostgresPro работает над данной проблемой)
- Отсутствие компрессии данных (у PostgresPro есть решение, компрессия ~ 3-4 раза на реальных данных 1С)

Конвертация данных и кода (продолжение)

Инструмент	Достоинства	Недостатки
Сервис PostgresPro	Результат под ключ, необходимость внесения изменения в приложение	
Конвертор Диасофт	Короткий срок, высокая скорость выгрузки данных из Oracle	Пока не пробовали окончательной версии продукта
Самостоятельно, с использованием ora2pg	Контроль процесса	Необходим тщательный контроль конечного результата. Медленная работа с LOBs

Конвертация данных и кода (продолжение)

- Стопроцентной автоматике перевода пока не придумали
- Ручной труд необходим
- Проверка автоматически сконвертированного кода всё равно необходима

Функциональное тестирование

- Результат работы функции $a(x)$ на Oracle и Postgres должен совпасть при одинаковых аргументах
- Если нет совпадения результатов, то зачем всё это затевалось =)

Нагрузочное тестирование

- Необходимо понять работает ли новая система под нагрузкой
- Postgres работает и ведет себя не так как Oracle и это надо учитывать при проведении тестирования
- Выявление узких мест в производительности системы

- PostgreSQL устроен не так как Oracle
- Vacuum и мультиверсионность
- Работа планировщика запросов отличается от его работы в Oracle

Оптимизация производительности (продолжение)

- Запросы, перенесенные один в один могут работать на Postgres медленнее чем они работают на Oracle
- Иногда необходимо подправить SQL запросы для достижения большей их производительности на Postgres

- Необходимо технологическое окно для окончательного переноса данных
- Сбор статистики для данных после их переноса
- Перенастройка внешних подсистем
- Запуск заранее подготовленных функциональных тестов
- Запуск системы тестирования приложения если таковая система имеется

Направления работ, сопутствующих миграции

- Работа с технической поддержкой
- Устранение проблем с «забытым», редко используемым функционалом
- Тюнинг системы под реальной нагрузкой
- Обеспечение отказоустойчивости

Направления работ, сопутствующих миграции (продолжение)

- Организация резервного копирования
- Обеспечению безопасности
- Обеспечение мониторинга системы
- Разработка необходимых расширений к СУБД
- Аудит состояния системы через полгода/год

Примеры миграций на PostgreSQL



Выводы

- Непреодолимых препятствий в миграции нет, необходимо вовремя и корректно оценить трудозатраты на неё
- Поначалу может показаться что миграция невозможна. Это не так. Всегда можно полностью переписать код. Шутка ;)

Выводы (продолжение)

- Процесс миграции нужно рассматривать в комплексе с обеспечением мер по отказоустойчивости, резервированию и безопасности новой системы
- Попутно с миграцией, необходимо провести обучение СУБД PostgreSQL для обслуживающего персонала и приобрести техническую поддержку

Вопросы?



Спасибо за внимание!

Контакты:

a.ignatov@postgrespro.ru

www.postgrespro.ru