



DISTILLERY

Business clarity. Software craft.

PGDAY'
RUSSIA 17

КОНФЕРЕНЦИЯ
ПО БАЗАМ ДАННЫХ

Миграция OLTP с MSSQL на PostgreSQL

Кристина Кучерова

DB engineer, Distillery

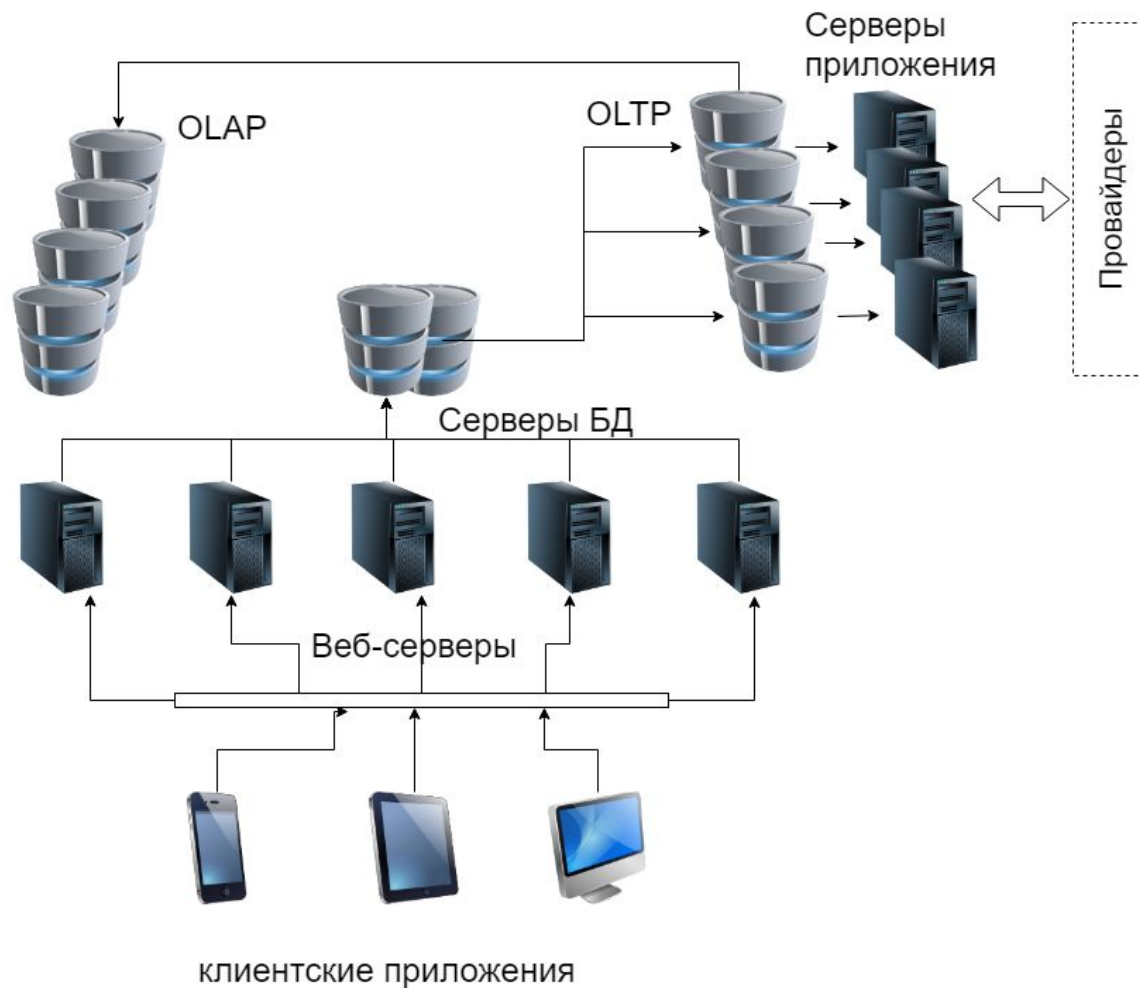
k.kucherova@distillery.com



Разработка стартап проектов,
участие в co-source проектах

В 2014 году начали реализацию
проекта по переносу базы данных
с MS SQL на PostgreSQL

Архитектура системы до миграции



Описание системы до миграции

- Все серверы на Windows + MS SQL
- Все серверы в Amazon Cloud
- 12 серверов (m3.medium, m3.large, m4.xlarge c3.xlarge)
- Бизнес логика в ХП в БД

Зачем нужно мигрировать?

1. Увеличение стоимости \$ → почти 2х кратное увеличение стоимости аренды серверов в Amazon
2. На чем можно сэкономить?
 - Оборудование — сократить количество серверов
 - Soft — перейти на MS SQL Standard Edition или на другую БД
 - Open Source — аренда Linux сервера в 8 раз дешевле, чем Windows
 - Люди – снизить зарплаты\уволить

Если не MS SQL, то что?

Open Source:

- MySQL
- **PostgreSQL**
- NoSQL
- In memory



Почему PostgreSQL

1. Близкая архитектура
2. Высокая скорость работы для OLTP
3. Миграция на близкий SQL код
4. Документация подробна, доступна, на русском
5. Сообщество разработчиков
6. Милый логотип



В чем выгода?

- Оплата за каждый сервер на Windows + MS SQL **1.94\$** в час
- На Postgres оплата за каждый сервер **снижается в 8 раз**
Linux + PostgreSQL **0.24\$** в час



План перехода

- Мигрировать часть системы на Linux + Postgres и нагрузку переводить поэтапно
- Выбрали OLTP серверы - меньше по объему кода и данных, и их больше всего
- 2 разработчика из команды

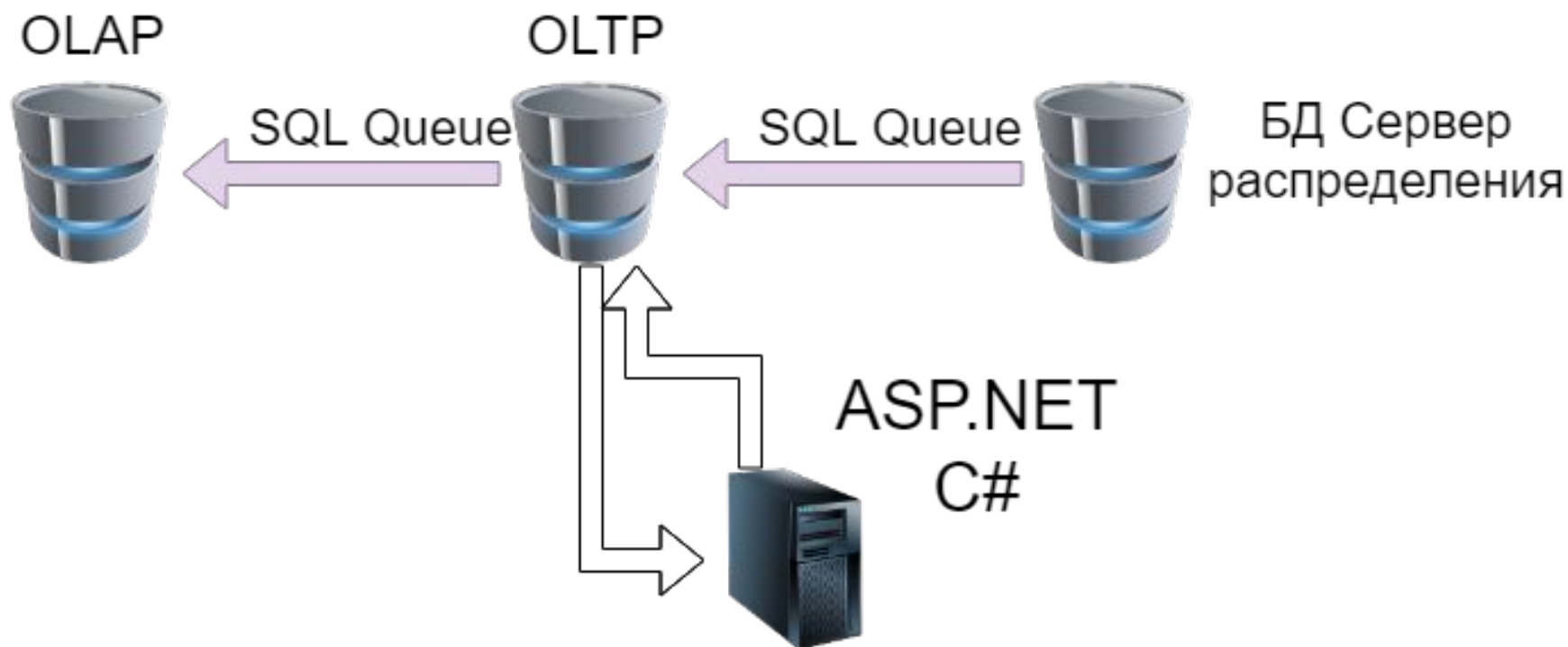
Описание БД для миграции

- 30 таблиц (7 из них справочные)
- 5 таблиц по 20 млн записей
- Транзакции хранятся в течение суток (до 4 млн записей)
- до 2000 в минуту бизнес транзакций в пиковую нагрузку

Процесс обработки

- Данные добавляются в БД из очереди с MS SQL сервера
- Периодическая выдача информации C# приложению
- После обработки, C# приложение обновляет данные в БД
- Транзакции через очередь на сервер OLAP

Схема обработки запроса



Задачи

1. Первичный перенос данных
2. Перенос бизнес логики в хранимых процедурах
3. Репликация данных с MS SQL серверов
4. Ассинхронная отправка на MS SQL серверы
5. Linked серверы – с MS SQL на Postgres
6. Минимальной изменение кода C# приложения

План миграции

- Новый OLTP сервер на Postgres
- Перенос больших таблиц
- Репликация данных на 2 недели для теста
- Данные на MS SQL и Postgres совпадают?
- Небольшой объем транзакций на новый сервер на отчетный месяц
- Формирование и проверка отчета
- Рабочий объем на новый сервер



Первоначальная загрузка данных

Через приложение Kettle By Pentaho

[http://wiki.pentaho.com/display/
EAI/PostgreSQL+Bulk+Loader](http://wiki.pentaho.com/display/EAI/PostgreSQL+Bulk+Loader)



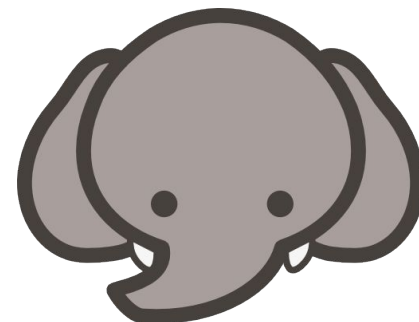
Репликация данных справочников

Собственное решение:

- Триггеры с ведением изменений по таблицам на MS SQL
- Процедура на SQL в Postgres для применения изменений через Linked server
- Объем данных до 500 тыс./24 ч

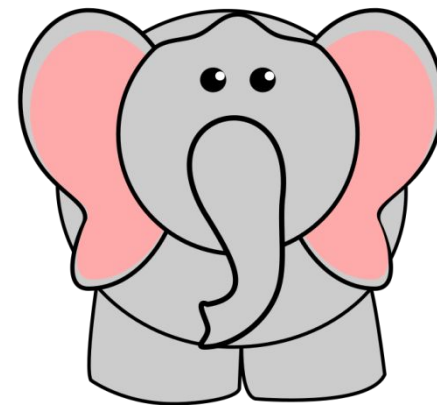
Репликация больших объемов

1. Перенос больших объемов данных в **MS SQL** через **очередь**
2. Использовали **PG queue**
3. При **длительном** простое **очередь** лучше почистить или выключить **перед простоем**



Изменения в приложении

- Приложение на **C# ASP.NET**
- **В MS SQL** ХП, возвращающие **несколько наборов данных**
- На Postgres **только 1** результирующий набор



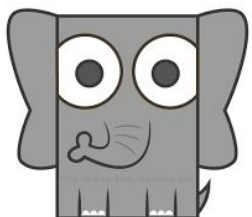
Проблемы

- Распределенные транзакции с MS SQL на Postgres для Linked серверов
- Очереди на стороне Postgres → были подвисания
- Процедуры в БД менялись быстрее, чем их переносили на Postgres
- Выбор кодировки драйвера для организации Linked сервера



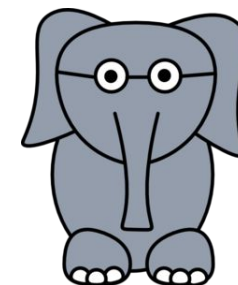
Грабли

1. Система менялась быстрее, чем шла миграция (**23** разработчика против **2**-х)
2. При миграции решили отрефакторить код
3. Отсутствие эксперта в выбранной технологии
4. Не было планового обучения команды



Уроки

- **Поэтапный переход**, разбитый на простые шаги
- Перевод части на Postgres + Linux, остальная на **Windows + MS SQL**
- БД с **бизнес логикой в ХП** – до **80% SQL кода переносимо**



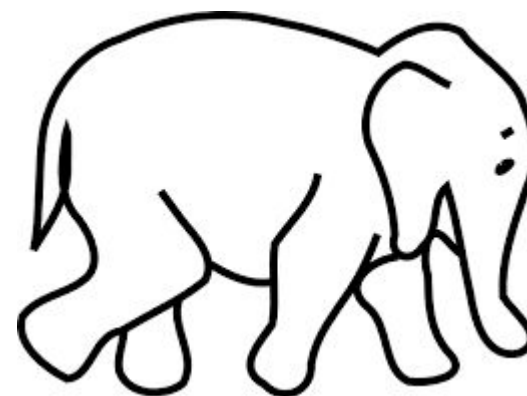
Мигрируем?

- ✓ **Часть для переноса**
- ✓ **Команда** на миграцию
- ✓ **Эксперт** в выбранной технологии
- ✓ **Тесты**
- ✓ **Документация** либо команда, знает как работает система



ИТОГИ

- Postgres подключен вместе с MS SQL
- Планировали **завершить** за полгода
- Первичное включение - через 1 год
- Полная нагрузка - через 1,5 года



ИТОГИ

- Миграция заняла в **3 раза больше времени**
- Postgres обучилось **30% команды**
- **Сервер**, при той же нагрузкой, **стоит дешевле в 8 раз**

