Проект по миграции СУБД

Антон Немцев
netrika.ru

Пациент

Data Warehouse система в АО «Росагролизинг»

- Получает данные из других систем РАЛ.
- Обрабатывает, пересчитывает и готовит Data Warehouse.
- Источник данных для аналитических отчетов и других систем РАЛ.
- Хранит финансовые данные, важна полная сходимость.
- Исходно
  MS SQL Server 2008 + MS SSIS 2008.
- После миграции
  Postgres Pro Standard ver. 14 + Apache Airflow.

Количественные показатели

- 6 Тб данных.
- 9 ETL пакетов.
- 131 процедура, от 500 до 12 000 строк кода.
- 17 функций.
- 114 представлений.
- 341 таблица, в каждой:
  - от 20 до 450 колонок,
  - от 20 тысяч до 4 миллиардов строк.
- 3 внешние CLR функции.
- Мощность ЦПУ: 2.1 GHz в 8 потоков.
- Объем оперативной памяти 48 Gb.
- Объем жесткого диска 7.5 Тб.

Этапы работ

~ 1 месяц на обследование.

~ 1 месяц на первую итерацию создания инструментов для миграции.

~ 3 месяца на миграцию + отладку + запуск в прод.

Этап	Наименование работ
Миграция пакета №1	Миграция общей структуры пакета и разработка в рамках DAG файла
	Тестирование пакета и SQL скриптов
	Оптимизация производительности процедур
Миграция пакета №2 (идентичен №1)	Копирование структуры и тестирование пакета и SQL скриптов
Миграция пакета №3	Миграция общей структуры пакета и разработка в рамках DAG файла
	Тестирование пакета и SQL скриптов
	Оптимизация производительности процедур
Миграция пакета №4	Миграция общей структуры пакета и разработка в рамках DAG файла.
	Тестирование пакета и SQL скриптов
	Оптимизация производительности процедур
Миграция пакета №5	Миграция общей структуры пакета и разработка в рамках DAG файла.
	Тестирование пакета и SQL скриптов
	Оптимизация производительности процедур
Миграция пакета №6	Миграция общей структуры пакета и разработка в рамках DAG файла.
	Тестирование пакета и SQL скриптов
Миграция пакета №7	Миграция общей структуры пакетаи разработка в рамках DAG файла.
	Тестирование SQL скриптов и пакета
Миграция пакета №8	Миграция общей структуры пакета и разработка в рамках DAG файла.
	Тестирование скриптов и пакета
Миграция пакета №9	Миграция общей структуры пакета и разработка в рамках DAG файла.
	Тестирование пакета и SQL скриптов
Финальное тестирование	

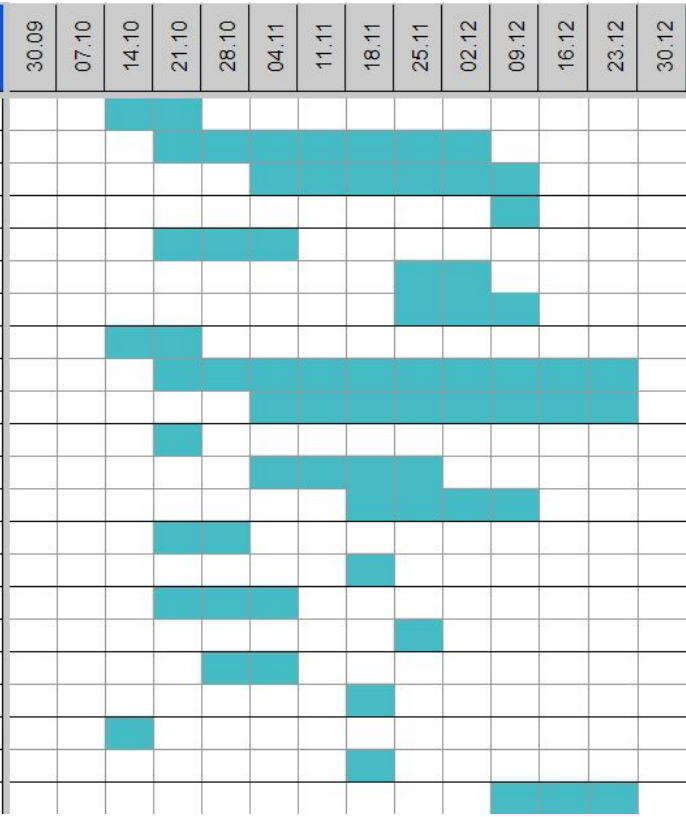

График работ по миграции

Команда проекта

- 2 эникейщика, которые мигрировали SQL.
- 2 эникейщика, которые мигрировали SSIS.
- 1 технический специалист заказчика.
- несколько менеджеров с обеих сторон.

Что мы сделали на обследовании

1. Вникли в систему.
2. Зафиксировали скоуп.
3. Выявили первые “мины”.
4. Зафиксировали требования в формате ТЗ.
5. Сделали оценку.
6. Поняли, как автоматизировать процесс.

Инструменты для миграции

- Скрипт переписи SQL кода in-place.
- Скрипт миграции SSIS пакетов путем генерации DAG файла с тем же наборов step’ов и task’ов.
- Скрипт копирования данных для одинаковой структуры БД, применяется бинарный BULK copy.
- Скрипт сравнения 100% соответствия данных в БД MS SQL и Postgres, а также проверки на 20 контрольных примерах.

Всё самописное. Найденные готовые решения теряют комментарии и форматирование кода.

Скрытые мины

1. Функциональные баги в исходном коде.
2. Скрытые баги в исходном коде, например, недетерменированность.
3. Имена объектов длинные + на русском языке.
4. Отсутствие возможностей в Postgres 14: возврат нескольких наборов данных из процедур, merge и др.
5. Различия приведения типов данных (float -> decimal) и в операции округления.
6. Скрытые зависимости.

Итерационная работа по исправлению кода и в MS SQL, и в Postgres.

Оптимизация производительности

1. Изучали планы выполнения по шагам и выявляли проседания.
2. Заменяли непроизводительные запросы. Outer apply -> оконная функция.
3. Убирали лишние индексы, добавляли нужные.
4. Вводили дополнительную статистику для планировщика запросов.
5. Тюнили сервера Linux и Postgres. Например, параметр read-ahead-kb значительно увеличивает производительность последовательного чтения.

(!) Следите за актуальностью visibility map в Postgres. Иначе потеряете эффект от покрывающих индексов.

Таким образом оптимизировали один запрос с 16 часов до 30 секунд

Железо

Сохранили аналогичные мощности.

Оптимизацию приостановили когда исполнение стало медленнее исходного в среднем на 20-30%. Это допустимо для заказчика.

Можно лучше? Однозначно да.

Перенос данных, ввод в прод

1. Сначала 2 дня переносили текущие данные в новую систему + 1 день строили индексы (не копируйте с включенными индексами - МЕДЛЕННО).
2. На выходных:
   a. остановили систему,
   b. дозалили новые данные,
   c. обновили старые данные,
   d. переключили взаимосвязи с внешними системами.
3. Включение обоих инстансов системы:
   a. PG работает полноценно, с ним работают внешние системы и аналитика,
   b. MS работает на тех же самых входных данных как эталон.
4. Отключение MS через месяц опытной эксплуатации.

Ключевые выводы

1. Миграция СУБД это не Rocket Science, а объемный кропотливый труд команды с прямыми руками и светлой головой.
2. Трудоемкость проекта по миграции это функция от количества ошибок в исходном коде и несоответствия возможностей СУБД.

Вопросы?

tg: **@anton_nemtsev**

netrika.ru