Миграция данных с MS SQL Server на PostgreSQL

PostgreSQL для администраторов баз данных и разработчиков

Меня хорошо видно
& слышно?

Защита проекта
Тема: Миграция данных с MS SQL Server на PostgreSQL

Константин Кузнецов

Разработчик баз данных MS SQL Server

План защиты

- Цель и задачи проекта
- Какие технологии использовались
- Что получилось
- Выводы
- Вопросы и рекомендации

Цель и задачи проекта

Цель проекта: хранить и обрабатывать данные о технических характеристиках транспортных средств средствами PostgreSQL

1. Выбрать инструменты для миграции данных;
2. Импортировать данные из базы MS SQL Server в PostgreSQL;
3. Сравнить быстродействие заполнения справочника на PostgreSQL.

Какие технологии использовались

1. Apple iMac (CPU Intel Core i5 2.7 MHz 4 core; memory 24 GB, SSD 1TB; macOS 10.13.6, VMware Fusion 11.5);
2. Windows 10 Pro на VMware Fusion 11.5:
    1. CPU 2 core; memory 8 GB; Vmware Virtual NVMe Disk 160 GB;
    2. MS SQL Server 2019 (15.0.2155.2) Developer Edition;
    3. psqlodbc 17.0.7;
3. Ubuntu 24.04.3 server на Vmware Fusion 11.5:
    1. CPU 2 core; memory 6 GB; Vmware Virtual NVMe Disk 100 GB;
    2. PostgreSQL 18.1;
    3. pgloader 3.6.10;

Что получилось

Выбран инструмент для переноса данных.

Выбор осуществлялся из следующих бесплатных инструментов:

- Драйвер ***ODBC PostgreSQL Unicode*** для Windows;
- Утилита ***pgloader*** для Linux;

Выбран pgloader, справившийся с большим объёмом данных.

```
sudo apt-get install pgloader
```

Что получилось

Настроены параметры импорта в файле /tmp/pgloader/otusproject.load

```
load database
                from mssql://xxxxxxx:xxxxxxx@192.168.10.139:53969/OtusProject
                into pgsql://xxxxxxxx:xxxxxxxxx@192.168.10.158/otusproject

-- исключить таблицы по имени
excluding table names like 'sys%' in schema 'dbo'

-- отобрать таблицы по имени
including only table names like '%' in schema 'dbo'

-- создавать таблицы в схеме с новым именем
alter schema 'dbo' rename to 'dbo2'

set mssql parameters textsize to '104857600'

-- задать опции
with create schemas, include drop, truncate, disable triggers, create tables, create indexes, drop indexes, reset sequences,
prefetch rows = 1000

-- настроить параметры работы по нагрузке при переносе данных
set work_mem to '16MB', maintenance_work_mem to '512MB', timezone to 'UTC', client_encoding to 'UTF-8'

-- задать правила преобразования типов
cast type bigint to bigint, type geometry to bytea, type geography to bytea, type smallmoney to money, type tinyint to smallint,
type smallint to smallint, type date to date

-- если схема dbo существует, то удалить ее, зарегистрировать расширение для ограничения по умолчанию типа uuid
before load do $$ drop schema if exists dbo2 cascade; $$, $$ CREATE EXTENSION IF NOT EXISTS "uuid-ossp"; $$
```

Что получилось

Создан файл настроек провайдера FreeTDS ~/.freetds.conf

```
[global]
    tds version = 7.4
    client charset = UTF-8
```

Что получилось

Произведён импорт.

```
pgloader /tmp/pgloader/otusproject.load
```

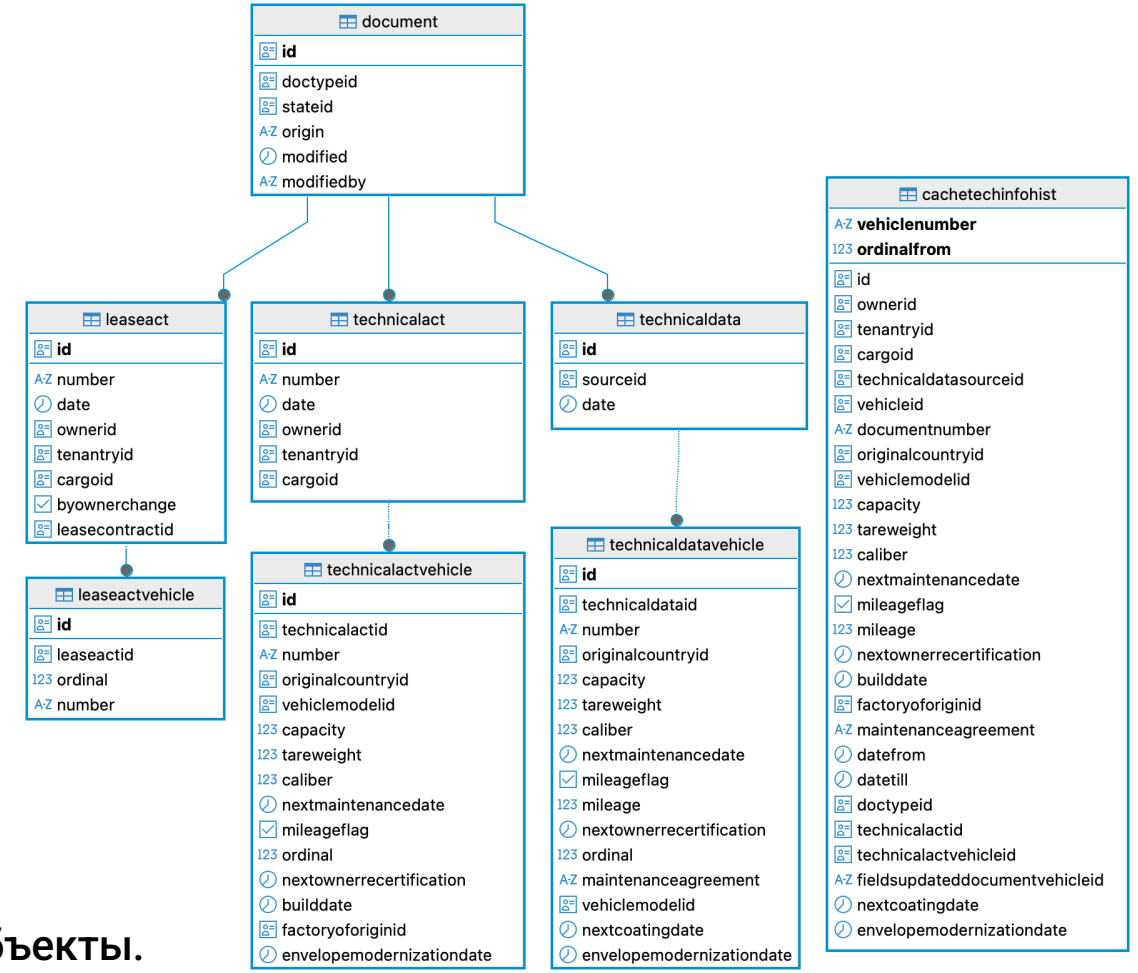
table name	errors	rows	bytes	total time	
before load		0	2		0.132s
fetch meta data		0	85		0.428s
Create Schemas		0	0		0.002s
Create SQL Types		0	0		0.006s
Create tables		0	26		0.064s
Set Table OIDs		0	13		0.006s
dbo2.cachetechinfohist		0	0		0.151s
dbo2.documenterrordatalog		0	24580	3.0 MB	1.567s
dbo2.document		0	78517	10.9 MB	5.470s
dbo2.documenttype		0	3	0.4 kB	0.600s
dbo2.leaseact		0	7224	1.3 MB	1.442s
dbo2.technicalact		0	6163	1.0 MB	1.075s
dbo2.technicaldata		0	65130	5.9 MB	3.093s
dbo2.documenterrortype		0	26	2.9 kB	0.391s
dbo2.enum		0	92	9.6 kB	0.965s
dbo2.leaseactvehicle		0	143032	11.8 MB	6.110s
dbo2.technicaldatavehicle		0	7286807	992.7 MB	3m17.710s
dbo2.technicalactvehicle		0	80226	17.0 MB	6.944s
dbo2.technicaldatasource		0	6	0.4 kB	0.335s

table name	errors	rows	bytes	total time	
COPY Threads Completion		0	4		3m24.658s
Create Indexes		0	60		3m42.752s
Index Build Completion		0	60		1m1.699s
Reset Sequences		0	0		0.120s
Primary Keys		0	13		0.018s
Create Foreign Keys		0	12		8.299s
Create Triggers		0	0		0.001s
Install Comments		0	0		0.000s
Total import time		✓	7691806	1.0 GB	8m17.547s

Результат:
схема ‘dbo2’, таблицы, индексы, ограничения; загружены данные таблиц.

Что получилось

Новая схема ‘dbo2’ содержит таблицы с данными для заполнения справочника.



… а также другие вспомогательные объекты.

Что получилось

Заполнены справочники в обеих СУБД

Заполнение справочника в PostgreSQL (psql):

```
call dbo2.cachetechinfohist_calc(null, false, false, false);
```

2 дня 11 ч 48 мин 50 сек

Заполнение справочника в MS SQL Server (sqlcmd):

```
sqlcmd -S 192.168.10.139\A -d OtusProject -E
    -Q "EXEC [dbo].[CacheTechInfoHist_Calc] NULL, 0, 0, 0"
```

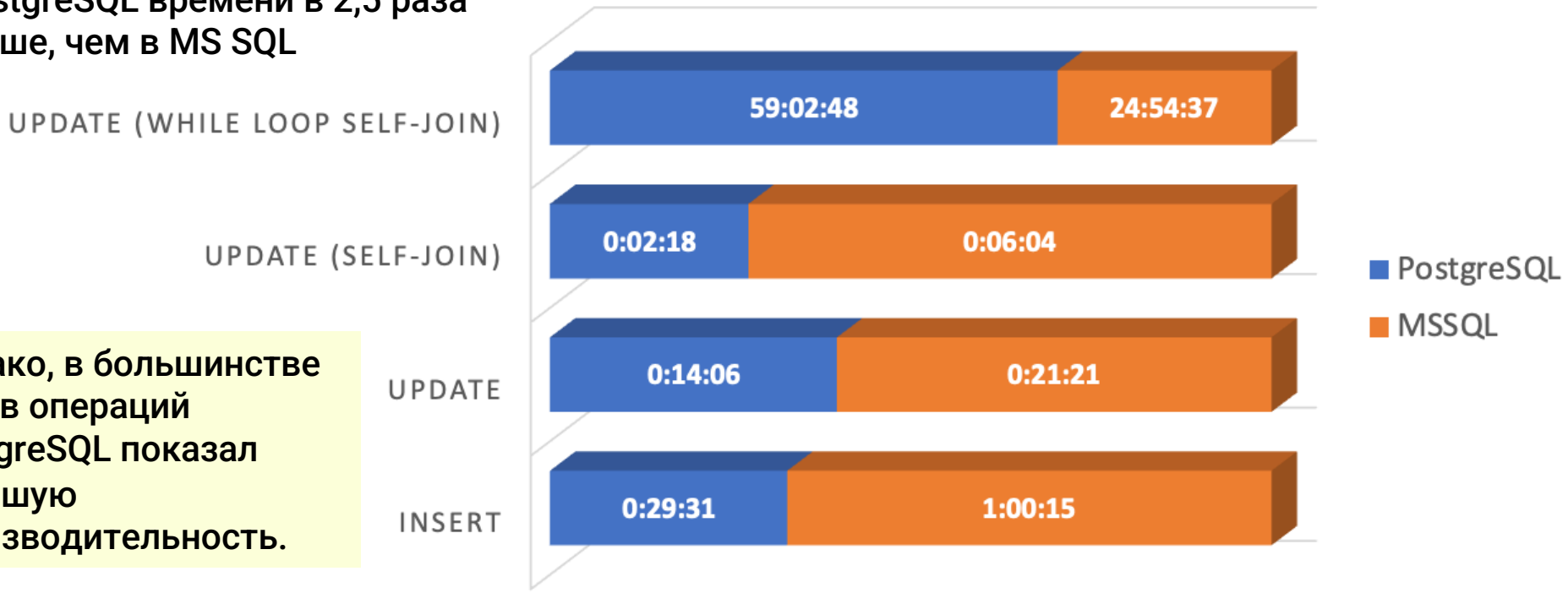
1 день 2 ч 22 мин 18 сек

Обработка данных справочника в PostgreSQL
заняла в 2,5 раза больше времени, чем в MS SQL.

Что получилось

Собрана статистика производительности схожих операций в разных СУБД

Выявлен тип операций, требующий в PostgreSQL времени в 2,5 раза больше, чем в MS SQL



Однако, в большинстве типов операций PostgreSQL показал большую производительность.

Выводы

1. Цель достигнута – данные импортированы, все задачи выполнены;
2. Легче получился физический перенос данных. Проблемы в основном описаны. Главные трудности скрываются за небольшими проблемами из-за недостатка навыков и незнакомых инструментов.
3. Работа над проектом заняла около 2х недель.
4. Проект оказался очень полезным. Стал виднее круг проблем, которые придётся решать при переносе решений на PostgreSQL.

Подробнее в README.md

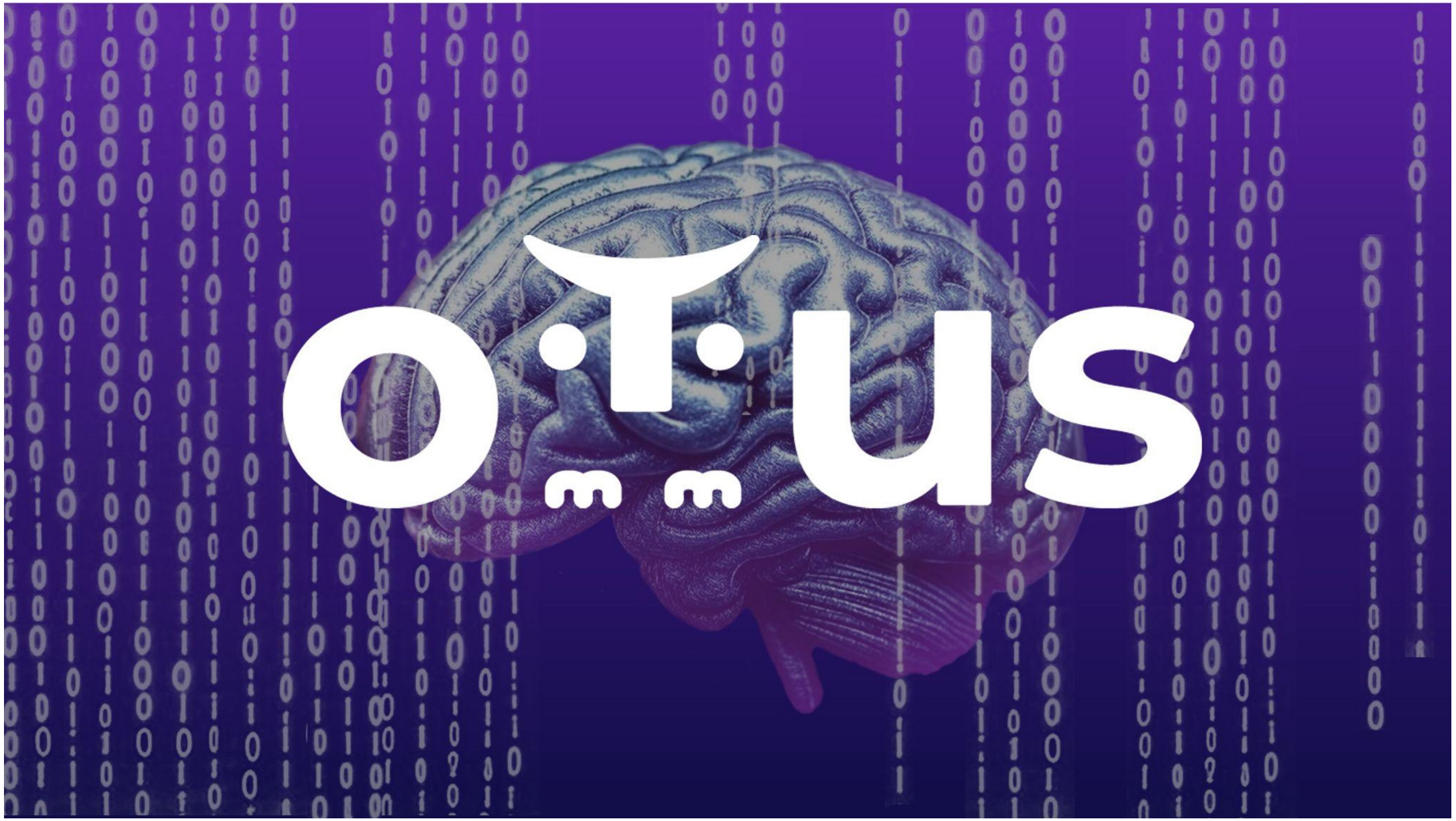
Ответьте на вопросы одногруппников и преподавателей и получите обратную связь на свою работу

Вопросы и рекомендации

\+ если есть вопросы

— если вопросов нет

Спасибо за внимание!


OTUS