



# Highload Architect

## Greenplum в высоконагруженных системах



Проверить, идет ли запись

**Напишите «+» в чат, если меня  
слышно и видно**





Тема открытого урока

# Greenplum в высоконагруженных системах



**Дмитрий Золотов**

*Flutter Mobile Developer @ Yandex, DevOps, Highload Architect, Fullstack*

- преподаватель курса "Highload Architect", "DevOps - практики и инструменты" в ОТУС

- руководитель курса "Kotlin QA Engineer", "Flutter Mobile Developer" в ОТУС

Контакты:

Telegram: @dmitriizolotov



# Маршрут вебинара

1. Знакомство

2. Об OTUS

3. Архитектура GreenPlum

4. Оптимизация для Highload

5. Команда курса

6. О курсе, программа обучения

7. Бонус: карьерная информация

8. Рефлексия

# Расскажите о себе

- Как вас зовут? Откуда вы?
- Ваш опыт работы в IT?
- С какой основной целью вы записались на занятие?



# Правила вебинара



Активно  
участвуем



Задаем вопрос  
в чат



Вопросы вижу в чате,  
могу ответить не сразу

## Условные обозначения



Индивидуально



Документ



Ответьте себе или  
задайте вопрос

# 06 OTUS

# О компании



## Сфера

ОТУС специализируется на обучении в IT.

Наша фишка — продвинутые программы для специалистов с опытом и быстрый запуск курсов по новым набирающим популярность технологиям.



## Клиенты

Наши партнеры современные технологичные компании.

А обучение и открытые материалы привлекают специалистов разных грейдов: junior, middle, senior, lead.





# Образование в ОТУС



## Программы курсов

OTUS имеет образовательную лицензию, поэтому наши курсы являются программами повышения квалификации и профессиональной переподготовки.

# Направления курсов

Обучение специалистов разных грейдов: junior, middle, senior, lead



- Программирование
- Инфраструктура
- Тестирование
- Аналитика



- Data Science
- Управление
- GameDev
- Информационная безопасность

# Мы в цифрах

130+

**курсов** для junior, middle, senior специалистов и менеджеров

600+

**преподавателей** делятся актуальными знаниями и реальными кейсами, востребованными в IT-индустрии



6

**лет** со дня основания компании

20 000+

**выпускников** уже прошли обучение по программам, адаптированным под запросы ведущих работодателей

430 000+

**ИТ-специалистов** в нашем сообществе, читают наши материалы, учатся и общаются на наших площадках



# Напишите, пожалуйста, в чат подходящую цифру

- 1 - если уже учились у нас в компании
- 2 - если НЕ учились, но слышали о нас
- 3 - если впервые знакомитесь с OTUS



# **Greenplum в высоконагруженных системах**

# Цели вебинара

После занятия вы сможете



1. Настраивать простой кластер GreenPlum
2. Применять правила настройки GreenPlum для высокой нагрузки
3. Осмысленно выбирать решение для хранилища больших данных



# Что такое GreenPlum?

# GreenPlum - PostgreSQL + MPP

- распределенная Open Source реляционная база данных с параллельным выполнением запросов (совместимая с PostgreSQL)
- для хранения данных использует модифицированный PostgreSQL 12.12 (в GreenPlum 7.0) для сегментов и coordinator (ранее назывался master) для планирования и оптимизации запросов (GPORCA)
- обеспечивает отказоустойчивость (координатор + standby-сервер, репликация между сегментами)





# Особенности хранения данных

- поддерживает две модели хранения данных - по рядам (как в классических реляционных БД) и по колонкам (как, например, в ClickHouse)
- возможно включение сжатия по алгоритмам LZW для содержания колонок (например, для исключения дублирующихся строк)
- базы данных могут быть оптимизированы для добавления (уменьшает количество операций VACUUM для очистки помеченных на удаление записей)
- поддерживается безопасность на уровне записей
- как и в PostgreSQL поддерживается SQL/JSONPath (7.0+)

# Отличия от PostgreSQL

- обеспечивает контроль ресурсов для заданий (возникают после планирования запроса), при этом ресурсы назначаются на пользователя или роль (через CGroups)
- при необходимости координатор перепланирует выполнение заданий между другими серверами, содержащими реплику сегмента
- изменение топологии кластера может быть выполнено “на лету” с добавлением новых standby-серверов и реплик
- JIT-компиляция запросов (7.0+), интегрирован pgvector (поиск векторного эмбединга по сходству)
- Поддерживаются индексы B-Tree, Bitmap, Gist, GIN index, BRIN index (7.0)
- Утилиты (gp) подключаются к серверам через ssh с управляющего сервера

# Установка

- DEB/RPM-пакет
- Docker-контейнер (есть только для версии 6)
- Из исходных текстов

После установки:

- `/usr/local/gpdb/greenplum_path.sh`
- `make create-demo-cluster`
- `source gpAux/gpdemo/gpdemo-env.sh`
- `createdb -h localhost -p 7000 gpdb`
- `(make installcheck-world для бенчмарка/тестов)`

# LIVE

# Распределенное хранение

- при создании можно указать распределение по сегментам по хэшу DISTRIBUTED BY(column), случайно DISTRIBUTED RANDOMLY, также можно задать репликацию
- при очистке через VACUUM/FULL VACUUM операция выполняется на всех сегментах
- поддерживаются распределенные транзакции (с MVCC), каждая транзакция имеет идентификатор (могут переполниться, надо чистить VACUUM'ом)

# Обработка данных в GreenPlum

- поддерживаются хранимые процедуры (PL/Container) на Java, C, Perl, Python, R
- может интегрироваться с Apache MADLib для задач машинного обучения
- через модуль postgresml можно подключать ML-модели и использовать их совместно с данными
- совместно с GPSS (StreamingServer) можно делать ETL-конвейер прямо внутри кластера GreenPlum
- можно загружать данные из внешних источников параллельно (через gpfdist)
- поддерживается интеграция с внешними базами данных (PXF), в том числе через Foreign Tables
- поддерживается полнотекстовый поиск и PostGIS

# Вопросы?



Задаем  
вопросы в чат



Ставим “-”,  
если вопросов нет



# Оптимизации



# Оптимизации под Highload

- GreenPlum поддерживает несколько видов таблиц (обычные/колоночные, ориентированные на добавление), правильный выбор влияет на производительность агрегаций и на время ожидания при изменении схемы (добавлении/удалении столбцов) `CREATE TABLE ... () WITH (appendonly=true)`, также можно указать `orientation=column` и сжатие: `HEAP` - для часто изменяемых, `COLUMN` - для фактов
  - в 7.0+ не копируется таблица при добавлении столбца

# Оптимизации под Highload

- Ограничение ресурсов через группы ресурсов
- Уменьшаем количество full scan, как можно больше храним в памяти, избегаем лишних запросов в плане выполнения (explain) и повторного обращения к таблицам (можно использовать временные таблицы для исключения передачи данных по сети)
- Инструменты: gpcheckperf, <https://github.com/pivotal/TPC-DS>

# Ограничение ресурсов

- `gpconfig -c gp_resource_manager -v "group"`
- затем `create resource group rg_new with (concurrency=5, cpu_rate_limit =20, memory_limit =30, memory_shared_quota =20, memory_spill_ratio=30);`
- `select * from gp_toolkit.gp_resgroup_config;`
- `CREATE ROLE r1;`
- `ALTER ROLE r1 RESOURCE GROUP rg_new;`

# Ограничение ресурсов

- **CONCURRENCY** - максимальное количество одновременных транзакций
- **CPU\_RATE\_LIMIT** - процент загрузки процессора
- **MEMORY\_LIMIT** - ограничение по размеру памяти
- **MEMORY\_SHARED\_QUOTA** - процент памяти для разделяемых транзакций в группе
- **MEMORY\_SPILL\_RATIO** - предел для транзакций в памяти до отправки на диск

# Оптимизации

- pg\_tune для настройки локальных PostgreSQL (сегментов)
- sysctl для высокой нагрузки  
(<https://www.thegeekdiary.com/sysctl-setting-for-high-load-and-prevent-ddos/>)
- сеть сильно влияет на производительность (особенно в broadcast-плане, когда таблица распространяется по всем сегментам)
- ориентируемся на bench + canary query

# УТИЛИТЫ

- **gpconfig** - управление postgresql.conf (например, тюнинг) и кластером
- **gpcheckperf** - проверка производительности
- **gpbackup** - резервное копирование
- **gpcheckcat -O** - проверка консистентности каталога
- **gpcopy** - копирование между кластерами
- **gpstate** - состояние кластера
- **gpmemwatcher / gpmemreport** - отчет об использовании памяти
- **gpfdist** - сервер параллельного скачивания
- **analyzedb** - статистика обращений к таблицам

# Служебные tables/views

- **gp\_segment\_configuration** - настройка сегментов
- **gp\_stats** - измеренная статистика
- **gp\_config** - эффективная конфигурация (изменения в 7.0

[https://docs.vmware.com/en/VMware-Greenplum/7/greenplum-database/ref\\_guide-guc-changes-6to7.html](https://docs.vmware.com/en/VMware-Greenplum/7/greenplum-database/ref_guide-guc-changes-6to7.html))

[https://docs.vmware.com/en/VMware-Greenplum/7/greenplum-database/ref\\_guide-system\\_catalogs-catalog\\_ref-views.html](https://docs.vmware.com/en/VMware-Greenplum/7/greenplum-database/ref_guide-system_catalogs-catalog_ref-views.html)

# Обслуживание

- **VACUUM catalog** - очистка каталога
- **REINDEX catalog** - переиндексация
- **ANALYZE catalog** - исследование ресурсов каталога
- **VACUUM FULL catalog** - убирает лишнее (перестраивает базу данных)
- **REORGANIZE table** - упорядочивание таблицы
- **VACUUM table** - очистка таблицы
- Удаление idle connections



# Вопросы?



Задаем  
вопросы в чат



Ставим “-”,  
если вопросов нет

# Ключевые тезисы

1. GreenPlum позволяет создать отказоустойчивую распределенную базу данных, совместимую с PostgreSQL
2. Для оптимизации можно использовать как общие highload-рекомендации, настройки PostgreSQL (сегментов) и оптимизации схемы данных в GreenPlum
3. Для распределенной системы важно непрерывно мониторить (с использованием внутренней статистики и технических метрик) сервера кластера

# Вопросы?



Задаем  
вопросы в чат



Ставим “-”,  
если вопросов нет



# Знакомство с командой и программой курса

# Процесс обучения



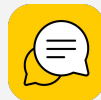
Обучение выстроено в формате вебинаров (онлайн). Онлайн-вебинары проводятся по вечерам или в выходные дни



Все записи занятий и материалы, предоставляемые преподавателями, сохраняются в личном кабинете и остаются доступны даже после окончания обучения



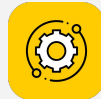
Домашние задания позволят Вам применить на практике полученные во время вебинаров знания. По каждому домашнему заданию преподаватель дает развернутый фидбек



В процессе обучения Вы можете задавать преподавателю вопросы по материалам лекций и домашних заданий, уточнять моменты, которые были непонятны на уроке



Время на обучение: от 4 ак. часов на занятия и 4-8 часов на домашнюю работу в неделю



Программа обучения на курсах обновляется каждый запуск в зависимости от актуальных запросов в сфере IT-технологий

# Карьерная информация



# Анализ позиции Highload Architect

# 2219

Вакансий Highload Architect в  
октябре 2023 г. для соискателей  
уровня Middle в России.

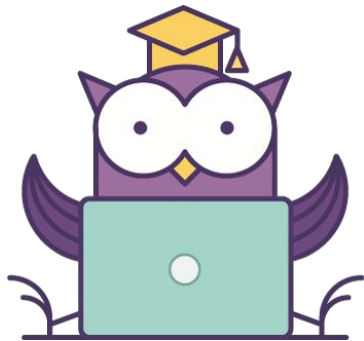
\*вакансии для аудитории с опытом работы от года до трех лет. Источник — hh

# Аналитика зарплатных предложений

(Junior - Senior)

## Медиана

170 000



Источник — hh





# Вакансии работодателей

## PHP-разработчик (Middle+/Senior backend developer)

до 150 000 ₽ на руки

### Требования:

- PHP 7;
- Опыт работы над коммерческими проектами от 2-х лет;
- Опыт работы с фреймворками Yii/Laravel;
- Опыт работы Git и GitLab;
- Знание работы реляционных СУБД. Опыт разработки структур баз данных;
- Знание языка запросов SQL (join, group by, вложенные select запросы, union), профилирование запросов;
- Базы данных MySQL, PostgreSQL;
- Знание HTML, CSS на уровне внести простые правки в готовый код;
- Unit tests;
- Опыт работы с менеджерами очередей, понимание концепции асинхронной работы;
- Опыт работы с Docker;



# Вакансии работодателей

## PHP-разработчик

от 150 000 до 200 000 ₽ на руки

### Требования:

- опыт разработки на php/golang (приоритет - php фреймворк symfony);
- знание SQL, понимание реляционных и NoSQL БД;
- знания алгоритмов, структур данных;
- уверенное знание git, опыт совместной разработки;
- умение разбираться в чужом коде;
- понимание работы docker, kubernetes;
- базовые знания linux;
- понимание процессов разработки и их устройства: code review, CI/CD;
- работа с REST;

Источник — hh

# Вакансии работодателей

## Highload PHP Developer

от 200 000 до 500 000 руб. на руки

### Требования:

- Отличное знание PHP 7 от 3-х лет (важно умение программировать, а не только знание фреймворков)
- Хорошее знание MySQL / SQL
- Хорошее знание HTML, CSS, JS (фронтом заниматься не придётся, но разбираться нужно)
- Практический опыт нагруженных баз данных (шардинг, репликация и всё такое)
- Опыт работы с высоконагруженными сервисами (highload)
- Фокус на легко поддерживаемую простую архитектуру
- Принципы REST API, SOLID, DDD
- Готовность работать в условиях нечетких и меняющихся требований
- Умение работать в командной строке
- Здравомыслие

Источник — hh

# Рефлексия

# Цели вебинара

## Проверка достижения целей

1. Настраивать простой кластер GreenPlum
2. Применять правила настройки GreenPlum для высокой нагрузки
3. Осмысленно выбирать решение для хранилища больших данных

# Список материалов для изучения

1. «Оптимизация запросов PostgreSQL» — Г. Домбровская, Б. Новиков, А. Бейликова
2. <https://postgrespro.ru/education/books/dbtech>
3. <https://pgconf.ru/2017/94457>

# 0 курсе



## *Highload Architect*



старт обучения: 31.10.2023

# Заполните, пожалуйста, опрос о занятии

Важно! Пройти опрос могут только залогиненные пользователи платформы OTUS



# Спасибо за внимание!