



721.07

Рейтинг

OTUS

Цифровые навыки от ведущих экспертов

[Подписаться](#)

MaxRokatansky

8 часов назад

5 основополагающих советов по управлению базами данных Greenplum

Средний 9 мин 395

Блог компании OTUS, Администрирование баз данных*

[Перевод](#)

Автор оригинала: Jingyu Wang

Greenplum — это массивно-параллельная (MPP) база данных с открытым исходным кодом, предназначенная для организации хранилищ данных и высокопроизводительной аналитики. Как и в случае с другими MPP-СУБД, она требует регулярной оптимизации запросов, корректировки распределения ресурсов и защиты данных. В этой статье мы рассмотрим пять рекомендаций, обязательных для эффективного управления Greenplum.

1. Правильно подбирайте политику распределения данных

Greenplum можно рассматривать как MPP-адаптацию PostgreSQL. Она расширяет возможности PostgreSQL, внедряя в ее операции параллелизм. И чтобы обеспечить параллельное выполнение запросов Greenplum распределяет данные по множеству машин, называемых сегментами. Greenplum предоставляет пользователям возможность выбрать один из трех методов распределения данных: хэш, случайное и реплицированное распределение.

По умолчанию Greenplum использует **хэш-распределение** (hash distribution). Если пользователь создает таблицы без указания метода распределения, система автоматически распределяет кортежи на основе первого хэшируемого столбца. Однако рекомендуется явно определять политику распределения. При выборе хэш-распределения важно выбрать столбец, который часто используется в операциях объединения

или в качестве условия фильтрации. Наиболее распространенными вариантами обычно служат первичные ключи или часто используемые внешние ключи.

Сценарий: База данных интернет-магазина, содержащая таблицу «Sales».

Для максимального распараллеливания запросов (например, операций объединения со связанными таблицами, такими как «Customers» и «Products») следует использовать хэш-распределение по «Customer ID» или составному ключу, включающему «Customer ID» и «Date of Purchase».

```
CREATE TABLE "Sales" (  
    sale_id SERIAL,  
    customer_id NUMERIC,  
    customer_name VARCHAR(255),  
    product_name VARCHAR(255),  
    purchase_date DATE,  
    sales_amount DECIMAL(10, 2)  
) DISTRIBUTED BY (customer_id);
```

Случайное распределение (random distribution) — менее распространенная опция Greenplum. В режиме случайного распределения данные распределяются по сегментам классическим алгоритмом round-robin. Однако никаких механизмов гарантирующих равномерность или какой-либо другой шаблон распределения системой не предусмотрено. Случайное распределение может подойти, если у вас нет какого-либо преобладающего шаблона запросов, зависящего от конкретных столбцов.

Сценарий: Мобильное приложение, которое ведет таблицу «UserEvents».

В таблицу «UserEvents» записываются различные пользовательские события, такие как клики, запуски приложения и транзакции. События генерируются случайным образом, без какого-либо определенного паттерна. Для эффективного хранения пользовательских событий и предотвращения перекосов данных следует использовать случайное распределение — это позволит подстроиться под нестабильную частоту возникновения пользовательских событий.

```
CREATE TABLE "UserEvents" (  
    event_id SERIAL,  
    timestamp TIMESTAMP NOT NULL,  
    user_id INT NOT NULL,
```

```
event_type VARCHAR(255) NOT NULL  
) DISTRIBUTED RANDOMLY;
```

Распространение с репликацией (replicated distribution) — третий вариант, предлагаемый Greenplum. В этом режиме каждый сегмент сохраняет полную копию всех кортежей в таблице. Преимущество ведения компактной реплицированной таблицы становится очевидным, когда она используется в сочетании с более крупной таблицей. В таких случаях нет необходимости перераспределять данные между сегментами по сети. Вместо этого операции объединения могут параллельно выполняться непосредственно в сегментах.

Сценарий: Организация здравоохранения, обрабатывающая таблицу «HospitalContacts».

При ограниченном количестве больниц и минимальных изменениях данных задача состоит в том, чтобы уменьшить время выполнения запросов и избежать перераспределения данных при выполнении операций объединения. Использование реплицированного распределения дает нам мгновенный доступ к данным на всех сегментах.

```
CREATE TABLE "HospitalContacts" (  
    hospital_name VARCHAR(255) NOT NULL,  
    contact_name VARCHAR(100),  
    phone_number VARCHAR(20),  
    email_address VARCHAR(255),  
    specialized_department_contacts TEXT[]  
) DISTRIBUTED REPLICATED;
```

2. Выберите правильную стратегию секционирования

Секционирование (или партиционирование) данных — процесс разбиение одной большой таблицы на несколько меньших частей, известных как разделы (или секции/партиции). Greenplum предоставляет пользователям два варианта стратегий секционирования на выбор: секционирование на основе диапазонов (range partitioning), которое часто используется для организации данных временных рядов, и секционирование на основе списков (list partitioning), которое позволяет создавать разделы на основе предопределенных списков значений.

Оптимизатор запросов Greenplum осведомлен о схемах секционирования. Он генерирует планы запросов, в которых используется фильтрация разделов (partition elimination). Это означает, что оптимизатор может игнорировать ненужные разделы при выполнении запросов. Фильтрация разделов может происходить во время обработки запроса — это

статическая фильтрация разделов. Или же она может происходить во время выполнения — это динамическая фильтрация разделов.

Сочетание секционирования и распределения дает значительное преимущество в оптимизации производительности запросов. Распределение данных максимизирует параллелизм, при этом каждый сегмент обрабатывает лишь часть всех кортежей. Одновременно с этим секционирование данных еще больше сокращает их объем за счет исключения нерелевантных разделов. В итоге уменьшение объема сканирования таблицы приводит к сокращению времени выполнения запроса.

Сценарий: Данные транзакций онлайн-продаж за несколько лет

Таблица «SalesData» распределяется по столбцу «customer_id». Это гарантирует, что кортежи с одинаковыми ID клиентов будут храниться в одном сегменте. Этот метод распределения данных оптимизирован под запросы, включающие клиентскую аналитику. В то же время таблица «SalesData» разбита на разделы по дате, чтобы эффективно управлять историческими данными. Эта схема секционирования особенно полезна для архивирования старых данных и ускорения запросов, которые связаны с определенным временным интервалом.

```
CREATE TABLE SalesData (  
    transaction_id SERIAL,  
    customer_id INT,  
    transaction_date DATE,  
    product_id INT,  
    order_amount DECIMAL(10, 2)  
) DISTRIBUTED BY (customer_id)  
PARTITION BY RANGE (transaction_date);;  
  
-- Создание разделов для разных лет  
CREATE TABLE SalesData_2021 PARTITION OF SalesData  
    FOR VALUES FROM ('2021-01-01') TO ('2021-12-31');  
CREATE TABLE SalesData_2022 PARTITION OF SalesData  
    FOR VALUES FROM ('2022-01-01') TO ('2022-12-31');  
-- Дополнительные разделы для последующих лет могут быть созданы аналогичным образом
```

3. Подходите к индексам с умом

Greenplum предоставляет поддержку различных типов индексов, включая B-tree, bitmap, GIN, BRIN, хэш, а также пространственное индексирование с помощью расширения PostGIS. Хотя индексы и дают существенные преимущества, они несут за собой определенные накладные расходы. Помимо того, что они занимают дополнительное

дисковое пространство, они требуют регулярного обслуживания для поддержания их актуальности. Тем не менее, при продуманном построении индексы могут значительно повысить производительность запросов. Это особенно заметно при выполнении `index only scan`ов, когда Greenplum извлекает данные непосредственно из индекса, минуя основную таблицу. Хорошо спроектированные индексы должны включать столбцы, которые часто встречаются в запросах.

Greenplum поддерживает как `index scan`, так и `index-only scan`. Более того, индексы Greenplum учитывают секционирование данных. Наряду со стандартными `index` и `index only scan`ами, которые выполняются на всех сегментах, Greenplum предлагает `dynamic index scan`ы и `dynamic index-only scan`ы, которые выполняются только на неисключенных (`unpruned`) разделах.

Сценарий: Данные транзакций онлайн-продаж за несколько лет

Продолжая пример с «SalesData», мы создаем B-tree индекс на основе столбца суммы заказа. Это позволяет использовать `dynamic index scan`, поскольку критерии фильтрации включают как дату транзакции (ключ раздела), так и сумму заказа (индексируемый столбец).

```
CREATE INDEX idx_order_amount ON SalesData USING btree (order_amount);

EXPLAIN SELECT product_id FROM SalesData WHERE transaction_date = '2022-01-01' AND orde
```

4. Регулярно используйте VACUUM и ANALYZE

До сих пор мы обсуждали важность обдуманного выбора распределения, секционирования и индексирования данных в Greenplum. Чтобы эти решения понастоящему продемонстрировали свою ценность, необходимо, чтобы Greenplum точно отражал эти аспекты данных. А это значит, что статистика данных должна оставаться актуальной.

VACUUM и ANALYZE — это две служебные операции, каждая из которых делает очень важную работу. VACUUM освобождает место в хранилище, высвобождая области, помеченные как устаревшие. Этот процесс очень важен для предотвращения чрезмерного разрастания размера базы данных со временем. В процессе работы VACUUM обновляет карту видимости, помечая блоки данных как «set» (указывая на «живые» строки) или «clear» (указывая на «мертвые» строки).

Информация о видимости важна для `index only scan`ов. Если блок данных «set», оптимизатор запросов уверенно извлекает все данные из индекса. И напротив, он пропускает блоки данных, содержащие исключительно мертвые строки. И только когда

карта видимости сигнализирует о наличии смеси живых и мертвых строк, `index only scan` обращается к данным таблицы.

`ANALYZE`, с другой стороны, отвечает за сбор целого набора статистических данных, включая гистограммы столбцов, наиболее распространенные значения (MCV), количество строк, количество различных значений (NDV) и корреляции между столбцами. Эти статистические данные важны для оценки кардинальности, избирательности и перекаса данных. Оптимизатор запросов опирается на эти оценки для определения оптимального использования индексов, стратегий объединения, условий фильтрации и методов доступа.

Обычно `ANALYZE` выполняется автоматически фоновым демоном автоанализа, который проверяет таблицы, превышающие заданный порог обновлений и удалений. Однако для больших таблиц этот порог достигается редко. Поэтому возникает необходимость вручную выполнять `ANALYZE`, в идеале в сочетании с `VACUUM FULL`, через регулярные промежутки времени, чтобы статистика данных оставалась точной.

5. Оптимизация запросов

Greenplum в значительной степени полагается на оптимизатор запросов — важнейший компонент, отвечающий за выбор наиболее эффективного маршрута операций, называемого планом запроса. Чтобы добиться оптимальной производительности запросов, необходимо понимать, как работает оптимизатор запросов Greenplum.

Когда в Greenplum поступает запрос, он проходит этап парсинга и переписывания, на котором и происходят основные преобразования. Затем оптимизатор берет разобранный запрос и исследует множество возможностей его выполнения. При этом учитываются различные факторы, такие как стратегии объединения, методы агрегирования, наличие индексов и методы доступа. В результате этого исследования генерируется несколько потенциальных планов выполнения запроса, удовлетворяющих его требованиям.

Greenplum использует стратегию оптимизации на основе затрат, при которой стоимость различных операций назначается на основе статистических данных. Расчет стоимости помогает оптимизатору выбрать наиболее эффективный план. Как упоминалось ранее, поддержание точной статистики с помощью регулярного выполнения операций `VACUUM` и `ANALYZE` является критически важным для эффективной работы оптимизатора запросов.

Теперь, когда у нас есть представление о работе оптимизатора запросов, мы можем использовать команду `EXPLAIN` для анализа планов выполнения. Она даст нам представление о том, как Greenplum планирует выполнить запрос, и поможет выявить потенциальные узкие места. Если выполнение запроса занимает больше времени, чем предполагалось, обратите внимание на распространенные проблемы, которые могут включать в себя:

- Высокий уровень дискового ввода-вывода, когда на диск записывается значительный объем данных
- План не использует доступные индексы
- Неоптимальные стратегии объединения, например, использование вложенных циклических джоинов, когда более эффективным было бы использование хэш-джоинов
- Большой промежуточный вывод при объединении, сортировке или агрегировании
- Высокий уровень перераспределения или трансляции данных
- Недооценка или переоценка кардинальности
- Отсутствие параллелизма, когда выполнение происходит на координаторе, а не в сегментах
- Условия фильтрации не продвинуты вниз
- Излишнее сканирование разделов или полных таблиц

Иногда изменение запроса в эквивалентную форму может помочь оптимизатору запросов сгенерировать более эффективный план. Это может включать в себя пересмотр объединений, упрощение условий фильтрации или декомпозицию сложных запросов на несколько более простых запросов. Если вы сталкиваетесь с постоянными проблемами производительности, обратитесь в службу поддержки или в сообщество Greenplum. Оптимизация запросов — это непрерывный процесс, часто требующий нескольких итераций анализа, корректировок и тестирования для достижения желаемой производительности.

Станьте мастером интеграции: PostgreSQL, ArenadataDB и Airflow за один урок! 18 марта в 20:00. Зарегистрироваться бесплатно можно на странице курса «Greenplum для разработчиков и архитекторов баз данных».

Теги: базы данных, Greenplum, mpp, субд

Хабы: Блог компании OTUS, Администрирование баз данных



Редакторский дайджест

Присылаем лучшие статьи раз в месяц





OTUS

Цифровые навыки от ведущих экспертов

[Сайт](#) [ВКонтакте](#) [Telegram](#)

113

62.3

Карма Рейтинг

OTUS @MaxRokatansky

Редактор

[Подписаться](#)

Комментарии 1

Публикации

ЛУЧШИЕ ЗА СУТКИ

ПОХОЖИЕ



kulyanin

13 часов назад

Б/У жёсткие диски Western Digital под видом новых в крупном ритейлере



Простой



3 мин



15K

Из песочницы



+86



40



106



yanakuklina

21 час назад

Как мы автоматизировали обжарку кофе и доставляем 100 тысяч заказов в месяц почти без ошибок



Простой



10 мин



7.3K

Кейс



+75



43



23



EI_Gato_Grande

20 часов назад

Need for Speed: Most Wanted. Как легендарная игра уничтожила всю серию

🕒 10 мин 👁 21K

📌 +62

🔖 20

💬 72



MaFrance351

20 часов назад

Кто сообщает картам, где едет ваш транспорт

💧 Средний 🕒 8 мин 👁 4.6K

Обзор

📌 +44

🔖 29

💬 22



EgorKotkin

13 часов назад

Как Кремниевая долина превращает бум ИИ в новый пузырь доткомов

🕒 8 мин 👁 2.8K

📌 +31

🔖 8

💬 17



mikhailsudakov

20 часов назад

История Ultimate Play the Game — легендарного разработчика игр для ZX Spectrum

🕒 26 мин 👁 2.7K

📌 +31

🔖 14

💬 18



km1337

17 часов назад

Что вам надо знать в 2025 году про контейнеры, чтобы не пропустить важное

🕒 12 мин 👁 2.4K

📌 +29

🔖 59

💬 3



vorobeevich

15 часов назад

GigaChat 2.0 в API



Средний



14 мин



4.2K

[Обзор](#)

+28



20



12

ANGIE

VBart

18 часов назад

Стильный современный «autoindex» в Angie/nginx без sms и сторонних модулей



Простой



24 мин



1.2K

[Тutorial](#)

+28



18



4



viktoshaf

15 часов назад

Настройка BGP с Anycast: высокая доступность и отказоустойчивость



Простой



10 мин



2.3K

[Тutorial](#)

+26



13



0

[Показать еще](#)

ВАКАНСИИ КОМПАНИИ «OTUS»

Преподаватель онлайн-курса Разработчик Битрикс24

OTUS · Можно удаленно

[Больше вакансий на Хабр Карьере](#)

ИНФОРМАЦИЯ

Сайт

otus.ru

Дата регистрации	22 марта 2017
Дата основания	1 апреля 2017
Численность	101–200 человек
Местоположение	Россия
Представитель	OTUS

ВИДЖЕТ



ВКОНТАКТЕ

**OTUS. Онлайн-образование**

23 882 подписчика

[Подписаться на новости](#)

БЛОГ НА ХАБРЕ

8 часов назад

5 основополагающих советов по управлению базами данных Greenplum

395 1

10 часов назад

unittest.mock: 5 вопросов на собеседовании

874 0

13 часов назад

Аннотации типов в Python: кратко о главном

2.2K 8

17 часов назад

A/B-тесты в PostgreSQL

1.2K 0

20 часов назад

Разработчики 1C – как с ними дружить?

2.6K 2

Ваш аккаунт	Разделы	Информация	Услуги
Войти	Статьи	Устройство сайта	Корпоративный блог
Регистрация	Новости	Для авторов	Медийная реклама
	Хабы	Для компаний	Нативные проекты
	Компании	Документы	Образовательные программы
	Авторы	Соглашение	Стартапам
	Песочница	Конфиденциальность	



Настройка языка

Техническая поддержка

© 2006–2025, Habr