

Greenplum для разработчиков и архитекторов баз данных

Оптимизация сложных запросов в MPP-кластерах: Greenplum, Arenadata DB, Cloudberry Database



**Меня хорошо видно
& слышно?**



Защита проекта

Тема: Оптимизация сложных запросов в MPP-кластерах: Greenplum, Arenadata DB, Cloudberry Database



Игорь Щербаков

Разработчик баз данных

План защиты

Цель и задачи проекта

Какие технологии использовались

Что получилось

Выводы

Вопросы и рекомендации

Цель и задачи проекта

Цель проекта:

Разбор методов оптимизации сложных запросов в MPP-кластерах

1. Собрать методы оптимизации запросов из разных источников
2. Подготовить стенды MPP-кластеров: Greenplum, Arenadata DB, Cloudberry Database
3. Разобрать методы оптимизации на различных примерах
4. Проанализировать планы выполнения запросов



Какие технологии использовались

1. MPP - Massively Parallel Processing
2. Virtualization, Docker
3. Greenplum, Arenadata DB, Cloudberry Database
4. PXF - Platform Extension Framework
5. Dbeaver, PostgreSQL



Источники по методам оптимизации

1. Лекции
2. Документация по Greenplum: <https://techdocs.broadcom.com/>
3. Документация по Arenadata: <https://docs.arenadata.io>
4. 5 лайфхаков оптимизации SQL-запросов в Greenplum
<https://habr.com/ru/companies/rostelecom/articles/442758/>
- 5.



Методы оптимизации запросов GP (1)

1. Распределение данных, соединения по ключам дистрибуции. Для ключей дистрибуции, по которым будут соединения, использовать одинаковые типы данных.
2. Партиционирование, partition elimination
3. Использование append optimized таблиц, использование колоночной ориентации
4. Использование distributed replicated для маленьких таблиц (справочников)
5. Использование unlogged таблиц
6. Использование temporary таблиц для хранения промежуточных результатов вычислений



Методы оптимизации запросов GP (2)

7. Добиваться равномерного распределения данных между сегментами. Избегать skew – перекосов.
8. По возможности, избегать переноса недостающих данных с одного сегмента на другой: broadcast motion и redistribute motion.
9. По возможности, избегать операций сортировки (order by в запросе, sort в плане запроса)
10. Поддержание статистики в актуальном состоянии, регулярный сбор



Сложные случаи оптимизации

1. Использование Nested Loop:
 - одна из соединяемых таблиц – маленькая,
 - соединение не по равенству: $>$, $<$, $>=$, $<=$, ...
2. Сканирование только индекса.
3. Запросы, для оптимизации которых требуются специальные виды статистики.
4. Использование параметров для управления оптимизацией

Что получилось

1. Развернуты кластеры: Greenplum, Arenadata DB, Cloudberry Database
2. Базы данных заполнены тестовыми данными
3. Из разных источников собраны методы оптимизации запросов
4. Произведен разбор некоторых сложных запросов и планов их выполнения



Выводы

1. MPP-кластеры отстают от БД PostgreSQL в оптимизации запросов
2. Чем больше версия MPP-кластера и БД, входящих в его состав, тем больше возможностей по оптимизации запросов
3. Методы оптимизации, собранные при выполнении данной проектной работы, позволят ускорять запросы в предстоящих проектах
4. Полученные навыки создания MPP-кластеров позволят эффективно разворачивать стенды для разработки и тестирования



Вопросы и рекомендации



если есть вопросы



если вопросов нет

Спасибо за внимание!

