Мысин Н.О., Кузьменко А.Е., Котилевец И.Д., Ксенофонтов Н.В.

Возможности применения Apache Spark с PostgreSQL.

Аннотация В статье рассматривается возможности использования стека технологий Big Data для работы с большими данными. Проведён обзор инструмента Apache Spark. Выявлены основные достоинства. Приведенная практическая реализация использования.

Ключевые слова: PostgreSQL, Apache Spark, крупномасштабная обработка информации, компоненты Apache Spark.

Введение

В последние годы, в связи с непрерывным экспоненциальным ростом количества информации как в частных так и в открытых источниках, классические средства по хранению и обработке структурированных данных такие как СУБД перестали удовлетворять пользователей в части быстродействия и производительности. Потребность поиска решения этой проблемы привела к возникновению ряда легко масштабируемых технологий позволяющих справится с обработкой огромных массивов данных за приемлемое время. Наиболее актуальными в настоящее время считаются следующие из них:

- NoSQL СУБД;
- Google MapReduce;
- Apache Hadoop и входящий в его состав Apache Spark;

В данной работе мы будем сравнивать быстродействие выполнения простых и сложных агрегирующих SQL запросов для таблиц различных размеров средствами классической СУБД PostgreSQL и с помощью параллельных вычислений Apache Spark. Первый раздел будет посвящен теоретическому обзору технологий, второй экспериментальной части - сравнению динамики выполнения различных запросов при изменении объемов данных и степени параллелизма.

Apache Spark

Он представляет из себя фреймворк с открытым исходным кодом для реализации распределенной обработки неструктурированных и слабоструктурированных данных, Spark работает в парадигме резидентных вычислений и обрабатывает данные в ОП (Оперативной Памяти), благодаря чему получает выигрыш в скорости работ некоторых классов задач.

Драйвер Spark - это процесс, который распределяет задачи, поступающие от пользователя по действующим исполнителям. Таким образом, Spark - драйвер преобразует пользовательское приложение на единицы исполнения, которые называются задачи (tasks). Экземпляр Spark - драйвера запускается во время запуска сессии (драйвер Spark создает сессию) Spark при первом запуске приложения и остается активным до тех пор, пока эта сессия активна (приложение работает и сбой не произошел).

Spark - исполнители (Spark executors) - это рабочие процессы, которые

отвечают за выполнение задач, приходящих из драйвера. Исполнители запускаются только один раз при запуске приложения Spark и продолжают свою работу на протяжении всего жизненного цикла программы. Они выполняют задачи, приходящие от драйвера и возвращают результат обратно драйверу Spark. Таким образом, распределенная архитектура приложения Spark позволяет выполнять большие объемы Big Data задач, требующих высокое количество вычислений. Все это делает framework Apache Spark весьма полезным средством для Data Scientist'а и разработчика Big Data приложений.

Плюсы:

1. Apache Spark делает возможной обработку очень больших наборов данных. Он обрабатывает эти наборы данных довольно быстро.

2. Apache Spark довольно хорошо справляется с реализацией моделей машинного обучения для больших наборов данных.

3. Apache Spark, похоже, является быстро развивающимся программным обеспечением, с новыми функциями, делающими программное обеспечение еще более простым в использовании.

Из минусов выделяют следующие:

1. Сложно конфигурировать, нужны дополнительные механизмы и трудозатраты на распределение задач.

2. Плохо справляется при стримминговой обработке данных свыше 2-3 террабайт

PostgreSQL

Система управления базами данных Postgres (она же PostgreSQL) является свободной объектно-реляционной СУБД. Наряду с MySQL, это хорошая альтернатива коммерческим СУБД, таким как Oracle Database или Microsoft SQL Server. Сегодня система управления базами данных PostgreSQL существует в реализациях для разных платформ, включая Linux, Win32, Mac OS X, Solaris/OpenSolaris, FreeBSD, QNX 4.25, QNX 6. Одной из наиболее сильных сторон СУБД PostgreSQL является архитектура. Как и в случаях со многими коммерческими СУБД, PostgreSQL можно применять в среде клиент - сервер - это предоставляет множество преимуществ и пользователям, и разработчикам. В основе PostgreSQL - серверный процесс базы данных, выполняемый на одном сервере. Также стоит сказать, что в Postgres пока не реализована технология высокой готовности, как это сделано в ряде других коммерческих систем управления базами данных уровня предприятия. Доступ из приложений к данным базы PostgreSQL производится с помощью специального процесса базы данных. То есть клиентские программы не могут получать самостоятельный доступ к данным даже в том случае, если они функционируют на том же ПК, на котором осуществляется серверный процесс. Таким образом мы получаем разделение клиентов и сервера, что даёт возможность создавать распределённые системы. Так-же PostgreSQL имеет свои плюсы и минусы:

Из плюсов выделяют:

1. Он хорошо работает с внешними источниками данных и работает на платформах со стабильной производительностью.

2. Клиенты могут быть уверены, что их личная информация будет в безопасности.

3. PostgreSQL работает на многих платформах ОС и поддерживает ANSI SQL, хранимые процедуры и триггеры

Из Минусов выделяют:

1. Увеличение горизонтального масштабирования является сложной задачей, но PostgreSQL может иметь решение для всех реплик для принятия операций.

2. Не требуется изменение порядка столбцов и лучшее сжатие данных.

3. PostgreSQL часто критикуют за то, что он медленный и непригодный для крупномасштабных корпоративных приложений.

Тесты

Тестирование обеих технологий производилось на ПК: (Установленная ОП 16ГБ, SSD 512ГБ, 8 ЦП, ОС Linux Ubuntu 22.04.4)

Таблица к которой писали запросы:

Атрибут	**Тип**	**Как заполняли**
user_id	Serial primery key	Автоинкрементирующийся индекс
name	varchar(20)	Случайное Имя из списка
surname	varchar(20)	Случайная Фамилия из списка
age	integer	Случайные числа от 1 до 120
country	varchar(20)	Случайные Страны из списка
city	varchar(20)	Случайные Города из списка
rating	real	Случайные числа от 1.0 до 10.0

Эксперимент 1.

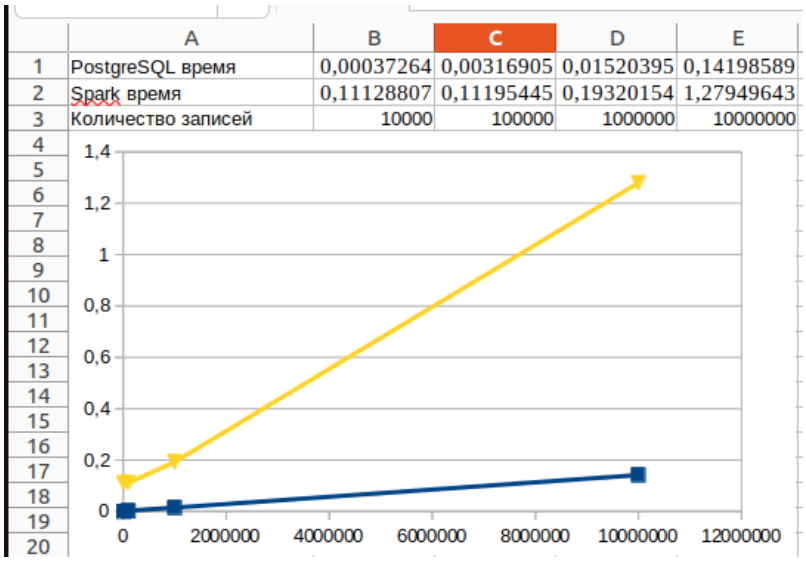
Простой подсчет количества записей: Select count(*) From users;

Сначала зафиксируем количество исполнителей и будем варьировать размеры таблицы. Будем использовать двух исполнителей, выделим каждому по 1ГБ и по 1ЦП).

Количество записей	PostgreSQL(time)	Spark(time)
10000	0.00037264	0.11128807
100000	0.00316905	0.11195445
1000000	0.01520395	0.19320154

10000000	0.14198589	1.27949643

График:



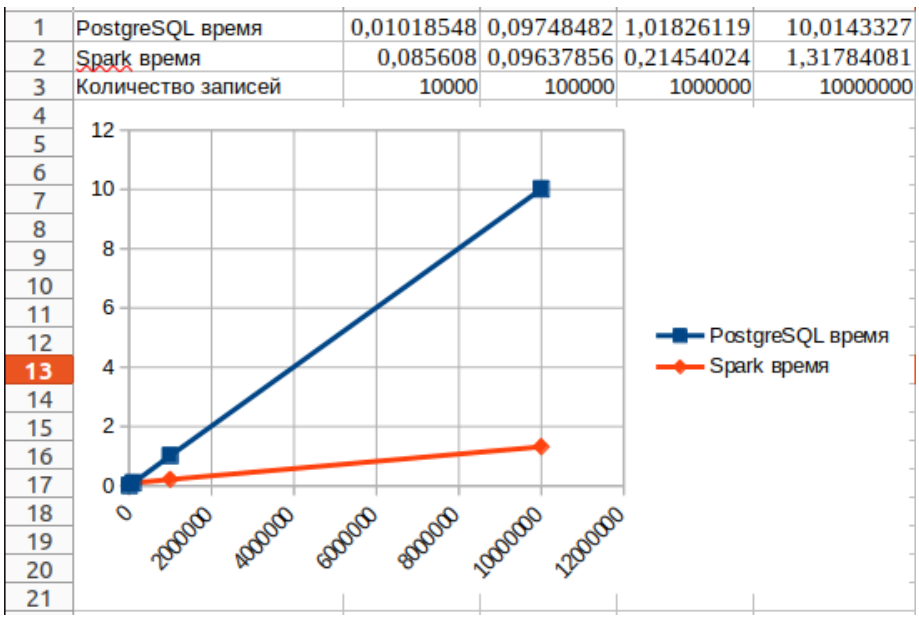
Эксперимент 2.

Вывести все Имена рейтинг которых меньше 5.0: Select name From users Where rating < 5.0;

Сначала зафиксируем количество исполнителей и будем варьировать размеры таблицы. Будем использовать двух исполнителей, выделим каждому по 1ГБ и по 1ЦП).

Количество записей	PostgreSQL(time)	Spark(time)
10000	0.01018548	0.08560800
100000	0.09748482	0.09637856
1000000	1.01826119	0.21454024
10000000	10.0143327	1.31784081

График:



Заключение

На сегодняшний день Apache Spark является одним из основных инструментов Big Data для крупномасштабной обработки информации. Благодаря своей универсальности и скорости работы, Apache Spark отлично может обрабатывать терабайты памяти. Мы исследовали и использовали модуль в Apache Spark, Spark SQL, который обеспечивает множество операций реляционной обработки. Spark SQL использует декларативный API DataFrame для обеспечения реляционных операций и функций, таких как автоматическая оптимизация. В то же время он позволяет пользователям смешивать системные операции и сложные операции анализа в конвейерные операции. Он поддерживает широкий спектр настраиваемой крупномасштабной обработки данных, включая полуструктурированные данные, объединение запросов и типы данных в машинном обучении. В ходе тестов было выявлено, что чем сложнее запрос и чем больше количество данных, тем хуже справляется СУБД PostgreSQL без использования Apache Spark (Spark SQL).

Список используемых источников

1. Дэви Силен, Арно Мейсман Основы Data Science и Big Data. Санкт-Петербург: Библиотека программиста, 2017.

2. Md. Rezaul Karim, Scala and Spark for Big Data Analytics, Книга по Требованию, 2017.

3. Что такое архитектура распределенной среды Spark: [Электронный ресурс]. [сайт]. URL: https://spark-school.ru/blogs/spark-parallel-architecture/ (дата обращения: 10.05.2022).

4. Джошуа Д. Дрейк, John C. Worsley Practical PostgreSQL, 2018.

5. Obe Regina, Hsu Leo PostgreSQL Up & Running, O'Reilly Media, 2014.

6. Документация PostgreSQL 13.7 The PostgreSQL Global Development Group: [Электронный ресурс]. [сайт]. URL: https://postgrespro.ru/docs/postgresql/13/index (дата: 01.05.2022).