УДК 004.652

Чіома Е.В., *студентка 3 курсу спеціальності 122 «Комп'ютерні науки»*
Ніколюк П.К., *д.ф.-м.н., професор кафедри комп'ютерних наук та інформаційних технологій*

GOOGLE BIGQUERY: ХМАРНЕ СХОВИЩЕ ДАНИХ

Донецький національний університет імені Василя Стуса, м. Вінниця

BigQuery від Google – це хмарне сховище даних корпоративного рівня. З моменту створення, сервіс перетворився в економічне і повністю кероване сховище даних, яке може виконувати неймовірно швидкі інтерактивні та спеціальні запити з наборами даних петабайтного масштабу. Крім того, BigQuery інтегрується з різними сервісами Google Cloud Platform (GCP) та сторонніми інструментами.

BigQuery – це безсерверна система, сховище даних представлене як сервіс, в якому відсутнє обладнання та програмне забезпечення для встановлення баз даних. Сервіс BigQuery керує інфраструктурою, включаючи масштабованість, високу доступність, надає простий клієнтський інтерфейс, який дозволяє користувачам виконувати інтерактивні запити. Для роботи з BigQuery, необхідно імпортувати дані в систему, а потім створити запити з використанням діалектів SQL. При цьому розуміння архітектури BigQuery є необов'язковим, але корисним при реалізації різних передових практик, включаючи контроль витрат, оптимізацію продуктивності запитів та сховища. Наприклад, корисним є розуміння розподілення ресурсів і взаємозв'язок між кількістю слотів і продуктивністю запиту [1].

Сховище BigQuery побудовано на основі технології Dremel, яка знаходиться в розробці компанії Google. Dremel – це інтерактивна система спеціальних запитів для аналізу вкладених даних. Завдяки об'єднанню стовпчатого сховища і деревовидної архітектури Dremel BigQuery забезпечує безпрецедентну продуктивність. Як показано на рис. 1, клієнт BigQuery взаємодіє з механізмом Dremel через клієнтський інтерфейс. Borg – система управління великомасштабними кластерами Google розподіляє обчислювальні потужності для завдань Dremel. В свою чергу ці завдання зчитують дані з файлових систем Google Colossus, використовуючи мережу Jupiter, виконують різні операції SQL і повертають результати клієнту.

Найдорожчою частиною будь-якої платформи для аналітики великих даних є забезпечення швидкості передачі даних між жорстким диском та оперативною пам'яттю. Для цього BigQuery зберігає дані в стопчиковому форматі, відомому як Capacitor. Кожне поле таблиці утворює стовпець та

зберігається в окремому файлі, що дозволяє досягти дуже високого рівня стиснення даних і швидкості їх сканування.

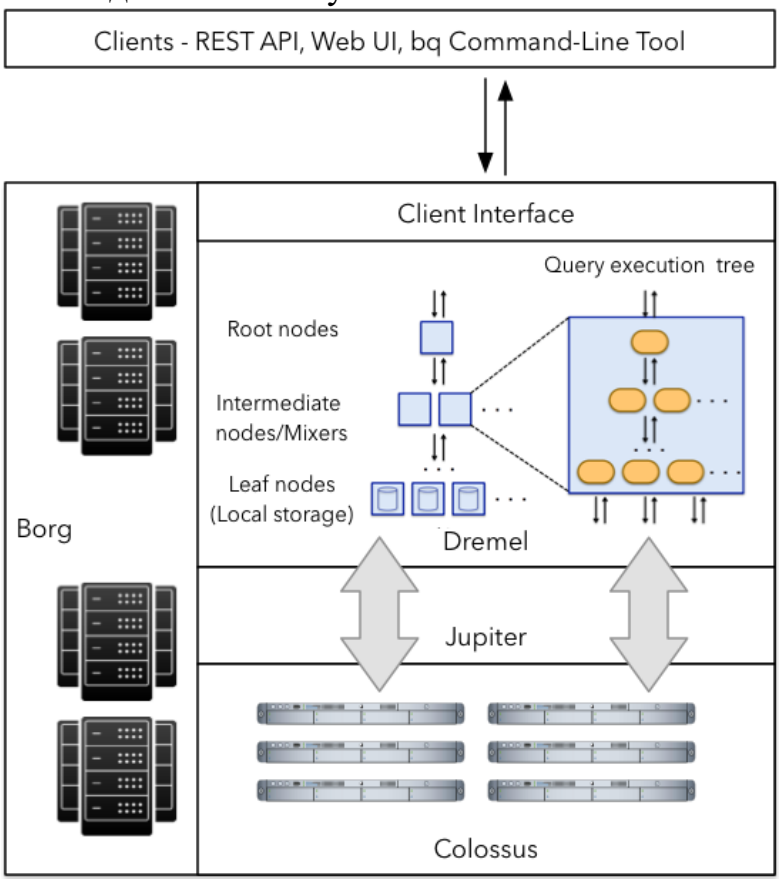

Рисунок 1 - Архітектура високого рівня сервісу BigQuery

Імпортування даних в сховище BigQuery відбувається за допомогою пакетного завантаження або потокової передачі. В процесі імпорту BigQuery кодує кожний стовпець окремо в формат Capacitor. Після того, як всі дані стовпця закодовані, вони записуються назад в Colossus. Під час кодування збирається різна статистика про дані, яка пізніше використовується для планування запитів.

BigQuery використовує Capacitor для зберігання даних в Colossus (розподілена файлова система Google останнього покоління). Colossus виконує реплікацію, відновлення і розподілене управління в масштабі кластера, забезпечує керовану клієнтом реплікацію і кодування. При записі даних в Colossus BigQuery приймає рішення про початкову стратегію сегментування, яка розвивається в залежності від шаблонів запитів і доступу. Після запису даних для забезпечення максимальної доступності BigQuery ініціює геореплікацію даних в різних центрах обробки даних.

Capacitor і Colossus є ключовими складовими в галузі характеристик продуктивності, запропонованих BigQuery. Colossus дозволяє розбивати дані на декілька розділів, щоб забезпечити неймовірно швидке паралельне читання, тоді як Capacitor знижує необхідну пропускну здатність сканування. Разом вони дозволяють обробляти терабайт даних в секунду [2].

Незважаючи на те, що існує декілька альтернатив BigQuery як в домені з відкритим вихідним кодом, так і у вигляді хмарних сервісів, є неможливим відтворити масштаб і продуктивність BigQuery. В першу чергу тому, що Google відмінно об'єднанує інфраструктуру з програмним забезпеченням. Рішення з відкритим вихідним кодом, такі як Apache Drill і Presto, вимагають масштабного проектування інфраструктури та постійних операційних витрат, щоб відповідати продуктивності BigQuery.

Висновок. BigQuery призначений для створення запитів структурованих і напівструктурованих даних з використанням стандартної мови SQL. Сервіс оптимізований до використання і забезпечує надзвичайно високу рентабельність. BigQuery – це повністю кероване хмарне сховище, яке не потребує додаткових витрат на експлуатацію. Підходить для інтерактивних запитів і сценаріїв використання OLAP / BI. Технології хмарної інфраструктури Google є ключовими відмінностями сервісу BigQuery в порівнянні з іншими аналогами.

Список використаної літератури:

1. *Глибоке занурення в архітектуру Google BigQuery – [Електронний ресурс]. Режим доступу: https://panoply.io/data-warehouse-guide/bigquery-architecture/*
2. *Лакшманан В., Тайджані Д. – «Google BigQuery. Все про сховищах даних, аналітику та машинне навчання» - O'Reilly, 2021. - 496 с.*

УДК 004.04

Шевчук В.Ю., *студент 1 курсу спеціальності 122 «Комп'ютерні науки»*
Бабаков Р.М., *к.т.н., доцент, доцент кафедри комп'ютерних наук та інформаційних технологій*

DATA QUALITY ЯК НЕВІД'ЄМНА ЧАСТИНА АНАЛІЗУ ДАНИХ В ПРИКЛАДНИХ ПРОЄКТАХ BIG DATA

Донецький національний університет імені Василя Стуса, м. Вінниця

В основі прикладних проєктів Big Data лежить аналіз і обробка великих об'ємів даних. Проте, перед тим як обробляти певні дані потрібно перевірити чи взагалі дані відповідають певним критеріям, саме цим і займаються спеціалісти з Data Quality.

Data Quality стосується стану якісної чи кількісної інформації. Існує багато визначень Data Quality, але дані, як правило, вважаються високоякісними, якщо вони "придатні для [їх] передбачуваного використання в операціях, прийнятті рішень та плануванні". [1] [2] Більше того, дані вважаються високоякісними, якщо вони правильно відображають реальну конструкцію, на яку вони посилаються. Крім того, крім цих визначень, із збільшенням кількості джерел