Проєктування високонавантаженних систем. Лабораторна 5, звіт

Михайло Голуб

15 грудня 2025 р.

1 Завдання лабораторної роботи

1.1 Частина 1. Робота зі структурами даних у Cassandra

1. Ознайомитись з особливістю моделювання даних у Cassandra;
2. Створити `keyspace` з найпростішою стратегією реплікації;
3. Створити дві таблиці: `items` та `orders`;
4. Таблиця `items` містить різноманітні товари (тобто у яких різний набір властивостей). Для набору властивостей товару виберіть базові характеристики однакові для всіх товарів (`name, category, price, producer,` ...), а для властивостей які відрізняються використовуйте тип map (створивши індекс для можливості пошуку по її вмісту) Необхідно, щоб пошук швидко працював для категорії (`category`) товарів. Ця вимога має бути врахована при створенні ключа для таблиці (тобто, `category` має бути `partition key`);
5. Наповнити таблицю тестовими данними;
6. Написати запит, який показує структуру створеної таблиці (команда `DESCRIBE`);
7. Написати запит, який виводить усі товари в певній категорії відсортовані за ціною;
8. Напишіть запити, які вибирають товари за різними критеріями в межах певної категорії (тут де треба замість індексу використайте Matirialized view): назва, ціна (в проміжку), ціна та виробник
9. Створіть таблицю orders в якій міститься ім'я замовника і інформація про замовлення: перелік id-товарів у замовленні, вартість замовлення, дата замовлення, ... Для кожного замовника повинна бути можливість швидко шукати його замовлення і виконувати по них запити. Ця вимога має бути врахована при створенні ключа для таблиці (аналогічно як для items);

10. Напишіть запит, який показує структуру створеної таблиці (команда DESCRIBE);
11. Для замовника виведіть всі його замовлення відсортовані за часом коли вони були зроблені;
12. Для кожного замовників підрахуйте загальну суму на яку були зроблені усі його замовлення;
13. Для кожного замовлення виведіть час коли його ціна були занесена в базу (SELECT WRITETIME);

!!! У запитах заборонено використовувати ALLOW FILTERING !!!

1.2 Частина 2. Налаштування реплікації у Cassandra

1. Сконфігурувати кластер з 3-х нод;
2. Перевірити правильність конфігурації за допомогою `nodetool status`;
3. Використовуючи `cqlsh`, створити три `keyspace` з `replication factor` 1, 2, 3 з `SimpleStrategy`;
4. В кожному з кейспейсів створити прості таблиці;
5. Спробувати писати і читати в ці таблиці підключаючись до різних нод через `cqlsh`;
6. Вставте дані в створені таблиці і подивіться на їх розподіл по вузлах кластера окремо; для кожного з кейспейсів (команда `nodetool status`) – має бути видно відсоток даних який зберігається на ноді;
7. Для якогось запису з кожного з кейспейсу виведіть ноди на яких зберігаються дані – має бути видно ip-адреси вузлів на яких зберігається даний рядок;
8. Відключити одну з нод. Для кожного з кейспейсів перевірити з якими рівнями consistency можемо читати та писати: для `replication factor` 1 – `CONSISTENCY ONE`, для 2 – `CONSISTENCY ONE/TWO`, для 3 – `CONSISTENCY ONE/TWO/THREE`;
9. Зробити так щоб три ноди працювали, але не бачили одна одну по мережі (заблокувати чи відключити зв'язок між ними);
10. Для кейспейсу з replication factor 3 задати рівень consistency рівним 1. Виконати по черзі запис значення з однаковим primary key, але різними іншими значенням окремо на кожну з нод (тобто створіть конфлікт);
11. Відновити зв'язок між нодами, перевірити що вони знову об'єднались у кластер. Визначити яким чином був вирішений конфлікт даних та яке значення було прийнято кластером та за яким принципом.

1.3 Частина 3. Аналіз продуктивності та перевірка цілісності

Аналогічно попереднім завданням, необхідно, для кластеру налаштованому у попередній частині, створити таблицю з каунтером лайків. Далі з 10 окремих клієнтів одночасно запустити інкерементацію каунтеру лайків по 10_000 на кожного клієнта з різними опціями взаємодії з Cassandra. Таблиця має бути створена у Keyspace з replication factor 3. Для створення каунтеру використовуйте спеціальний тип колонки – counter (цей тип буде підтримувати операції increment/decrement in-place):

- Вказавши у параметрах запиту `Consistency Level One` (це буде означати, що запис відбувається синхронно тільки на одну ноду), запустіть 10 клієнтів з інкрементом по 10_000 на кожному з них. Виміряйте час виконання та перевірте чи кінцеве значення буде дорівнювати очікуваному – 100K
- Вказавши у параметрах запиту `Consistency Level QUORUM` (це буде означати, що запис відбувається синхронно на більшість нод), запустіть 10 клієнтів з інкрементом по 10_000 на кожному з них. Виміряйте час виконання та перевірте чи кінцеве значення буде дорівнювати очікуваному – 100K

1.4 Вимоги до оформлення протоколу

Завдання здається особисто без протоколу, або надсилається протокол який має містити:

- команди та результати їх виконання у вигляді скріншотів
- аналіз отриманих результатів

2 Хід роботи

2.1 Кластер cassandra

Кластер запускається на Docker з наступним docker-compose:

Лістинг 1: docker-compose.yml

```
services:
  cassandra1:
    image: cassandra:4.1
    container_name: cassandra1
    hostname: cassandra1
    networks:
      - cassandra-net
    ports:
```

```
      - "9042:9042"
    environment:
      CASSANDRA_CLUSTER_NAME: "TestCluster"
      CASSANDRA_SEEDS: "cassandra1"
      CASSANDRA_DC: "dc1"
      CASSANDRA_RACK: "rack1"
      CASSANDRA_ENDPOINT_SNITCH:
  GossipingPropertyFileSnitch
      CASSANDRA_NUM_TOKENS: 256
    volumes:
      - cassandra1-data:/var/lib/cassandra
    restart: unless-stopped

  cassandra2:
    image: cassandra:4.1
    container_name: cassandra2
    hostname: cassandra2
    networks:
      - cassandra-net
    depends_on:
      - cassandra1
    environment:
      CASSANDRA_CLUSTER_NAME: "TestCluster"
      CASSANDRA_SEEDS: "cassandra1"
      CASSANDRA_DC: "dc1"
      CASSANDRA_RACK: "rack1"
      CASSANDRA_ENDPOINT_SNITCH:
  GossipingPropertyFileSnitch
      CASSANDRA_NUM_TOKENS: 256
    volumes:
      - cassandra2-data:/var/lib/cassandra
    restart: unless-stopped

  cassandra3:
    image: cassandra:4.1
    container_name: cassandra3
    hostname: cassandra3
    networks:
      - cassandra-net
    depends_on:
      - cassandra1
    environment:
      CASSANDRA_CLUSTER_NAME: "TestCluster"
      CASSANDRA_SEEDS: "cassandra1"
      CASSANDRA_DC: "dc1"
      CASSANDRA_RACK: "rack1"
```

```
      CASSANDRA_ENDPOINT_SNITCH:
   GossipingPropertyFileSnitch
      CASSANDRA_NUM_TOKENS: 256
    volumes:
      - cassandra3-data:/var/lib/cassandra
    restart: unless-stopped

networks:
  cassandra-net:
    driver: bridge

volumes:
  cassandra1-data:
  cassandra2-data:
  cassandra3-data:
```

2.2 Частина 1. Робота зі структурами даних у Cassandra

Для повторюванності виконання запитів до Cassandra, запити виконуються pyton скриптом. Також, це дозволяє створити генератор даних для таблиць. Окрім вказаного в завданні функціоналу, скрипт видаляє keyspace, якщо він вже існує.

Лістинг 2: part_1.py

```python
from cassandra.cluster import Cluster
from cassandra.query import SimpleStatement

KEYSPACE = "lab_keyspace"

def main():
    cluster = Cluster(["127.0.0.1"])
    session = cluster.connect()

    session.execute(f"DROP KEYSPACE IF EXISTS {KEYSPACE};")
    print(f"Keyspace '{KEYSPACE}' dropped")

    session.execute(f"""
        CREATE KEYSPACE {KEYSPACE}
        WITH replication = {{
            'class': 'SimpleStrategy',
            'replication_factor': 3
        }};
    """)
    print(f"Keyspace '{KEYSPACE}' created")

    session.set_keyspace(KEYSPACE)

    session.execute("""
        CREATE TABLE items (
            category text,
            item_id uuid,
```

```
            name text,
            price decimal,
            producer text,
            attributes map<text, text>,
            PRIMARY KEY ((category), item_id)
        );
    """)
    print("Table 'items' created")

    session.execute("""
        CREATE INDEX items_attributes_idx
        ON items (ENTRIES(attributes));
    """)
    print("Index on attributes created")

    cluster.shutdown()


if __name__ == "__main__":
    main()
    input()
```

Результати тестування:
Час роботи: 59.51 секунд
Пропускна здатність: 1680.42 запитів/с
Кінцеве значення лічильника: 100к

3 Результати

Варіант	Час роботи	Пропускна здатність	Значення лічильника
1	26.37	3792.65	100к
majority	76.2	1312.37	100к
1 та падіння	23.05	4338.28	100к
majority та падіння	59.51	1680.42	100k