УДК 004

ВИКОРИСТАННЯ NOSQL БАЗИ ДАНИХ REDIS У СУЧАСНИХ ДОДАТКАХ

Крупіна К.А., Сватко В.В.

Національний транспортний університет, Київ

У сучасному світі все більше і більше з’являється потреба у збережені та моніторингу даних. Наприклад, для успішної практичної діяльності людини у сфері бізнесу або виробництва є залежність від ефективності організації у рамках обміну інформації. Чим більше виникало такої інформації, тим більше з’являлося потреби щодо її збережені у певному місці, яке буде захищене від третіх осіб за потребою. Саме для цього були створені такі програмні продукти як бази даних.

На фізичному рівні БД являє собою файл певного формату. Але, теоретично, зберігати дані можна і в звичайному файлі формату TXT якщо є можливість щоб їх прочитати та використовувати.

Функціонал забезпечення незалежності та цілісності даних реалізує система управління базами даних (СУБД). Кожна СУБД має певний інтерфейс для взаємодії з базами даних, за допомогою якого надається доступ до модифікації чи інших операцій з інформацією яка там знаходиться. Завдяки цьому СУБД відіграє важливу роль для взаємодії користувача системи та даних.

На логічному рівні кожна СУБД має свої відмінності, але ці відмінності не дуже значні. Вони мають вплив лише при тонкому налаштуванню під конкретну задачу та спосіб взаємодії з базою даних.

Фізичний рівень – це фактично і є база даних. На даному рівні розглядається структура файлів, а також їх взаємодія між собою.

Для створення сучасних додатків все більше та більше виникає потреба у невеликому сховищу даних, яке буде зберігати інформацію, яка буде швидко доступна. Саме для цієї потреби були створені NoSQL бази даних. Ця технологія була створена для спрощення стеження для певних моделей даних і мають гнучкі схеми, що

дозволяють розробляти сучасні та ефективні додатки. Типовий представник NoSQL бази даних – Redis.

Redis – доволі популярний інструмент, який підтримує велику кількість різноманітних методів та типів даних, які найчастіше використовуються у сучасних додатках. Типовий спосіб використання цієї БД – кешування для зберігання даних про користувача: сесії, кукі, JSON токен для авторизації та багато іншого. Доступ до них можливий у будь-який момент часу.

Для демонстрації кешування використаємо Docker-образ Redis для PHP.

```
redis:
    image: redis:latest
    ports:
        - 6379:6379
```

Рисунок 1 – Образ Redis у файлі docker-compose

Піднімемо контейнер Docker та підключимось до Redis у контролері. У нас є два варіанти підключення connect та pconnect. Connect – щоразу створює нове підключення, а pconnect – підключається к вже наявному каналу або створює нове якщо його немає.

```
$redis = new Redis();
$redis->pconnect( host: 'redis', port: 6379);
```

Рисунок 2 – Підключення до Redis

Кешування відбувається максимально просто. Є ключ, а є значення яке ми кладемо під певним ключем. Ключем виступає певне строкове значення.

```
$redis->set( key: 'rate:GBP', value: 92);
$redis->mset([
    'rate:EUR' => 80,
    'rate:JPY' => 60
]);
$redis->get('rate:GBP');
```

Рисунок 3 – Встановлення та отримання значення

Для встановлення значення є методи set та mset. Set приймає два аргументи: ключ та значення. Mset приймає одне значення – список(масив). За допомогою цього методу можна добавити велику кількість даних щоб кожного разу не визивати set. Також існує метод setnx, який встановлює значення за ключем якщо він існує. За допомогою методу get можна отримати значення по ключу, а за допомогою del – видалити.

```
$redis->persist( key: 'rate:GBP');
$redis->expire( key: 'rate:USD', ttl: 3600);
```

Рисунок 4 – Встановлення часу життя значення

Можна встановити час життя значення за допомогою методів persist та expire. Persist – значення буде зберігатись нескінченно, а за допомогою метода expire та встановлення значення у секундах.

Це лише невелика частина можливостей. Слід зазначити, що зберігання відбувається у операційній системі носія, тому великі об’єми даних не слід зберігати за допомогою Redis. Існує функціонал, який в залежності від сценарію може виконуватись перекидання даних на інший носій.

Redis включає у себе:

1. Кешування.
2. Використання як типової БД.
3. Транзакції (послідовне виконання раніше записаних команд).

4. Pub/Sub (користувач підписується на канал, де публікатор відправляє повідомлення всім підписникам).

Redis орієнтований на досягнення максимальної продуктивності на атомарних операціях. Заявлено про 100 тис. запитів за секунду на базових серверах Linux.

Популярність та простота Redis підтверджена тим, що його використовують декілька великих та провідних компаній: Twitter, GitHub, Uber, Pinterest, Snapchat, StackOverflow та багато інших.

Література:

1. Інформатика: Комп'ютерна техніка. Комп'ютерні технології. Посіб./ За ред. О.І. Пушкаря – К.: Видав-ничий центр "Академія", 2001. – 696 с.
2. Нирва Мориссо-Леруа, Мартин К. Соломон, Джеральд Момплезир. 2001. Oracle9i Программирование на языке SQLJ. Издательство «Лори», Москва.
3. Визначення бази даних (БД) і банку даних (БНД). Склад і структура банку даних. Призначення основних компонентів банку даних [електронний ресурс]. — Режим доступу: http://e-educ.ru/bd1.html.
4. NoSQL технології на прикладі MongoDB [електронний ресурс]. — Режим доступу: https://dou.ua/forums/topic/33453/.
5. Розбираємось з Redis [електронний ресурс]. — Режим доступу: https://habr.com/ru/company/wunderfund/blog/685894/.
6. Redis Documentation [електронний ресурс]. — Режим доступу: https://docs.redis.com/latest/rs/.