

Реалізація каунтера з використанням PostgreSQL

Мета

Необхідно декількома способами реалізувати оновлення значення каунтера в СКБД PostgreSQL та оцінити час кожного із варіантів.

Структура таблиці

user_counter

USER_ID	Counter	Version
1	1	0
...	...	

Нехай маємо таблицю **user_counter** наступної структури

- **USER_ID** - ключ з ідентифікатором користувача
- **Counter** - Integer - значення каунтера
- **Version** - Integer - допоміжне поле, буде використовуватись в одному з варіантів реалізації

Варіанти реалізації

1. Lost-update (реалізація що втрачатиме значення)

З 10 потоків чи окремих клієнтів одночасно запустити

```
for (i in 1..10_000) {  
    counter = cursor.execute("SELECT counter FROM user_counter  
WHERE user_id = 1").fetchone()  
    counter = counter + 1  
    cursor.execute("update user_counter set counter = %s where  
user_id = %s", (counter, 1))  
    conn.commit()  
}
```

Необхідно заміряти час виконання

2. In-place update

З 10 потоків чи окремих клієнтів одночасно запустити

```
for (i in 1..10_000) {  
    cursor.execute("update user_counter set counter = counter + 1  
where user_id = %s", (1))  
    conn.commit()  
}
```

Необхідно заміряти час виконання

3. Row-level locking

Row level locking with *SELECT ... FOR UPDATE*:

<https://www.postgresql.org/docs/current/explicit-locking.html#LOCKING-ROWS>

<https://www.postgresql.org/docs/current/applevel-consistency.html#NON-SERIALIZABLE-CONSISTENCY>

З 10 потоків чи окремих клієнтів одночасно запустити

```
for (i in 1..10_000) {
    counter = cursor.execute("SELECT counter FROM user_counter
WHERE user_id = 1 FOR UPDATE").fetchone()
    counter = counter + 1
    cursor.execute(("update user_counter set counter = %s where
user_id = %s", (counter, 1))
    conn.commit()
}
```

Необхідно створювати окреме підключення та курсор для кожного потоку, щоб *"select ... for update"* мав ефект.

Необхідно заміряти час виконання

4. Optimistic concurrency control

З 10 потоків чи окремих клієнтів одночасно запустити

```
for (i in 1..10_000) {
    while (True) {
        (counter, version) = cursor.execute("SELECT counter,
version FROM user_counter WHERE user_id = 1").fetchone()
        counter = counter + 1
        cursor.execute(("update user_counter set counter = %s,
version = %s where user_id = %s" and version = %s, (counter, version
+ 1, 1, version))
        conn.commit()
        count = cursor.rowcount
        if (count > 0) break
    }
}
```

Необхідно заміряти час виконання

Вимоги до реалізації

- Мова реалізації будь-яка
- Не використовувати ORM-фреймворки (Hibernate, SQLAlchemy, ...)

- Не забувати про необхідність окремої транзакції на кожен запис

Звіт

У результаті виконання роботи має бути надано код скрипта та отримані результати тестування для різних сценаріїв.

В усіх варіантах, окрім першого, кінцеве значення каунтера має бути 100K

Посилання

- Взаємодія з БД для Python <https://www.psycopg.org/docs/usage.html>