



*Міністерство освіти і науки України Національний технічний університет України
«Київський політехнічний інститут ім. І. Сікорського» Фізико-технічний інститут*

ПРОЕКТУВАННЯ ВИСОКОНАВАНТАЖЕНИХ СИСТЕМ

ЛАБОРАТОРНА РОБОТА №2

РЕАЛІЗАЦІЯ КАУНТЕРА З ВИКОРИСТАННЯМ POSTGRESQL

Виконав:

Студент групи ФБ-41мп

Заріцький О. В.

Київ 2024 р.

Мета роботи:

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

1. Lost-update (реалізація що втрачатиме значення)

Виконання:

```
def part1(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    for _ in range(10000):
        cur.execute(f"SELECT counter FROM user_counter WHERE user_id = {user_id};")
        counter = cur.fetchone()[0] + 1
        cur.execute(f"UPDATE user_counter SET counter = {counter} WHERE user_id =
{user_id};")
        conn.commit()
    cur.close()
    conn.close()
```

Результат:

```
PS D:\KPI\mag\hisp\lab2> & C:/Users/
/// Part #1 ///
Time: 201.6s
Final counter: 10191
```

2. In-place update

Виконання:

```
def part2(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    for _ in range(10000):
        cur.execute(f"UPDATE user_counter SET counter = counter + 1 WHERE user_id =
{user_id};")
        conn.commit()
    cur.close()
    conn.close()
```

Результат:

```
/// Part #2 ///
Time: 340.2s
Final counter: 100000
```

3. Row-level locking

Виконання:

```
def part3(user_id):
    for _ in range(10000):
        conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
        cur = conn.cursor()
        cur.execute(f"SELECT counter FROM user_counter WHERE user_id = {user_id} FOR
UPDATE;")
        counter = cur.fetchone()[0] + 1
        cur.execute(f"UPDATE user_counter SET counter = {counter} WHERE user_id =
{user_id};")
        conn.commit()
        cur.close()
        conn.close()
```

Результат:

```
/// Part #3 ///
```

| | |
|----------------|--------|
| Time: | 516.4s |
| Final counter: | 100000 |

4. Optimistic concurrency control

Виконання:

```
def part4(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    for _ in range(10000):
        while True:
            cur.execute(f"SELECT counter, version FROM user_counter WHERE user_id =
{user_id};")
            counter, version = cur.fetchone()
            counter += 1
            cur.execute(f"UPDATE user_counter SET counter = {counter}, version = {version
+ 1} WHERE user_id = {user_id} AND version = {version};")
            conn.commit()
            if cur.rowcount > 0: break
        cur.close()
    conn.close()
```

Результат:

```
/// Part #4 ///
```

| | |
|----------------|---------|
| Time: | 3236.2s |
| Final counter: | 100000 |

Повний код виконання

compose.yml

```
services:
  postgres:
    container_name: oz_lab2
    environment:
      - POSTGRES_DB=oz_lab2
      - POSTGRES_USER=zaritsky_fb41mp
      - POSTGRES_PASSWORD=bober0_0
    ports:
      - '5432:5432'
    image: 'postgres:latest'
```

lab2.py

```
import time
import psycopg2
import threading

def init_table():
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    cur.execute('''CREATE TABLE IF NOT EXISTS user_counter (
                    user_id serial PRIMARY KEY,
                    counter integer,
                    version integer);
                ''')
    for i in range(1, 5):
        cur.execute(f"INSERT INTO user_counter (user_id, counter, version) VALUES ({i},
0, 0) ON CONFLICT (user_id) DO NOTHING;")
        conn.commit()
    cur.close()
    conn.close()

def reset_counter(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    cur.execute(f"UPDATE user_counter SET counter = 0, version = 0 WHERE user_id =
{user_id};")
    conn.commit()
    cur.close()
    conn.close()

def part1(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    for _ in range(10000):
        cur.execute(f"SELECT counter FROM user_counter WHERE user_id = {user_id};")
        counter = cur.fetchone()[0] + 1
        cur.execute(f"UPDATE user_counter SET counter = {counter} WHERE user_id =
{user_id};")
        conn.commit()
    cur.close()
    conn.close()

def part2(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    for _ in range(10000):
        cur.execute(f"UPDATE user_counter SET counter = counter + 1 WHERE user_id =
{user_id};")
        conn.commit()
```

```

cur.close()
conn.close()

def part3(user_id):
    for _ in range(10000):
        conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
        cur = conn.cursor()
        cur.execute(f"SELECT counter FROM user_counter WHERE user_id = {user_id} FOR
UPDATE;")
        counter = cur.fetchone()[0] + 1
        cur.execute(f"UPDATE user_counter SET counter = {counter} WHERE user_id =
{user_id};")
        conn.commit()
        cur.close()
        conn.close()

def part4(user_id):
    conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
    cur = conn.cursor()
    for _ in range(10000):
        while True:
            cur.execute(f"SELECT counter, version FROM user_counter WHERE user_id =
{user_id};")
            counter, version = cur.fetchone()
            counter += 1
            cur.execute(f"UPDATE user_counter SET counter = {counter}, version = {version
+ 1} WHERE user_id = {user_id} AND version = {version};")
            conn.commit()
            if cur.rowcount > 0: break
    cur.close()
    conn.close()

def runAllTasks():
    parts = [part1, part2, part3, part4]
    init_table()

    for idx, part in enumerate(parts, start=1):
        print(f"\n/// Part #{idx} ///")
        reset_counter(idx)
        start = time.time()
        threads = [threading.Thread(target=part, args=(idx,)) for _ in range(10)]
        for t in threads: t.start()
        for t in threads: t.join()
        elapsed_time = round(time.time() - start, 1)
        print(f"Time: {elapsed_time}s")

        conn = psycopg2.connect('host=localhost dbname=oz_lab2 user=zaritsky_fb41mp
password=bober0_0')
        cur = conn.cursor()
        cur.execute(f"SELECT counter FROM user_counter WHERE user_id = {idx};")
        counter = cur.fetchone()[0]
        print(f"Final counter: {counter}")
        cur.close()
        conn.close()

if __name__ == "__main__":
    runAllTasks()

```