



Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут ім. Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

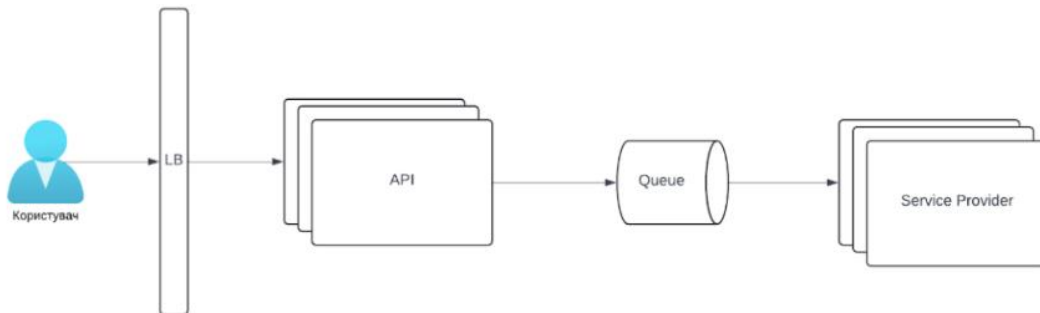
ЛАБОРАТОРНА РОБОТА №2
З ДИСЦИПЛІНИ “ Проектування розподілених систем”

Виконав:
студент групи ІО-31мн
Чередник В.Ю.
Перевірив:
Обозний Д. М.

м. Київ – 2024 р.

Лабораторна робота № 2

- Реалізувати асинхронну комунікацію між Постачальником Сервісу і Споживачем Сервісу за допомогою Брокера Повідомлень
- Постачальник Сервісу має підраховувати час обчислення і логувати його для подальшого аналізу
- Споживач Сервісу має підраховувати час виконання запиту і логувати його для подальшого аналізу
- Реалізувати горизонтальне масштабування засобами Брокера Повідомлень
- Реалізувати чергу с пріоритетами
- Реалізувати request-reply паттерн в асинхронній комунікації
- Порівняти результати синхронної і асинхронної комунікації



Опис програмної реалізації

1. Користувач надсилає HTTP-запит до Споживача Сервісу (consumer_service/app.py) через маршрут /calculate/async. Споживач приймає запит і обробляє його. Потім формує повідомлення (завдання), яке відправляється до Брокера.
2. Споживач використовує RabbitMQ як Брокер повідомлень. Завдання, яке містить числа для обчислення та пріоритет, передається до черги RabbitMQ (calculation_queue).

3. RabbitMQ черга передає повідомлення Постачальнику Сервісу, який є споживачем цієї черги. Постачальник обчислює результат додавання двох чисел.
4. Після обчислення результату Постачальник надсилає відповідь назад у RabbitMQ через механізм `reply_to` (callback черга), вказаний Споживачем.
5. RabbitMQ повертає відповідь із результатом до Споживача Сервісу, використовуючи callback чергу.
6. Споживач отримує результат із RabbitMQ, обчислює час виконання, формує відповідь і відправляє її Користувачу.

Механізм асинхронної обробки реалізований через RabbitMQ з використанням `reply_to` для request-reply патерну. Пріоритети у черзі обробляються за допомогою RabbitMQ, що дозволяє Постачальнику отримувати завдання у відповідному порядку.

Результати роботи

Приклади запитів

```
C:\Users\vital\PycharmProjects\DSP_Lab2>curl -X POST http://localhost:8000/calculate/async -H "Content-Type: application/json" -d '{"num1": 35, "num2": 71, "priority": 1}'
{"calculation_time":3.170967102050781e-05,"request_time":0.019062280654907227,"result":106}

C:\Users\vital\PycharmProjects\DSP_Lab2>curl -X POST http://localhost:8000/calculate/async -H "Content-Type: application/json" -d '{"num1": 35, "num2": 71, "priority": 10}'
{"calculation_time":2.5987625122070312e-05,"request_time":0.004058837890625,"result":106}

C:\Users\vital\PycharmProjects\DSP_Lab2>curl -X POST http://localhost:8000/calculate/async -H "Content-Type: application/json" -d '{"num1": 35, "num2": 71, "priority": 5}'
{"calculation_time":6.60419464113281e-05,"request_time":0.005578517913818359,"result":106}
```

Логи

```
rabbitmq-1 | 2024-12-30 18:51:39.856577+00:00 [notice] <0.86.0> alarm_handler: {set,{system_memory_high_watermark,[]}}
service2_1-1 | 2024-12-30 18:51:47,251 - service2_1 - INFO - [service2_1] Calculation completed in 0.0000 seconds
service1_1-1 | 2024-12-30 18:51:47,254 - service1_1 - INFO - [service1_1] Async request completed in 0.0191 seconds
service1_1-1 | 2024-12-30 18:51:47,255 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:51:47] "POST /calculate/async HTTP/1.0" 200 -
nginx-1 | 172.31.0.1 - - [30/Dec/2024:18:51:47 +0000] "POST /calculate/async HTTP/1.1" 200 92 "-" "curl/8.9.1"
service2_3-1 | 2024-12-30 18:51:49,274 - service2_3 - INFO - [service2_3] Calculation completed in 0.0000 seconds
service1_2-1 | 2024-12-30 18:51:49,275 - service1_2 - INFO - [service1_2] Async request completed in 0.0041 seconds
service1_2-1 | 2024-12-30 18:51:49,277 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:51:49] "POST /calculate/async HTTP/1.0" 200 -
nginx-1 | 172.31.0.1 - - [30/Dec/2024:18:51:49 +0000] "POST /calculate/async HTTP/1.1" 200 90 "-" "curl/8.9.1"
service2_2-1 | 2024-12-30 18:52:12,064 - service2_2 - INFO - [service2_2] Calculation completed in 0.0001 seconds
service1_1-1 | 2024-12-30 18:52:12,067 - service1_1 - INFO - [service1_1] Async request completed in 0.0056 seconds
service1_1-1 | 2024-12-30 18:52:12,067 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:52:12] "POST /calculate/async HTTP/1.0" 200 -
nginx-1 | 172.31.0.1 - - [30/Dec/2024:18:52:12 +0000] "POST /calculate/async HTTP/1.1" 200 92 "-" "curl/8.9.1"
```

Створимо декілька паралельних запитів

```
#!/usr/bin/perl -w
my $url = "http://localhost:8000/";
my $count = 10;
my $delay = 0.1;

for ($i = 0; $i < $count; $i++) {
    my $curl_cmd = "curl -X POST -H 'Content-Type: application/json' -d '{\"num1\": $i, \"num2\": \"${i+1}\", \"priority\": \"$i\"} $url/async & done";
    system($curl_cmd);
    sleep($delay);
}
```

Логи

```
2024-12-30 18:52:52,779 - service2_1 - INFO - [service2_1] Calculation completed in 0.0000 seconds
2024-12-30 18:52:52,781 - service1_1 - INFO - [service1_1] Async request completed in 0.0044 seconds
2024-12-30 18:52:52,782 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:52:52] "POST /calculate/async HTTP/1.0" 200 -
2024-12-30 18:52:52,815 - service2_3 - INFO - [service2_3] Calculation completed in 0.0000 seconds
2024-12-30 18:52:52,816 - service1_2 - INFO - [service1_2] Async request completed in 0.0032 seconds
2024-12-30 18:52:52,940 - service2_2 - INFO - [service2_2] Calculation completed in 0.0000 seconds
2024-12-30 18:52:52,940 - service2_1 - INFO - [service2_1] Calculation completed in 0.0000 seconds
2024-12-30 18:52:52,942 - service1_2 - INFO - [service1_2] Async request completed in 0.0041 seconds
2024-12-30 18:52:52,942 - service1_1 - INFO - [service1_1] Async request completed in 0.0047 seconds
2024-12-30 18:52:52,943 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:52:52] "POST /calculate/async HTTP/1.0" 200 -
```

Логи з першої лабораторної

```
2024-12-30 21:15:47,534 - INFO - 192.168.16.4 - - [30/Dec/2024 21:15:47] "POST /calculate HTTP/1.0" 401 -
2024-12-30 21:15:47,537 - INFO - 192.168.16.1 - - - to: 192.168.16.2:5000: POST /calculate HTTP/1.1 upstream_response_time 0.011 msec 1735593347.537 request_time 0.010
2024-12-30 21:20:14,562 - INFO - 192.168.16.4 - - [30/Dec/2024 21:20:14] "POST /token HTTP/1.0" 200 -
2024-12-30 21:20:14,563 - INFO - 192.168.16.1 - - - to: 192.168.16.3:5000: POST /token HTTP/1.1 upstream_response_time 0.008 msec 1735593614.562 request_time 0.008
2024-12-30 21:24:11,128 - INFO - Computation duration: 0.0053 seconds
2024-12-30 21:24:11,129 - INFO - Computed result: 95
2024-12-30 21:24:11,136 - INFO - 192.168.0.4 - - [30/Dec/2024 21:24:11] "POST /compute HTTP/1.0" 200 -
2024-12-30 21:24:11,143 - INFO - Request duration: 0.0052 seconds
2024-12-30 21:24:11,145 - INFO - 192.168.16.4 - - [30/Dec/2024 21:24:11] "POST /calculate HTTP/1.0" 200 -
2024-12-30 21:26:05,550 - INFO - Computation duration: 0.0082 seconds
2024-12-30 21:26:05,552 - INFO - Computed result: 110
2024-12-30 21:26:05,562 - INFO - 192.168.0.4 - - [30/Dec/2024 21:26:05] "POST /compute HTTP/1.0" 200 -
2024-12-30 21:26:05,563 - INFO - 192.168.16.3 - - - to: 192.168.0.3:5000: POST /compute HTTP/1.1 upstream_response_time 0.061 msec 1735593965.563 request_time 0.060
2024-12-30 21:26:05,567 - INFO - Request duration: 0.0944 seconds
2024-12-30 21:26:05,568 - INFO - 192.168.16.4 - - [30/Dec/2024 21:26:05] "POST /calculate HTTP/1.0" 200 -
2024-12-30 21:26:05,568 - INFO - 192.168.16.3 - - - to: 192.168.16.3:5000: POST /calculate HTTP/1.1 upstream_response_time 0.009 msec 1735593965.568 request_time 0.009
2024-12-30 21:26:29,765 - INFO - Computation duration: 0.0003 seconds
2024-12-30 21:26:29,766 - INFO - Computed result: 19
2024-12-30 21:26:29,766 - INFO - 192.168.0.4 - - [30/Dec/2024 21:26:29] "POST /compute HTTP/1.0" 200 -
2024-12-30 21:26:29,769 - INFO - Request duration: 0.0086 seconds
2024-12-30 21:26:29,769 - INFO - 192.168.16.4 - - [30/Dec/2024 21:26:29] "POST /calculate HTTP/1.0" 200 -
2024-12-30 21:26:29,769 - INFO - 192.168.16.1 - - - to: 192.168.16.2:5000: POST /calculate HTTP/1.1 upstream_response_time 0.012 msec 1735593989.770 request_time 0.013
2024-12-30 21:26:51,633 - INFO - Computation duration: 0.0002 seconds
2024-12-30 21:26:51,633 - INFO - Computed result: 863
2024-12-30 21:26:51,634 - INFO - 192.168.0.4 - - [30/Dec/2024 21:26:51] "POST /compute HTTP/1.0" 200 -
2024-12-30 21:26:51,634 - INFO - 192.168.16.3 - - - to: 192.168.0.3:5000: POST /compute HTTP/1.1 upstream_response_time 0.004 msec 1735594011.634 request_time 0.004
2024-12-30 21:26:51,636 - INFO - Request duration: 0.0093 seconds
2024-12-30 21:26:51,637 - INFO - 192.168.16.4 - - [30/Dec/2024 21:26:51] "POST /calculate HTTP/1.0" 200 -
2024-12-30 21:26:51,637 - INFO - 192.168.16.1 - - - to: 192.168.16.3:5000: POST /calculate HTTP/1.1 upstream_response_time 0.014 msec 1735594011.638 request_time 0.014
2024-12-30 21:31:55,413 - INFO - Computation duration: 0.0004 seconds
2024-12-30 21:31:55,414 - INFO - Computed result: 114
2024-12-30 21:31:55,414 - INFO - 192.168.0.4 - - [30/Dec/2024 21:31:55] "POST /compute HTTP/1.0" 200 -
2024-12-30 21:31:55,415 - INFO - 192.168.16.2 - - - to: 192.168.0.2:5000: POST /compute HTTP/1.1 upstream_response_time 0.003 msec 1735594315.415 request_time 0.004
2024-12-30 21:31:55,416 - INFO - Request duration: 0.0075 seconds
2024-12-30 21:31:55,417 - INFO - 192.168.16.4 - - [30/Dec/2024 21:31:55] "POST /calculate HTTP/1.0" 200 -
2024-12-30 21:31:55,417 - INFO - 192.168.16.1 - - - to: 192.168.16.2:5000: POST /calculate HTTP/1.1 upstream_response_time 0.011 msec 1735594315.418 request_time 0.011
2024-12-30 21:32:14,017 - INFO - Computation duration: 0.0003 seconds
2024-12-30 21:32:14,017 - INFO - Computed result: 143
2024-12-30 21:32:14,018 - INFO - 192.168.0.4 - - [30/Dec/2024 21:32:14] "POST /compute HTTP/1.0" 200 -
2024-12-30 21:32:14,021 - INFO - Request duration: 0.0168 seconds
2024-12-30 21:32:14,021 - INFO - 192.168.16.1 - - - to: 192.168.16.3:5000: POST /calculate HTTP/1.1 upstream_response_time 0.021 msec 1735594334.023 request_time 0.021
2024-12-30 21:32:14,022 - INFO - 192.168.16.4 - - [30/Dec/2024 21:32:14] "POST /calculate HTTP/1.0" 200 -
```

Порівняння результатів

В асинхронній комунікації ми бачимо в логах паралельну обробку (різні споживачі одночасно отримують запити).

```

service2_2-1 | 2024-12-30 18:52:52,940 - service2_2 - INFO - [service2_2] Calculation completed in 0.0000 seconds
service2_1-1 | 2024-12-30 18:52:52,940 - service2_1 - INFO - [service2_1] Calculation completed in 0.0000 seconds
service1_2-1 | 2024-12-30 18:52:52,942 - service1_2 - INFO - [service1_2] Async request completed in 0.0041 seconds
service1_1-1 | 2024-12-30 18:52:52,942 - service1_1 - INFO - [service1_1] Async request completed in 0.0047 seconds
nginx-1 | 172.31.0.1 - - [30/Dec/2024:18:52:52 +0000] "POST /calculate/async HTTP/1.1" 200 91 "-" "curl/8.11.0"
service1_2-1 | 2024-12-30 18:52:52,943 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:52:52] "POST /calculate/async HTTP/1.0" 200 -
service1_1-1 | 2024-12-30 18:52:52,943 - werkzeug - INFO - 172.31.0.8 - - [30/Dec/2024 18:52:52] "POST /calculate/async HTTP/1.0" 200 -

```

Також видно, що запити обробляються майже одночасно. Різні створені при масштабуванні репліки працюють паралельно. Також порядок завершення запитів може відрізнятись від порядку їх надходження, що видно на скріншоті.

Для порівняння, при синхронному виконанні ми бачимо послідовне виконання запитів без паралелізму, і загальний час виконання, хоч і незначно, але більший. Якщо б ми мали справу з більш складними системами та більш важкими запитами, то це було б помітніше. При асинхронній моделі час обробки для кожного запиту може бути меншим через те, що Споживач не очікує завершення виконання. А Постачальник працює незалежно, обробляючи повідомлення в черзі.