НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ

імені ІГОРЯ СІКОРСЬКОГО”

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Лабораторна робота №2

з предмету «Проектування розподілених систем»

Виконав:

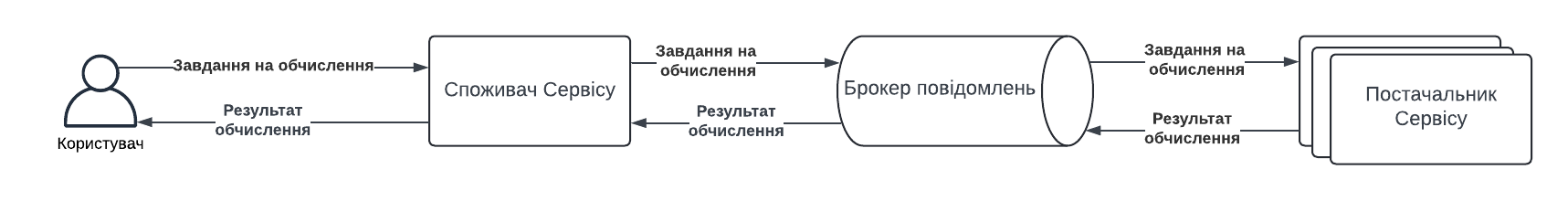
студент групи ІМ-31мн

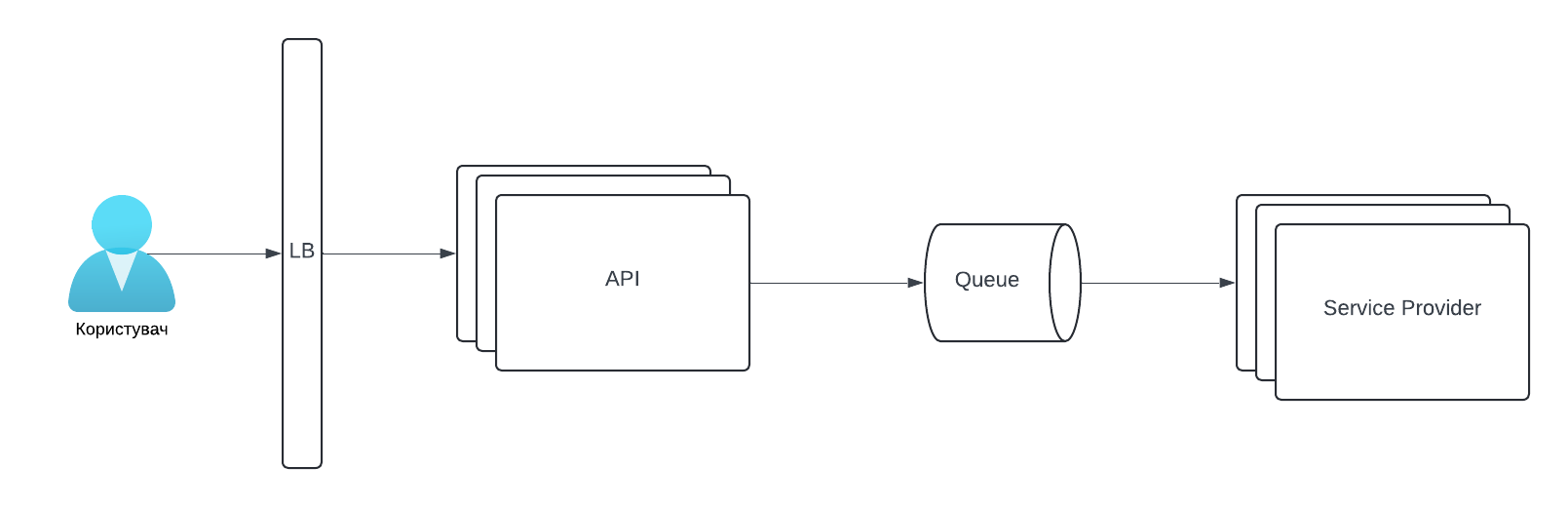
Рекечинський Дмитро

Київ 2024

Завдання

* 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
* Cпоживач Cервісу 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





Виконання завдання

Для виконання завдання було створено 4 типи сервісів (всього 6 інстансів):

* Споживач сервісу consumer-service
  + Інстанс consumer-service-1
  + Інстанс consumer-service-2
* Постачальник сервісу provider-service
  + Інстанс provider-service-1
  + Інстанс provider-service-2
* Load balancer для consumer-service (lb\_consumer)
* Брокер повідомлень RabbitMQ (rabbitmq)

RabbitMQ ідеально підходить в якості брокера повідомлень, тим паче, що у ньому реалізована концепція черги з пріоритетами.

Доволі тяжкою частиною було забезпечення того, що спочатку запуститься готовий для підключень сервіс RabbitMQ, а після нього — усі залежні сервіси (споживач сервісу та постачальник сервісу).

Втім, рішення знайшлось: це healthcheck. Суть така: вписуємо команду, яка перевіряє, чи готовий сервіс до роботи. Якщо ні, перевіримо ще раз через вказану кількість секунд. Якщо так, оповіщуємо про це всі залежні сервіси, і вони після цього запускаються.

Конфігурація healthcheck для RabbitMQ виглядає таким чином:

healthcheck:

test: rabbitmq-diagnostics check\_port\_connectivity

interval: 10s

timeout: 5s

retries: 10

start\_period: 5s

Для того, щоб враховувати статус healthcheck для залежних сервісів, слід змінити конфігурацію з такої форми:

depends\_on:

- rabbitmq

на таку:

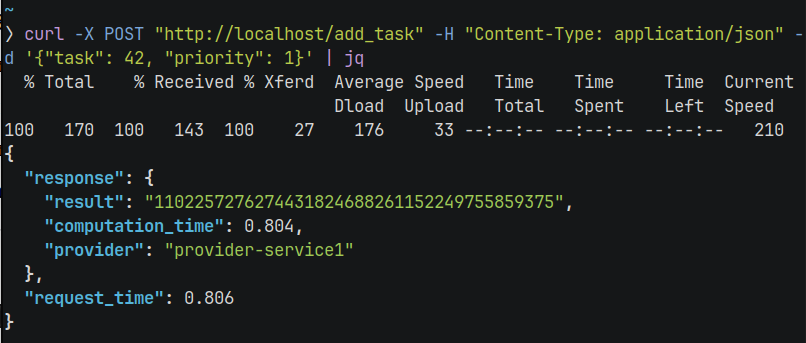
depends\_on:

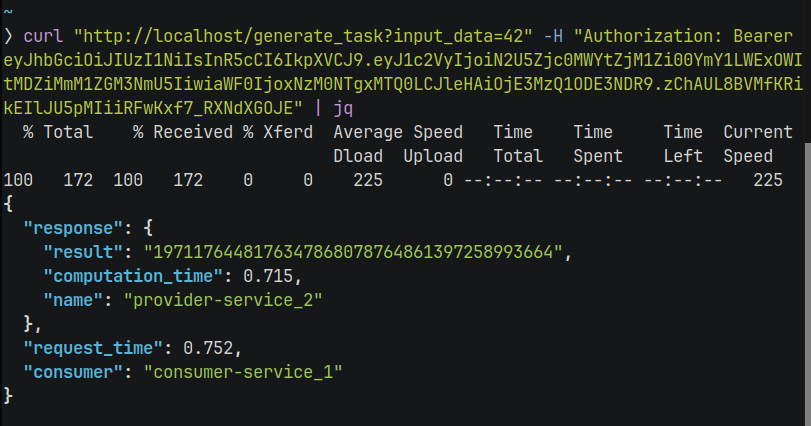
rabbitmq:

condition: service\_healthy

Якщо не вказувати condition явно, то його значення за замовчуванням є service\_started, що буквально означає «як тільки сервіс запустився». У випадку із RabbitMQ це значення не підходить, оскільки ми орієнтуємось саме на статус готовності до підключень.

Демонстрація результатів:

Рис. 1 — Виконання асинхронної комунікації за допомогою брокера повідомлень

Рис. 2 — Виконання синхронної комунікації за допомогою балансувальника

Ефективність можна легко визначити за різницею між часом обчислення та часом на запит. У синхронній комунікації різниця складає 37 мс, в той же час, як брокер повідомлень зайняв 2 мс, і це лише для виконання одного запиту. При використанні більших навантажень ця різниця стане більш суттєвою.

Повна версія коду проекту розміщена за веб-адресою: <https://github.com/rocket111185/distribution-systems/tree/release/lab2>