

УДК 681.1

Д.В. Мальцев, студент гр. ПІ-11мп, к.т.н., доц. Богомазов С.А.
КПІ ім. Ігоря Сікорського

СИСТЕМА ВІДДАЛЕНОГО ЗБОРУ СЕНСОРНИХ ДАНИХ НА ОСНОВІ АСИНХРОННИХ ПРОГРАМНИХ ТЕХНОЛОГІЙ

Анотація. У статті наведено аналіз особливостей організації систем віддаленого збору сенсорних даних на основі асинхронних програмних технологій із використанням мікрофреймворку Spring WebFlux, який забезпечує підтримку реактивного програмування для веб-додатків, написаних із використанням Spring-framework.

Ключові слова: Spring mvc, Spring webflux, Java

ПОСТАНОВКА ПРОБЛЕМИ

На сьогоднішній день системи віддаленого збору інформації відіграють важливу роль у нашому житті. Вони використовуються як у побуті, так і на великих підприємствах у якості систем технічного контролю і діагностики.

Важливу роль у таких системах відіграє здатність серверного програмного забезпечення до швидкої обробки і передачі отриманих даних.

За імперативного підходу до розробки програмного забезпечення, програма — це послідовність інструкцій, які виконуються послідовно в тому порядку, в якому вони були написані розробником. Програма очікує повного завершення попередніх дій перед тим, як почати виконання наступних кроків. Під час виконання операції, особливо якщо це операція зберігання в базу даних чи взаємодія з користувачем, потік, який викликав цю задачу, блокується і очікує [1]. Цей процес дуже ресурсомісткий, і тому виникає задача аналізу й дослідження систем збору сенсорних даних із використанням асинхронних програмних технологій.

ОСНОВНІ ПОЛОЖЕННЯ

Більшість мов програмування підтримують паралельне виконання, але створення і синхронізація паралельних потоків - досить складна задача, і тому для реалізації паралелізму було використано веб-фреймворк із реактивним стеком Spring WebFlux.

Spring WebFlux – це мікрофреймворк для мови програмування Java, який забезпечує підтримку повністю асинхронного і неблокуючого веб-стеку, що дозволяє обробляти більшу кількість одночасно отриманих запитів порівняно зі стандартним MVC [2].

Термін “реактивний” відноситься до моделей програмування, які побудовані, як реакції на зміни. Наприклад: мережеві компоненти, які реагують на завершення операцій введення-виведення або контролери інтерфейсу користувача, які реагують на натискання клавіші. У цьому сенсі неблокуючі потоки забезпечують реактивність, тому що замість того, щоб бути заблокованими й чекати на завершення поточної операції, фреймворк реагує на подію і передає задачу обробки інформації будь-якому вільному у той час потоку.

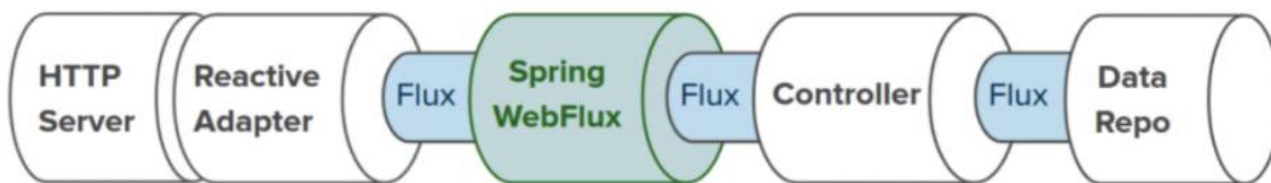


Рисунок 1. Архітектура реактивного програмного забезпечення з використанням Spring WebFlux

Будь-який додаток, із використанням даного фреймворку, складається з 5 основних частин:

1. HTTP Server;
2. Реактивний адаптер (WebFlux не сумісний із веб-сервером Netty, який за замовчуванням поставляється разом із фреймворком);
3. Spring WebFlux;
4. Контролер;
5. Репозиторій для роботи з базою даних.

У якості HTTP сервера за замовчуванням використовується веб-сервер Netty. Netty – це клієнт-серверне середовище неблокуючого введення-виведення для розробки мережевих додатків Java [3].

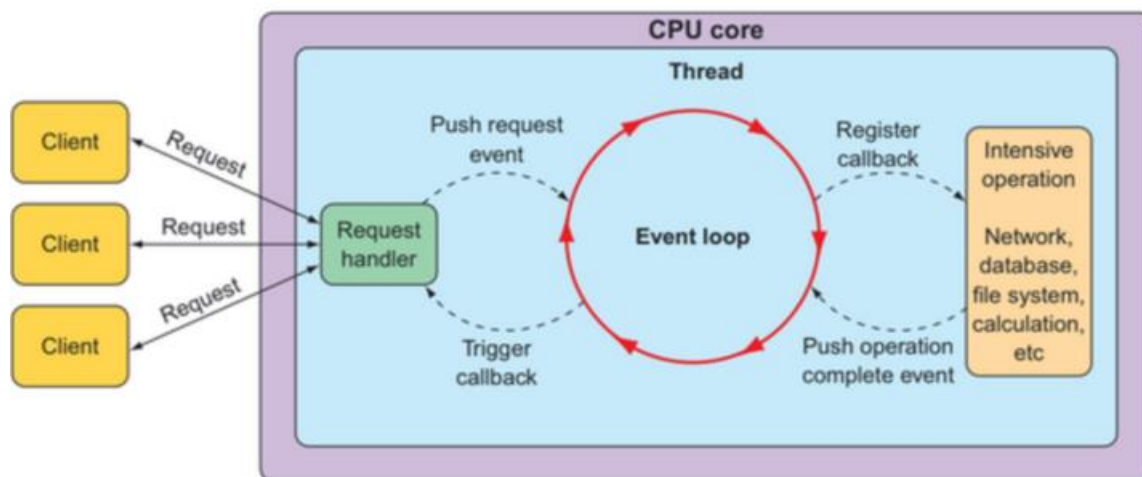


Рисунок 2. Принцип роботи асинхронних систем для виконання великої кількості операцій з обмеженою кількістю потоків

Для обробки великої кількості запитів на вході в Netty в нескінченному циклі знаходиться потік виконання. Цей цикл делегує виконання всіх операцій в пул асинхронних потоків. У той самий час відбувається реєстрація функції зворотного виклику (колбек, callback). Ця функція буде викликана після того, як потік, якому було передано інформацію на обробку, завершить свою роботу. Колбеки потрібні для передачі обробленої інформації назад до клієнта. Після делегування операції в пул потоків, головний потік продовжує виконання циклу й оброблює наступні запити.

На основі асинхронного підходу розроблено програмне забезпечення демонстраційної мережевої системи моніторингу температурних параметрів дизельних двигунів. Розроблений пристрій реалізує п'ять вимірювальних каналів (температура води на вході дизеля, температура води на виході правого

і лівого блоків дизеля, температури масла на вході і температури масла на виході дизеля). В якості датчиків температури було використано платинові термометри опору. Мікропроцесорний блок вимірювання температури виконує вимірювання опору та розраховує температуру за відповідною залежністю. Отримані значення температури передаються по інтерфейсу I2C на модуль Electric Imp [4]. Пристрій imp001 зчитує інформацію від блоку вимірювання і надсилає її агенту в хмарне середовище для подальшої обробки і передачі на асинхронний сервер.

Програмне забезпечення асинхронної системи віддаленого збору даних з використанням платформи Electric Imp складається з трьох частин:

1. Код пристрою;
2. Програма, яка працює на сервері Electric Imp Cloud;
3. Програмне забезпечення асинхронного сервера та сервера бази даних.

Кожен пристрій має власного агента, який розташований в impCloud. Він забезпечує надійний і безпечний зв'язок між пристроєм і користувацькими серверами. Для зв'язку агенту з хмарою використовується протокол HTTP, що полегшує взаємодію з іншими програмами. Програмне забезпечення пристрою imp001 і агента написано на мові Squirrel. Для написання програмного забезпечення для системи віддаленого збору температурних параметрів дизельних двигунів використано середовище розробки impCentral. Пристрій отримує дані від блоку вимірювання температури через протокол I2C і відправляє отримані дані власному агенту. Агент отримує дані, які йому надіслав пристрій, і передає їх далі на асинхронний сервер для подальшої обробки та зберігання

ВИСНОВОК

Таким чином, була реалізована асинхронна програмна система, яка буде виконувати велику кількість паралельних операцій із меншою кількістю потоків. Це дозволить розробляти великі системи обробки даних із використанням менш потужних пристроїв, не погіршуючи загальну продуктивність системи.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

- [1] Craig Walls “Spring in Action, 5th edition” — Shelter Island, NY, 2019. - 499p.
- [2] Web on Reactive Stack. [Online]. Available: <https://docs.spring.io/spring-framework/docs/current/reference/html/web-reactive.html#webflux-why-reactive>. Accessed on: November 27, 2021.
- [3] Netty [Online]. Available: [https://en.wikipedia.org/wiki/Netty_\(software\)](https://en.wikipedia.org/wiki/Netty_(software)) Accessed on: November 27, 2021.
- [4] Мальцев Д.В., Богомазов С.А. Організація системи Інтернету речей на базі платформи ElectricImp // Збірник праць XVI Всеукраїнської науково-практичної конференції “Ефективність та автоматизація інженерних рішень у приладобудуванні”. – К.: ПБФ, КПІ ім. Ігоря Сікорського. – 2020. – С. 419-421

Наук. керівник – к.т.н., доц. Богомазов С.А.