



**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ПОЛТАВСЬКА ПОЛІТЕХНІКА  
ІМЕНІ ЮРІЯ КОНДРАТЮКА**

**ЗБІРНИК МАТЕРІАЛІВ**

**76-ї НАУКОВОЇ КОНФЕРЕНЦІЇ ПРОФЕСОРІВ,  
ВИКЛАДАЧІВ, НАУКОВИХ ПРАЦІВНИКІВ,  
АСПІРАНТІВ ТА СТУДЕНТІВ УНІВЕРСИТЕТУ**

**ТОМ 1**

**14 травня – 23 травня 2024 р.**

## РОЗРОБКА ПРОГРАМНИХ ДОДАТКІВ РЕАЛЬНОГО ЧАСУ НА ОСНОВІ ВИКОРИСТАННЯ МІКРОСЕРВІСНОЇ АРХІТЕКТУРИ

Пріоритетним шляхом розробки програмного забезпечення для інформаційних систем реального часу є використання мікросервісної архітектури. Аналіз даних про системи реального часу показав, що вони характеризуються підвищеними вимогами щодо часових показників при виконанні запитів користувачів. Монолітна архітектура, яка є характерною для переважної більшості існуючих додатків, передбачає зосередження в межах однієї структури великої кількості модулів з можливістю їх загального доступу до спільної бази даних. Такий підхід, окрім того, що не забезпечує масштабованість такої системи, також може призвести до зниження продуктивності системи. Такий стан можливий у випадку потреби одночасного використання спільних апаратних ресурсів (процесорного часу, оперативної пам'яті, даних з БД) багатьма модулями одного монолітного додатку. Так, монолітний додаток важко масштабувати, коли різні модулі мають конфліктні вимоги до спільних апаратних ресурсів. Наприклад, один модуль може бути задіяний для обробки графіки з інтенсивним використанням центрального процесору. Другий модуль, в цей період часу, може мати підвищені вимоги до використання оперативної пам'яті. Однак, оскільки ці модулі розгорнуті разом, то необхідно вирішувати задачу з вибором апаратного забезпечення. Масштабувати в даному випадку необхідно весь додаток, навіть якщо це потрібно тільки для одного модуля. Тому для систем реального часу така архітектура не завжди може бути використана. Одним з перспективних шляхів підвищення продуктивності веб-додатків є використання мікросервісного підходу. При виборі такої архітектури для рішення кожної задачі розробляється окремий сервіс, який розгортається в окремому сервері та взаємодіє з іншими сервісами через стандартизований API. При такій архітектурі кожний мікросервіс має власну базу даних. Для обміну даними між мікросервісами пропонується використовувати формат JSON.

Так, для реалізації веб-додатку інформаційної системи електронного інтернет-аукціону пропонується використати мікросервісний підхід. Інтернет-аукціон є системою реального часу оскільки потребує мінімальних затримок при виконанні запитів користувачів. Тому для реалізації мікросервісів пропонується використати технології: Python FastAPI, Vue.js, WebSockets. В якості бази даних – PostgreSQL.

### *Література*

1. *Vue.js - The Progressive JavaScript Framework // Vue.js [Електронний ресурс] Vue.js. – Режим доступу: <https://vuejs.org/guide/introduction>*
2. *WebSockets [Електронний ресурс] Режим доступу: [https://developer.mozilla.org/en-US/docs/Web/API/WebSockets\\_API](https://developer.mozilla.org/en-US/docs/Web/API/WebSockets_API)*