

УДК 519.6

doi: 10.30970/vam.2023.31.11965

ПОРІВНЯЛЬНИЙ АНАЛІЗ ВИКОРИСТАННЯ СПЕЦІАЛІЗОВАНИХ МОВ ПРОГРАМУВАННЯ У ВИСОКОНАВАНТАЖУВАНИХ ДОДАТКАХ

І. Дияк, Я. Марчук

*Львівський національний університет імені Івана Франка,
вул. Університетська 1, Львів, 79000
e-mail: ivan.dyjak@lnu.edu.ua*

Розроблений процес реалізації REST (**R**epresentational **S**tate **T**ransfer “передача репрезентативного стану”) API (**A**pplication **P**rogramming **I**nterface “програмний інтерфейс застосування”) з використанням різних мов програмування. Для реалізації сервера на Python використано бібліотеку Django. Сервер на Golang реалізований на підставі спеціалізованого пакета net/http. На TypeScript використаний підхід реактивного розширення. Реалізована версія на спеціалізованій швидкісній мові програмування Erlang. Проведено порівняльний аналіз швидкодії розроблених бек-енд додатків.

Ключові слова: REST API, RESTful, Python, Golang, net/http, TypeScript, Erlang.

1. ВСТУП

Швидкодія високонавантажуваних бек-енд додатків є важливою характеристикою програмних систем. Наприклад, система обслуговування фізичних та юридичних осіб банку, прогноз погоди, новинні системи та торгівельні онлайн-майданчики. Розглянемо бек-енд розкладу занять для студентів, який також потребує високої швидкодії.

Одним з найпоширеніших варіантів розробки бек-енд застосунків є REST API підхід [1]. API системи потребують будь-які компанії та організації, які навчаються правильно управляти власними цифровими можливостями в умовах високо швидкісних змін цифрових середовищ. Збереження даних додатку організовано в документо-орієнтованій системі управління базами даних – MongoDB [2].

Важливо для швидкодії бек-енд додатку використати архітектуру мікросервісів як розподілену систему простих модулів, кожен з яких виконує окремий функціонал. Відокремленість кожного мікросервісу реалізується за допомогою програмної платформи Docker, що гарантує роботу кожної частини додатку у власному середовищі та цілком відокремлено від операційної системи [3]. Крім відокремленості, мікросервісна архітектура забезпечує простоту масштабування найбільш навантажуваних частин системи.

Відомо, що будь-яка сучасна мова програмування має кілька бібліотек для реалізації REST API підходу. Ми реалізуємо підхід з використанням мов – Python, Golang, TypeScript та Erlang [4]. Python був використаний як найбільш популярна мова програмування [5]. Golang використали тому, що найбільш швидко розвивається [6]. Erlang як предстанник функціональних мов програмування виявив найбільшу стабільність за останні 20 років у таких компаніях: Nortel, T-Mobile, RabbitMQ, WhatsApp, Nintendo, Samsung та багато інших [7]. Хорошими прикладами використання Erlang в Україні є додатки Приват24 та eHealth.

JavaScript/TypeScript – одна з найнадійніших мов програмування для бек-енд-середовища [8]. За основу процесу реалізації REST API було використано найпоширенішу для бек-енд програмування мову Python. Бібліотека Django добре пристосована для розробки високонавантажених серверів. Інша використана мова програмування, яка набуває шаленої популярності – Golang. Вона також чудово справляється з великою кількістю запитів, адже була розроблена не так давно, та під час її проектування було враховано більшість недоліків конкурентів. Використання TypeScript для REST API стало суттєво ефективнішим з застосуванням реактивних бібліотек, що також дало змогу TypeScript бути на рівні з іншими використаними мовами програмування у сфері бек-енду. Також розглядається реалізація на спеціалізованій мові програмування Erlang, яка була розроблена саме для бек-енд програмування компанією Ericsson [9]. Швидкодія розроблених додатків перевіряється універсальним інструментом тестування **JMeter**. **JMeter** – це інструмент для виконання навантажувальних тестів. Він написаний на мові програмування Java [10].

2. РЕАЛІЗАЦІЯ НА PYTHON

Python було вибрано у зв'язку з його розповсюдженістю та легкістю освоєння для початківців. Важливу роль популярності цієї мови демонструє величезна кількість розроблених бібліотек і продовження стабільного вдосконалення мови та бібліотек.

REST API реалізований мовою Python під мікрофреймворком Django. Головна перевага використання Django у цій праці – можливість управління RESTful запитами та детальна документація. RESTful – це інтерфейс, що підтримує безпечний, надійний та ефективний обмін інформацією завдяки стандартам програмної взаємодії [11]. Користувачів фреймворку понад мільйон. Це забезпечує постійне оновлення фреймворку, що і підтримує популярність та потребу у такому інструменті.

3. РЕАЛІЗАЦІЯ НА GOLANG

Golang розроблялася як спеціалізована мова для бек-енд програмування. Важливість її розробки у вирішенні розповсюджених проблем сучасного програмування, таких як неконтрольовані залежності, повільна обробка(побудова) програм, труднощі розробки програмного забезпечення та проблеми міжмовної взаємодії [12]. Отож основними вимогами, як наслідок перевагами, став спрощений і зрозумілий синтаксис. Особливу область застосування Golang становлять запити протоколу HTTP. Протокол HTTP працює над TCP. Ми можемо написати програму, яка приймає або надсилає запити за протоколом TCP і надсилати та отримувати запити за протоколом HTTP. Однак у зв'язку з тим, що цей протокол, в цілому, сфера Інтернет відіграє важливу роль, усі відповідні функції роботи з **http** були виділені в окремий пакет **net/http** [13]. Далі наведено фрагмент коду на Go обробки запитів:

```
// функція створення нового маршруту для HTTP-запитів
func HandleRequests() \{
    router := mux.NewRouter().StrictSlash(true)
    router.HandleFunc(
        "api/schedule/{group}", GetSchedule).Methods("GET")
}
```

```
log.Fatal(http.ListenAndServe(":port", router))
\}
```

4. ВИКОРИСТАННЯ РЕАКТИВНОГО ПІДХОДУ НА TYPESCRIPT

TypeScript, у наш час, є однією з найуніверсальніших мов програмування, адже з її допомогою можна розробляти фронт – і бек-енд додатки. Тут використано серверну платформу Node.js, призначену для виконання мережевих високопродуктивних додатків, які написані на мові TypeScript [14]. Сервер REST API реалізований засобами модуля **Express** та підходу реактивного програмування. Реактивність – це асинхронність, поєднана з потоковою обробкою даних [15]. Далі наведено фрагмент коду на TypeScript, що реалізує обробку запитів:

```
// реалізація отримання даних засобами реактивного підходу
app.get('/', async (req: Request, res: Response) => \{
  const url = mongoURL;
  const client = new MongoClient(url);
  try \{
    await client.connect();
    const database = client.db(dbName);
    const collection = database.collection(collectionName);
    const query = \{group\_name: groupName\};
    const document = collection.find(query).toArray();
    const result = rxjs.from(document);
    result.subscribe((x: any[]) => \{
      res.send(x);
    \});
  \} catch(error) \{
    console.error('Error:', error);
    res.status(500).send('Internal Server Error');
  \} finally \{
    client.close();
  \}
\});
```

5. РЕАЛІЗАЦІЯ НА ERLANG

У момент постійного розвитку технологій важливо використовувати потужності доступних апаратних можливостей з найбільшою ефективністю. Для максимально корисного використання ресурсів розроблена вузькоспеціалізована мова програмування – Erlang [9]. Ця мова програмування вирішує проблеми розподілених, відмовостійких і паралельних систем, які працюють при критичних обмеженнях. Для простішої роботи з мовою було додано стандартну бібліотеку модулів і шаблонних рішень – фреймворк **Open Telecom Platform**. Варто зазначити, що надважливою здатністю Erlang є заміна коду під час безперервної експлуатації додатків [9].

6. АНАЛІЗ ОБЧИСЛЮВАЛЬНИХ ЕКСПЕРИМЕНТІВ

Для зручного тестування навантаження розроблених серверів використовується популярний інструмент **JMeter** [10].

У табл. 1 наведено результати виконання 2000 запитів для 5 з'єднань при максимальному часі очікування 10 секунд для кожного реалізованого сервера.

Таблиця 1

Результати виконання 2000 запитів для 5 з'єднань

	Python	Golang	TypeScript	Erlang
Загальний час, хв	23,63	24,9	12,48	0,026
Запитів/сек	2,59	2,53	2,87	923
Затримка, сек	3,91	3,29	2,01	0,00352
Пропускна здатність, КБ/сек	282	564,3	244,34	2510

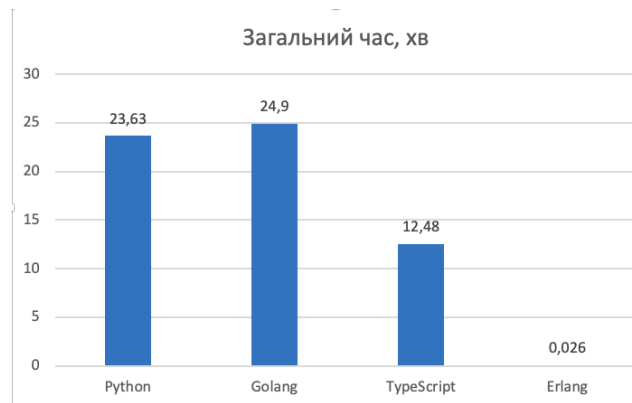


Рис. 1. Гістограма часу виконання запитів

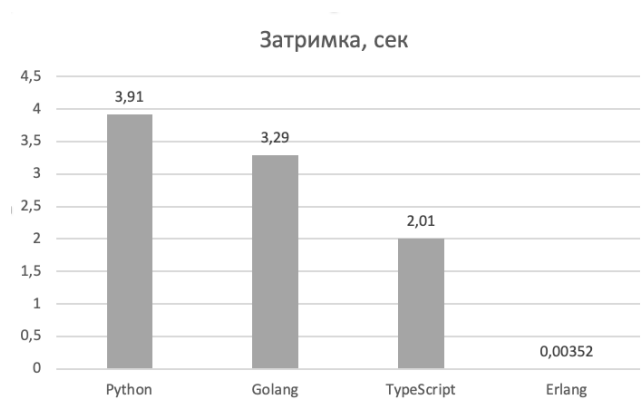


Рис. 2. Гістограма затримки надсилання відповідей

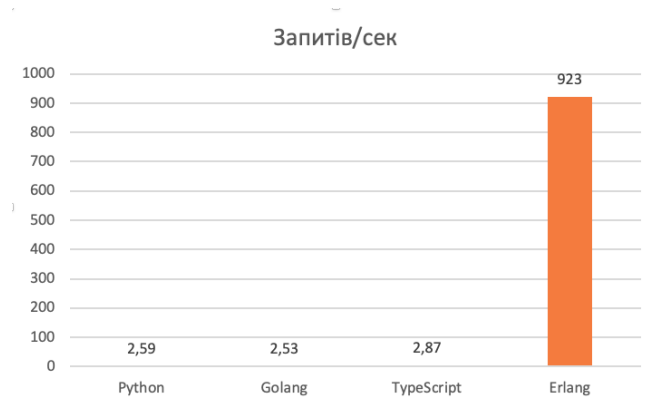


Рис. 3. Гістограма кількості запитів за секунду



Рис. 4. Гістограма пропускної здатності серверів

Таблиця 2

Результати виконання 4000 запитів для 10 з'єднань

	Python	Golang	TypeScript	Erlang
Загальний час, хв	24,15	25,08	30,18	0,16
Запитів/сек	2,53	2,47	2,89	309,1
Затримка, сек	3,98	3,41	4,7	0,02527
Пропускна здатність, КБ/сек	269,75	558,1	195,4	971,9

Розглянемо графіки загального часу виконання запитів серверами (рис. 1), затримки відповідей від серверів (рис. 2), кількість запитів за секунду (рис. 3) і пропускну здатність серверів (рис. 4).

У табл. 2 наведено результати виконання 4000 запитів для 10 з'єднань при максимальному часі очікування 10 секунд для кожного реалізованого сервера.

7. ВИСНОВОК

Результати тестування засвідчують, що бек-енд, створений спеціалізованою мовою **Erlang**, має найкращу швидкодію. Функціональний підхід до реалізації REST API з використанням інтерфейсу передачі повідомлень в реалізації паралельної обробки даних пояснює перевагу цієї мови над іншими мовами щодо швидкості виконання. Реалізація REST API засобами універсальних мов програмування **Python**, **Golang** і **TypeScript** продемонстрували майже однакову швидкодію. В наступних дослідженнях планується використати розподілені системи кешування бек-енд додатків з хмарними технологіями. Очікується, що такі підходи значно підвищать швидкодію REST API додатку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Masse M. REST API Design Rulebook / M. Masse – O'Reilly Media, 2011. – 112 с.
2. Bradshaw S. MongoDB: The Definitive Guide, 3rd Edition / S. Bradshaw, E. Brazil, K. Chodorow. – O'Reilly Media, 2019. – 514 с.
3. Poulton N. Docker Deep Dive / N. Poulton. – Packt Publishing, 2020. – 250 с.
4. Top back-end programming languages for 2023 (2022, Nov 10) [Online]. Available: <https://blog.boot.dev/backend/best-backend-programming-languages/>
5. Python (2023, Jan 5) [Online]. Available: <https://www.aclti.com/en/blog/the-most-popular-language-is-python>
6. Go (2020, Jul 6) [Online]. Available: <https://medium.com/javarevisited/what-is-go-or-golang-programming-language-why-learn-go-in-2020-1cbf0afc71db>
7. Erlang (2023, June 20) [Online]. Available: [https://en.wikipedia.org/wiki/Erlang_\(programming_language\)](https://en.wikipedia.org/wiki/Erlang_(programming_language))
8. JavaScript/TypeScript (2020) [Online]. Available: <https://blog.back4app.com/backend-programming-languages-list/>
9. Laurent S.St. Introducing Erlang, 2nd Edition / S.St. Laurent. – O'Reilly Media, March 2017. – 168 с.
10. Jain J. Learn API Testing: Norms, Practices, and Guidelines for Building Effective Test Automation / J. Jain. – Apress, July 2022. – 235 с.
11. Sarda S. Build REST APIs with Django REST Framework and Python / S. Sarda. – Packt Publishing, June 2021. – 302 с.
12. Anagnostopoulos A. Hands-On Software Engineering with Golang / A. Anagnostopoulos. – Packt Publishing, January 2020. – 460 с.
13. Yellavula N. Building RESTful Web services with Go / N. Yellavula. – Packt Publishing, 2017. – 316 с.
14. Goldberg J. Learning TypeScript / J. Goldberg. – O'Reilly Media, June 2022. – 318 с.
15. Blackheath S. Functional Reactive Programming / S. Blackheath, A. Jones. – Manning Publications, 2016. – 360 с.

Стаття: надійшла до редколегії 31.07.2023

доопрацьована 04.09.2023

прийнята до друку 20.09.2023

**COMPARATIVE ANALYSIS OF THE USE
OF SPECIALIZED PROGRAMMING LANGUAGES
IN HIGH-LOAD APPLICATIONS**

I. Dyyak, Y. Marchuk

*Ivan Franko National University of Lviv,
1, Universytetska str., 79000, Lviv, Ukraine
e-mail: ivan.dyyak@lnu.edu.ua*

Performance is the key characteristic of high-load back-end applications, such as finance systems, news services, social networks, Internet of Things systems, online trading, online gaming, and educational system. The process of implementing REST (Representational State Transfer) and API (Application Programming Interface) using various programming languages was developed. To implement the server in Python, we used the Django library, which is suitable for developing high-load servers. The Golang server is implemented on the basis of a specialized net/http package. The TypeScript approach is based on reactive extension and the Express module. The version in a specialized high-speed programming language Erlang is implemented using a standard library of modules and template solutions – Open Telecom Platform. Data storage is organized in the document-oriented MongoDB system. A comparative analysis of the performance of the developed back-end applications was carried out. For testing, a tool with the function of performing load tests – JMeter – is used. The choice of programming language depends on the specific requirements of the project. Each language has its own advantages and disadvantages.

Key words: REST API, RESTful, Python, Golang, net/http, TypeScript, Erlang.