

УДК 004.7

## ОСОБЛИВОСТІ ДІАГНОСТУВАННЯ МЕРЕЖЕВИХ РЕСУРСІВ В ОС WINDOWS

Христинець Н.А., к. т. н., доцент, [n.khrystynets@lutsk-ntu.com.ua](mailto:n.khrystynets@lutsk-ntu.com.ua), ЛНТУ  
Рибачук Р.О., магістрант, [lo886690@gmail.com](mailto:lo886690@gmail.com), ЛНТУ

Діагностика мережевих ресурсів в операційній системі Windows є критично важливою для забезпечення ефективності та надійності не лише мережевого підключення, а й функціонування усіх компонент комп'ютерної системи. Цей процес взаємодії тісно пов'язаний з різними аспектами операційної системи та користувачів, а також впливає на загальну продуктивність та безпеку інформаційно-комунікаційного середовища [1]. Діагностика запитів надає інформацію адміністраторам мережі для прийняття рішень щодо розширення мережі, покращення безпеки та вдосконалення її продуктивності. Водночас, така діагностика мереж має великий вплив на веб-представництва компаній, які реалізуються за допомогою веб-запитів до відповідних служб [2] та безпосередньо залежать від надійного та ефективного мережевого з'єднання для свого функціонування і надання послуг [3].

На практиці, робота веб-додатків часто супроводжується парсингом даних для отримання, аналізу та обробки інформації, а асинхронність є одним із способів оптимізації під час обробки багаточисельних запитів бази даних.

Для вирішення питання багатопоточної обробки запитів використовують різні алгоритми.

Розглядаючи загальну схему діагностування мережевих ресурсів [4] засобами найпоширенішої операційної системи Windows, можна виділити такі етапи: перевірка стану мережевого пристрою; застосування консольних команд, або мережевих утиліт для тестування доступності і відстеження маршруту запиту; перевірка налаштувань IP та Winsock класичними методами адміністрування мереж; моніторинг завантаженості мережі та ідентифікації процесів, які використовують мережеві ресурси; аналіз стану налаштувань мережевих служб DHCP, DNS, WLAN AutoConfig. Така діагностика дозволяє ідентифікувати та усувати проблеми мережевого трафіку в операційній системі Windows.

Розглядаючи в контексті комп'ютерних мереж та систем запит, як мережевий ресурс чи, точніше, як взаємодію з мережевим ресурсом, підтримка асинхронного мультипроцесингу забезпечується через використання функціональних вбудованих механізмів [5]. Якщо розглянути асинхронність запитів засобами фреймворку Laravel, то тут для керування чергою задач у базі даних можна використовувати таблицю Jobs, в якій зберігаються інформація про задачі, що очікують виконання. Для ефективності їх використання і зв'язки

використано механізми connects. Варто зазначити, що механізм впровадження класу Jobs можливий в фоновому (Background Jobs) і нефоновому (Non-Background/Synchronous Jobs) режимі. Черги (queues) працюють на основі концепції «виробник-споживач». Це дозволяє додавати задачі (jobs) до черги, а потім виконувати їх асинхронно. Перевірка черги на наявність задач виконується у фоновому режимі за схемою, як на рисунку 1.

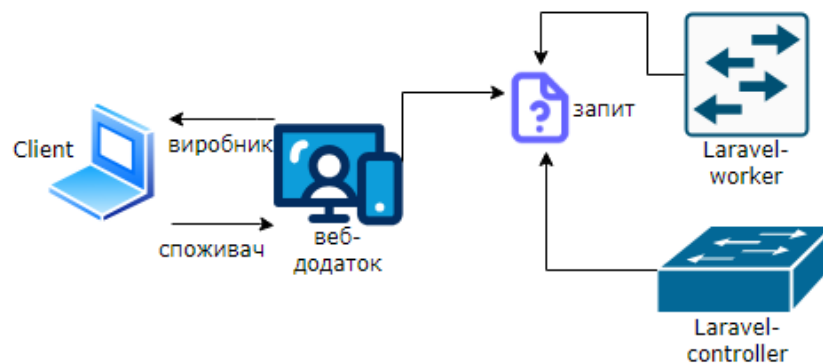


Рисунок 1 – Використання асинхронних запитів у фреймворку Laravel

Ця схема використана для вирішення питання багатопоточної обробки запитів з підтримкою асинхронного мультипроцесингу через використання функціональних вбудованих механізмів фреймворку Laravel.

**Висновок.** У результаті дослідження підтверджено розв'язання питання діагностики мережевих ресурсів та взаємодії з ними. Використано класичні методи адміністрування операційної системи Windows та асинхронні методи взаємодії з мережевими ресурсами засобами фреймворку Laravel з підтримкою мультипроцесингу.

### Список використаних джерел

1. Христинець Н.А. Практичне дослідження програмних засобів кіберзахисту в операційній системі Manjaro. *Електронне моделювання*. Київ, 2022. Т. 44, № 4. С. 55-63.
2. Коцюба А.Ю., Цяпич Я.П., Лавренчук С.В. Про методику оптимізації відмовостійкості веб-серверів на одночасну кількість запитів. *Комп'ютерно-інтегровані технології: освіта, наука, виробництво*. 2016. Т. 24-25. С. 37-41
3. Use of bem-blocks when creating a site. *Computer-integrated technologies: education, science, production* / Khrystynets N. A. et al. 2019. V. 35. P. 206-210.
4. Melnyk K., Melnyk V., Hryhoryshyn A. Automatic collection of information (parsing) in the network. *Journal of Computing and Information Technology*. 2020. V.39. P. 151-156.
5. Multiprocessing as a Way to Optimize Queries / N. Khrystynets et al. *Advances in Transdisciplinary Engineering*. 2024. Vol. 48. P. 455–464. DOI: 10.3233/ATDE231357