

## Особливості процесу автоматизованого тестування WEB-орієнтованого програмного забезпечення із застосуванням паралельності тестів на основі Selenium Grid

Вінницький національний технічний університет

### Анотація

Наведено результати дослідження процесу автоматизованого тестування WEB-орієнтованого програмного забезпечення із застосуванням паралельності тестів на основі Selenium Grid та його особливості. Відзначено перспективи застосування даного підходу для оптимізації процесу автоматизованого тестування. Описано роботу Selenium Grid при паралельності тестів. Виділено недоліки та переваги паралельності тестів.

**Ключові слова:** програмне забезпечення, паралельність, автоматизоване тестування, оптимізація.

### Abstract

The results of research into the process of automated testing of WEB-oriented software using the parallelism of tests based on Selenium Grid and its features are given. The prospects of using this approach to optimize the process of automated testing are noted. The operation of Selenium Grid with parallel tests is described. Disadvantages and advantages of parallel tests are highlighted.

**Keywords:** software, concurrency, automated testing, optimization.

### Вступ

Автоматизоване тестування – важливий етап у розробці програмного забезпечення (ПЗ). Його мета – забезпечити високу якість продукту, виявляючи помилки та недоліки. Однак, якщо кількість тестів зростає, час їх виконання може стати обмежуючим фактором. Тут на допомогу приходить паралелізація тестів [1,2].

Паралелізація тестів при автоматизованому тестуванні ПЗ є важливою, зважаючи на такі особливості:

1. Збільшення швидкості: паралельне виконання тестів на різних пристроях або в різних потоках дозволяє значно зменшити час виконання [3].
2. Ефективність ресурсів: паралельне виконання дозволяє використовувати ресурси (пристрої, сервери) більш ефективно.
3. Виявлення конкурентних проблем: паралельне тестування допомагає виявити проблеми, пов'язані з конкурентністю, які можуть впливати на продуктивність [4].

### Результати дослідження

Розглянемо детальніше особливості організації процесу паралелізації тестів.

1. Розподіл тестів. Тести необхідно розділити на групи і виконувати їх паралельно. Наприклад, UI-тести можна виконувати на одному пристрої, а API-тести – на іншому.
2. Використання фреймворків. Багато автоматизованих фреймворків підтримують паралельне виконання тестів. Вони дозволяють запускати тести в різних потоках або в різних процесах.
3. Хмарні рішення. Використання хмарних сервісів для паралельного виконання тестів на різних пристроях.

У даному дослідженні проведено дослідження паралельності тестів за допомогою проксі сервера Selenium Grid. Розглянемо детальніше паралельність тестів за допомогою Selenium Grid.

Selenium Grid – інструмент, який дозволяє виконувати тести на різних конфігураціях та пристроях одночасно [5].

Основні компоненти Selenium Grid:

1. Hub(головний вузол) – центральна точка у Selenium Grid, яка приймає запити на виконання тестів і розподіляє їх між доступними вузлами.

2. Вузли – машини або пристрої, на яких виконуються тести. Кілька вузлів можуть бути підключені до головного вузла (Hub).

Процес паралельного виконання тестів з використанням Selenium Grid включає наступні кроки:

1. Налаштування Hub. Спочатку необхідно запустити Hub, який буде відслідковувати запити на виконання тестів і розподіляти їх між вузлами.

2. Налаштування вузлів. Після запуску головного вузла (Hub) необхідно запустити один чи кілька вузлів, які будуть виконувати тести. Кожен вузол може мати свої власні конфігурації, такі як операційна система, браузер та його версія, розмір вікна браузера тощо.

3. Написання тестів. Розробляються тести з використанням Selenium WebDriver. Тести можуть бути написані на мовах програмування, які підтримуються Selenium, таких як Java, Python, C#, Ruby тощо.

4. Конфігурація тестового середовища. Тести налаштовуються для виконання на Selenium Grid, зазначаючи адресу головного вузла (Hub) та конфігурацію вузлів.

5. Запуск тестів. Тести запускаються з вказівкою адреси головного вузла (Hub). Головний вузол розподіляє тести між вузлами, які відповідають вказаним критеріям.

6. Перегляд результатів. Після завершення тестів можна переглянути результати виконання, які можуть містити інформацію про успішність або невдачі кожного тесту, а також «лог-звіти» тестів.

Застосування паралельного виконання тестів з допомогою Selenium Grid дозволяє ефективно використовувати ресурси та прискорює процес тестування, особливо в умовах великого обсягу тестів або вимог до підтримки різних конфігурацій пристроїв та середовищ. Особливістю Selenium Grid також є те що цей інструмент сам вираховує потрібну кількість тестів, а це в свою чергу залежить від можливостей пристроїв або середовищ та їх потужностей. Схематично узагальнену роботу Selenium Grid зображено на рис. 1. На рис. 1 проілюстровано перевагу паралельності тестів, що оптимізує велику частину ресурсів та часу для проходження тестів. Адже за допомогою вузлів ми можемо запустити одні і ті ж самі тести в паралелі на різних пристроях та різних браузерах (кросбраузерність та кросплатформеність). А це, в свою чергу, досить сильно оптимізує час проходження тестів.

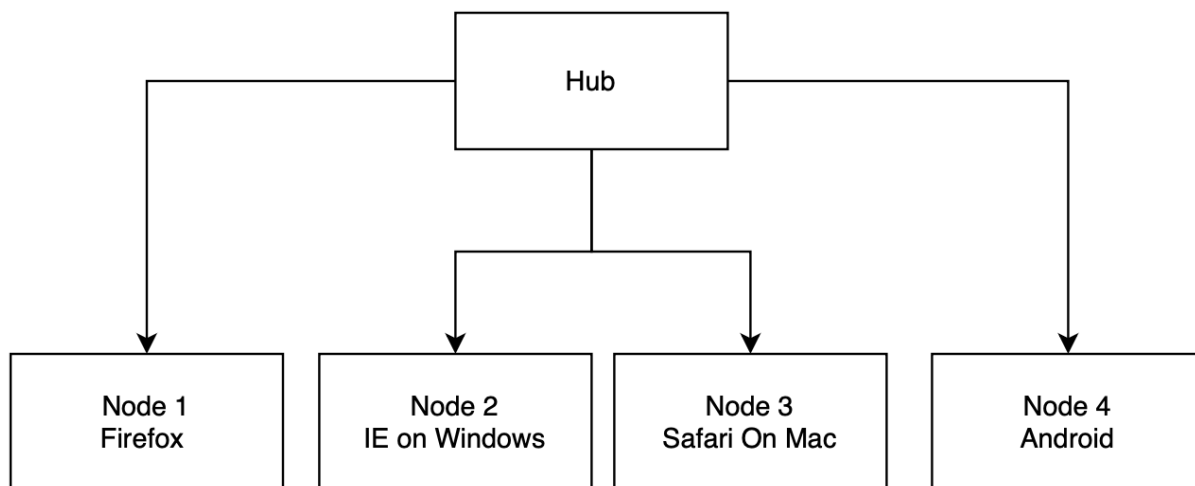


Рисунок 1 – Узагальнена схема роботи проксі-сервера Selenium Grid

Разом з тим, варто виділити деякі проблеми паралельності тестів:

1. Залежності між тестами. Деякі тести можуть мати залежності між собою. При паралельному виконанні це може призвести до неправильних результатів.

2. Ресурси. Паралельне виконання вимагає більше ресурсів (пристроїв, серверів), що може бути вартим уваги.

## Висновки

Таким чином, паралелізація тестів є потужним інструментом для оптимізації процесу автоматизованого тестування ПЗ, особливо WEB-орієнтованого ПЗ. Вона дозволяє збільшити швидкість виконання тестів та забезпечити високу якість продукту. У даній роботі розглянуто переваги та недоліки паралельності тестів, виділено їх переваги та недоліки, а також описано основні компоненти та принцип роботи Selenium Grid.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Аналіз предметної області автоматизованого тестування WEB-ресурсів в контексті оптимізації процесів / О.С. Морозов, А.А. Яровий, А.В. Козловський : Збірник матеріалів Всеукраїнської науково-практичної конференції "Молодь в науці: дослідження, проблеми, перспективи (МН-2023)". – В.: ВНТУ, 2023. – С. 1-3. [Електронний ресурс] – Режим доступу: <https://conferences.vntu.edu.ua/index.php/mn/mn2023/paper/viewFile/18706/15503>
2. Елфрід Дастін, Джефф Решкі, Джон Пол. Автоматизоване тестування програмного забезпечення. Лори – 2009. – 589 с.
3. The Art of Software Testing / Glenford J. Myers, Revised and Updated by Tom Badgett, Todd M. Thomas, Corey Sandler. – 2nd ed. – Hoboken, New Jersey: John Wiley & Sons, Inc., 2004 – 234 p.
4. Скорочуємо час виконання UI-тестів з використання Selenium Grid у Minikube-кластері [Електронний ресурс]. – 2020. – Режим доступу: <https://dou.ua/lenta/articles/autotesting-with-selenium-grid-and-minikube/>
5. 10 вразливих місць в інфраструктурі для автотестів, на які варто звернути увагу [Електронний ресурс]. – 2022. – Режим доступу: <https://dou.ua/forums/topic/41147/>

**Морозов Олександр Сергійович** – аспірант кафедри комп'ютерних наук, факультет інтелектуальних інформаційних технологій та автоматизації, Вінницький національний технічний університет, м. Вінниця.

**Яровий Андрій Анатолійович** – д.т.н., проф., завідувач кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

**Козловський Андрій Володимирович** – к.т.н., доц., доцент кафедри комп'ютерних наук, Вінницький національний технічний університет, м. Вінниця.

**Oleksandr S. Morozov** – Post-Graduate Student of the Department for Computer Science, Faculty of Intelligent Information Technologies and Automation, Vinnytsia National Technical University, Vinnytsia.

**Andrii A. Yarovyi** – Dr. Sc. (Eng), Professor, Head of the Department for Computer Science, Vinnytsia National Technical University, Vinnytsia.

**Andrii V. Kozlovskiy** – Ph. D. (Eng), Associate Professor of the Department for Computer Science, Vinnytsia National Technical University, Vinnytsia.