

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ ІМЕНІ
ІВАНА ПУЛЮЯ
ФАКУЛЬТЕТ КОМП'ЮТЕРНО ІНФОРМАЦІЙНИХ СИСТЕМ І
ПРОГРАМНОЇ ІНЖЕНЕРІЇ
КАФЕДРА КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

СТЯГАР АНДРІЙ ІВАНОВИЧ

УДК 004.442

ПРОДУКТИВНІСТЬ СЕРВЕРНИХ JAVASCRIPT ПРОГРАМ ДЛЯ
КОМП'ЮТЕРНИХ СИСТЕМ З КЛІЄНТ-СЕРВЕРНОЮ АРХІТЕКТУРОЮ.

123 «Комп'ютерна інженерія»

Автореферат
дипломної роботи на здобуття освітнього ступеня «магістр»

Тернопіль 2018

Роботу виконано на кафедрі комп'ютерних систем та мереж Тернопільського національного технічного університету імені Івана Пулюя Міністерства освіти і науки України

Керівник роботи: д. т. н, професор кафедри математичних методів в інженерії, Ясній Олег Петрович, Тернопільський національний технічний університет імені Івана Пулюя,

Рецензент: кандидат технічних наук, доцент кафедри інформатики і математичного моделювання, Гащин Надія Богданівна, Тернопільський національний технічний університет імені Івана Пулюя,

Захист відбудеться 22 лютого 2018 р. о 9:00 годині на засіданні екзаменаційної комісії № 34 у Тернопільському національному університеті імені Івана Пулюя за адресою: 46001, м. Тернопіль, вул. Руська, 56, навчальний корпус №1, аудиторія 603.

ЗАГАЛЬНА ХАРАКТЕРИСТИКА РОБОТИ

Актуальність теми роботи. Використання веб-технологій помітно зростає в усіх сферах людського життя. Тому є потреба підвищувати основні показники продуктивності серверів, для забезпечення комфортної роботи веб-додатків. На даний момент мова програмування JavaScript широко використовується для написання серверних програм. Поряд з іншими технологіями дана мова є однією з найкращих для створення високопродуктивних програм. Проте вона, як і інші має свої переваги та недоліки і для різних задач показує різні показники продуктивності. Тому при розробці програми необхідно володіти цією інформацією для забезпечення максимально ефективної її роботи.

Мета і завдання роботи. Основною метою даної роботи є дослідити продуктивність серверних JavaScript платформ та фреймворків та виявити основні недоліки даних технологій. Основними завданнями даного дослідження є:

- огляд актуальних проблем продуктивності сучасних серверних платформ та фреймворків;
- аналіз особливостей архітектури серверних JavaScript фреймворків та порівняння її з іншими технологіями
- розробити веб-додаток для запуску на серверній JavaScript платформі Node.js;
- за допомогою програми Jmeter провести ряд навантажувальних тестів на додаток та відобразити отримані результати за допомогою графіків;
- оцінити продуктивність при багатопотоковій реалізації додатку та з використанням балансувальника навантаження.

Об'єкт, предмет та методи дослідження. Об'єкт наукового дослідження – серверні JavaScript платформи та фреймворки.

Предмет наукового дослідження – продуктивність серверних JavaScript програм

Дослідження проводилось методом тестування продуктивності веб-додатку за допомогою програми Jmeter

Наукова новизна та практичне значення отриманих результатів.

В результаті, аналізу архітектури платформи Node.js було виявлено, що асинхронна обробка запитів дозволяє забезпечити більшу продуктивність і масштабованість при типових мережевих навантаженнях, ніж типова реалізація на основі потоків. Навантажувальне тестування показало, що дана технологія дуже добре справляється з великою кількістю одночасних запитів і при цьому використовує мінімальну кількість оперативної пам'яті. Проте для запитів з інтенсивним використанням центрального процесора спостерігається помітне падіння продуктивності. Вирішенням цієї проблеми може бути запуск кількох екземплярів додатку та балансування навантаження між ними. Тестування продуктивності додатку, масштабованого за допомогою модуля Cluster та сервера Nginx, показали у два рази кращий результат ніж при однопотоковій реалізації. Тому для забезпечення максимальної продуктивності додатків необхідно використовувати дані інструменти масштабування.

Апробація. Результати дипломної роботи апробовані на V науково-технічній конференції «Інформаційні моделі, системи та технології» (Тернопіль 2018 р.).

Структура роботи. Робота складається з пояснювальної записки та графічної частини. Робота складається зі вступу, трьох розділів, висновка, списку літератури та найменувань, додатків. Загальний обсяг текстової частини – 107 сторінок, 7 таблиць, 54 рисунки, бібліографічний опис – 19.

ОСНОВНИЙ ЗМІСТ РОБОТИ

У вступі подано загальну характеристику роботи: стан розробки наукової проблеми й актуальності роботи, об'єкт, предмет, описано наукову новизну і практичну значимість отриманих результатів.

У першому розділі під назвою «ОГЛЯД СУЧАСНИХ ВЕБ-ТЕХНОЛОГІЙ» проведено огляд архітектури сучасних веб-додатків та основних технологій, які використовуються для їх створення, а також актуальних проблем продуктивності веб.

У другому розділі «ПРОДУКТИВНІСТЬ СЕРВЕРНОГО JAVASCRIPT» проведено аналіз архітектури серверної платформи Node.js та здійснено порівняння даної платформи з іншими серверними технологіями. Також проводиться характеристика основних методів та інструментів для дослідження продуктивності веб-додатків.

У третьому розділі під назвою «ПРАКТИЧНА РЕАЛІЗАЦІЯ ДОСЛІДЖЕННЯ» проведено опис додатку, який використовувався для реалізації дослідження. Представлено опис процедури налаштування програми Jmeter, створення скриптів для тестування продуктивності, а також опис процедури тестування з результатами у вигляді графіків та таблиць.

У четвертому розділі під назвою «ОБГРУНТУВАННЯ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ» представлено економічне обґрунтування та оцінку економічних показників дослідження.

У п'ятому розділі під назвою «ОХОРОНА ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ» представлено опис основних вимог та правил техніки безпеки під час експлуатації електронно-обчислювальних машин.

У шостому розділі під назвою «ЕКОЛОГІЯ» представлено опис методів узагальнення екологічної інформації та описано процес формування бази статистичних даних в екології.

ВИСНОВКИ

В ході проведення дослідження було виконано наступні завдання:

- проведено огляд сучасних серверних технологій для розробки веб-додатків;
- проведено аналіз архітектури серверної платформи Node.js та здійснено порівняння її з іншими технологіями;
- розроблено веб-додаток з використанням фреймворку Express.js, та бази даних MongoDB;
- за допомогою програми Jmeter проведено ряд навантажувальних тестів на додаток;
- проведено тестування продуктивності при багатопотоковій реалізації додатку та з використанням балансувальника навантаження .

В результаті, аналізу архітектури платформи Node.js було виявлено, що асинхронна обробка запитів дозволяє забезпечити більшу продуктивність і масштабованість при типових мережевих навантаженнях, ніж типова реалізація на основі потоків. Навантажувальне тестування показало, що дана технологія дуже добре справляється з великою кількістю одночасних запитів і при цьому використовує мінімальну кількість оперативної пам'яті. Проте для запитів з інтенсивним використанням центрального процесора спостерігається помітне падіння продуктивності. Вирішенням цієї проблеми може бути запуск кількох екземплярів додатку та балансування навантаження між ними. Тестування продуктивності додатку, масштабованого за допомогою модуля Cluster та сервера Nginx, показали у два рази кращий результат ніж при однопотоковій реалізації. Тому для забезпечення максимальної продуктивності додатків необхідно використовувати дані інструменти масштабування.

СПИСОК ОПУБЛІКОВАНИХ АВТОРОМ НАУКОВИХ ПРАЦЬ ЗА ТЕМОЮ РОБОТИ

1. Стягар А.І. Продуктивність веб-додатків / Матеріали V науково-технічної конференції ТНТУ ім. І. Пулюя, 2018. – 82 с.
2. Стягар А.І. Серверна платформа NODE.JS/ Матеріали V науково-технічної конференції ТНТУ ім. І. Пулюя, 2018. – 83 с.

АНОТАЦІЯ

Дипломна робота на тему “Продуктивність серверних JavaScript програм для комп’ютерних систем з клієнт-серверною архітектурою” на здобуття освітньо-кваліфікаційного рівня “Магістр” із спеціальності “Комп’ютерна інженерія” написана обсягом 107 сторінок і містить 54 ілюстрацій, додатки та 19 джерел з переліком посилань.

Дипломна робота магістра присвячена аналізу архітектури серверних JavaScript платформ та проведенні експериментальних досліджень продуктивності платформи Node.js за допомогою програми Jmeter.

В результаті, аналізу архітектури платформи Node.js було виявлено, що асинхронна обробка запитів дозволяє забезпечити більшу продуктивність і масштабованість при типових мережевих навантаженнях, ніж типова реалізація на основі потоків. Навантажувальне тестування показало, що дана технологія дуже добре справляється з великою кількістю одночасних запитів і при цьому використовує мінімальну кількість оперативної пам’яті. Проте для запитів з інтенсивним використанням центрального процесора спостерігається помітне падіння продуктивності. Вирішенням цієї проблеми може бути запуск кількох екземплярів додатку та балансування навантаження між ними. Тестування продуктивності додатку, масштабованого за допомогою модуля cluster та сервера Nginx, показали у два рази кращий результат ніж при однопотоківій реалізації.

Тому для забезпечення максимальної продуктивності додатків необхідно використовувати дані інструменти масштабування.

Ключові слова: ПРОГРАМНА ПЛАТФОРМА, ПРОДУКТИВНІСТЬ, ФРЕЙМВОРК, СЕРВЕР, АРХІТЕКТУРА, ВЕБ-ДОДАТОК.

ABSTRACT

Thesis on the theme "Server JavaScript software performance for computer systems with client-server architecture" for obtaining an educational qualification level "Master" in the specialty "Computer Engineering" is written in volume of 107 pages and contains 54 illustrations, applications and 19 sources from a list of links.

The master's thesis is devoted to analyzing the architecture of server-side JavaScript platforms and conducting experimental research on the performance of the Node.js platform using the Jmeter program.

As a result of analysis of the Node.js platform architecture, it was found that asynchronous query processing allows for higher performance and scalability in typical network loadings than typical implementation based on flows. Load testing showed that this technology is very well coped with a large number of simultaneous requests and thus uses a minimum amount of RAM. However, for intensive CPU usage, there is a noticeable drop in performance. Solving this problem may be to run multiple instances of the application and load balancing between them. Testing the performance of the application, scaled using the Cluster module and the Nginx server, showed twice as good a result than one-threaded implementation. Therefore, you need to use these scaling tools to maximize the performance of applications.

Keywords: SOFTWARE PLATFORM, PERFORMANCE, FRAMEWORK, SERVER, ARCHITECTURE, WEB-APPLICATION.