

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

ЯРЕМЧУК ЯНА ІГОРІВНА

УДК 004.3

**ПРОГРАМНА СТРЕС-СИСТЕМА ТЕСТУВАННЯ
ОБЧИСЛЮВАЛЬНИХ РЕСУРСІВ АСИНХРОННИМИ ЗАСОБАМИ НА
ОСНОВІ ВИКОРИСТАННЯ ПРОТОКОЛУ WEBSOCKET**

121 «Інженерія програмного забезпечення»

Автореферат

дипломного проекту на здобуття освітнього ступеня «магістр»

Тернопіль 2018

Проект виконано на кафедрі програмної інженерії Тернопільського національного технічного університету імені Івана Пулюя

Керівник проекту: Доцент, канд. тех. наук,
Кінах Ярослав Ігорович,
Тернопільський національний технічний університет
імені Івана Пулюя

Рецензент: доктор технічних наук,
професор кафедри комп'ютерних систем та мереж,
Лупенко Сергій Анатолійович,
Тернопільський національний технічний університет
імені Івана Пулюя

Захист відбудеться 22 лютого 2017 р. о 9³⁰ годині на засіданні екзаменаційної комісії №1 у Тернопільському національному технічному університеті імені Івана Пулюя за адресою: 46001, м. Тернопіль, вул. Руська, 56, навчальний корпус №1, ауд.

101

ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРОЕКТУ

Актуальність теми проекту. Стресове тестування дозволяє перевірити наскільки додаток і система в цілому працездатні в умовах стресу і також оцінити здатність системи до регенерації, тобто до повернення до нормального стану після припинення впливу стресу. Стрес тестування включає в себе тестування за межами нормального робочого потенціалу, часто до моменту збою, для того, щоб спостерігати за результатами.

Мета проекту: Розробка розподіленої системи для проведення стрес-тестування веб-серверів, які використовують технологію WebSocket та асинхронні засоби.

Об'єкт, методи та джерела дослідження: одночасна робота розподілених додатків, які навантажують серверну частину тестованої системи, та подальший збір статистики.

Практичне значення отриманих результатів: Розроблена система використовується для проведення стрес-тестування, перевіряє наскільки додаток і система працездатні в умовах стресу.

Структура проекту. Проект складається з розрахунково-пояснювальної записки та графічної частини. Розрахунково-пояснювальна записка складається з вступу, 4 частин, висновків, переліку посилань. Обсяг проекту: розрахунково-пояснювальна записка – арк. формату А4, графічна частина – 15 слайдів.

ОСНОВНИЙ ЗМІСТ ПРОЕКТУ

У вступі проведено аналіз актуальності та мети проекту, поставлено задачі дослідження, наведена наукова новизна та практичне значення одержаних результатів.

В розділі «Розробка програмної системи», розглянуто популярні системи для проведення стресс-тестування, проведено аналіз систем. Обгрунтовано метод асинхронного підходу та розроблено архітектуру системи навантажувального стресс-тестування. .

В розділі «**Спеціальна частина**» описаний процес тестування розробленої програмної системи. Проведено реалізацію архітектурної та прикладної частини застосунка.

В розділі «**Обґрунтування економічної ефективності**» розглянуто питання організації виробництва і проведено розрахунки техніко-економічної ефективності проектних рішень з огляду двох підходів розробки – об’єктно-орієнтованого та процедурного. Також проаналізовано економічні складові, що виникають в ході розробки, та чинники, які впливають на виконання проекту.

В розділі «**Охорона праці та безпека в надзвичайних ситуаціях**» розглянуто питання специфіки дотримання норм та правил Охорони праці в галузі розробки ПЗ з використанням персональних комп’ютерів. Також проаналізовано вплив здорового способу життя на професійну діяльність людей. Досліджено негативний вплив іонізуючого випромінювання та дієві засоби захисту працівників від нього.

У **загальних висновках щодо дипломного проекту** описано результати проектування та розробки інформаційної системи для стрес-тестування обчислювальних ресурсів.

ВИСНОВКИ

В результаті виконання дипломного проекту було вирішено проблему , яка стала актуальною при розробці серверів, які використовують бінарний протокол передачі даних, які передають через WebSocket.

У процесі роботи було отримано наступні результати:

- досліджено поняття навантажувального тестування, його особливості та цілі;
- детально розглянуто та проаналізовано функціональні можливості кількох існуючих систем для проведення навантажувального тестування;
- проведено аналіз моделей навантажувального тестування;
- на основі теоретичних міркувань розроблено алгоритмічне забезпечення для проведення розподіленого процесу навантажувального тестування;

- розроблено новий підхід для проведення розподіленого стрес-тестування додатку шляхом використання моделі акторів.

АНОТАЦІЯ

Дипломна робота на тему «Програмна стрес-система тестування обчислювальних ресурсів асинхронними засобами на основі використання протоколу WebSocket» Яремчук Яна Ігорівна. – Тернопільський національний технічний університет імені Івана Пулюя, Факультет комп'ютерно-інформаційних систем і програмної інженерії, Кафедра програмної інженерії, група СПМ-61 // Тернопіль, 2018.

Метою дипломної роботи є розробка розподіленої системи для проведення стрес-тестування побудованої на технології WebSocket, що призначена для обміну інформацією між клієнтом та веб-сервером в режимі реального часу, щоб визначити продуктивність під час пікових навантажень.

Ключові слова: СТРЕС-ТЕСТУВАННЯ, WEBSOCKET, AJAX, JAVA, МОДЕЛЬ АКТОРІВ, ПРОГРАМНА СИСТЕМА, НАВАНТАЖЕННЯ.

SUMMARY

Diploma work on theme "Software stress testing system of computing resources by asynchronous means using the WebSocket protocol" Yaremchuk Yana Igorivna. - Ternopil National Technical University named after Ivan Puluj, Faculty of Computer Information Systems and Program Engineering, Department of Software Engineering, SP-61 Group // Ternopil, 2018.

The purpose of the thesis is to develop a distributed stress testing system built on WebSocket technology that is designed to exchange information between the client and the web server in real time, to determine performance during peak loadings.

Keywords: STRESS-TESTING, WEBSOCKET, AJAX, JAVA, MODEL OF ACTORS, SOFTWARE, LOADING.