MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA

Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică și Microelectronică
Departamentul Informatică și Ingineria Sistemelor

Admis la susținere

Șef Departament:

Sudacevschi Viorica, conf. univ., dr.

_____

„___”_____2022

Cadru software de testare a performanței sistemelor informatice de tip frontend/backend

Teză de master

Student: **Plamadeala Dumitru, CRI-211M**

Conducător: **Moraru Victor, conf. univ., dr.**

Chișinău, 2022

ADNOTARE

La proiectul de master: „Cadru software de testare a performanței sistemelor informatice de tip frontend/backend”,

elaborat de Plamadeala Dumitru. Chișinău, 2022.

Cuvinte cheie: testarea de performanță, cadru software, analiză și raportare, acord de nivel de servicii, indici de performanță, viteză, stabilitate, scalabilitate, fiabilitate, resurse hardware, backend, frontend, plan de testare, generator de sarcină.

Teza de masterat este dedicată dezvoltării unui cadru software de testare a performanței sistemelor informatice, care permite efectuarea testării backend și frontend cu posibilitate de a monitoriza diverse metrici (disc I/O, consumul de resurse CPU, consumul de memorie RAM, numărul de cereri HTTP/S acceptate/eșuate, timpul de procesare a cererii de către server, latența rețelei, indexul de viteză) și funcții precum generarea de rapoarte în tablouri scriptate moderne, depanarea cererilor eșuate, eșantionarea consumului de memorie, de a scoate în evidență punctele slabe și de a îmbunătăți comportamentul general al aplicațiilor sub utilizare intensă.

Cadrul de testare permite efectuarea testelor de încărcătură cu posibilitatea de analiză detaliată a cauzei erorilor. Pentru backend, testele de încărcătură au fost efectuate cu ajutorul instrumentului Apache JMeter. Pentru frontend, testele au fost efectuate cu ajutorul platformei sitespeed.io și a instrumentului Google Lighthouse. Iar pentru monitorizarea metricilor serverului/aplicației au fost utilizate tablouri de bord scriptate utilizând instrumnetul Grafana. Cadrul software dezvoltat este creat într-un mediu containerizat utilizând platforma Docker, fiecare instrument utilizat poate fi ușor schimbat, prin adăugarea la două fișiere de configurare, fișier docker și fișier docker-compose. Aplicația care a fost aleasă pentru a fi supusă testării este „OpenCart”, un soft de comerț online cu sursă deschisă. Pentru aplicare în practică și o simulare cât mai aproape de realitate, a fost dezvoltată o strategie de testare adecvată și clară, au fost definite și proiectate trei profile operaționale pe baza cărora au fost construite profilele de încărcare. Aceste profile de încărcare reprezintă, simularea unor scenarii realizate de un utilizator sub o anumită sarcină de încărcare. În timpul efectuării testelor de performanță backend/frontend asupra aplicației „OpenCart” care a fost instalată în mediul de lucru intern, nu au fost depistate probleme majore de performanță. Pe durata întregului ciclu de testare au fost luate în calcul caracteristici, cum ar fi distribuția sarcinii, modele de testare, cerințele de disponibilitate, cerințele de rezistență, cerințele de fiabilitate, stiva de tehnologii, numărul de utilizatori concurenți, standarde industriale al timpilor de răspuns și al caracteristicilor mediului testat. Pentru raportare a fost adus un exemplu de raport care include cele mai importante aspecte ale testării de performanță.

Tehnologii utilizate: Docker – folosit pentru a crea, implementa și gestiona containere de aplicații, Grafana pentru vizualizarea datelor și monitorizare, InfluxDB – bază de date pentru serii cronologice de valori și evenimente, Telegraf – agent server pentru colectarea și raportarea indicilor, Jenkins – server de integrare continuă pentru executarea testelor, Portainer – serviciu pentru gestionarea mediului containerizat, Sitespeed.io și Google Lighthouse – set de instrumente pentru testarea încărcării interfeței, Apache JMeter – instrument pentru testarea încărcăturii backend.

ANNOTATION

On the Master thesis "Software framework for testing the performance of frontend/backend IT systems"

elaborated by Plamadeala Dumitru. Chișinău, 2022.

Keywords: performance testing, software framework, analysis and reporting, service level agreement, performance metrics, speed index, stability, scalability, reliability, hardware resources, backend, frontend, test plan, load generator.

The purpose of master's thesis is dedicated to the development of a performance testing framework of information systems, which allows performing backend and frontend testing with the possibility of monitoring various metrics (disk I/O, CPU resource consumption, RAM memory consumption, number of HTTP/S requests accepted/failed, server requests processing time, network latency, speed index) and features such as reporting in modern dashboards, debugging of failed requests, this framework helps in finding of bootlenecks and improve the overall behavior of applications under heavy load.

This performance testing framework have the posibility to identify and analyze failed samples. For backend testing, tests were performed using the Apache JMeter. For the frontend, the tests were performed using the sitespeed.io platform and the Google Lighthouse tool. Grafana scripted dashboards were used for application server metrics monitoring and for general reporting. The developed software framework is created in a containerized environment using the Docker platform, each tool used, can be easily changed by adding two configuration files, docker file and docker-compose file. The application under test is "OpenCart", an open source e-commerce software. An appropriate test strategy was developed for practical and simulation module, three operational profiles were designed, based on which the load profiles were built. These load profiles represent the simulation of a certain scenarios executed by a real user under specific load. The "OpenCart" platform was installed in the internal working environment, no major performance issues were detected while performing backend/frontend performance tests. During the entire test cycle, were taken into account characteristics, such as load distribution, test patterns, availability, endurance, reliability, technology stack, number of concurrent users, industrial standards for the response times. A sample report that includes the most important aspects of performance testing has been provided for reporting.

Technologies which were used: Docker – used to create, deploy and manage virtualized containers, the Grafana tool was used for data visualization and monitoring, InfluxDB – time series database platform for metrics and events, Telegraf – server agent for collecting and reporting metrics, Jenkins – continuous integration server for test execution, Portainer – service for managing docker environment, Sitespeed.io and Google Lighthouse – frontend load testing toolkit, Apache JMeter – tool for backend load testing.

CUPRINS

INTRODUCERE

Din primăvara anului 2020, cuvântul „testare” a căpătat semnificații neașteptate și conotații ambigue aproape peste tot în lume, dar nu și în domeniul tehnologiilor informaționale. În domeniul dat, dezvoltarea unui soft nu este suficient, pe lângă toate el mai trebuie și testat. Fără testare nu este posibil de creat o aplicație calitativă și așa a fost întotdeauna. În zilele noastre, aplicațiile software sunt folosite în toate domeniile vieții, sănătate, agricultură, arhitectură, aviație, constructoare de mașini etc., prin urmare, absența completă a testării software-ului este periculoasă. Este periculos în sensul că poate provoca vătămări grave, cum ar fi o încălcare a securității, pierderea de bani și chiar moartea în unele cazuri. Livrarea sau rularea unei aplicații fără a o testa temeinic va cauza multe probleme mici sau mari pentru utilizatori. Testarea software poate fi reprezentată prin procesul evaluării și verificării unui produs software. Unele dintre scopurile de bază ale testării software ar fi : eliminarea defectelor, reducerea costurilor de dezvoltare, îmbunătățirea diferitor aspecte tehnice și non-tehnice ale aplicație testate precum performanța, experiența utilizatorilor, securitatea și altele. O mare cantitate de testare poate îmbunătăți uimitor calitatea finală a software-ului, ceea ce va aduce la o mare satisfacție cât a clienților atât și a utilizatorilor finali.

Teza de masterat este dedicată dezvoltării unui cadru software de testare a performanței ale sistemelor informatice, care permite efectuarea testării back-end și front-end ale aplicațiilor cu posibilitate de a monitoriza diverse metrici, de a scoate în evidență punctele slabe și de a îmbunătăți comportamentul general al aplicațiilor sub utilizare intensă. Acest sistem de testare ne va mai permite analiza detaliată a cauzei erorilor survenite în urma testării cu detaliere pentru cererile eșuate pe un anumit nod al aplicației, efectuarea testelor de încărcătură pentru evidențierea punctelor tari și a punctelor slabe ale softului cu versiunile sale anterioare pe tablouri de bord scriptate. Pentru back-end testele de încărcătura vor fi efectuate cu ajutorul softului Apache Jmeter. Pentru front-end testele vor fi efectuate cu ajutorul platformei sitespeed.io și Google Lighthouse. Iar pentru monitorizarea a metricilor serverului / aplicației vor fi utilizate tablouri de bord alte softului Grafana. De menționat că toate softurile utilizate pentru dezvoltare sistemului de testare a performanței aplicațiilor sunt cu sursă deschisă.

Testarea performanței este un proces de testare a software-ului utilizat pentru a testa viteza, timpul de răspuns, stabilitatea, fiabilitatea, scalabilitatea și utilizarea resurselor unei aplicații software sub o anumită sarcină de lucru. Scopul principal al testării performanței este de a identifica și elimina blocajele de performanță dintr-o aplicație software.

BIBLIOGRAFIE

1. *Courseware: Centurion University of Technology and Management,* Assessing the Financial Impact of Downtime. [citat 12.09.2022].
   Disponibil : https://courseware.cutm.ac.in/Financial-Impact-of-Downtime
2. WEBB, G. Kent. Predicting Processor Performance. Issues in Information Systems [online]. *Scholarworks*, 2004, pp. 340-346. [citat 13.09.2022]
   Disponibil : https://scholarworks.sjsu.edu/mis_pub/Issues-In-Information-Systems/
3. ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation – System and software quality models.
4. YORKSTON, Keith. *Performance Testing: An ISTQB Certified Tester Foundation Level Specialist Certification Review.* Apress, 2021. p37. ISBN: 978-1-4842-7254-1
5. *IBM: Start performance testing early.* IBM Cloud Garage Method, IBM Corporation. [citat 26.09.2020].
   Disponibil : https://www.ibm.com/method/practices/code/performance_testing/
6. SITARAMAN, M., KULCZYCKI, G., KRONE, J., OGDEN, W.F., REDDY, A.N. *Performance Specification of software Components.* ACM SIGSOFT Software Engineering Notes, 2001, nr. 26, pp. 3-10.
   Disponibil: https://doi.org/10.1145/379377.375223
7. PRADEL, M., SCHUH, P., NECULA, G., SEN, K. *EventBreak: Analyzing the Responsiveness of User Interfaces through Performance-Guided Test Generation*. ACM SIGPLAN Notices, 2014, nr. 49, pp. 33-47.
   Disponibil: https://doi.org/10.1145/2714064.2660233
8. WALLEN, Jack. Portainer, a GUI for Docker Management [online]. *Thenewstack*, 2022. [citat 01.11.2022]
   Disponibil: https://thenewstack.io/portainer-a-gui-for-docker-management/
9. SHARMA, Vivek. Route Flow Of Opencart [online]. *Webkul*, 2017 [citat 03.11.2022]
   Disponibil: https://webkul.com/blog/route-flow-opencart/
10. HAMBLING, Brian. *An ISTQB-BCS Certified Tester Foundation Guide*. BCS Learning & Development Ltd, 2019. ISBN: 978-1-78017-493-8
11. IEEE STANDARD FOR SOFTWARE AND SYSTEM TEST DOCUMENTATION. IEEE 829-2008. Institute of Electrical and Electronics Engineers Inc. ISBN 978-0-7381-5746-7
12. PerfMatrix: UI Performance Testing, © 2022 citat [15.11.2022] Disponibil : https://www.perfmatrix.com/first-paint-ui-metric/