

# **RELIABLE PERFORMANCE METRICS FOR MICROSERVICE- BASED APPLICATION**

This dissertation submitted in partial fulfilment of the requirements for the  
Degree of MSc in Computer Science specialising in Cloud Computing

G.W.G.S.V Amaradasa  
179302N

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

March 2022

**DECLARATION**

I declare that this is my own work and this MSc Thesis Project Report does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning, and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to the University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other media. I retain the right to use this content in whole or part in future works.

***UOM Verified Signature***

**31-03-2021**  
-----

G.W.G.S.V. Amaradasa

Date

The above candidate has carried out research for the Masters under my supervision

Name of the supervisor :

-----

-----

Signature of the supervisor

Date

Name of the supervisor : L C T Silva

***UOM Verified Signature***

**31-03-2021**  
-----

Signature of the supervisor (External)

Date

## **ABSTRACT**

Microservice is the most popular technology nowadays. But unfortunately, performance testing with microservice-based applications was comparatively more challenging than testing. As a result, many studies have focused on the performance testing bottleneck in the Microservice-based application. Furthermore, performance attributes were challenging to measure due to the unique features inherited from microservices, such as auto load balancing auto-provisioning elements.

The research aims to identify the best way to reduce the variation in performance test execution and propose more stable performance metrics that stand more reliable in microservice-based applications.

Performance tests execution uses different load patterns to identify the best way to reduce variations in a different implementation. Stress, endurance, and load tests were used primarily as the load patterns.

Key metrics were identified beforehand and evaluated to determine the most reliable metrics.

The thesis refers to the suggested future works of the existing research. In addition, refer the same analysis to the detailed studies on the performance characteristics of microservice applications.

The research's scope is to identify the possibility of measuring the microservice-based applications' performance and finding the best metrics with each load pattern.

The Benchmark application was TeaStore for the proposed evaluations. Further, the Sock Shop application illustrates the performance aspects comparatively against the TeaStore applications.

The proposed evaluation method used the TeaStore and Sock shop applications deployed on Google Kubernetes Engine and JMeter Scripts.

The analysis evaluates metrics collected, graphs, and other statistics in JMeter and Google Kubernetes dashboard.

## **ACKNOWLEDGEMENTS**

I would like to express my great appreciation to my research supervisor Dr. Adeesha Wijayasiri for his guidance, enthusiastic encouragement, and valuable critiques on this research work. His valuable insight and expertise greatly assisted the research.

I am also grateful to Dr. Dilum Bandara, Dr. Gihan Dias, and Dr. Indika Perera, Dr Sanath Jayasena who initially guided me to start the proper research. Mrs. Chamadenri Silva and Mr. Suren Rodrigo supported me with all the aspects and necessary guidance. I express my sincere thanks to them for their help.

Last but not least, my sincere gratitude to my family for their continuous support and sacrifices. I would like to extend gratitude to my loving husband and mother-in-law, for their love toward my studies by caring for my little son, and my mother's and father's well wishes are with me in whatever I achieve.

# TABLE OF CONTENTS

DECLARATION .....	2
ABSTRACT.....	3
ACKNOWLEDGEMENTS .....	4
TABLE OF CONTENTS .....	5
LIST OF FIGURES.....	9
LIST OF TABLES .....	12
LIST OF ABBREVIATIONS .....	13
1. INTRODUCTION.....	14
1.1. Product Quality of Microservice-Based Application.....	14
1.2. The Problem Statement .....	15
1.3. Objectives .....	17
1.4. Outline .....	17
2. LITERATURE REVIEW .....	18
2.1. The Experimental Setup .....	18
2.2. TeaStore Application .....	19
2.3. TeaStore Application Architecture .....	20
2.4 TeaStore Setup for Experimental Execution .....	21
2.5 Sock Shop application.....	21
2.6 Sock Shop Architecture.....	21
2.7 Sock Shop Setup for Experimental Execution.....	22
2.8. Use of Google Kubernetes Engine Cluster for the Experimental Execution.....	22
2.9. What is Kubernetes? .....	23

2.10. What is Google Kubernetes Engine (GKE)? .....	23
2.11. Google Kubernetes Engine[7] .....	23
2.12. Benefit of Google Kubernetes Engine .....	23
2.13. Key Feature of Google Kubernetes Engine [3].....	23
2.14. Performance Testing Levels .....	25
2.14.2 Client-Server (Mainly used).....	26
2.14.3 Code Level.....	27
2.12. Type of Performance Testing .....	27
2.12.1 Load Test .....	28
2.12.2 Endurance Test .....	28
2.12.3 Stress Test .....	29
2.12.4 Spike Test .....	30
2.12.5 Breakpoint Test.....	31
2.13. Performance Measurements and Matrices .....	31
2.13.1 Measurements .....	31
2.13.2 Metrics .....	32
2.14. Performance Testing Process .....	32
2.15. Microservice- based Application Architecture and Related Researches .....	33
2.16. Performance Testing in Microservices- Based Applications .....	33
2.16. Conclusions Based on Literature Review .....	35
3. METHODOLOGY .....	36
3.1 Chapter Overview .....	36
3.2. Setup Application Under Test .....	36

3.3. Run the TeaStore on a Kubernetes Cluster.....	37
3.4. Run the Sock shop on a Kubernetes Cluster.....	38
3.4.1.The GKE Cluster .....	38
3.4.2.Modes of operations.....	39
3.5. Deployment Steps .....	41
3.6. TeaSotre application UI .....	44
3.6. Sock Shop application UI.....	45
4. IMPLEMENTATION .....	46
4.1. Chapter Overview .....	46
4.2. The Workload Modelling and Execution .....	46
4.3. Performance Metrics for the Evaluation.....	48
4.4. Test Scripts and Execution .....	50
5. EVALUATION.....	51
5.1. Chapter Overview .....	51
5.2 .Evaluation of Load Test Results.....	51
5.3. Resource Utilisation - Load test .....	58
5.2 .Evaluation of Endurance Test Results .....	62
5.5. Comparisons - Load test statistics vs endurance test statistics.....	66
5.6. Stress Test Result and Metrics Evaluation .....	68
5.6.1 Stress Test - TeaStore Observations :.....	68
5.6.1.1 Stress test Response time - TeaStore.....	69
5.6.1.2 Stress Test - Sock Shop - Observations :.....	70
5.6.1.3 Stress test Response time - Sock Shop.....	70

5.7. Overall Resource utilisation ..... 76

6. CONCLUSION ..... 77

6.1. Chapter Overview ..... 77

7. REFERENCES ..... 80

## LIST OF FIGURES

<i>Figure 2-1: TeaStore Application Architecture, Source[2]</i> .....	20
<i>Figure 2-2 : Sock Shop Application Architecture, Source[52]</i> .....	22
<i>Figure 2-3:Performance Testing Levels, Source [8]</i> .....	26
<i>Figure 2-4 :Load Testing Pattern, Source[12]</i> .....	28
<i>Figure 2-5:Endurance Test Pattern, Source[13]</i> .....	28
<i>Figure 2-6:Stress Test Pattern, Source [14]</i> .....	29
<i>Figure 2-7:Spike Testing Patterns, Source [15]</i> .....	30
<i>Figure 2-8:Breakpoint Testing Pattern, Source[16]</i> .....	31
<i>Figure 2-9:Testing Approaches</i> .....	33
<i>Figure 3-1:Overview of GKE Cluster, Source[18]</i> .....	39
<i>Figure 3-2 : Cluster Setup for the experiment</i> .....	40
<i>Figure 3-3 :Cluster configuration</i> .....	40
<i>Figure 3-4 :Node Overview</i> .....	41
<i>Figure 3-5 :Workload tab with TeaStore application</i> .....	42
<i>Figure 3-6 : Deployment details of the TeaStore UI</i> .....	42
<i>Figure 3-7 : Services of the TeaStore application</i> .....	43
<i>Figure 3-8:TeaStore: WebUI Node Port</i> .....	43
<i>Figure 3-9 :TeaStore UI</i> .....	44
<i>Figure 3-10 :Sock Shop UI</i> .....	45

<i>Figure 4-1:Load Testing Setup.....</i>	<i>47</i>
<i>Figure 4-2 :Endurance Testing Setup .....</i>	<i>47</i>
<i>Figure 4-3: Stress Testing Setup .....</i>	<i>48</i>
<i>Figure 4-4 :JMeter script overview.....</i>	<i>50</i>
<i>Figure 5-1 :Response time over time graph.....</i>	<i>53</i>
<i>Figure 5-2 : All key metrics , vs data .....</i>	<i>53</i>
<i>Figure 5-3 : the response time behaviour vs the load.....</i>	<i>54</i>
<i>Figure 5-4 Sock Shop Response Time behaviour.....</i>	<i>55</i>
<i>Figure 5-5:the Throughput behaviour during the load test .....</i>	<i>56</i>
<i>Figure 5-6 :the Throughput behaviour during the load test - Sock Shop .....</i>	<i>58</i>
<i>Figure 5-7:Cluster CPU utilisation, Load Test.....</i>	<i>59</i>
<i>Figure 5-8 : Node level Resource utilisation: Node 1.....</i>	<i>59</i>
<i>Figure 5-9 :Node Level Resource utilisation : Node 2 .....</i>	<i>60</i>
<i>Figure 5-10 : Node Level Resource Utilisation :Node 03.....</i>	<i>60</i>
<i>Figure 5-11:Resource Utilisation of Registry Services: Load Test .....</i>	<i>61</i>
<i>Figure 5-12 :Resource Utilisation of DB Services: Load Test .....</i>	<i>61</i>
<i>Figure 5-13 Response Time Behaviour of Endurance Test.....</i>	<i>63</i>
<i>Figure 5-14 : Endurance Test - Error Log.....</i>	<i>64</i>
<i>Figure 5-15:Overview of Response time : Stress test.....</i>	<i>71</i>
<i>Figure 5-16 :Response Time Over Time .....</i>	<i>72</i>

*Figure 5-17: The active threads graphs..... 73*

*Figure 5-18 :Response Times over Time :After the Test : Stress Test ..... 74*

*Figure 5-19 : Response Times over Time 2 : After the Test : Stress Test ..... 74*

*Figure 5-20 : Error details : Stress Test..... 75*

*Figure 5-21:Throughput over time..... 75*

*Figure 5-22 : Overall usage : Node Level ..... 76*

**LIST OF TABLES**

*Table 5-1: Response time summary in two execution cycles..... 52*

*Table 5-2 : Sock Shop Load Test Execution summary ..... 55*

*Table 5-3: Sock shop Throughput summary..... 56*

*Table 5-4:Average Throughput in Load test of Sock shop..... 57*

*Table 5-5 : Response Time summary for Sock Shop ..... 65*

*Table 5-6 :Throughput Summary - Endurance Test -TeaStore..... 65*

*Table 5-7 :Throughput Summary - Endurance Test - Sock shop ..... 66*

## LIST OF ABBREVIATIONS

<b>Abbreviation</b>	<b>Description</b>
SOA	Service Oriented Architecture
AUT	Application Under Test
GKE	Google Kubernetes Engine
VM	Virtual Machine
JS	Java Scripts
DOM	Document Object Model
DB	Data Base
HIPAA	Health Insurance Portability and Accountability Act
YAML	A recursive acronym for "YAML Aren't Markup Language."
PCIDSS	Payment Card Industry Data Security Standard
RAM	Random-Access Memory
WAR	Web Archive
API	Application Programming Interface
URL	A Uniform Resource Locator
IP	Internet Protocol
IOPS	Input/ Output Operations Per Second
HDD	Hard Disk Drive
SSD	Solid-State Drive
CSS	Cascading Style Sheets
CPU	Central Processing Unit