

ANALYZING AND MODELLING WEB SERVER BASED SYSTEMS

Pasindu Nivanthaka Tennage

188012C

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

July 2020

ANALYZING AND MODELLING WEB SERVER BASED SYSTEMS

Pasindu Nivanthaka Tennage

188012C

Thesis submitted in partial Fulfillment of the Requirements for the Degree Master of
Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

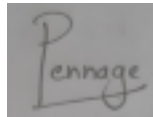
July 2020

Declaration

“I declare that this is my own work and this thesis 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 University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books)”.

Signature:



Date: 2020-06-15

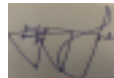
The above candidate has carried out research for the Masters thesis under my supervision.

Signature (Sanath Jayasena):



Date: 2020-06-15

Signature (Malith Jayasinghe):



Date: 2020-06-15

Abstract

Server based systems are widely used in modern computer systems. Understanding the performance of web server based systems, under different conditions is important. This requires a step by step approach that includes modelling, designing, implementing, performance testing and analyzing of results. In this research, we aim at characterizing the web server systems under different configurations. We present a summary of prevalent server architectures, provide a systematic approach for performance testing, and present a novel open source Python library for latency analysis. We experiment on existing server architectures, and propose eight new server architectures. Our analysis shows that under different conditions the new architectures outperform the existing architectures. Moreover we do an extensive tail latency analysis of Java microservices.

Key words: Server architectures, tail index, performance, latency, throughput, web

Acknowledgements

I would like to acknowledge with greatest gratitude the help and guidance I received to conduct this research, from these respected persons. I would like to show my deepest gratitude to project supervisors Professor Sanath Jayasena and Dr Malith Jayasinghe for the thorough guidance and the consistent assistance I received throughout this research. My gratitude goes to Dr Srinath Perera for guiding me in the research through numerous consultations.

Table of Contents

Declaration	i
Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vii
List of Tables	viii
List of Abbreviations	ix
1. INTRODUCTION	1
1.1. Research Problem	3
1.1.1 Motivation and overview	3
1.1.2 Problem statement	3
1.2. Research Objectives	3
2. RELATED WORK	4
2.1 Web Services	4
2.2 Concurrency	6
2.3 Scalability	6
2.4 Web Server Architectures	7
2.5 Message Passing Architectures	8
2.6 Microservices	9
2.7 Summary	9
3. SERVER ARCHITECTURES	10
3.1 Introduction	10
3.2 Client Server Paradigm	10
3.3 Mobile Agents	10
3.4 Service Oriented Architecture	10
3.5 Microservices	11
4. PERFORMANCE ENGINEERING	12
4.1 Introduction	12
4.2 Benchmarks	13
4.3 Workload Generation	14

4.3.1 Real workloads	14
4.3.2 Synthetic workloads	14
4.4 Performance Models	15
4.4.1 Open loop model	15
4.4.2 Closed loop model	15
4.4.3 Half open model	15
4.5 Tools	16
4.6 Performance Measurement	18
4.7 Latency Analysis Methods	18
4.7.1 Average latency	18
4.7.2 Latency percentiles	18
4.7.3 Distribution analysis	19
4.7.4 Theoretical distributions	23
4.7.5 Long tail distribution analysis	26
5. WEB SERVER ARCHITECTURES	33
5.1 Web Server Architectures	33
5.1.1 Thread per request architecture	33
5.1.2 Event driven architecture	34
5.1.3 Staged event driven architecture	36
5.2 Message Passing Architectures	37
5.2.1 Queue	37
5.2.2 Disruptor	37
5.2.3 Actors	39
5.3 Methodology and Implementation	40
5.3.1 Micro benchmark applications	42
5.3.2 Workload generation	42
5.4 Experiment Setup	44
5.5 Results	44
5.6 Discussion	44
5.6.1 Blocking architectures	45
5.6.2 NIO architectures	49
5.6.3 NIO2 architectures	50
5.6.4 SEDA architectures	51
5.7 Summary	51
6. JAVA MICROSERVICES TAIL LATENCY ANALYSIS	53
7. SCALABILITY	54

7.1 Introduction	54
7.2 Amdahl's Law for Software Scalability	55
7.3 Universal Scalability Law for Software Scalability	56
7.4 WSO2 Enterprise Integrator Dataset	57
7.5 Experimental Setup	57
7.6 Results and Discussion	58
7.7 Summary	60
8. DISCRETE EVENT SIMULATION	61
8.1 Introduction	61
8.2 Definitions	61
8.3 SimPy	62
8.3.1 Major concepts	62
8.4 Closed System DES Simulation	63
8.4.1 Client process	65
8.4.2 Server process	65
8.4.3 Results and discussion	66
8.5 Modelling Interservice Calls	68
8.5.1 Results and analysis	70
8.6 Summary	71
9. LOAD BALANCING	72
9.1 Introduction	72
9.2 Definition	72
9.3 Experiment Setup	73
9.4 Results and Discussion	75
9.5 Summary	79
10. CONCLUSION	80
Appendix A: Server Architecture Results	93

List of Figures

Figure 4.1	JMeter Experimental Setup	16
Figure 4.2	Histogram	20
Figure 4.3	Probability Density Function	21
Figure 4.4	Cumulative Distribution Function	22
Figure 4.5	Maximum Likelihood Pareto Fit of Data	24
Figure 4.6	Mass Count Disparity	27
Figure 4.7	Lorenz Curve	29
Figure 4.8	Heavy Tailed Distributions	29
Figure 4.9	LLCD	31
Figure 4.10	Hill Plot	32
Figure 5.1	Thread per Request Class Diagram	34
Figure 5.2	Reactor Pattern	35
Figure 5.3	Proactor Pattern	35
Figure 5.4	SEDA Architecture	36
Figure 5.5	Disruptor Structure	39
Figure 7.1	EI setup	57
Figure 7.2	USL curves	60
Figure 8.1	DES Abstraction	63
Figure 8.2	Single Server Python Code	64
Figure 8.3	Interservice Calls DES Abstraction	68
Figure 8.4	Interservice Calls, Python Code	69
Figure 9.1	Single Service	74
Figure 9.2	Two Services	74
Figure 9.3	Three Services	74

List of Tables

Table 4.1	Workload Generation Tools	17
Table 5.1	Mechanical Sympathy	38
Table 5.2	Server Architectures	40
Table 7.1	Universal Law of Scalability Performance Results	58
Table 7.2	USL Parameters	59
Table 8.1	Closed System DES Results	66
Table 8.2	Interservice calls DES results	70
Table 9.1	Hardware Configurations	75
Table 9.2	Load Balancing Results	76

List of Abbreviations

Abbreviation	Description
CCDF	Complementary Cumulative Distribution Function
CDF	Cumulative Distribution Function
DES	Discrete Event Simulation
HTTP	HyperText Transfer Protocol
LLCD	Log Log Complementary Graphs
PDF	Probability Density Function
REST	Representational State Transfer
RPC	Remote Procedure Call
SEDA	Staged Event-driven Architecture
SOAP	Simple Object Access Protocol
UDDI	Universal Description, Discovery, and Integration
URI	Uniform Resource Identifier
WSDL	Web Services Description Language
XML-RPC	XML Based RPC