

**HYBRID DYNAMIC LOAD BALANCING ALGORITHM FOR CLOUD  
PERFORMANCE IMPROVEMENT**

**OBIASI, OGHORCHUKWUYEM ORIEKOSE  
(20PCJ02084)**

**JANUARY, 2023**

**HYBRID DYNAMIC LOAD BALANCING ALGORITHM FOR CLOUD  
PERFORMANCE IMPROVEMENT**

**BY**

**OBIAZI, OGHORCHUKWUYEM ORIEKOSE  
(20PCJ02084)**

**B.Eng Computer Engineering, University of Benin, Benin**

**A DISSERTATION SUBMITTED TO THE SCHOOL OF  
POSTGRADUATE STUDIES IN PARTIAL FULFILLMENT OF THE  
REQUIREMENTS FOR THE AWARD OF MASTER OF ENGINEERING  
(M.Eng.) DEGREE IN COMPUTER ENGINEERING IN THE  
DEPARTMENT OF ELECTRICAL AND INFORMATION  
ENGINEERING, COLLEGE OF ENGINEERING, COVENANT  
UNIVERSITY, OTA, OGUN STATE**

**JANUARY, 2023**

## **ACCEPTANCE**

This is to attest that this dissertation is accepted in partial fulfilment of the requirements for the award of the degree of Master of Engineering in Computer Engineering in the Department of Electrical and Information Engineering, College of Engineering, Covenant University, Ota, Nigeria.

**Miss Adefunke F. Oyinloye**  
**(Secretary, School of Postgraduate Studies)**

**Signature and Date**

**Prof. Akan B. Williams**  
**(Dean, School of Postgraduate Studies)**

**Signature and Date**

## **DECLARATION**

**I, OBIAZI, OGHORCHUKWUYEM ORIEKOSE (20PCJ02084)** declare that this dissertation is a representation of my work and is written and implemented by me under the supervision of Dr. Adewale, Adeyinka A. of the Department of Electrical and Information Engineering, Covenant University. I attest that this dissertation has in no way been submitted either wholly or partially to any other university or institution of higher learning for the award of a masters' degree. All information cited from published and unpublished literature has been duly referenced.

**OBIAZI, OGHORCHUKWUYEM ORIEKOSE**



12/01/2023

**Signature and Date**

## CERTIFICATION

This is to certify that this dissertation titled “**HYBRID DYNAMIC LOAD BALANCING ALGORITHM FOR CLOUD PERFORMANCE IMPROVEMENT**” is an original research work carried out by **OBIAZI, OGHORCHUKWUYEM ORIEKOSE** meets the requirements and regulations governing the award of Master of Engineering (M.Eng) degree in Computer Engineering from the Department of Electrical and Information Engineering, College of Engineering, Covenant University, Ota, and is approved for its contributions to knowledge and literary presentation.

**Dr. Adewale Adeyinka A.**  
(Supervisor)



30/01/2023

**Signature and Date**

**Prof. Emmanuel Adetiba**  
(Head of Department)

**Signature and Date**

**Prof. Jonathan Gana Kolo**  
(External Examiner)

**Signature and Date**

**Prof. Akan B. Williams**  
(Dean, School of Postgraduate Studies)

**Signature and Date**

## **DEDICATION**

This dissertation is dedicated to God. His consuming love, favour, and grace were especially evident throughout this research. To my parents, Engr. Prof. And Dr. Mrs. Obiazi, Thank you for everything. For the encouragement, the love, the support, the advice. Thank you for going above and beyond the standard definition of parenthood. To my husband Peter Idoko Alexander, Thanks for being my strength throughout this research.

## ACKNOWLEDGEMENTS

I want to thank God for his divine direction and the mental and physical fortitude He granted me over the course of writing this research dissertation.

My deepest gratitude is extended to Dr. David O. Oyedepo, Chancellor of Covenant University, whose comments have served as a continual reminder that anything is doable and attainable if you set your mind to it. A special thanks to Covenant University's Vice-Chancellor, Professor Abiodun H. Adebayo, Dean of the School of Postgraduate Studies, Professor Akan Williams, the Dean of the College of Engineering, Prof David O. Olukanni as well as the Head of the Electrical and Information Engineering Department, Professor Emmanuel Adetiba, for their unwavering efforts in realizing the university's goal of ranking among the top ten universities worldwide by the year 2022. God the Almighty be with you always in blessing and favour.

My sincere gratitude to Dr. Adewale Adeyinka A., my wonderful supervisor, for his generosity, support, guidance, and counsel throughout this research project. I owe him a great deal for his advice on the dissertation and for providing helpful guidance during different phases of my work. I am thankful for the constant cloud computing and AI resources he graciously shared with me. My academic thinking, problem-solving skills, and technical writing have all improved thanks to him, and they will be instrumental in my future career and research. My deep appreciation goes to the postgraduate Coordinator for the department of Electrical and Information Engineering, Dr. Kennedy Okokpujie. Thank you for your deep interest in the welfare of the students and how you constantly try to guide us. I do not have enough words to say thank you. Additionally, a big thank you to the entire academic staff of the electrical and information engineering department. May the Almighty pay you all back for your contributions to my life.

My sincere thanks go to my parents, Engr. Prof. & Dr. Obiazi, who have always been there for me, prayed for me, supported me in every way they could, and provided me with their wise counsel. Words cannot adequately express my gratitude to you.

Finally, I want to thank my friends and colleagues, Mrs. Tola Omotoye, Mr. Kennedy Ekong, Mr. Andrew Omiloli, and Mr. Emmanuel Mbaya, for their support and contributions throughout my postgraduate studies. To my friends, Ann Anyanwu and Eniola Akinwunmi, Thank you for the support throughout the duration of this project.

# TABLE OF CONTENT

<b>ACCEPTANCE</b>	<b>iii</b>
<b>DECLARATION</b>	<b>iv</b>
<b>CERTIFICATION</b>	<b>v</b>
<b>DEDICATION</b>	<b>vi</b>
<b>ACKNOWLEDGEMENTS</b>	<b>vii</b>
<b>TABLE OF CONTENT</b>	<b>viii</b>
<b>LIST OF FIGURES</b>	<b>xi</b>
<b>LIST OF TABLES</b>	<b>xii</b>
<b>LIST OF ABBREVIATIONS</b>	<b>xiii</b>
<b>ABSTRACT</b>	<b>xv</b>
<b>CHAPTER ONE</b>	<b>1</b>
<b>INTRODUCTION</b>	<b>1</b>
1.1 Introduction	1
1.2 Background of Study	2
1.2.1 Cloud Computing Overview	2
1.2.2 Virtual machines	5
1.2.3 Load Balancing	6
1.3 Statement of the Problem	8
1.4 Aim and Objectives	9
1.4.1 Aim	9
1.4.2 Objectives	9
1.5 Scope of Study	10
1.6 Limitation of Study	10
1.7 Justification for Study	10
1.8 Dissertation Organization	11
<b>CHAPTER TWO</b>	<b>12</b>
<b>LITERATURE REVIEW</b>	<b>12</b>
2.1 Preamble	12
2.2 Cloud Computing	12
2.2.1 Cloud Computing Service Models	14
2.2.2 Cloud Deployment Models	14
2.3 Virtualization in Cloud Computing	17
2.3.1 Approaches to Virtualization in Cloud Computing	17
2.3.2 Virtual Machines	18



2.4	Load Balancing	19
2.4.1	Load Balancing Metrics	21
2.5	Load Balancing Classifications	23
2.5.1	Static Load Balancing Algorithm	23
2.5.2	Dynamic Load Balancing Algorithm	24
2.5.3	Challenges Identified in Cloud Load balancing	26
2.6	Review of Previous Work Done	26
2.6.1	Round-Robin Load Balancing Algorithm	26
2.6.2	Weighted Round-Robin Load Balancing Algorithm	28
2.6.3	Min-Min Load Balancing Algorithm	30
2.6.4	First Come First Serve Load Balancing Algorithm	31
2.6.5	Shortest Job First Load Balancing Algorithm	32
2.6.6	Ant Colony Optimization	33
2.6.7	Honey Bee Foraging Load Balancing algorithm	35
2.6.8	Q-Learning	37
2.7	Critique of Previous work	39
2.7.1	Q-learning Algorithm	40
2.7.2	Honeybee Foraging LBA	43
2.8	Summary of Some Related Works	46
2.9	Chapter Summary	53
<b>CHAPTER THREE</b>		<b>54</b>
<b>METHODOLOGY</b>		<b>54</b>
3.1	Preamble	54
3.2	Cloud Computing Environment	54
3.2.1	Cloudlets	56
3.2.2	Virtual Machines (VMs)	56
3.2.3	Data Center	57
3.3	CloudSim Framework	58
3.3.1	Scheduling in CloudSim	60
3.3.2	Cloudlet scheduling	60
3.3.3	Cloudlet Scheduler Space Shared	60
3.4	Novelty of research	60
3.5	Problem Formulation	61
3.6	Design of a hybrid Load balancing algorithm	63

3.7	Implementation Resources	68
3.8	Method of Evaluation	70
3.9	Evaluation experiments	71
3.9.1	Experiment one	71
3.9.2	Experiment two	71
3.9.3	Experiment three	72
3.9.4	Experiment Specifications.	72
3.10	Chapter Summary	74
<b>CHAPTER FOUR</b>		<b>75</b>
<b>RESULTS AND DISCUSSION</b>		<b>75</b>
4.1	Preamble	75
4.2	Performance metrics of the algorithm with varying tasks	75
4.2.1	Results and analysis for experiment one	77
4.3	Performance metrics of the algorithm with varying VMs	77
4.3.1	Results and analysis for experiment two	79
4.4	Performance metrics of the algorithm with constant tasks and VMs	80
4.4.1	Results and analysis for experiment 3	82
4.5	Discussion	82
4.6	Chapter Summary	83
<b>CHAPTER FIVE</b>		<b>84</b>
<b>CONCLUSION AND RECOMMENDATION</b>		<b>84</b>
5.1	Summary	84
5.2	Conclusion	84
5.3	Contribution to knowledge	85
5.4	Strengths and Weaknesses of the HBFQL algorithm	85
5.4.1	Strengths of algorithm	85
5.4.2	Weakness of the algorithm	85
5.5	Recommendation	86
<b>REFERENCES</b>		<b>87</b>
<b>APPENDIX</b>		<b>94</b>

## LIST OF FIGURES

FIGURES	LIST OF FIGURES	PAGES
Figure 1.1:	Cloud Computing Overview	3
Figure 1.2	Cloud Deployment Models	4
Figure 1.3:	Virtual Machines (Masdari et al., 2016)	6
Figure 1.4:	Load Balancing in Cloud Computing (Afzal & Kavitha, 2019)	8
Figure 2.1:	Cloud Service Models (Mishra et al., 2020)	16
Figure 2.2:	Cloud Deployment Models	16
Figure 2.3:	Virtual Machine (Han et al., 2016)	19
Figure 2.4:	Load Balancing Model (Jafarnejad Ghomi et al., 2017)	21
Figure 2.5:	Round Robin algorithm	28
Figure 2.6:	Weighted Round Robin	29
Figure 2.7:	Ant Colony Optimization LBA (Shafiq et al., 2021)	34
Figure 3.1:	Load balancing overview	55
Figure 3.2:	The cloud sim framework (Sankaran et al., 2021)	59
Figure 3.3:	Q-learning algorithm block diagram	63
Figure 3.4:	HBF algorithm block diagram	64
Figure 3.6:	Proposed HBFQL algorithm block diagram	65
Figure 3.5:	Architectural model for HBFQL algorithm	67
Figure 3.7:	HBFQL Algorithm Flowchart	68
Figure 4.1:	Throughput result for experiment one	75
Figure 4.2:	Makespan from experiment one	76
Figure 4.3:	Runtime results for experiment one	76
Figure 4.4:	Throughput for experiment two	78
Figure 4.5:	Makespan from experiment two	78
Figure 4.6:	Runtime of different algorithms in experiment two	79
Figure 4.7:	Throughput in experiment three	80
Figure 4.8:	Experiment three makespan	81
Figure 4.9:	Algorithm runtime in experiment three	81

## LIST OF TABLES

<b>TABLES</b>	<b>LIST OF TABLES</b>	<b>PAGES</b>
Table 2.1	Characteristics of cloud computing	13
Table 2.2:	Comparison between Static and Dynamic Load Balancing Algorithms	25
Table 2.3:	Summary of Some Related Works.	46
Table 3.1:	CloudSim system requirements	59
Table 3.2:	Specifications of system used	69
Table 3.3:	Java files in CloudSim modified to implement the developed load balancing algorithm	69
Table 3.4:	Algorithms used for evaluation	70
Table 3.5:	Simulated Data Centre specification	72
Table 3.6:	Simulated VM specification	73
Table 3.7:	Cloudlet specification	73
Table A1:	Exp (1) Throughput of HBFQL compared with SJF and ACO	94
Table A2:	Exp (1) Makespan of HBFQL compared with SJF and ACO	94
Table A3:	Exp (1) Runtime of HBFQL compared with SJF and ACO	95
Table A4:	Exp (2) Throughput of HBFQL compared with SJF and ACO	95
Table A5:	Exp (2) Makespan of HBFQL compared with SJF and ACO	96
Table A6:	Exp (3) Runtime of HBFQL compared with SJF and ACO	96
Table A7:	Exp (3) Throughput of HBFQL compared with SJF and ACO	97
Table A8:	Exp (3) Makespan of HBFQL compared with SJF and ACO	97
Table A9:	Exp (3) Runtime of HBFQL compared with SJF and ACO	97

## LIST OF ABBREVIATIONS

Ant Colony Optimization (ACO)

Application Specific Integrated Circuits (ASICs)

Artificial intelligence (AI)

Client Service Side (CSS)

Cloud Service Provider (CSP)

Data Centre (DC)

Deep Q-Learning Task Scheduling (DQTS)

First Come First Serve (FCFS)

Genetic Algorithm (GA)

Honey Bee Foraging (HBF)

Information and communication technology (ICT)

Infrastructure-as-a-Service (IAAS)

Load Balancing as a Service (LBaaS)

Machine learning (ML)

Max-Min Scheduling Algorithm (MMSIA)

Million Instructions Per Second (MIPS)

Millions of Instructions (MI)

Opportunistic Load Balancing (OLB)

Physical Machine (PM)

Platform As a Service (PAAS)

Q-learning based task scheduling framework for energy-efficient cloud computing (QEEC)

Quality of Service (QoS)

Round Robin (RR)

Service Level Agreements (SLAs)

Shortest Job First - Extreme Learning Machine (SJF-ELM)

Shortest Job First (SJF)

Software As a Service (SAAS).

Virtual Machine (VM)

Virtual Passive Optical Network (VPON)

Virtual Private Networks (VPN)

Weighted Round Robin (WRR)

## ABSTRACT

Cloud computing is a modern robust approach, enabling individuals and businesses to buy the services they need per their demands over the internet. It offers a wide range of amenities such as easy access to online applications and services, storage, deployment platforms, and much more. Load balancing is a crucial component of cloud computing, and it prevents the overburdening of nodes while others are idle or underutilized. Maintaining the Quality of Service (QoS) parameters can be difficult for cloud providers when equal workload distribution across servers is a challenge. An effective Load Balancing (LB) approach should enhance and provide a high level of customer satisfaction by effectively utilizing Virtual Machines across servers. Even though load balancing algorithms (LBA) have been the subject of much research, efforts to decrease runtime, makespan, and boost throughput have not yielded satisfactory results. Through the hybridization of a dynamic load balancing algorithm and a machine learning algorithm, this research intends to decrease the runtime of load balancing activities, decrease makespan, and boost task throughput in a cloud computing environment. This study combines the Q-learning algorithm with the Honeybee Foraging Load Balancing Algorithm (HBF-LBA). The proposed Honeybee Foraging Q-Learning algorithm (HBFQL) was implemented in the CloudSim simulation environment. The suggested solution successfully decreased runtime by 13.1% and makespan by 8.95% while enhancing throughput by 8.37% during routing operations compared to the Shortest Job First (SJF) algorithm. Compared with the Ant Colony Optimization (ACO), the proposed algorithm reduced runtime by 14.57% and makespan by 13.71% while increasing throughput by 3.43%. This research improved task execution speed by continually monitoring the virtual machine usage history to route tasks to the best available virtual machine and ensure effective task distribution.

***Keywords: Cloud Computing, Load balancing, Virtual Machines, Honeybee Foraging, Q-learning***