A New Priority Rule Cloud Scheduling Technique That Utilizes Gaps to Increase The Efficiency of Jobs Distribution.

SAYDUL AKBAR MURAD

MASTER OF SCIENCE

UNIVERSITI MALAYSIA PAHANG
SUPERVISOR’S DECLARATION

I hereby declare that I have checked this thesis, and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science.

(Supervisor’s Signature)

Name of Supervisor: Dr. Zafril Rizal Bin M Azmi
Position: Senior Lecturer
Date: 07/07/2023
STUDENTS’S DECLARATION

I declare that this thesis entitled “A New Priority Rule Cloud Scheduling Technique That Utilizes Gaps to Increase The Efficiency of Jobs Distribution” is the result of my own re-search except as cited in the reference. The thesis has not been accepted for any degree and is not concurrently submitted in the candidature of any other degree

__________________________

Murad

(Student’s Signature)

Name: Saydul Akbar Murad
ID number: MCN21001
Date: 07/07/2023
A New Priority Rule Cloud Scheduling Technique That Utilizes Gaps to Increase
The Efficiency of Jobs Distribution

Saydul Akbar Murad

Report submitted in fulfillment of the requirements for the Viva of
Master of Science

Faculty of Computing
UNIVERSITI MALAYSIA PAHANG

July 2023
ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep thanks to ALLAH Almighty for the grace he blessed me throughout my life.

I am grateful and would like to express my sincere gratitude to my supervisor Dr. Zafril Rizal Bin M Azmi, for his ideas, invaluable guidance, and constant support in making this viva possible. He has always impressed me with his outstanding professional conduct and his belief in me to successfully make the report.

My sincere thanks go to all the staff of the Faculty of Computing, UMP, who helped me in many ways and made my stay at UMP pleasant and unforgettable.
ABSTRAK

Dalam beberapa tahun terakhir ini, konsep komputasi awan telah semakin menda-
pat perhatian untuk menyediakan akses yang meningkat secara dinamik kepada sumber
daya komputasi bersama (perisian dan perkakas) melalui internet. Tidak dirahsia-
lagi bahawa keupayaan komputasi awan untuk menyediakan perkhidmatan yang penting
bagi sesuatu misi telah menjadikan penjadualan kerja sebagai topik panas dalam industri
pada masa ini. Walau bagaimanapun, penggunaan sumber daya awan ini secara efisien
telah menjadi cabaran, sering kali mengakibatkan pembaziran atau prestasi perkhidmatan
yang terjejas disebabkan penjadualan yang lemah. Bagi menyelesaikan isu ini, penyel-
idikan sedia ada telah memberi tumpuan kepada teknik penjadualan kerja berdasarkan
antinan, di mana kerja-kerja dijadualkan berdasarkan tarikh akhir atau panjang kerja yang
tertentu. Bagi mengatasi cabaran ini, ramai penyelidik telah memberi tumpuan kepada
peningkatan penjadual awan Peraturan Keutamaan (PR) sedia ada dengan membangunkan
algoritma penjadualan dinamik, tetapi mereka belum dapat memenuhi kepuasan pengguna
seperti masa aliran kerja, masa penyelesaian keseluruhan, dan masa kelewatan. Ini adalah
batasan pelaksanaan semasa penjadualan berdasarkan Peraturan Keutamaan (PR) yang sedia ada, terutama
disebabkan oleh halangan yang disebabkan oleh kerja-kerja di bahagian hadapan antinan. Batasan-batasan ini mengakibatkan prestasi
yang lemah dalam aplikasi mudah alih berdasarkan awan dan perkhidmatan awan lain.
Bagi menangani isu ini, objektif utama penyelidikan ini adalah untuk meningkatkan pen-
jadual awan PR sedia ada dengan membangunkan algoritma penjadualan dinamik baru
dengan memanipulasi kesenjangan dalam jadual kerja kerja-kerja awan. Dalam tesis ini,
pertamanya diperkenalkan algoritma Penjadualan Adil Berdasarkan Keutamaan (PBFS)
untuk menjadualkan kerja-kerja supaya kerja-kerja mendapat akses kepada sumber daya
yang diperlukan pada waktu yang optimal. Kemudian, satu strategi pengisian semula yang
dipanggil Penjadualan Adil Berdasarkan Keutamaan Kesenjangan Terpendek (SG-PBFS)
dicadangkan yang cuba memanipulasi kesenjangan dalam jadual kerja kerja-kerja awan.
Akhirnya, penilaian prestasi menunjukkan bahawa algoritma SG-PBFS yang dicadangkan
mengungguli SG-SJF, SG-LIF, SG-FCFS, SG-EDF, dan SG-(MAX-MIN) dari segi masa
aliran kerja, masa penyelesaian keseluruhan, dan masa kelewatan yang dengan tegas mem-
buktikan keberkesanannya. Keputusan eksperimen menunjukkan bahawa bagi 500 kerja,
SG-PBFS masa aliran kerja, masa penyelesaian keseluruhan, dan masa kelewatan adalah
9%, 4%, dan 7% lebih rendah.
ABSTRACT

In recent years, the concept of cloud computing has been gaining traction to provide dynamically increasing access to shared computing resources (software and hardware) via the internet. It’s no secret that cloud computing’s ability to supply mission-critical services has made job scheduling a hot subject in the industry right now. However, the efficient utilization of these cloud resources has been a challenge, often resulting in wastage or degraded service performance due to poor scheduling. To solve this issue, existing research has been focused on queue-based job scheduling techniques, where jobs are scheduled based on specific deadlines or job lengths. To overcome this challenge, numerous researchers have focused on improving existing Priority Rule (PR) cloud schedulers by developing dynamic scheduling algorithms, but they have fallen short of meeting user satisfaction, such as flowtime, makespan, and total tardiness. These are the limitations of the current implementation of existing Priority Rule (PR) schedulers, mainly caused by blocking made by jobs at the head of the queue. These limitations lead to the poor performance of cloud-based mobile applications and other cloud services. To address this issue, the main objective of this research is to improve the existing PR cloud schedulers by developing a new dynamic scheduling algorithm by manipulating the gaps in the cloud job schedule. In this thesis, first a Priority-Based Fair Scheduling (PBFS) algorithm has been introduced to schedule jobs so that jobs get access to the required resources at optimal times. Then, a backfilling strategy called Shortest Gap Priority-Based Fair Scheduling (SG-PBFS) is proposed that attempts to manipulate the gaps in the schedule of cloud jobs. Finally, the performance evaluation demonstrates that the proposed SG-PBFS algorithm outperforms SG-SJF, SG-LJF, SG-FCFS, SG-EDF, and SG-(MAX-MIN) in terms of flow time, makespan time, and total tardiness, which conclusively demonstrates its effectiveness. The experiment result shows that for 500 jobs, SG-PBFS flow time, makespan time, and tardiness time are 9%, 4%, and 7% less than PBFS gradually.
# TABLE OF CONTENTS

## DECLARATION

## TITLE PAGE

## ACKNOWLEDGEMENTS

## ABSTRAK

## ABSTRACT

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ABBREVIATIONS

## LIST OF APPENDICES

### CHAPTER 1 INTRODUCTION

1.1 Overview

1.2 Problem Statement

  1.2.1 Priority Rules Scheduler in Cloud Computing

  1.2.2 Gap Filling Technique in Cloud Computing

1.3 Hypotheses

1.4 Research Question

1.5 Research Objectives

1.6 Research Scopes

1.7 Research Contributions

1.8 Thesis Organization

### CHAPTER 2 LITERATURE REVIEW

2.1 Overview of JST in Cloud Computing

  2.1.1 User

iv
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1.2 Submitted Job</td>
<td>11</td>
</tr>
<tr>
<td>2.1.3 Resource Management</td>
<td>11</td>
</tr>
<tr>
<td>2.1.4 Cloud Information Service (CIS)</td>
<td>14</td>
</tr>
<tr>
<td>2.1.5 Datacenter</td>
<td>14</td>
</tr>
<tr>
<td>2.2 Taxonomy of JST in Cloud Computing</td>
<td>15</td>
</tr>
<tr>
<td>2.2.1 Allocation of Resources</td>
<td>15</td>
</tr>
<tr>
<td>2.2.2 Task Scheduling</td>
<td>25</td>
</tr>
<tr>
<td>2.2.3 Resource Mapping</td>
<td>40</td>
</tr>
<tr>
<td>2.3 Summary</td>
<td>42</td>
</tr>
<tr>
<td>CHAPTER 3 METHODOLOGY</td>
<td>43</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>43</td>
</tr>
<tr>
<td>3.2 Operational Framework</td>
<td>43</td>
</tr>
<tr>
<td>3.3 Research Design and Procedure</td>
<td>43</td>
</tr>
<tr>
<td>3.3.1 Proposed Priority Rule Design in Cloud Computing</td>
<td>44</td>
</tr>
<tr>
<td>3.3.2 Proposed Backfilling Technique (SG-PBFS) Algorithm</td>
<td>48</td>
</tr>
<tr>
<td>3.3.3 Gap Searching</td>
<td>52</td>
</tr>
<tr>
<td>3.4 Performance Evaluation</td>
<td>57</td>
</tr>
<tr>
<td>3.5 Time Complexity</td>
<td>60</td>
</tr>
<tr>
<td>3.6 Experimental Setup</td>
<td>60</td>
</tr>
<tr>
<td>3.6.1 Prototype development</td>
<td>63</td>
</tr>
<tr>
<td>3.6.2 Environmental setup</td>
<td>64</td>
</tr>
<tr>
<td>3.7 Summary</td>
<td>69</td>
</tr>
<tr>
<td>CHAPTER 4 RESULT AND DISCUSSION</td>
<td>71</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>71</td>
</tr>
<tr>
<td>4.1.1 Experimental result of proposed PBFS, EG-PBFS and SG-PBFS using low job load</td>
<td>71</td>
</tr>
<tr>
<td>4.1.2 Experimental result of proposed PBFS, EG-PBFS and SG-PBFS using median job load</td>
<td>78</td>
</tr>
</tbody>
</table>
4.1.3 Experimental result of proposed PBFS, EG-PBFS and SG-PBFS using higher number job load 84
4.1.4 Performance Comparison between PBFS, EG-PBFS and SG-PBFS 89

CHAPTER 5 CONCLUSION AND FUTURE WORK 94
5.1 Introduction 94
5.2 Research Summary 94
5.3 Future Directions and Research Opportunities 95

REFERENCES 97

APPENDICES 115
REFERENCES


Alkayal, E. S., Jennings, N. R., & Abulkhair, M. F. (2016). Efficient task scheduling
multi-objective particle swarm optimization in cloud computing. In 2016 ieee 41st conference on local computer networks workshops (lcn workshops) (pp. 17–24).


Lavanya, M., Shanthi, B., & Saravanan, S. (2020). Multi objective task scheduling algo-


Mambretti, J., Chen, J., & Yeh, F. (2015). Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn). In *2015 international conference on cloud computing research and innovation (icccri)* (pp. 73–79).


Praveenchandar, J., & Tamilarasi, A. (2021b). Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing. *Jour-


Wang, C.-F., Hung, W.-Y., & Yang, C.-S. (2014). A prediction based energy conserv-
ing resources allocation scheme for cloud computing. In *2014 ieee international conference on granular computing (grc)* (pp. 320–324).


