Priority Based Fair Scheduling: Enhancing Efficiency in Cloud Job Distribution

Saydul Akbar Murad
Faculty of Computing, College of Computing & Applied Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah,
Pahang, Malaysia

Zafril Rizal M Azmi*
Faculty of Computing, College of Computing & Applied Sciences, Universiti Malaysia Pahang Al-Sultan Abdullah,
Pahang, Malaysia

Faria Jerin Brishti
Computer Science and Engineering, International Islamic University
Chittagong, Chattogram, Bangladesh

Md Saib
South China University of Technology
Guangzhou, China

Anupam Kumar Bairagi
Khulna University
Khulna, Bangladesh

2023 IEEE 8th International Conference On Software Engineering and Computer Systems (ICSECS) | 979-8-3503-1093-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICSECS58457.2023.10256273

***Abstract*—In recent years, there has been a growing interest in cloud computing as a means to enhance user access to shared computing resources, including software and hardware, through the internet. However, the efficient u t ilization o f t h ese cloud resources has been a challenge, often resulting in wastage or degraded service performance due to inadequate scheduling. 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. In this study, we introduce a new PR scheduler called Priority Based Fair Scheduling (PBFS), which takes into account key parameters such as CPU Time, Job Arrival Time, and Job Length. We evaluate the performance of PBFS by comparing it with five e x isting a l gorithms, a n d the results demonstrate that PBFS surpasses the performance of the other algorithms. The experiment was conducted using the CloudSim simulator, utilizing a dataset of 300 and 400 jobs. In order to assess the performance, three key metrics were employed: flow t i me, m a kespan t i me, a n d t o tal t a rdiness. These metrics were chosen to evaluate and analyze the effectiveness of the proposed scheduling algorithm.**

***Keywords*—Cloud Computing, Job Scheduling, Priority Rule, Makespan Time, Flow Time, PBFS.**

I. Introduction

The main goal of cloud computing is to make it easy to use resources that are far away or in different places. Scheduling is one of the issues that have arisen as the cloud environment evolves and encounters new challenges. The job allocation problem is described as scheduling jobs and resources such that users can complete their jobs in a short amount of time (before the deadline) and at a cheap cost, while also increasing user and cloud resource vendor throughput satisfaction in a cloud environment [1]. In a cloud context, there is currently no established norm for job scheduling. Distributed system job scheduling has been the subject of numerous in-depth research.

The problem with current job scheduling techniques, such as in [2]–[4], is that they only focus on the power consumption of jobs in cloud systems, but they never consider any performance matrices like makespan time or flow t i me. T h is i s because the jobs mentioned in the above references used PR(Priority Rules) such as FCFS (First Come First Serve) and SJF (Shortest Job First) which is proven to be not optimized. The most significant criteria for job scheduling in cloud computing are tardiness time, makespan time, and flow time. Makespan time defines the total amount of time taken by the machine to accomplish all jobs. If the makespan time is higher, it indicates the poor performance of this algorithm [5]. Flow time, on the other hand, is the overall time required to go from one operation to the next, including any time spent waiting for equipment or job orders to arrive, as well as any time wasted due to machine breakdowns, process delays, or component shortages. The lower the flow time, the better the algorithm's performance. A delayed job indicates that the job was not completed on time due to poor scheduling. Some researchers in existing research try to reduce the makespan time but do not care about flow time, while others are only concerned with flow time and never mention total tardiness. Therefore, to improve the performance of PR in terms of makespan time, flow time, and total tardiness, the development of a high-performance job scheduler is required.

In order to solve the above challenges, this research proposed a PR algorithm called Priority-based Fair Scheduling (PBFS). This proposed algorithm improves the performance of makespan time, flow time, and total tardiness. To measure the performance of the proposed algorithm, we have utilized an existing dataset. This research presents diverse fundamental contributions that can be summarized as follows:

- Firstly, a priority-based fair scheduling (PBFS) algorithm has been proposed to increase the efficiency of job scheduling.
- Secondly, this research improved the performance of makespan time, flow time, and total tardiness.
- Thirdly, compare the performance of the proposed PBFS with five existing algorithms.

The structure of this research paper is as follows: Section 2 presents a comprehensive literature review, examining previous studies relevant to the topic. In Section 3, the research methodology employed to achieve the research objectives is outlined. The results and discussion are presented in Section 4, where the performance of the proposed priority algorithm is assessed. Finally, Section 5 concludes the paper, summarizing the findings and offering insights based on the obtained results.

The authors would like to thank Universiti Malaysia Pahang Al-Sultan Abdullah for their financial support through UMPSA Tabung Persidangan Dalam Negara (TPDN) and research fund RDU222415..