SG-PBFS : Shortest Gap-Priority Based Fair Scheduling technique for job scheduling in cloud environment

Murad, Saydul Akbar[a, d]; Azmi, Zafril Rizal M.[a]; Muzahid, Abu Jafar Md[a]; Bhuiyan, MD. Khairul Bashar[b]; Saib, Md[c]; Rahimi, Nick[d]; Prottasha, Nusrat Jahan[e]; Bairagi, Anupam Kumar[f]

[a] Faculty of Computing, College of Computing & Applied Sciences, Universiti Malaysia Pahang, Pahang, Pekan, 26600, Malaysia
[b] Brac University, Dhaka, 1212, Bangladesh
[c] South China University of Technology, Guangzhou, 510006, China
[d] School of Computing Sciences & Computer Engineering, University of Southern Mississippi, Hattiesburg, 39401, United States
[e] Stevens Institute of Technology, 07030, NJ, United States
[f] Khulna University, New Khulna, 9208, Bangladesh

ABSTRACT

Job scheduling in cloud computing plays a crucial role in optimizing resource utilization and ensuring efficient job allocation. But cloud resources may be wasted, or service performance may suffer because of under-utilization or over-utilization because of poor scheduling. Existing approaches often overlook the dynamic nature of cloud environments, resulting in issues like missed deadlines, prolonged flowtime, extended makespan, and unacceptable total tardiness. To address this issue, the main objective of this research is to improve the existing Priority Rules (PR) cloud schedulers by developing a new dynamic scheduling algorithm by manipulating the gaps in the cloud job schedule. Firstly, a Priority-Based Fair Scheduling (PBFS) algorithm has been introduced to schedule jobs so that jobs can access the required resources at optimal times. Then, a backfilling strategy called Shortest Gap - Priority-Based Fair Scheduling (SG-PBFS) is developed 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) regarding flow time, makespan time, and total tardiness, which conclusively demonstrates its effectiveness. To conduct this experiment, we employed the CloudSim simulator, which is implemented using the Java programming language.

KEYWORDS

Algorithm; Backfilling technique; Cloud computing; FCFS; Flow time; Gap search; Job scheduling; Makespan time

REFERENCES

[1]A. Jakóbik, D. Grzonka, F. Palmieri, Non-deterministic scheduler for distributed cloud organizations, 76 (2017) 67–81.
[2]Y. Pachipala, J. Alzubi, Managing the cloud secured fuzzy keyword search for multiple, (2018) 563–565.
[3]M. Maray, J. Shuja, Computation offloading mobile edge computing: Survey, taxonomy, 2022 (2022).
[4]Y.M. Ren, Y. Ding, Y. Zhang, P.D. Christofides, framework for additive manufacturing,
[5]L.G. Garcia, E. Montoya, S. Isaza, R.A. Velasquez, parallel programming assignments with Int. J. Eng. Pedagogy 11 (4) (2021).