Egyptian Informatics Journal (2016) 17, 125–137

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

FULL-LENGTH ARTICLE

Designing of vague logic based multilevel feedback queue scheduler

Supriya Raheja [a,*], Reena Dadhich [b], Smita Rajpal [c]

[a] *Department of Computer Science & Engineering, School of Engineering & Technology, NCU (Formerly ITM University), Gurgaon, Haryana, India*
[b] *Department of Computer Science & Informatics, University of Kota, Kota, Rajasthan, India*
[c] *Alpha Global IT, Toronto, Canada*

Received 28 January 2015; revised 2 August 2015; accepted 5 September 2015
Available online 19 October 2015

KEYWORDS

Vague based Multilevel Feedback Queue (VMLFQ) Scheduler;
Scheduler;
Multilevel Feedback Queue (MLFQ) Scheduling;
Vague set theory;
Vague inference system

Abstract Multilevel feedback queue scheduler suffers from major issues of scheduling such as starvation for long tasks, fixed number of queues, and static length of time quantum in each queue. These factors directly affect the performance of the scheduler. At many times impreciseness exists in attributes of tasks which make the performance even worse. In this paper, our intent is to improve the performance by providing a solution to these issues. We design a multilevel feedback queue scheduler using a vague set which we call as VMLFQ scheduler. VMLFQ scheduler intelligently handles the impreciseness and defines the optimum number of queues as well as the optimal size of time quantum for each queue. It also resolves the problem of starvation. This paper simulates and analyzes the performance of VMLFQ scheduler with the other multilevel feedback queue techniques using MatLab.

© 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

1. Introduction

A scheduler is the key module of any contemporary operating system that manages the concurrent execution of active tasks by sharing the CPU time among these tasks. To achieve these goals it runs a scheduling algorithm which selects the next task to run as well as divide the CPU time. In a productive system, scheduler should be fair and efficient [1,2]. Efficiency and fairness can be considered in terms of different parameters such as average waiting time, average turnaround time, average response time, and starvation. These goals vary with the system being used. Keeping these goals, operating system's designers prefer to use Multilevel Feedback Queue (MLFQ) scheduling algorithm for scheduler over other scheduling algorithms.

The fundamental problems with the MLFQ scheduling are threefold: first is how to assign the parameters to the scheduler, such as how to decide the optimum number of queues, how much length of time quantum for each queue and how the priority is assigned to each task, so that starvation will not occur

[*] Corresponding author.

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

http://dx.doi.org/10.1016/j.eij.2015.09.003
1110-8665 © 2015 Production and hosting by Elsevier B.V. on behalf of Faculty of Computers and Information, Cairo University.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

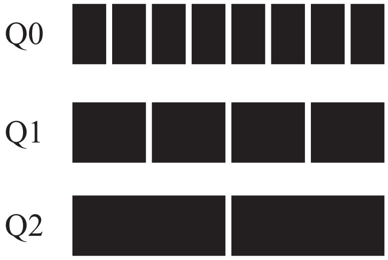

Figure 1 Variation in the size of time quantum within different queues.

[3]. Most of the MLFQ schedulers allow variable length of time quantum to the queues. Often higher priority queues are assigned short time quantum for interactive tasks and the lower priority queues are assigned long time quantum as they contain background tasks. Fig. 1 illustrates the variation in size of time quantum within each queue. Each black box represents the one time quantum.

Second, it tries to optimize the average turnaround time. Third, MLFQ desires that the system should be more responsive, thus to minimize the response time [4]. However, the algorithms such as Round Robin minimize the response time but unfortunately increase the turnaround time [5]. Moreover, task's attributes can be having imprecise data which further affect these issues and make performance even worse. Hence, the focus of an operating system designer is to build a scheduler that achieves all the desired goals of scheduling and at the same time handles the impreciseness.

In this paper we introduce a vague logic based new multilevel feedback queue CPU scheduler and call it as VMLFQ scheduler. VMLFQ scheduler considers all the above mentioned problems with MLFQ and provides solutions to all. VMLFQ scheduler dynamically calculates the length of time quantum for each queue which makes the scheduler flexible. Hence, it can take decisions at run time. With all these, it also improves the performance of a system in terms of average waiting time, average turnaround time, average normalized turnaround time and average response time.

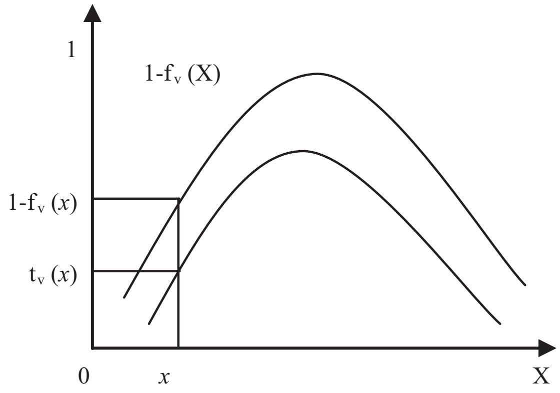

Figure 3 Vague member functions.

This paper is organized as follows. Section 2 gives brief explanation of the task scheduling algorithm and the Multilevel Feedback Queue Scheduling. This section also discusses the related work of MLFQ. Section 3 provides the reader with the background information on vague set theory. Section 4 describes VMLFQ scheduler in detail. Section 5 discusses the simulation with the help of sample task sets and results. Finally, Section 6 concludes the work.

2. Related work

Scheduling algorithm is the technique that a scheduler uses to decide the next task to run. The performance of operating system mainly depends on the scheduling algorithm used by scheduler. There have been a number of scheduling algorithms proposed in the literature such as First Come First Serve, Priority, Shortest Job First, Round Robin, Multilevel Queue, and Multilevel FeedBack Queue scheduling algorithm [6–9]. However, out of all, multitasking systems prefer Multilevel Feedback Queue Scheduling algorithm [10,11]. As our focus is on Multilevel Feedback Queue, all these algorithms are out of scope of this paper.

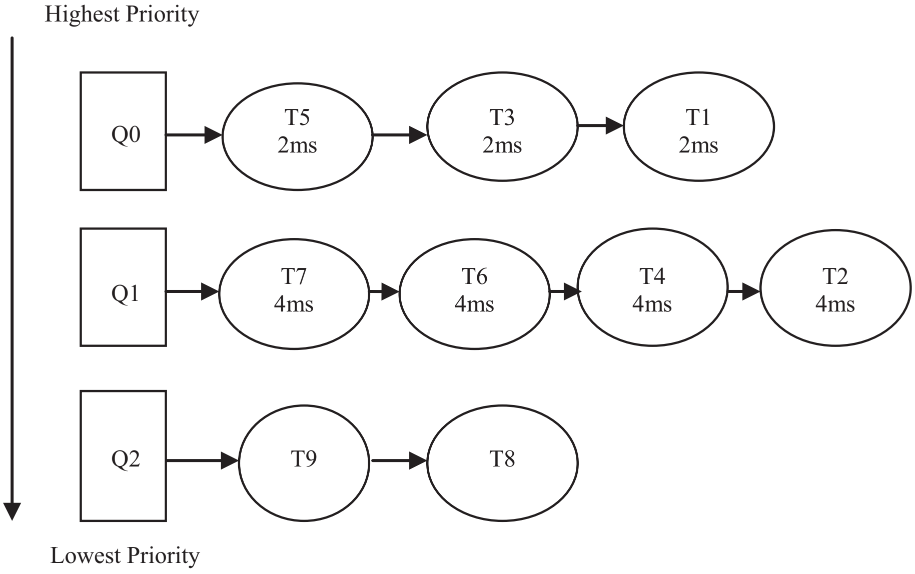

Figure 2 Pictorial representation of multilevel feedback queue example.

Download English Version:

https://daneshyari.com/en/article/476488

Download Persian Version:

https://daneshyari.com/article/476488

Daneshyari.com