Transactions on Machine
Learning and Data Mining
Vol. 14, No. 2 (2021) 47 - 54
© 2021, ibai-publishing
P-ISSN: 1865-6781, E-ISSN 2509-9337
ISBN 978-3-942952-89-7

Machine learning methods for CPU scheduling

Majed Alsanea,
Arab East Colleges, Riyadh, Saudi Arabia
malsanea@arabeast.edu.sa

Abstract. CPU scheduling is an important component of modern operating systems. The performance of any operating system is highly dependent on the scheduling discipline utilized. There are relatively few scheduling disciplines still in use. Almost all modern operating systems utilize a Multilevel Feedback Queue scheduling algorithm (MLFQ). However, a fixed time slice set for each queue poses a performance-inhibiting problem that afflicts this discipline. The assignment of fixed time slices negatively influences the performance of the scheduler. In this paper, we try to tackle this issue and enhance the efficiency of the MLFQ scheduler by applying SMO-reg, Support Vector Machines and Random Tree classification methods to approximate the time slice required for each queue. We apply these methods to three artificial datasets, and the performance results are noted and compared.

Keywords: Support Vector Machine, Machine Learning, Random Tree Classification

1 Introduction

Process or CPU scheduling [1] [2] is an important part of a modern operating systems. CPU scheduling changes the state of a process from ready to running. The main goal of a scheduler is to enhance the efficiency of the operating system. Over the years, several scheduling disciplines have been introduced and used [1] [2]. Currently, all modern operating systems such as Windows and Linux utilize the Multilevel Feedback Queue scheduling algorithm (MLFQ) [3, 4], which is the industry standard scheduling discipline. Nevertheless, new challenges have surfaced after the widespread deployment and use of this algorithm [5, 6]. Fixed time slices, allocated for each queue, are the main challenges now facing this scheduling discipline.

Recently, numerous methodologies have been introduced to enhance the efficiency of the Multilevel Feedback Queue scheduling algorithm [7, 8, 9, 10]. In this paper, we introduce a methodology based on machine learning (ML), Machine Learning Based Multilevel Scheduling (MLBMS). This new methodology utilizes seven phases. The advantages of this methodology are 1) the turnaround time is reduced; and 2) the response time is reduced. Reducing the turnaround time and response time leads to increased efficiency of the scheduling process. We performed comprehensive simulation experiments on different types of datasets to evaluate the efficiency of this newly introduced methodology.

The rest of this paper is organized as follows. Section 2 briefly introduces the MLFQ discipline. Section 3 briefly introduces the methods used. Section 4 presents the newly developed methodology. Section 5 presents an overview of the datasets utilized. Section 6 provides the results and analysis of the experiments. Conclusions and future-plans are given in Section 7.

2 Multilevel Feedback Queue

Multilevel feedback queue scheduling [3, 4] is an extension of the multilevel queue scheduling discipline. The MLFQ scheduler was developed in the early 1960s under the name Compatible Time-Sharing System (CTSS). Over the years, the scheduler has evolved to what we find in today's sophisticated operating systems. This scheduler is a familiar and well-known scheduling discipline utilized by many modern operating systems such as Windows and Linux. The MLFQ design is partitioned into several queues, as shown in Figure 1, and each queue utilizes the scheduling discipline.

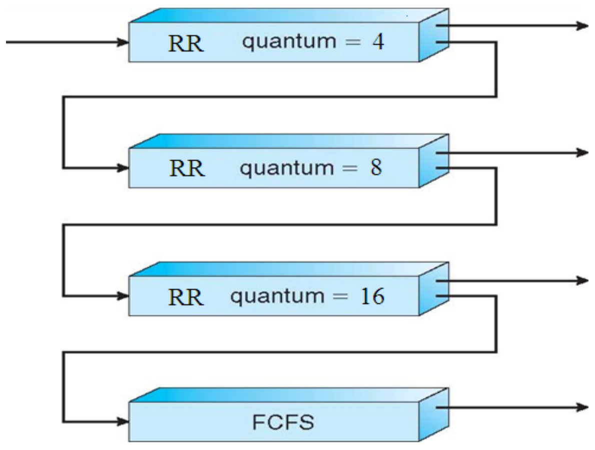

Fig. 1. Multilevel Feedback Queue (MLFQ)

This design permits processes to wander between queues. This algorithm allows the processes to flexibly alter their foreground or background states. Nevertheless, setting a fixed time slice for each queue hinders performance and is the main challenge for this scheduling discipline.

3 Methods Utilized

Several techniques have been developed for data pre-processing and machine learning [11, 12, 13, 14]. In this section, we give a brief introduction on two feature selection techniques (Relief-F and Information gain) and three classification techniques (RandomTree, SMOreg and Support Vector Machine), which were applied in the experiments. These techniques are implemented in the WEKA package [15, 16]. Relief-F has been developed as a non-myopic feature selection method [17], i.e., it computes the quality of a given feature in the context of other features. Information gain (IG) [18] is a popularly used feature selection technique in several applications [19, 20]. IG is utilized for discrete features only. The system entropy is a common measure for Information Gain. Decision Tree (Random Tree) works by creating a tree to evaluate an instance of data. Different types of decision trees exist. Random Tree is a commonly used decision tree. Random Tree is a type of supervised learning technique, and it creates several different learners. SMOreg [21] is a commonly utilized classification method. Support Vector Machine (SVM) [22], which is considered a supervised learning method, is used for linear and non-linear classification and also for regression analysis. SVM is a commonly utilized clustering technique in industrial, biological, and other fields. The efficiency of SVM relies on the kernel value, which is an important parameter in this technique. During experiments, this parameter is set to a Gaussian kernel. All other values related to other methods are set to default values.

4 Proposed Approach

This work presents an ML-based methodology to calculate the running time of a process for every queue. Machine learning permits the system to learn from the data at hand. Machine learning is an iterative process that leads to enhancements in the types of associations made between data elements. ML has been used in several computer engineering and science areas. The proposed approach instantly calculates the time slice for each queue and makes the decision at run time. This approach enhances the effectiveness of the system in terms of the average waiting time, turnaround time, and response time. This methodology consists of seven phases, as shown in Figure 1, and is briefly presented as follows:

1. Preparing the dataset: This stage is divided into two sub-steps: first, expressing the attributes as feature vectors; second, analysing the synthetic dataset and dividing it into training data (60%) and testing data (40%).
2. Selecting the attributes**:** In this step, two feature selection methods, namely, Relief-F and Information gain, are utilized to select the most important attributes.
3. Building the model**:** This step trains DT (REPtree), SVMs and SMOreg ML methods using the training dataset.
4. Assessing the model**:** The testing dataset is utilized to assess the developed model. Correlation Coefficient (CC) and Mean Absolute Error (MAE) are utilized to evaluate the developed models.

5. Calculating the time slice**:** In this step, the developed model is utilized to approximate the necessary time slice, which is given to the process that is waiting in the active queue.
6. Running the process: In this step, the process is executed for the calculated time slice.
7. Assess the outputs**:** In this step, the calculated time slice is assessed for further analysis.

5 Dataset Description

To test the new methodology ability to effectively calculate the approximate time slice, we utilized three artificial datasets consisting of 3000, 4000 and 5000 processes. Each process has 8 properties, which are User ID, Process ID, Time of submission, Time of arrival, Time of waiting, Time of execution, Group ID, and partition number. Since artificial datasets are utilized for testing, we are sure not to have missing data. Hence, the attributes need not be analyzed. Furthermore, since we are expecting a discrete output, a discrete distribution is utilized for the goal attribute.

6 Experiments and Results

In this work, three ML methods are applied, i.e., Decision Tree (DT), K-NN, and support vector machines (SVMs). These techniques are implemented in the WEKA package [13]. The kernel value of the Support Vector Machine method is set to the Gaussian kernel. All other values related to other methods are set to their default values. RandomTree was used with its default settings. Three artificial training data sets were used by each ML method to compare the consistency of the methods among all datasets. The Mean Absolute Error (MAE), which is computed by WEKA software, is selected as the norm error model. Furthermore, the Correlation Coefficient (CC) is used to evaluate the introduced methodology. Table 1 presents the results of computing CC MAE for each machine learning method.

Table 1. Results of computing CC and MAE

	CC			MAE		
Method	Dataset 1	Dataset 2	Dataset 3	Dataset 1	Dataset 2	Dataset 3
Decision Tree	0.8401	0.8363	0.8491	0.3621	0.3542	0.3694
SMO-reg	0.8218	0.8389	0.8267	0.3486	0.3711	0.3627
SVMs	0.8641	0.8527	0.8366	0.3743	0.3763	0.3552

Figure 2 presents the results of computing the error for each machine learning method on various datasets. It is clear from Figure 2 that SMOreg achieved excellent results on dataset 1 as well dataset 3 and that Decision Tree produced excellent results on dataset 2. SMOreg obtained excellent results in 2 datasets and was second in the third dataset. Decision Tree was first and second in 2 datasets, while it was not first for the other two datasets. SVM was second in one dataset and was third in the remaining datasets. It is clear that SMOreg generated the best outcomes.

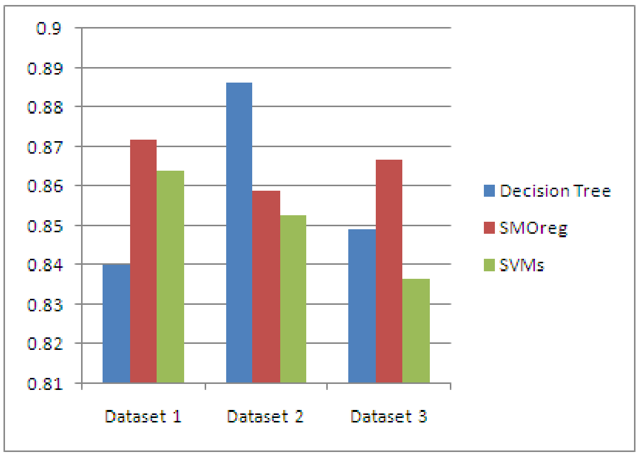

Fig. 2. Results of Computing CC

Figure 3 presents the results of computing the Correlation Coefficient for each machine learning method on various datasets. We notice from Figure 3 that SMOreg obtained less errors on dataset 1 and dataset 3, and SVM obtained the lowest errors on dataset 2. Decision Tree obtained the lowest errors on dataset 3 and was second in the remaining datasets.

To compare the new methodology with the MLFQ scheduling discipline, we utilized the CPU scheduling simulator provided by [23]. Table 2 presents the performance resulting from running the MLFQ scheduling discipline using random values for the time slice and then running the discipline using the values of the time slice calculated by the new methodology. We used turnaround time (TT) and response time (RT) as the performance criteria for comparisons. The MLFQ scheduling discipline was run using four artificial datasets consisting of 50, 100, 150, 2000 processes. Five runs were performed on each dataset, and the mean value of the results was output. From the table, we notice that the new methodology obtained the best results in dataset 1, dataset 2, and dataset 4 for TT and RR criteria. Furthermore, the new methodology obtained results close to those from the MLFQ scheduling discipline.

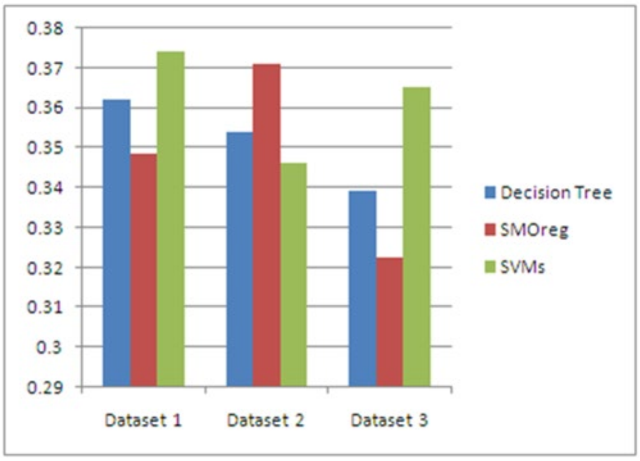

Fig. 3. Results of Computing MAE

Table 2. Results of TT and RT

	New Methodology		MLFQ	
	TT	RT	TT	RT
Dataset 1	66.3	54.8	78.2	67.4
Dataset 2	112.9	88.3	119.6	97.7
Dataset 3	181.2	108.7	178.4	109.3
Dataset 4	228.7	134.3	234.8	141.8

7 Conclusion and Future Work

In this work, we introduced a new methodology that utilizes two feature selection methods, Relief-F and Information gain, as well as three classification methods, SMOreg, Support Vector Machines and Random Tree, to approximately calculate the time slice required for each queue in the Multilevel Feedback Queue system. Three artificial datasets were utilized for experimentation. We performed several experiments, and the results are presented for the three artificial datasets. The SMOreg classification method achieved excellent results compared to the other methods. Subsequently, we performed several experiments to compare the new methodology with the MLFQ scheduling discipline using different datasets. The results of the comparisons are provided. The new methodology produced good results for three out of four datasets. In future studies, real-world datasets and other machine learning methods will be utilized.

References

1. S. Abraham, Peter B. Galvin and Greg Gagne, “Operating System Concepts,” 9th ed., John Wiley & Sons, 2012.
2. S. William, “Operating systems: internal and design principles,” 8th ed., Prentice Hall, Person Education, 2015.
3. F. J. Corbato, M. M. Daggett, R. C. Daley,“An Experimental Time-Sharing System”, IFIPS 1962.
4. D.H.J. Epema, “An Analysis of Decay-Usage Scheduling in Multiprocessors”, SIGMETRICS ’95.
5. Torrey LA, Coleman J, Miller BP. A comparison of interactivity in the Linux 2.6 scheduler and an MLFQ scheduler. Softw Pract Experience 2007;37(4):347–64.
6. Arpaci-Dusseau RH, Arpaci-Dusseau AC. Operating Systems : three easy pieces, scheduling, multilevel feedback queue. Version 0.80; 2014.
7. Malhar Thombare; Rajiv Sukhwani; Priyam Shah; Sheetal Chaudhari; Pooja Raundale, “Efficient implementation of Multilevel Feedback Queue Scheduling”, 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET).
8. Rakesh Kumar Yadav, Anurag Upadhayay, "A fresh loom For Multilevel Feedback Queue Scheduling Algorithm", International Journal of Advances in Engineering Sciences, vol. 2, no. 3, July 2012.
9. Deepali Maste, Leena Ragha, Nilesh Marathe, "Intelligent Dynamic Time Quantum Allocation in MLFQ Scheduling" in International Journal of Information and Computation Technology, International Research Publications House, vol. 3, no. 4, pp. 311-322, 2013, ISBN 0974-2239.
10. H.S. Behera, Reena Kumari Naik, Suchilagna Parida "Improved Multilevel Feedback Queue Scheduling Using Dynamic Time Quantum and Its Performance Analysis". (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 3 (2), 2012, 3801-3807
11. Zimmerman J. Fuzzy set theory and its applications. Norwell, Massachusetts, U.S.A.: Kluwer Academic Publishers; 2001.
12. Zadeh LA. The role of fuzzy logic in the management of uncertainty in expert systems. Fuzzy Sets Syst 1983; 11:197–227.
13. Tom Mitchell, Machine Learning, 1st Ed, pp: 52-75, 154-183, 230-244, the Mc-Graw Hill Company. Inc. International Edition, 1997.
14. Shevade, S. K., S. S. Keerthi, C. Bhattacharyya, and K. R. K. Murthy. 2000. “Improvements to the SMO Algorithm for SVM Regression.” IEEE Transactions on Neural Networks 11(5):1188–93.
15. E. Frank, M. A. Hall, H. Witten, “Data Mining: Practical Machine Learning Tools and Techniques”, Morgan Kaufmann Publisher, Fourth Edition, 2016.
16. Holmes, A. Dokin, I. H. Witten, WEKA: A Machine Learning Workbench, In Proceedings of the Second Australian and New Zealand Conference on Intelligent Information Systems, 357-361, 1994.
17. Kononenko, I., Sikonja, M. R., 2008. Non-myopic feature quality evaluation with (r) relieff. Computational Methods of Feature Selection, 169–191.
18. T. Mitchell. Machine Learning. McGraw-Hill, New York, 1997
19. Fleuret,F.(2004).Fast binary feature selection with conditional mutual information, JournalofMachineLearningResearch,5,1531–1555.
20. J. Novakovic, “Using Information Gain Attribute Evaluation to Classify Sonar Targets”, 17th Telecommunications forum TELFOR, Belgrade, 2009.

21. Platt, John (1998), Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines, CiteSeerX 10.1.1.43.4376.
22. Cortes, Corinna; Vapnik, Vladimir N. (1995). "Support-vector networks". Machine Learning. 20 (3): 273–297. doi:10.1007/BF00994018
23. Yosef Hasan Jbara, “A New Visual tool to Improve the Effectiveness of Teaching and Learning CPU Scheduling Algorithms”, 2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT).