

STATISTICAL APPROACH TO DETERMINE MOST EFFICIENT VALUE FOR TIME QUANTUM IN ROUND ROBIN SCHEDULING

Shreyank N Gowda

Department of Computer Science And Automation, Indian Institute Of Science,
Bengaluru, India

ABSTRACT

Scheduling various processes is one of the most fundamental functions of the operating system. In that context one of the most common scheduling algorithms used in most operating systems is the Round Robin method in which, the ready processes waiting in the ready queue, take control of the processor for a short period of time known as the time quantum (or time slice) circularly. Here we discuss the use of statistics and develop a mathematical model to determine the most efficient value for time quantum. This is strictly theoretical as we do not know the values of times for the various processes beforehand. However the proposed approach is compared with the recent developed algorithms to this regard to determine the efficiency of the proposed algorithm.

KEYWORDS

Round Robin, Time Quantum, Process Scheduling, Statistics

1. INTRODUCTION

A process is defined as a program in execution. One of the most fundamental functions of the Operating Systems is scheduling various processes that are stored in the ready queue. To help in this functionality various scheduling algorithms are present.

Before we look at the fundamentals of the Round Robin Algorithm let us have a look at some important terms we need to know to understand the Round Robin algorithm better.

All the processes that are waiting to be assigned to a processor are first put into a queue called the ready queue. Here all the processes wait patiently for their turn before execution.

The time needed for a process to complete its execution or the time needed for a process to hold the CPU is called the burst time. The time at which the process arrives into the ready queue is called the arrival time.

The time needed to complete execution of a process after its arrival in the ready queue is called turnaround time. The amount of time the process waits in the ready queue is called waiting time.

The number of times the CPU switches from one process to the other is known as context switch. The most efficient scheduling algorithm will have minimum turnaround time, minimum waiting time and minimum number of context switches. Of the various algorithms, the most used in operating systems is the Round Robin Scheduling.

In this algorithm, processes are dispatched in a FIFO manner but are given a limited amount of time to access the CPU, this time is called a time-slice or a quantum. If a process does not complete before its time-slice expires, then the CPU is pre-empted and the next process waiting in the queue is given access to the CPU.

The pre-empted process is placed at the end of the ready queue. Round Robin Scheduling is effective in time-sharing environments in which the system needs to guarantee reasonable response times for interactive users, since it offers pre-emption.

The only issue with the round robin scheme is the length of the time-quantum. If we set the quantum value as too short it results in too many context switches and lowers the efficiency of the CPU. On the other hand, if we set the quantum too high a value, then it may cause poor response time and approximate First Come First Serve.

The time needed for all process execution is quite large in comparison with other algorithms such as Shortest Job first, but it performs better than the only practical algorithm the First Come First Serve.

2. BACKGROUND

To ensure that an algorithm is efficient for job scheduling it should have minimal waiting and turnaround time. The most important parameter in this scheduling algorithm, is the value of the time quantum[1].

If the value chosen is small then, the short processes will be able to pass the system very soon, and the overhead of this system will increase due to the increasing number of context switches.

On the other hand, if this amount of time quantum is bigger than the maximum burst time of the ready processes, this policy will execute like the FCFS scheduling algorithm.

The value of this parameter hence should be considered as a little more than a specific transaction or the time needed for a common execution[2].

Albielmona [3] did a complete review over different scheduling algorithms and showed the complexity of each whilst showing only the First Come First Serve algorithm and the Round Robin algorithm were the only practically implementable ones.

Proportional share scheduling algorithm was proposed in [4] and it combines the small overhead of the RR method by ensuring protection of the short processes is done. The capability of re-adjusting the weights also enables the algorithm to have a more fair behaviour.

Nieh et al. [5] proposed a scheduling algorithm known as Virtual Time Round-Robin (VTRR) with complexity of $O(1)$ by combining the fair queuing algorithms with the existing RR scheduling algorithm.

Zahedi et al. [6] optimized the existing RR algorithm by making use of the concept of fuzzy rules and introduced a novel algorithm as the Fuzzy rule based Round-Robin CPU Scheduling (FRRCS). They showed that using the proposed algorithm, managed to reduce the average waiting time.

The constraint of static time quantum was resolved in [7] by dynamically re-adjusting the value of the parameter which was the time quantum itself, repeatedly.

Singh et al. [8] proposed a new Round Robin algorithm and showed that the proposed method, reduced the problems of the simple Round Robin algorithm and also increases the system throughput.

Mostafa et al. [9] introduced a new algorithm based on an integer programming method used to calculate the appropriate value of the time quantum and proved that this showed very good results.

Mohanty et al. [10] also developed a new algorithm called Shortest Remaining Burst Round Robin (SRBRR) algorithm based on minimum remaining burst time and showed that their method is more effective than the common Round Robin method.

Shahram et al. [11] introduced a nonlinear programming mathematical model to determine the optimum value of the time quantum, in order to minimize the average waiting time of the processes.

3. PROPOSED METHOD

Before The proposed algorithm involves a few statistical terms namely mean and standard deviation. The formulae for both are given below in fig 1. along with formula for variance.

$$\begin{aligned} \text{Mean}(\text{sample}) &= \mu = \frac{\sum_{i=1}^k f_i x_i}{n-1} \\ \text{StandardDeviation}(\text{sample}) &= \sigma = \sqrt{\frac{\sum_{i=1}^k f_i (x_i - \mu)^2}{n-1}} \\ \text{Variance}(\text{sample}) &= \sigma^2 = \frac{\sum_{i=1}^k f_i (x_i - \mu)^2}{n-1} \end{aligned}$$

Figure 1. Formulae of common statistical terms

The Standard Deviation is a measure of how spread out numbers are i.e how far they are from the mean. The mean is just the average of all the numbers.

This proposed algorithm is purely theoretical since it should be knowing before hand the values of all the processes.

The formula obtained for the algorithm was done so following a series of trial and error attempts to find most efficient value for time quantum by testing about 50 different samples.

Following are the steps involved for the algorithm.

Step 1: Start

Step 2: Calculate number of processes as 'n'

Step 3: Find mean and standard deviation for all the values and store in variables 'm' and 's' respectively

Step 4: Calculate time quantum as $tq = \sqrt{n \times m \times s}$

Step 5: Execute completely using the same value for time quantum

Step 6: Stop

Fig.2 shows a flowchart of working for the proposed algorithm.

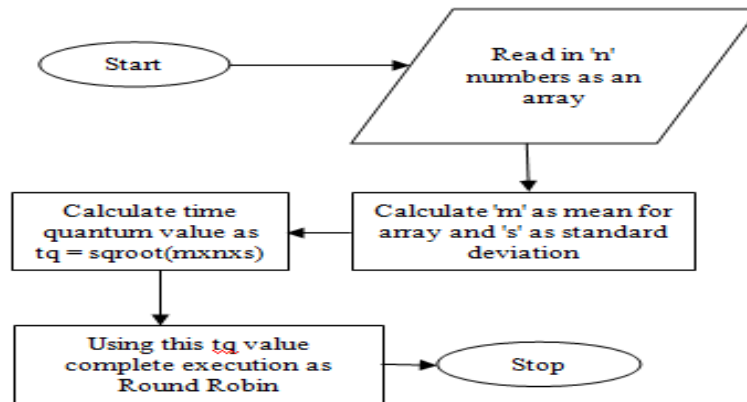


Figure 2. Flowchart of working of algorithm

4. EXPERIMENTAL ANALYSIS

The most efficient existing algorithm is the algorithm proposed in [10]. So in that regard the experimental analysis has been done as a comparison of our proposed algorithm with that proposed in [10] and also the standard Round Robin Algorithm.

Case 1:

Table I. Input for case 1

Process number	Arrival	Burst Time
1	0	13
2	0	35
3	0	46
4	0	63
5	0	97

In this case for the proposed algorithm we need to find value for time quantum. Mean for the given data is 50.8 hence approximated to 51. The Standard Deviation value is 31.56, hence approximated to 32.

Now value for time quantum can be calculated as approximately 90. Using this as the time quantum value we perform the Round Robin algorithm. Also in comparison the Time Quantum value obtained for [10] is 63.

We assume that for the standard Round Robin algorithm that the time quantum value is 25. Using these three values we perform the Round Robin algorithm on the input. We then compare the output.

Table II. Computational results for Case 1

Method	Time Quantum	Average Turnaround time	Average Waiting time
Round Robin	25	148.2	97.4
Linear Programming model	63	113.2	62.4
Proposed model	90	113.2	62.4

For case 1 it can be seen that the proposed model holds as good as the linear programming model. In case of the standard Round Robin model it took more time to execute because of the various overheads in calculation.

Case 2:

Table I. Input for case 1

Process number	Arrival	Burst Time
1	0	14
2	0	34
3	0	45
4	0	62
5	0	77

In this case for the proposed algorithm we need to find value for time quantum. Mean for the given data is 46.4 hence approximated to 46. The Standard Deviation value is 24.41, hence approximated to 24. Now value for time quantum can be calculated as approximately 74.

Using this as the time quantum value we perform the Round Robin algorithm. Also in comparison the Time Quantum value obtained for [10] is 62. We assume that for the standard Round Robin algorithm that the time quantum value is 25.

Using these three values we perform the Round Robin algorithm on the input. We then compare the output.

Table II. Computational results for Case 1

Method	Time Quantum	Average Turnaround time	Average Waiting time
Round Robin	25	143.4	97
Linear Programming model	62	108.2	62
Proposed model	74	108.2	62

For case 2 it can be seen that the proposed model holds as good as the linear programming model. In case of the linear programming model it took more time to execute because of the various overheads in calculation.

Case 3:

Table I. Input for case 1

Process number	Arrival	Burst Time
1	0	20
2	0	40
3	0	60
4	0	80

In this case for the proposed algorithm we need to find value for time quantum. Mean for the given data is 50. The Standard Deviation value is 25.81, hence approximated to 26. Now value for time quantum can be calculated as approximately 72.

Using this as the time quantum value we perform the Round Robin algorithm. Also in comparison the Time Quantum value obtained for [10] is 60. We assume that for the standard Round Robin algorithm that the time quantum value is 25.

Using these three values we perform the Round Robin algorithm on the input. We then compare the output.

Table II. Computational results for Case 1

Method	Time Quantum	Average Turnaround time	Average Waiting time
Round Robin	25	125	75
Linear Programming model	60	100	50
Proposed model	60	100	50

For case 3 again it can be seen that the proposed model holds as good as the linear programming model. In case of the linear programming model it took more time to execute because of the various overheads in calculation.

5. CONCLUSION

As could be seen from the experimental results that the proposed algorithm gives results as good as the current best algorithm. However the time needed in terms of execution for the current best algorithm is much larger in comparison with the proposed algorithm and hence we can say that the proposed algorithm is more efficient in this case.

ACKNOWLEDGEMENTS

Any work be it scholastic or otherwise does not depend solely on the person doing the work. Various people have given be intellectual, professional and emotional support during my time conducting this work. This is my opportunity to thank them all.

I would like to thank the Head of the Department of Computer Science And Engineering Dr. Shoba G for her support in doing this work. Also I would like to thank all the faculty members for their constant support to motivate me in performing this.

Last but not the least I would like to thank my family and my friends for being a strong emotional support throughout my time performing this project.

REFERENCES

- [1] W. Stallings, Operating Systems: Internals and Design Principles. 6th Edition, Prentice Hall, 2008.
- [2] M. Fahimi, Operating Systems, vol. 1, 1st Edition, Tehran: Jelve Publishing, 1992.
- [3] R. Abielmona, "Scheduling Algorithmic Research", Department of Electrical and Computer Engineering, Ottawa-Carleton Institute, 2000.
- [4] T. Helmy, and A. Dekdouk, "Burst Round Robin: As a Proportional-Share Scheduling Algorithm", IEEE, Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations, pp. 424-428, 2007.
- [5] J. Nieh, Ch. Vaill, and H. Zhong, "Virtual-Time Round-Robin: An O(1) Proportional Share Scheduler." Proceeding of the 2001 USENIX Annual Technical Conference, USA, 2001.
- [6] M.H. Zahedi, M. Ghazizadeh, and M. Naghibzadeh, "Fuzzy Round Robin CPU Scheduling (FRRCS) Algorithm", Advances in Computer and Information Sciences and Engineering. 348-353, 2008.
- [7] R.J. Matarneh, "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the now Running Processes", American Journal of Applied Sciences, 6(10), 1831- 1837, 2009.
- [8] A. Singh, P. Goyal, and S. Batra, "An Optimized Round Robin Scheduling Algorithm for CPU Scheduling", International Journal on Computer Science and Engineering. 2(7), 2382-2385, 2010.
- [9] S.M. Mostafa, , S.Z. Rida, and S.H. Hamad, "Finding time quantum of Round Robin CPU scheduling in general computing systems using integer programming", International Journal of Research and Review in Applied Science. 5(1), pp 64-71. 2010.
- [10] Mohanty, H. Behera, K. Patwari and M. Dash, "Design and performance evaluation of a new proposed shortest remaining burst round robin (SRBRR) scheduling algorithm" In Proceedings of International Symposium on Computer Engineering & Technology (ISCET) (Vol. 17, pp. 126-137), 2010.
- [11]S. Saeidi, H. A. Baktash, " Determining the Optimum Time Quantum Value in Round Robin Process Scheduling Method" International Journal Of Information Technology and Computer Science pp 67-73, 2012.

AUTHORS

Shreyank N Gowda, born on October 2nd 1993, is a graduate from the Computer Science And Engineering Department of R.V.College Of Engineering. He has published numerous papers in the field of Theoretical Computer Science and his ambition is to continue working in that field.

