Optimal Round Robin CPU Scheduling Algorithm using Euclidean Distance

Merwyn D'Souza
Assistant Professor
Department of Computer Engineering
Don Bosco College of Engineering, Goa -India

Fiona Caiero
Assistant Professor
Department of Computer Engineering
Padre Conceicao College of Engineering, Goa-India

Suwarna Surlakar
Assistant Professor
Department of Computer Engineering
Padre Conceicao College of Engineering, Goa-India

ABSTRACT

The performance of Round robin CPU scheduling algorithm is entirely dependent on the time quantum selected. This paper describes a new method to calculate the time quantum without user intervention by finding the relationship between the burst times of all processes in the ready queue using a similarity measure known as the Euclidean distance. Similarity measure is used to find patterns in the burst times of processes present in the ready queue.

General Terms

Operating Systems, Algorithms, Efficiency.

Keywords

Round Robin, Scheduling algorithms, Time Slice, Time Quantum, CPU scheduling, Turnaround Time, Waiting Time, Context Switch**.**

1. INTRODUCTION

CPU scheduling is an essential operating task. The problem of determining when processor should be assigned and to which processes is called processor scheduling or CPU scheduling. CPU scheduling is of two types preemptive scheduling and non-preemptive scheduling. Preemptive scheduling allows a process to be interrupted in the midst of execution. Non-preemptive scheduling does not allow a process to be interrupted in the midst of execution.

When more than one process is runnable, the operating system must decide which one must be scheduled first. The part of the operating system concerned with this decision is called the scheduler, and algorithm it uses is called the scheduling algorithm.

1.1 Scheduling Criteria

Different CPU scheduling algorithms have different properties, and the choice of a particular algorithm may favor one class of processes over another. In choosing which algorithm to use in a particular situation, one must consider the properties of the various algorithms. Many characteristics can be used to compare scheduling algorithms. The scheduling criteria include the following:

CPU utilization: Keep the CPU as busy as possible (ranges from 0 to 100 percent).

Throughput: It is the number of processes completed per unit of time.

Turnaround time: It is the interval from the time of submission of a process to the time of completion.

Waiting Time: It is the sum of the periods spent waiting in the ready queue.

Response time: It is the time from submission of a request until first response is produced.

1.2 Scheduling Algorithms

First Come First Served (FCFS) [1] is the simplest algorithm, the process that requests the CPU first is allocated the CPU first. Being a non-preemptive algorithm, once a process has the CPU, it runs to completion. Thus it is undesirable for time-sharing systems where each process needs a share of CPU at regular intervals.

Shortest-Job-First (SJF) algorithm [1] gives CPU to the processes having smallest next CPU burst. The SJF algorithm can be preemptive or non-preemptive. The main difficulty of SJF is to know in advance the length of the next CPU burst.

Priority scheduling algorithm [1] assigns priorities to ready processes and CPU is allocated to the process with highest priority. A major problem of priority based scheduling algorithms is that lower priority processes may be starved of CPU time due to frequent execution of higher priority processes.

Round Robin Scheduling algorithm [1] employs time-sharing, giving each process a certain amount of CPU time (a time slice or time quantum) and if it is not finished by the end of the time slice, the process is moved to the back of the process queue, and the next process in line is moved to the CPU.

2. RELATED WORK

Rami J. Matarneh [2] proposed a method that calculates median of burst time of all processes in ready queue. Mostafa et al. [3] proposed a new method using integer programming for finding the value of time quantum. The method depends on changing time quantum in each round over the cyclic queue. Ahad [4] proposed to modify the time quantum of a process based on some threshold value which is calculated by taking average of left out time of all processes in its last turn. Hiranwal et al. [5] introduced a concept of smart time slice which is calculated by taking the average of burst time of all processes in the ready queue if number of processes are even otherwise time slice is set to mid process burst time. Yaashuwanth et al [6] introduced a term intelligent time slice which allocates the frame exclusively for each task based on priority, shortest CPU burst time and context switch avoidance time. Behera et al. [7] proposed an approach in which the process are arranged in ascending order of the burst time and then an optimal time quantum is calculated using the median concept, which means that if the number of processes in the ready queue is odd, the burst time of the middle process will become the time quantum, otherwise the average of the two middle processes will become the time quantum. .Noon et al. [8] proposed to calculate the time quantum by taking average of the burst time of all the processes in ready queue. Alam [9] proposed a method to determine time quantum using fuzzy logic. It uses two fuzzy inference systems, one for finding the time quantum value and another for deciding the preemption. Banerjee et al.[10] proposed an algorithm which first sorts all the processes according to the burst time and then finds the time quantum by taking average of burst time of all process from mid to last. Nayak et al. [11] calculated the optimal time quantum by taking the average of highest burst and median of burst.

3. PROPOSED WORK

A major disadvantage of round robin is that a process is pre-empted and context switch occurs, even if the running process requires time (in fractions) which is slightly more than assigned time quantum. Another problem with round robin is the time quantum selection. If time quantum is too large, the response time of the processes is too much, the algorithm degenerates to FCFS which may not be tolerated in an interactive environment. If time quantum is too small, it causes unnecessarily frequent context switches leading to more overheads resulting in lesser throughput.

Since random selection of time quantum can lead to an inefficient round robin algorithm, the proposed new algorithm Optimal Round Robin (ORR) determines a dynamic time quantum without user intervention based on the similarity of the burst times of all processes present in the ready queue. Similarity is the measure of how much alike two data objects are. Similarity determines how the burst times of processes in ready queue are related to each other. Based on the relationship of burst times with each other, a time quantum is calculated. The similarity is calculated using Euclidean distance measure.

3.1 Euclidean Distance

In mathematics the Euclidean distance [12] or Euclidean metric is the "ordinary" distance between the two points that one would measure with a ruler, which can be proven by repeated application of the Pythagorean Theorem. The proposed new algorithm on round robin uses Euclidean distance to determine a time quantum based on similarity of burst times of all processes in the ready queue.

The Euclidean distance between point p and q is the length of the line segment connecting them (p,q). In Cartesian coordinates, if p = (p_1, p_2,..., p_n) and q = (q_1, q_2,..., q_n) are two points in Euclidean *n*-space, then the distance from p to q, or from q to p is given by:

$$d(p,q) = d(q,p)$$
$$= \sqrt{(q_{1-}p_1)^2 + (q_3 - p_3)^2 + \cdots + (q_n - p_n)^2}$$
$$= \sqrt{\sum_{i=1}^{n}(q_i - p_i\ )^2}$$

3.2 Proposed Methodology

The proposed algorithm known as Optimal Round Robin (ORR) algorithm selects a time quantum based on the Euclidean distance (ED) of burst times of all processes in ready queue and a factor known as Quantum factor (Qf).The ED finds the distance (similarity) between the burst times of all processes in ready queue. The Qf factor multiplied with the ED gives a time quantum large enough to schedules majority of the processes in a single time quantum thereby reducing the waiting times, turnaround times and number of context switches. Based on experiments it is found that best Qf value is 0.4. The time quantum chosen is large enough so that processes require less waiting times; small enough so that number of context switches are reduced.

3.3 Optimal Round Robin Algorithm

The following data structures are needed:

Process (Pi). Number of processes in ready queue for i=1, 2, 3,4,…...n

Burst Time (Bi): Processing time required by each Pi

1. Calculate the Euclidean Distance 'ED' of the cpu burst times of processes present in ready queue

2. time quantum = ED * Qf ; //consider as integer

3. Schedule processes according to the calculated time quantum.

4. EXPERIMENTS

For the purpose of simplicity, a demonstration is done using group of five processes in three different cases that the ORR algorithm is more efficient than the classic Simple Round Robin (SRR). For SRR, a time quantum is assumed in all cases in order to compare the two algorithms fairly.

Case 1: Assume five processes arrive at time 0 with following burst times: P1=24, P2=11, P3=31, P4=12, P5=20.

The Euclidean distance (ED) calculated is 37
Time quantum (ORR) = ED * Qf = 37 * 0.4 = 14

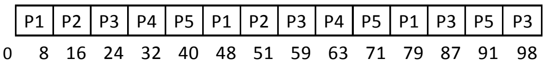

Fig 1: SRR Gantt Chart (case 1)

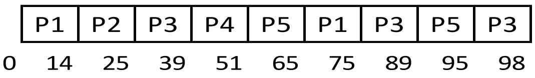

Fig 2: ORR Gantt Chart (case 1)

Table 1. Comparison between SRR and ORR (case 1)

Algorithm	Time Quantum	Waiting Time	Turnaround time	Context Switch
SRR	8	56.7	76.4	14
ORR	14	49.2	68.8	9

Case 2: Assume five processes arrive at time 0 with following burst times: P1=7, P2=13, P3=24, P4=10, P5=18.

The Euclidean distance (ED) calculated is 27
Time quantum (ORR) = ED * Qf = 27 * 0.4 = 10

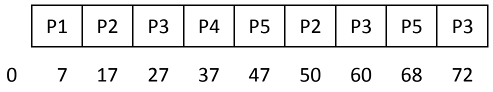

Fig 3: ORR Gantt Chart (case 2)

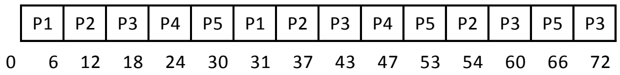

Fig 4: SRR Gantt Chart (case 2)

Table 2. Comparison between SRR and ORR (case 2)

Algorithm	Time Quantum	Waiting Time	Turnaround time	Context Switch
SRR	6	39.4	54	14
ORR	10	32.4	46.7	9

Case 3: Assume five processes arrive at time 0 with following burst times: P1=10, P2=14, P3=33, P4=29, P5=7.

The Euclidean distance (ED) calculated is 30
Time quantum (ORR) = ED * Qf = 30 * 0.4 = 12

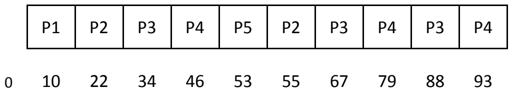

Fig 5: ORR Gantt Chart (case 3)

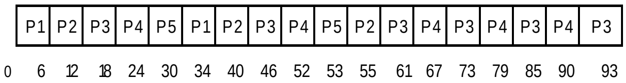

Fig 6: SRR Gantt Chart (case 3)

Table 3. Comparison between SRR and ORR (case 3)

Algorithm	Time Quantum	Waiting Time	Turnaround time	Context Switch
SRR	6	46.4	65	18
ORR	12	41.2	59.7	10

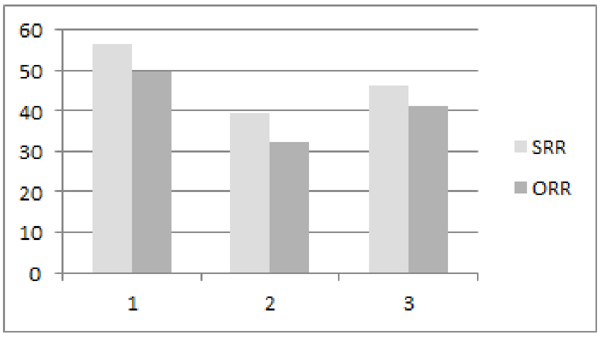

Fig 7: Difference in Waiting time between SRR and ORR

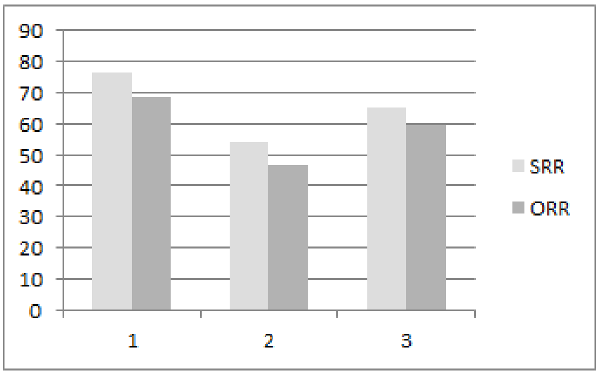

Fig 8: Difference in Turnaround time between SRR and ORR

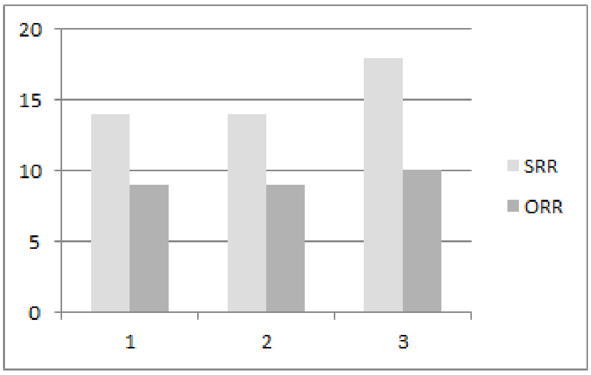

Fig 7: Difference in Context switch between SRR and ORR

From the above comparisons and as can be seen in fig 7, fig 8 and fig 9, the ORR algorithm using Euclidean distance method for calculating time quantum is clearly more efficient than the SRR algorithm resulting in reduction of turnaround time , waiting time and context switches.

Although three cases with each case having five processes are shown, the number of processes does not affect the working of ORR algorithm as it works well even with large number of processes.

5. CONCLUSION AND RESULTS

The performance of round robin algorithm is entirely dependent on the time quantum selected. Many attempts have been made in the past to select an optimum time quantum. Some approaches required making use of other algorithms like shortest job first or priority scheduling, thereby carries forward the deficiencies of those algorithms into round robin scheduling.

The Optimal Round Robin (ORR) determines the time quantum by taking account the similarity or differences of the burst times of all processes present in the ready queue. The ORR does not require priorities to be assigned to the jobs nor does it require the jobs to be sorted according to their burst times. It results in better performance of round robin algorithm with reduction in context switches, turnaround times and waiting times. The time quantum determined through ORR is dynamic in the sense that no user intervention is required and the time quantum is related to the burst times of processes present in the ready queue.

The ORR algorithm performs better than the classic round robin algorithm.

6. REFERENCES

[1] Operating System Concepts ,8th Ed.,Abraham Silberschatz, Peter B. Galvin, Grege Gagne . ISBN 978-81-265-2051-0.

[2] Rami J Matarneh. , "Self adjustment time quantum in round robin algorithm depending on burst time of the now running process", American Journal

[3]Samih M. Mostafa, S. Z. Rida, Safwat H. Hamad, "Finding time quantum of round robin CPU scheduling, algorithm in general computing systems using integer programming" October 2010, IJRRAS 5 (1)

[4] Mohd Abdul Ahad," Modifying round robin algorithm for process scheduling using dynamic quantum precision", International Journal of Computer applications(0975-8887) on Issues and Challenges in Networking, Intelligence and Computing Technologies- ICNICT 2012.

[5] Saroj Hiranwal and Dr. K.C. Roy, " Adaptive round robin scheduling using shortest burst approach based on smart time slice", International Journal of Data Engineering, volume 2, Issue.3, 2011.

[6] Yaashuwanth C. & R. Ramesh, " Intelligent time slice for round robin in real time operating system, IJRRAS 2 (2), February 2010.

[7] H.S.Behera, R. Mohanty, Debashree Nayak, "A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and its Performance Analysis", Volume 5-No. 5, August 2010, International Journal of Computer Application(0975-8887)

[8] Abbas Noon , Ali Kalakech and Saifedine Kadry, " A new round robin based scheduling algorithm for operating systems: dynamic quantum using the mean average" IJCSI International Journal of Computer Science Issues, Vol. 8, Issue 3, No. 1, May 2011.

[9] Bashir Alam, "Fuzzy round robin cpu scheduling algorithm", Journal of Computer Science 9 (8): 1079-1085, 2013.

[10] Pallab Banerjee, Probal Banerjee and Shweta Sonali Dhal, "Comparative performance analysis of mid average round robin scheduling (MARR) using dynamic time quantum with round robin scheduling algorithm having static time quantum", International Journal of Electronics and Computer Science Engineering, ISSN-2277-1956 2012.

[11] Debashree Nayak , Sanjeev Kumar Malla and Debashree Debadarshini, "Improved round robin scheduling using dynamic time quantum", International Journal of Computer Applications (0975-8887) Volume 38- No 5, January 2012.

[12] http://en.wikipedia.org/wiki/Euclidean_distance