

AN IMPROVED APPROACH TO MINIMIZE CONTEXT SWITCHING IN ROUND ROBIN SCHEDULING ALGORITHM USING OPTIMIZATION TECHNIQUES

Mahesh Kumar M R¹, Renuka Rajendra B², Sreenatha M³, Niranjan C K⁴

¹Asst. Professor, Dept. of CSE, JSSATE, Bangalore, India

²Asst. Professor, Dept. of CSE, JSSATE, Bangalore, India

³Asst. Professor, Dept. of CSE, JSSATE, Bangalore, India

⁴Asst. Professor, Dept. of CSE, JSSATE, Bangalore, India

Abstract

Scheduling is a fundamental operating system function. Almost all computer resources are scheduled before use. All major CPU scheduling algorithms concentrate more on to maximize CPU utilization, throughput, waiting time and turnaround time. In particular, the performance of round robin algorithm depends heavily on the size of the time quantum. To improve the performance of CPU and to minimize the overhead on the CPU, time quantum should be large with respect to the context switch time. Otherwise, context switching will be more. In this research paper, we propose a method to minimize the context switching and to break the fixed time quantum size in round robin scheduling algorithm using optimization techniques. Both results and calculations show that, our proposed method is more efficient than the existing round robin scheduling algorithm.

Keywords- Time quantum, CPU Scheduling, Context Switching, Round Robin (RR), Simplex Method, LPP

-----***-----

1. INTRODUCTION

In a multiprogrammed operating system, some process running at all the times to maximize CPU utilization. Several processes are kept in memory at one time. When one process has to wait, the operating system takes the CPU away from that process and gives the CPU to another process. This pattern continues [2]

Every time one process has to wait, another process can take over use of the CPU. To do this more effectively, some of the operating system works on FCFS, SJF, Priority and Round Robin (RR) scheduling algorithms. These algorithms decide when and for how long each process runs, they make choices about preemptibility, turnaround time, waiting time, response time and other process characteristics[3].

Round Robin (RR) scheduling algorithm, considered as the most widely adopted CPU scheduling algorithm, has severe problems related to quantum size. If the time quantum size is too large, it minimizes the process characteristics. If the time quantum is too small, context switching will be more as a result more overhead on the CPU.

In our proposed method, we will break the fixed time quantum size by using the Simplex method to improve the efficiency of the CPU.

1.1 CPU Scheduling Criteria

Generally a set of criteria is established against which various scheduling policies may be evaluated.

1) *CPU Utilization* In a multiprogrammed operating system, CPU should be as busy as possible so has to execute more jobs.

2) *Throughput* This is a measure of how much work is being performed per unit of time. This depends on the average length of the process.

3) *Turnaround Time* This is the sum of the periods spent waiting to get into memory, waiting in the ready queue, executing on the CPU and doing I/O.

4) *Waiting Time* This is the sum of the periods spent waiting in the ready queue.

5) *Response Time* This is the time from the submission of a request until response begins to be received.

1.2 CPU Scheduling Algorithms

There are many CPU scheduling algorithm such as FCFS, SJF non preemptive and preemptive, priority and round robin algorithm etc, which helps in scheduling the requests.

1) *First Come First Serve (FCFS)* processes are dispatched according to their arrival time at the ready queue. Once a process has a processor, the process runs to completion. It is unfair because long processes make short process to wait [3].

2) *SJF nonpreemptive* This algorithm associates each process with the length of the process's next CPU burst. The difficulty is that knowing the length of the next CPU burst time.

3) *SJF Preemptive* This algorithm will preempt the currently executing process and the processor is given to the next process arrived in the ready queue. it requires estimates of future process behavior to be effective.

4) *Priority Scheduling* This algorithm allocates the CPU to the process with the highest priority. A major problem is that it can leave some lower priority processes waiting indefinitely [2].

5) *Round Robin Scheduling* This algorithm, which is the main concern of this paper is one the widely used scheduling algorithm, designed especially for time sharing systems. It is designed to give better response time but the increase in the turnaround time and waiting time due to the fixed time quantum size as a result increase in the context switching.

2. RELATED WORK

In the recent years many researchers have come forward with their work and ideas of reducing the context switching and improving the time quantum in Round Robin scheduling algorithm.

Rakesh Kumar Verma, 2010, suggested that after allocating the time quantum to all the process only in first cycle, they use SJF to select the next process from the ready queue [1].

Maria ulfah siregar, 2012, suggested combines Round Robin with Genetic algorithm thereby increasing the CPU utilization [4].

Deabashree Nayak, Sanjeev Kumar Malla and Debashree Debadarshini, 2012, proposed an improved Round Robin (IRR) scheduling algorithm by arranging the processes according to their shortest burst time and assigning each of them with an optimal time quantum which is able to reduce the context switching [7].

Rami J Matameh, 2009, proposed an algorithm called Self Adjustment Round Robin to make the time quantum repeatedly adjusted according to the burst time of the now running process. Experiment and calculations solves the fixed time quantum problem [5].

Samih M Mostafa, Safwat H Hamad and S Z Rida, 2011, proposed a changeable time quantum that decides a value that is neither too large nor too small and every process has got reasonable response time and the throughput of the system is not decreased due to unnecessarily context switches [6].

Abbas Noon, Ali Kalakech and Seifedine Kadry, 2012, proposed a new approach called dynamic-time-quantum; the idea of this approach is to make the operating systems adjusts the time quantum according to the burst time of the set of waiting processes in the ready queue [8].

Tarek Helmy, Abdelkader Dekdouk, 2007, a proportional-share scheduling algorithm as an attempt to combine the low scheduling overhead of round robin algorithms and favor shortest jobs using a novel weight adjustment for processes that are blocked for I/O and lose some CPU time to assure proportional fairness. Experiment results showed that quickly knocking away shortest processes achieves better turnaround time, waiting time, and response time [9].

3. OPTIMIZATION TECHNIQUES

Operations research is considered as one of the best solution provides for the real world problems. In this case, we use simplex method of operations research to solve the problems of fixed time quantum and to reduce context switching in round robin algorithm. The graphical method is used when number of decision variables is two [10].

First, using the classical round robin scheduling algorithm we calculate the waiting time and turnaround time for the given burst size of individual process. Then we convert the given burst time, waiting time and turnaround time into Linear Programming Models (LPP) mathematical models in the form

$$Z = C_1X_1 + C_2X_2 + \dots + C_nX_n$$

Subject to

$$A_{11}X_1 + A_{12}X_2 + \dots + A_{1n}X_n \dots B_1$$

$$A_{21}X_1 + A_{22}X_2 + \dots + A_{2n}X_n \dots B_2$$

Then we solve the problem by applying simplex algorithm.

After applying the algorithm, we get the optimal solutions based on objective function and among these solutions we choose the best solution to reduce the context switching.

Using the existing round robin algorithm, we calculate the waiting time and turnaround time for each process and number of context switch for the given burst time of each process.

By applying the simplex method to the LPP containing waiting time and turnaround time of each process as constraints and burst time of each process in objective function, we get a new updated time quantum. Using this new quantum, we calculate the new waiting time, turnaround time and context switching of processes.

4. SIMPLEX ALGORITHM

1. Convert the given problem into LPP form as objective function containing process burst time.
2. The constraints contain the waiting time and turnaround time of each process and average waiting time and average turnaround time as right side of the constraints [10, 11].
3. Introduce slack variables to convert LPP into standard form

$$Z=10X_1+1X_2+2X_3+1X_4+5X+0S_1+0S_2$$

Subject to constraints

$$9X_1+1X_2+5X_3+3X_4+9X_5+1S_1=27$$

$$19X_1+2X_2+7X_3+4X_4+14X_5+1S_2=46$$

4. Obtain the starting basic feasible solution to create the simplex table.
5. Compute the net value of $Z_j - C_j$ to identify the entering and leaving variable in the simplex table.
6. Repeat the above procedure until we get all the values of $Z_j - C_j$ are positive.
7. Once we got all that all the values are positive check the values of Z_j , this gives new quantum time is 2 for the problem.
8. Calculate the waiting time and turnaround time for each process using new quantum size.

5. RESULTS AND OBSERVATIONS

Consider there are five processes P1, P2, P3 P4 and P5 having the burst time (10, 1, 2, 1, 5). Assume that time quantum is 1ms.

After applying the round robin algorithm, waiting time of each process is

P1=9ms, P2=1ms, P3=5ms, P4=3ms and P5=9ms
Average Waiting time=27/5=5.4ms

Turnaround time of each process is given by

P1=19ms, P2=2ms, P3=7ms, P4=4ms and P5=14ms

Average Turnaround time=46/5=9.2ms

From the above calculations, we observed that, the number of context switching for a set of 5 processes is 15.

5.1 Proposed Method

As per simplex method, the new quantum size for the above set of process is 2ms.

After applying the round robin algorithm with new quantum, waiting time of each process is

P1=9ms, P2=2ms, P3=3ms, P4=5ms and P5=10ms

Average Waiting time = 29/5=5.8ms

Turnaround time of each process is given by

P1=19ms, P2=3ms, P3=5ms, P4=6ms and P5=15ms

Average Turnaround Time = 48/5=9.6ms

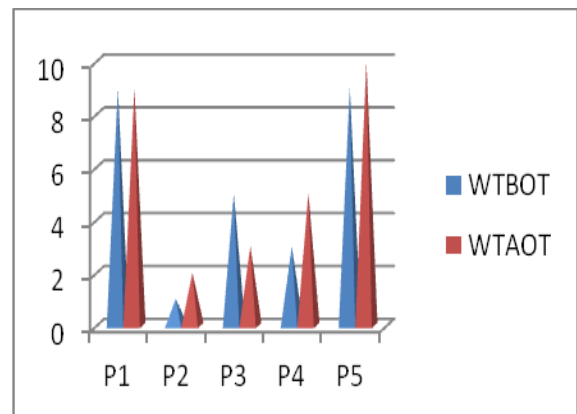
The above calculations shows that the context switching between the process is 9 with nominal increase/decrease in waiting time and turnaround time for few process for quantum=2ms.

5.2 Comparison

The comparison results of waiting time of existing round robin algorithm and proposed method along with graph is shown in table 1.

Table 1

Process	WTbOT	WTaOT
P1	9	9
P2	1	2
P3	5	3
P4	3	5
P5	9	10



WTbOT: Waiting Time before Optimization Techniques

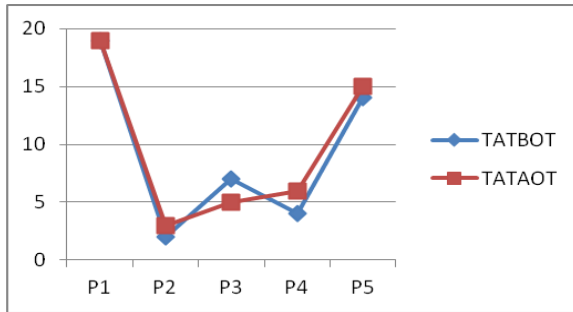
WTaOT: Waiting Time after Optimization Techniques

The above graph shows that there is a nominal increase and decrease in the waiting time for a few process after applying the proposed method.

The comparison results of Turnaround time of existing round robin algorithm and proposed method along with graph is shown in table 2.

Table 2

Process	TATbOT	TATaOT
P1	19	19
P2	2	3
P3	7	5
P4	4	6
P5	14	15



TATbOT: Turn Around Time before applying Optimization Techniques

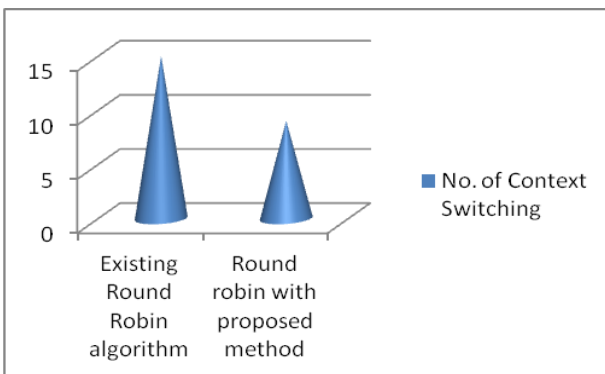
TATaOT: Turn Around Time after applying Optimization Techniques

The above graph shows that there is a nominal increase and decrease in the turnaround time for few processes after applying the proposed method.

The comparison results of number of context switching of existing round robin algorithm and proposed method is shown in table 2 along with the graph.

Table 3

Algorithm	No. of Context Switching
Existing Round Robin algorithm	15
Round robin with proposed method	9



The above graph shows that there is a decrease in the no. of context switching between the existing round robin algorithm and after applying the proposed method.

6. CONCLUSIONS

The performance of round robin algorithm depends heavily on the size of the time quantum and the effect of context switching between the processes. With the classical approach of round robin the quantum size is fixed. If the quantum size is too small, the number of context switching is more with increase in the waiting time and the turnaround time of each process.

Based on our experiment, we can conclude that the number of context switching for any set of processes can be reduced by obtaining the new quantum size using the simplex algorithm. With this new quantum we showed that the number of context switching will be less along with nominal increase/decrease in waiting time and turnaround time of each process. Thus minimizing the overhead on the CPU

REFERENCES

- [1]. Rakesh Kumar Yadav, Abhishek K Mishra, Navin Prakash and Himanshu Sharma, "An Improved Round Robin Scheduling Algorithm for CPU Scheduling", International Journal on Computer Science and Engineering, Vol. 02, No. 04, 2010, pp. 1064-1066.
- [2]. Silberchatz, Galvin and Gagne, operating systems concepts, 8th edition, John Wiley and Sons, 2012
- [3]. Dietel, Dietel and Choffnes, Operating Systems, 3rd edition, Pearson education, 2009.
- [4]. Maria Ulfah Siregar, "A New Approach to CPU Scheduling Algorithm: Genetic Round Robin", International Journal of Computer Applications Vol. 47 - Number 19, 2012
- [5]. Rami J. Matarneh, "Self-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Processes", American Journal of Applied Sciences, Vol 6, No. 10, 2009.
- [6]. Mostafa, S.M., Rida, S.Z. and Hamad, S.H. 2010. Finding time quantum of round robin CPU scheduling algorithm in general computing systems using integer programming IJRRAS
- [7]. Debashree Nayak, Sanjeev Kumar Malla Debashree Debadarshini, "Improved Round Robin Scheduling using Dynamic Time Quantum", International Journal of Computer Applications (0975 – 8887) Volume 38– No.5, January 2012.
- [8]. Abbas Noon, Ali Kalakech and Seifedine 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]. Tarek Helmy, Abdelkader Dekdouk, "Burst Round Robin as a Proportional-Share Scheduling Algorithm", IEEE-GCC

Conference on Towards Techno- Industrial Innovations, pp. 424-428, Bahrain, 2007.

[10]. H S Kasana and K D Kumar, "Introductory Operations Research, Springer, 2004.

[11]. J K Sengupta, "An iterative convex simplex method for geometric programming with applications", International Journal of System Science, 1977