IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Volume-4, Special Issue-1 E-ISSN: 2347-2693

Design and performance evaluation of Advanced Priority Based Dynamic Round Robin Scheduling Algorithm (APBDRR)

Debasmita Saha*
Department of Computer Science
University of GourBanga
Malda, India
debasmita.saha@hotmail.com

Ardhendu Mandal
Department of Computer Science and Application
University of North Bengal
Siliguri, India
am.csa.nbu@gmail.com

***Abstract*—In this paper we have proposed a improvised version of Round Robin Scheduling Algorithm by calculating Dynamic Time Quantum (DTQ) and taking into consideration the priorities assigned with the processes. We have compared the performance of the proposed Advanced Priority Based Dynamic Round Robin Scheduling Algorithm (APBDRR) with the performances of Round Robin Algorithm (RR), Improved Shortest Remaining Burst Round Robin Algorithm (ISRBRR) and Efficient Dynamic Round Robin Algorithm (EDRR). Experimental results show that the proposed algorithm performs better than these algorithms in terms of Average Waiting Time(AWT) and Average Turnaround Time(ATAT).**

Keywords-*CPU Scheduling; Round Robin Scheduling; Priority Scheduling; Waiting Time; Turnaround Time; Time Quantum; Priority, Advanced Priority Based Dynamic Round Robin Scheduling Algorithm*

I. INTRODUCTION

A process is an instance of a computer program in execution. Scheduling these processes is one of the most important jobs of operating system. Scheduling is the process of switching the Central Processing Unit(CPU) amongst the processes so that the CPU utilization can be optimized. There are different CPU Scheduling Algorithms which are used to accomplish this task. The optimum scheduling algorithm should have minimum waiting time, minimum turnaround time and should utilize the maximum CPU time.[7]

II. DEFINITION OF TERMS

- Ready Queue: The queue where the processes wait to be assigned to a processor.
- Burst Time: The time for which a process holds the CPU.
- Arrival Time: Time at which a process arrives at the ready queue.
- Throughput: Amount of work done per unit time by the processor.
- Waiting Time: Total time for which a process has been waiting in the ready queue.
- Turnaround Time:Total time taken between the submission of a program/process for execution and the completion of the process.[5]
- Response Time: Time needed by a system to respond to a particular process.

III. PREIMINARIES

There are various CPU scheduling algorithms each with different working mechanism. To understand the proposed algorithm, one needs to have the basic idea about two classical scheduling algorithms: Round Robin scheduling algorithm and Priority based scheduling algorithm.

- Round Robin scheduling: Round Robin scheduling is used in timesharing systems.[8] The basic mechanism is: a static time slot or time quantum is assigned to each process and after the quantum is over, the CPU time is given to the next process in the queue. The procedure continues until all the processes finish.The disadvantages of Round RobinScheduling are higher waiting and response time, low throughput.

- Priority based scheduling: In these types of algorithms, the operating system assigns a fixed priority to every process, and the scheduler arranges the processes in the ready queue in order of their priority. Lower priority processes get interrupted by incoming higher priority processes. Waiting time and response time depend on the priority of the process. Higher priority processes have smaller waiting and response times.

Most of the scheduling algorithms take into consideration one of the three parameters associated with the processes: *arrival time*, *burst time* and *priority assigned*. [6]

In this paper it is tried to take into consideration two parameters: burst time and priority. In the proposed algorithm a dynamic time quantum is calculated, which depends upon the burst time of the processes in the ready queue. After calculating the time quantum, the processes will be assigned the CPU according to their priority. Then the remaining burst times will be calculated and the priorities will change depending upon the remaining burst time. Process with less remaining burst time will get high priority. Then all the processes will get the CPU again according to their new priorities.

IV. Related Works

A major disadvantage of Round Robin scheduling algorithm is static time quantum. Different approaches were taken in last few years to improve the performance of Round Robin Scheduling by assigning dynamic time quantum like Min-Max Round Robin(MMRR)[1], Average Max Round Robin Algorithm (AMRR)[2], Shortest Remaining Burst Round Robin Algorithm (SRBRR), Improved Shortest Remaining Burst Round Robin Algorithm (ISRBRR)[3], Efficient Dynamic Round Robin Algorithm (EDRR)[4] etc. In this paper the proposed algorithm tries to give better turnaround time and average waiting time than ISRBRR and EDRR.

V. Proposed Scheduling Algorithm

A. Algorithm

The working principle of the proposed algorithm is as follows:

Step1: Sort the burst time in ascending order.

Step2: Get the Highest and Lowest burst time.

Step3: Calculate mean and median of the burst times.

Step4: If (mean>median)

TimeQuantum=ceil(sqrt((mean*highest)+(median*lowest))/2).

Else If(mean< median)

TimeQuantum=ceil(sqrt((median*highest)+(mean*lowest))/2).

Else

Time Quantum=ceil(mean/2).

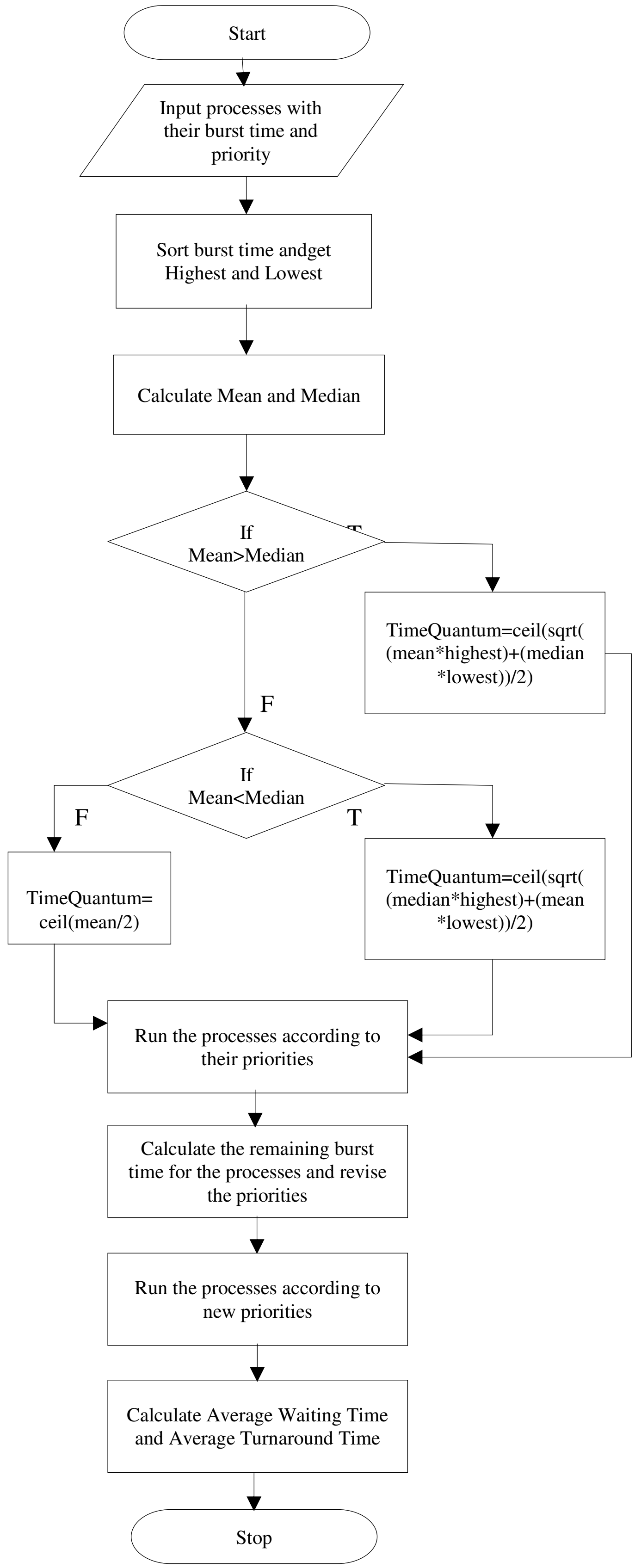
Step5: Run the processes according to their priority.

Step6: Priorities will be changed according to their remaining burst time. Process with lesser burst time will get the higher priority and process with higher burst time will get lower priority.

Step7: Repeat Step 6 until all the processes of ready queue complete their execution.

Step8: Calculate Average Waiting Time and Average Turnaround Time.

B. Flowchart


Start
Input processes with their burst time and priority
Sort burst time andget Highest and Lowest
Calculate Mean and Median
If Mean>Median
T
TimeQuantum=ceil(sqrt((mean*highest)+(median *lowest))/2)
F
If Mean<Median
F
T
TimeQuantum= ceil(mean/2)
TimeQuantum=ceil(sqrt((median*highest)+(mean *lowest))/2)
Run the processes according to their priorities
Calculate the remaining burst time for the processes and revise the priorities
Run the processes according to new priorities
Calculate Average Waiting Time and Average Turnaround Time
Stop

C. Illustration

Let us consider there are five processes (P0,P1,P2,P3,P4) with arriving time 0, Burst time (10,29,3,7,12) and Priority(1,4,2,3,5) respectively. According to our algorithm first of all the burst times will be sorted in ascending order in order to find the "Highest Burst Time" and "Lowest Burst Time". In this example,

Highest Burst Time= 29

Lowest Burst Time= 3

Now we need to calculate "Mean" and "Median" of burst times. In this example,

Mean= 12.2

Median= 10

Now, as here Mean > Median (12.2 > 10)

So, we will apply the following formula to calculate the "Time Quantum".

TimeQuantum=ceil(sqrt((mean*highest)+(median*lowest))/2)

In this example,

TimeQuantum = ceil(sqrt((12.2*29)+(10*3))/2) = 10

Now each process will get the CPU according to their priorities. Thus we get the following GANTT chart after all the processes get the CPU for the first time.

P0	P2	P3	P1	P4

0 10 13 20 30 40

In the next step, the remaining burst times of the processes will be calculated. Thus, the remaining burst time of the processes (P0,P1,P2,P3,P4) will be (0,19,0,0,2) respectively. As the process P4 has lower remaining burst time, it will be assigned higher priority, i.e, 1 and as process P1 has higher remaining burst time, it will be assigned lower priority, i.e, 2. As, rest of the processes have 0 remaining burst time, we can conclude that those processes finished their execution. So, only processes P1 and P4 will get the CPU for the next iteration. Thus the final GANTT chart is as follows

P0	P2	P3	P1	P4	P4	P1	P1

0 10 13 20 30 40 42 52 61

For this set of processes, after applying APBDRR we get the following Average Waiting Time (AWT) and Average Turnaround Time (ATAT).

AWT = 17.0

ATAT = 29.2

VI. EXPERIMENTAL RESULT

Case 1: We have applied the APBDRR Algorithm on a set of five processes (P0,P1,P2,P3,P4) with arriving time 0, Burst time (22,18,9,10,4) and Priority(4,2,1,3,5) respectively. Table-I shows the data set. Now, after using the algorithm we get the Gantt chart shown in Figure-I. Table-II shows the results obtained for this data after applying RR, ISRBRR, EDRR and APBDRR. Figure-II shows the comparison of AWT and ATAT for RR, ISRBRR, EDRR and APBDRR algorithms.

Here, TQ= Time Quantum

AWT= Average Waiting Time

ATAT= Average Turnaround Time

Table-I

Process	Arrival	Burst	Priority

Name	Time	Time	
P0	0	22	4
P1	0	18	2
P2	0	9	1
P3	0	10	3
P4	0	4	5

P2	P1	P3	P0	P4	P3	P1	P0	P0

0 9 18 27 36 40 41 50 59 63

Figure-I: GANTT CHART FOR APBDRR

Table-II

Algorithm	*TQ*	*AWT*	*ATAT*
RR	8	41.0	53.6
ISRBRR	15	40.2	52.8
EDRR	18	39.0	51.6
APBDRR	9	28.0	40.6

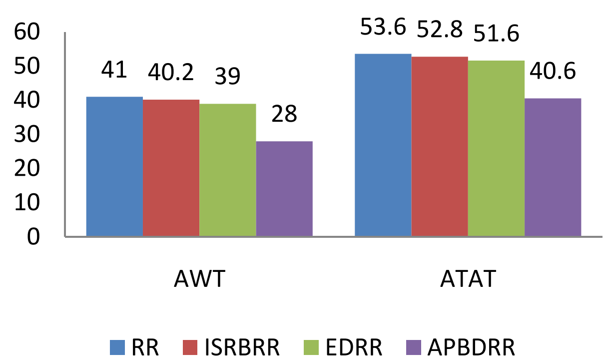


Figure-II: Comparative Analysis

Case 2:We have tested the proposed algorithm with 100 processes.We have fed the arrival time, burst time and priority against each process. The input data set is shown in Table-III. Table-IV shows the results obtained for 100 data after applying RR, ISRBRR, EDRR and APBDRR. Figure-II shows the comparison of AWT and ATAT for RR, ISRBRR, EDRR and APBDRR algorithms.

Table-III: Data set of 100 processes

AT	BT	PRIORITY	AT	BT	PRIORITY	AT	BT	PRIORITY	AT	BT	PRIORITY
0	13	1	25	45	8	50	99	16	75	49	50

1	38	53	26	37	49	51	66	66	76	95	62
2	83	74	27	45	39	52	74	52	77	26	64
3	93	10	28	66	5	53	90	69	78	4	8
4	50	4	29	72	36	54	57	79	79	7	9
5	23	70	30	14	3	55	54	32	80	51	10
6	87	78	31	70	30	56	45	22	81	67	79
7	98	4	32	77	15	57	25	76	82	63	51
8	39	54	33	98	24	58	53	78	83	21	11
9	12	44	34	73	73	59	100	34	84	47	26
10	83	38	35	35	16	60	19	52	85	35	21
11	31	67	36	55	11	61	88	30	86	93	67
12	4	15	37	15	4	62	78	35	87	74	76
13	82	35	38	55	67	63	4	5	88	50	15
14	22	68	39	24	13	64	93	13	89	44	64
15	13	49	40	33	60	65	93	59	90	60	36
16	65	2	41	44	36	66	47	31	91	10	12
17	25	53	42	82	45	67	60	57	92	45	72
18	5	2	43	60	24	68	20	22	93	12	69
19	96	35	44	82	51	69	6	7	94	39	70
20	53	5	45	8	16	70	39	11	95	5	6
21	97	5	46	24	37	71	90	8	96	46	36
22	39	19	47	79	21	72	51	42	97	72	56
23	66	18	48	60	36	73	3	6	98	15	13
24	13	65	49	21	2	74	65	63	99	49	51

Here, AT= Arrival Time, BT= Burst Time.

Table-IV

Algorithm	*TQ*	*AWT*	*ATAT*
RR	30	3222.055	3271.95
ISRBRR	70	3011.99	3271.95
EDRR	72	3004.59	3054.50
APBDRR	36	2853.36	2903.26

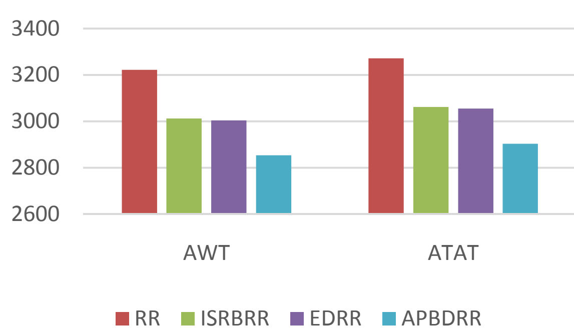


Figure-III: Comparative Analysis

VII. Conclusion

From the above experimental results we can conclude that the proposed APBDRR algorithm performs better than Round Robin Algorithm (RR), Improved Shortest Remaining Burst Round Robin Algorithm (ISRBRR) [3], and Efficient Dynamic Round Robin Algorithm (EDRR) [4] in terms of Average Waiting Time and Average Turnaround Time. Again we can also conclude that the algorithm works well and good for large number of processes also and is able to provide better Average Waiting Time and Average Turnaround Time for bulk of processes.

References

[1]. Sanjay Kumar Panda and Saurav Kumar Bhoi, "An Effective Round Robin Algorithm using Min-Max Dispersion Measure", International Journal on Computer Science and Engineering, 4(1), pp. 45-53, January 2012.

[2]. Pallab Banerjee, Probal Banerjee, Shweta Sonali Dhal, "Comparative Performance Analysis of Average Max Round Robin Scheduling Algorithm (AMRR) using Dynamic Time Quantum with Round Robin Scheduling Algorithm using Static Time Quanmtum", International Journal of Innovative Technology and Exploring Engineering, 1(3), pp. 56-62, August 2012.

[3]. P.Surendra Varma, "A Finest Time Quantum for Improving Shortest Remaining Burst Round Robin (SRBRR) Algorithm", Journal of Global Research in Computer Science, 4 (3), pp. 10-15, March 2013.

[4].

[5]. Raman, Dr.Pradeep Kumar Mittal, "An Efficient Dynamic Round Robin CPU Scheduling Algorithm (EDRR)", International Journal of Advanced Research in Computer Science and Software Engineering, 4(5), pp. 907-910, May 2014.

[6]. Silberschatz, A., P.B. Galvin and G. Gagne, Operating Systems Concepts. 7th Edn., John Wiley and Sons, USA., ISBN: 13: 978-0471694663, pp. 944.

[7]. Rakesh Mohanty, H. S. Behera, Khusbu Patwari, Monisha Dash, "Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm", Proc. of International Symposium on Computer Engineering & Technology 2010, Vol 17, pp. 126-137, 2010 .

[8]. R. J. Matarneh, "Seif-Adjustment Time Quantum in Round Robin Algorithm Depending on Burst Time of the Now Running Proceses", American Journal of Applied Sciences, 6(10), pp. 1831-1837, 2009.

[9]. H. S. Behera, R. Mohanty, and D. Nayak, "A New Proposed Dynamic Quantum with Re-Adjusted Round Robin Scheduling Algorithm and Its Performance Analysis", International Journal of Computer Applications, 5(5), pp. 10-15, August 2010.