Research Paper

by Hamza kabeer

"Dynamic Time Slicing Round Robin CPU Scheduling Algorithm"

M. Hamza Kabeer[1], Prof. Hassaan Rafiq[2], Amna Qudat[3], Maryum Saleem [4], Ahsan Mehmood[5], Umer Farooq[6]

hamzakabeercs@gmail.com

maryumsaleem390@gmail.com

amnaqudrat@outlook.com

Ahsanmahmood3210@gmail.com

umershaukat094@gmail.com

Department of Computer Science, Lahore Garrison University, DHA phase 6, Lahore, Pakistan

Abstract

Worldwide the most commonly used is the Round Robin algorithm that is considered to be the most efficient algorithm. But, as every algorithm has its limitations, it also do. Suppose if the quantum number that is selected, is too large than the expected response time is considered to be very high. On the other side of view, if the quantum number is supposed too small, then it might enhance the CPU overhead. Now we are proposing an algorithm that is not bound to follow the fixed quantum time, rather it creates a dynamic quantum number corresponding to process ready to be executed in the ready queue and hence in this way it increases the efficiency of the RR algorithm.

Key words: RR scheduling, CPU scheduling,

Dynamic Quantum, TAT, AWT.

1. Introduction:

A computer system performs a huge number of tasks, the efficiency of which depends upon various factors. Among the performance effective factors is at most the speed of a process that depends upon the algorithm used in the processor. Modern computing is moving towards the multitasking environment and core of this development depends on the speed of processor. Since speed depends upon the algorithm used in processor, so how efficient is the algorithm, so fast is your processor.

First Come First Serve (FCFS)-[1]. It is the type of algorithm in which the CPU serve the processes in accordance to their arrival. Let say, if the Process P1 reaches first then it will be served first and P2 reaches after the earlier P1, then it'll be served very next after P1 and in this way the CPU continues the facilitating these processes. Since this is the Non-Pre Emptive scheduling the CPU will perform fair scheduling.

Short-Job-First (SJF)-[2] It is the also one of the CPU scheduling strategy that is somehow considered to be more efficient then the FCFS, as it entertain the short jobs first that might be more important than the longer one and in FCSFS[5] this job remains in the waiting queue until the earlier task is performed. This situation causes the Convoy effect. In this Scheduling if the CPU is processing any task and there comes a task with burst time smaller than that of currently running task, the CPU will be Pre Empted and given to the job with smaller burst time.

This scheduling criteria is not considered to be fair as it creates a starvation which means if smaller processes are continuously entering in ready queue, then the process will be forced to ignore the processes with the greater burst time. And thus in this case ultimately the process will starvate.

Round-Robin (RR)-[3] [4] This is the algorithm that is more commonly used in devices as it keeps the CPU as busy as possible, but it also have some limitations that we have discussed earlier in abstract. This algorithm uses the phenomena of Time slicing and fairly schedule all processes to be executed unbiased of the length or burst time of the process.

2. Previous works:

Undoubtly, Round Robin becomes very famous algo throughout the systems, ignoring the discipline it follows of fixed quantum time. Almost this algorithm is used in every operating system.

Earlier, developers founded that an appropriate quantum time can be calculated by the median of time burst of all processes present in the ready queue, until the required median is less than 25ms. So, the quantum value should be modified to 25ms to avoid any overhead occurred of context switch time.

These efforts gave platform for the new and more efficient algorithms to be designed in order to utilize the CPU more and more and keep it busy as much as possible.

3. Algorithm presented:

In this offered paper the solution is presented to the problem of RR algorithm of being over headed due to the smaller burst time (SJF) [6] and too high response rate due to a larger process having a long burst time. The proposed algorithm is proved to be more efficient after several experiments.

3.1 Methodology:

This algorithm creates a dynamic time Quantum depending upon the average value of the time burst and the process placed in the queue that is ready to be executed. When a new process arrives it waits in this queue to be executed and when the quantum time over the CPU is preempted and given to that process which is waiting in the ready queue, after realizing the time burst of such process and considering it in the average of the total process in the ready queue.

4. Scheduling Criteria:

- A number of processes enter in the queue that is ready to be executed.
- The processes arrival time may differ or same to each other's arrival time.
- At the time Process**T1** enters the ready queue, if it is the very first and there's no other process then the quantum will vary to its burst time and the process will start it's execution in the processor.
- Meanwhile when another processes enter the ready queue during the execution of **T1**, then the average of all the time burst of processes waiting in the queue that is ready to be executed will be calculated.
- The next process after **T1** will be given the quantum number, the average calculated.
- In another case, if all processes arrives at the same time, let say at zero second, then the same criteria (taking average will be entertained) in the start.

5. Numerical Explanation:

CASE 1:

Suppose these four processes arrived at time = 0, with time burst (**T1** =10, **T2** = 20, **T3**= 30, **T4** = 20):

➔ Here PROCESS, P= T

a) Simple RR Algorithm

Process	Burst Time
T1	10
T2	20
T3	30
T4	20

Ready Queue:

T1	T2	T3	T4	T2	T3	T4	T3

Gantt chart:

T1	T2	T3	T4	T2	T3	T4	T3

0 10 20 30 40 50 60 70 80

Turn Around Time	52.5
Waiting Time	32.5

b) Dynamic Time Slicing RR algorithm:

Process	Burst Time
T1	10

T2	20
T3	30
T4	20

Ready Queue

T1	T2	T3	T4	T3	T4

Gantt chart:

T1	T2	T3	T4	T3	T4	
0	10	30	55	67.5	72.5	80

Turn Around Time	48.125
Waiting Time	28.125

CASE 2

Lets assume these four processes arrived at different time intervals:

a) Simple RR Algorithm

Process	Time Burst	Arrival Time
T1	10	0
T2	20	5
T3	30	15
T4	20	20

Ready Queue:

T1	T2	T3	T4	T2	T3	T4	T3

Gantt chart:

T1	T2	T3	T4	T2	T3	T4	T3	
0	10	20	30	40	50	60	70	80

Turn Around Time	42.5
Waiting Time	22.5

b) Dynamic Time Slicing RR algorithm:

Process	Burst Time	Arrival Time
T1	10	0
T2	20	5
T3	30	15

T4	20	20

Ready Queue

T2	T2	T3	T4	T3	T4

Gantt chart:

T1	T2	T3	T4	T3	T4	
0	10	30	55	67.5	72.5	80

Turn Around Time	38.125
Waiting Time	18.125

6. References:

[1] *F. Baccelli, P. Boyer, G. Hebuterne, "Single-Server Queue with Impatient Customers"*, Advance Appl. Probability, *vol. 16, 1984*

[2] *IJCSN - International Journal of Computer Science and Network, Volume 3, Issue 6, December 2014 ISSN (Online): 2277-5420 Impact Factor: 0.274*

[3] *Rakesh Kumar Yadav et. al. / (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 04, 2010, 1064-1066*

[4] *Mohanty and others also developed other algorithms in order to improve the scheduling algorithms performance. One of them is constructed as a combination of priority algorithm and RR while the other algorithm is much similar to a combination between SJF and RR.*

[5] *International Journal of Computer Applications (0975 – 888) Volume 47– No.13, June 2012*

Research Paper

ORIGINALITY REPORT

16%
SIMILARITY INDEX

15%
INTERNET SOURCES

7%
PUBLICATIONS

9%
STUDENT PAPERS

PRIMARY SOURCES

1 arxiv.org
Internet Source
7%

2 myslide.es
Internet Source
4%

3 www.enggjournals.com
Internet Source
1%

4 Sulochan Naik, Jayprakash T Lalchandani. "An Optimal Disk Scheduling Technique", 2018 3rd International Conference for Convergence in Technology (I2CT), 2018
Publication
1%

5 ijcsn.org
Internet
1

Source

%

6 www.dtic.mil

Internet Source

1 %

Exclude quotes Off

Exclude bibliography Off

Exclude matches Off