Shortest Job First Round Robin Burst Time based Time Quantum Scheduling Algorithm

ABSTRACT

Shortest Job First Round Robin (SJF RR) is a hybrid scheduling

algorithm that specializes in priority scheduling of lowest burst

time in a single CPU environment. Generally, Shortest Job First is

a scheduling algorithm in which the process with the smallest

execution time is selected for execution next. Owing to its simple

nature, the shortest job first is considered optimal. It also reduces

the average waiting time for other processes awaiting execution.

On the other hand, Round Robin provides better average response

time that typically comes at the cost of higher average turnaround

time and it also assigned a fixed time slot in a cyclic way. It is

pre-emptive as processes are assigned CPU only for a fixed slice

of time at most. In corresponds to the two algorithms, Round

Robin throughput largely depends on the choice of the length of

time algorithm. While the Shortest Job First selects the process

with the shortest time of execution and executes the process.

When the CPU is available, the process with the shortest CPU

burst is selected. Combining the two algorithms will reduce the

lowest burst time of both scheduling algorithms creating a hybrid

and improved scheduling algorithm.

Keywords

Scheduling Algorithm; Time Quantum; Burst Time;

Arrival Time; Turnaround Time; Round Robin; Shortest Job

First;.

1. INTRODUCTION

CPU scheduling is a technique that approves one

method to use the CPU whilst the execution of every other

system is on hold (in waiting for state) due to the

unavailability of any useful resource. The purpose of CPU

scheduling is to make the gadget efficient, quick and fair and

it is important because it can have a big effect on resource

utilization and the overall performance of the system. [1] As

such, the scheduling algorithm is a set of rules that determines

the task to be executed at a particular moment. In spite of the

fact that there are a few packet scheduling calculations that

have been proposed within the literature, the design of those

algorithms is challenged by the need for supporting different

levels of services, fairness, and in execution complexity and

so on.

This paper focuses on the improvement of the

existing scheduling algorithm based on a Shortest Job First

which selects the waiting process with the smallest execution

time to execute next and the Round Robin which assigns a

fixed priority rank to each process.

Shortest-Job-First (SJF) in this algorithm the process

which has minimum CPU burst time will schedule first. This

algorithm may lead to a problem that we cannot predict how

long a job will be executed. Round Robin (RR) algorithm is

one of the oldest, simplest, and most widely used scheduling

algorithms. A time slice is given to every process and every

process will be executed for a particular defined time slice.

New processes are added at the end of the ready queue. The

scheduler picks the process from the starting point of the ready

queue and sets the timer to a defined time slice and also sets an

interrupt. If the process is still not completed its complete

execution within a time slice it will be preempted after a time

slice and added at the end of the ready queue. [2]

Burst Time is the time when the process is being

executed in the CPU, i.e. CPU is the resource being used by the

process at that time. I/O burst- The time when the process

requests for I/O and is using I/O as a resource example.

Scheduling algorithms are the mechanism by which a useful

resource is allotted to a procedure or task. CPU scheduling is

the mechanism through which a resource is to a process or task

and executes in different ways. [3] There are several

scheduling algorithms available as FCFS (First Come First

Serve), SJF (Shortest Job First), Priority Scheduling and RR

(Round Robin) scheduling algorithm. These scheduling

Algorithms are used to minimize the turnaround time, response

time, waiting time and no. of context switching. [4] The

processes are scheduled in accordance with the given burst

time, arrival time and priority. The execution of approaches

used a number of resources such as Memory, CPU, etc. [5] A

scheduling decision implies to the theory of selecting the next

process to execute. In each scheduling decision, a context

switch may occur, meaning that the current process will stop

executing and put back to the ready queue (or some other

place) and another process will be dispatched. We define the

scheduling overhead cost which includes context switching

time. Since processes are switching the CPU performance will

be decreased. Scheduling algorithms are extensively used in

communication networks and in operating systems to assign

resources to competing processes.

The two algorithms will be combined with one

another to make a new and better version of the older version of

the algorithms thus calling it Shortest Job First Round Robin

Scheduling Algorithm using Priority Scheduling of Lowest

Burst Time.

II. RELATED WORKS

The Review of literature is the work done by the earlier authors

in the direction of the process scheduling algorithm and process

management is stated. E.g;

In the year 2012, Manish Kumar Mishra [6] describes

an improvement in RR. IRR picks the first process from the ready

queue and allocates the CPU to it for a time interval of up to

1-time quantum. After completion of the process‟s time quantum,

it checks the remaining CPU burst time of the currently running

process. If the remaining CPU burst time of the currently running

process is less than 1-time quantum, the CPU again allocated to

the currently running process for remaining CPU burst time. In

the year 2009, Rami J. Matarneh [7] performed a work SelfAdjustment-Round-Robin (SARR) based on a new approach

called dynamic time quantum, in this approach the time quantum

repeatedly adjusted according to the time of the now-running

processes. In the year 2011, Saroj Hiranwal in which first of all

we arrange the processes according to the execution time/burst

time in increasing order that is smallest the burst time higher the

priority of the running process. The smart time slice is equal to the

mid-process burst time of all CPU burst time.. The year 2013,

Aashna Bisht [8] performed a work Enhanced Round

Robin(ERR), which modify the time quantum of only those

processes which require a slightly greater time than the allotted

time quantum cycle. The remaining process will be executed in

the conventional Round robin manner. The year 2011, Rakash

Mohanty & Manas Das [9] performed work in which a new

variant of Round Robin scheduling algorithms by executing the

processes according to the new calculated Fit factor „f‟ and using

the concept of dynamic time quantum. In the year 2012,

Debashree Nayak [10] performed a work in which a median plus

some other value is added in time quantum. This scheduling

algorithm, which is a combination of SJF & RR. In the year 2012,

Ishwari Singh Rajput, Deepa Gupta [11] proposed priority-based

Round-robin CPU scheduling algorithms is based on the

integration of round-robin and priority scheduling. It retains the

advantage of a round-robin in reducing starvation and also

integrates the advantage of priority scheduling. The Proposed new

algorithm also implements the concepts of aging by assigning new

priorities to the processes. In the year 2012, P.SurendraVarma

[12] performed a work, In which the improved version of SRBRR

(Shortest Remaining Burst Round Robin) by assigning the

processor to processes with the shortest remaining burst in

round-robin manner using the best possible time quantum. In this

paper, the time quantum is computed with the help of the median

and highest burst time. In the year 2012, H.S. Behera & Brajendra

Kumar Swain [13] performed a work it gives precedence to all

processes according to their priority and burst time, then applies

the Round Robin algorithm on it. This Proposed algorithm is

developed by taking dynamic mean time quantum into account..

In the year 2010, RakeshkumarYadav, NavinPrakash and

Himanshu Sharma [14] performed a work. In this paper, the new

algorithm is used Round Robin with the shortest Job first

scheduling. In which allocate all processes to the CPU only one

time as like present RR Scheduling algorithm. After it, we select

the shortest job from the waiting queue and its shortest job assign

to the CPU. After that next shortest job is selected. In the year

2012, Ali Jbaeer Dawood [15] performed work. The processes

were ascending with the shortest remaining burst time and

calculate the TQ from multiplying the average summation of

minimum and maximum BT by (80) percentage. [16]: In this

algorithm, methods are accomplished in accordance to their

arrival times. Once a process is accomplished for the defined time

quantum, instead of switching to the next process, access whether

the resultant burst-time of the current process is less than two or

equal to two time-quantum value If yes, the identical procedure is

executed. Otherwise, process execution occurs as in regular RR.

This method reduces variety of context switches as compared to

conventional RR. Improved Mean Round Robin with Shortest Job

First Scheduling (IMRRSJF) [17]: This algorithm has combined

elements of each RR two and SJF, ie, the processes are arranged

according to the burst-time values and time-quantum is calculated

as the square root of the product of mean and highest burst

time.Whenever a new procedure comes in, the techniques are

again sorted in ascending order and time quantum is calculated

again.

.III. METHODOLOGY

3.1 Assumptions:

1. The arrival times of the set of processes are assumed to be 0.

2. The burst times of each process are assumed to be known.

3.2 Proposed Methodology:

The proposed methodology works similarly to

Round-Robin with added features of IMRRSJF to efficiently

compute for the Time-Quantum and ADRR to be able to lessen

the number of Context Switches.

3.3 Approach:

The processes present in the Ready Queue (RQ) are

sorted in ascending order. Time Quantum (TQ) is computed using

the formula used in IMRRSJF. Time Quantum = Square root of

(Mean of Burst-Times\*Highest Burst-Time). Each process will

now be executed for the TQ. Calculation of the remaining

Burst-Time will begin and if the remaining Burst-Time is less

than or equal to the TQ, the same process will be executed until it

completes. Then it will continue to execute the next process until

every process is done.

IV. PSEUDOCODE

Step 1: start

Step 2: sort the processes in increasing order of BT // BT =

Burst-Time

Step 3: compute for the TQ

TQ = √(mean of BT\*highest BT) // TQ = Time-Quantum

Step 4: keep doing steps 5 and 6 until all processes are done

Step 5a: if rem\_BT > 0 //rem\_BT = Remaining Burst-Time

allocate CPU for time quantum and move on to the

next process

Step 5b: if rem\_BT <= TQ

do step 6

Step 6: else

allocate CPU for remaining time then calculate for its

WT

// WT = Waiting-Time

Step 7: calculate for the processes' TAT // TAT =

Turnaround-Time

Step 8: calculate the AWT and ATAT

// AWT = Average Waiting-Time, ATAT = Average

Turnaround-Time

Step 9: stop

V. EXPERIMENTS AND RESULTS

In this section, we make a comparison of the performance

measures of the proposed algorithm with RR, ADRR, IMRRSJF

and SJFRRBTTQ algorithm.

Round Robin Algorithm:

Let us take the time quantum value as 25ms. Conventional RR

executes processes in the order of arrival time. For the processes

given in Table 1, the Gantt chart as per RR algorithm is as below:

P1 P2 P3 P4 P5

13 38 63 88 113

P2 P3 P4 P5 P2

138 163 188 195 211

P3 P4 P4

218 243 249

Average Waiting-Time = 127.4

Average Turnaround Time = 177.2

Context Switches = 12

ADRR:

Here also we assume the time quantum be 25ms. The Gantt chart

of process execution is as follows

P1 P2 P3 P4 P5

13 45 70 95 120

P3 P4 P5 P5

152 193 218 249

Average Waiting-Time = 80.6

Average Turnaround Time = 130.4

Context Switches = 8

IMRRSJF:

Here the processes are sorted in increasing order of burst-time and

the time-quantum is: TQ = √( mean\*highest burst-time) = 63

P1 P2 P3 P4 P5

13 45 102 165 228

P4 P5

231 249

Average Waiting Time = 78.2

Average Turnaround Time = 128.0

Context Switches = 6

SJFRRBTT:

Here the processes are also sorted in an increasing order of

burst-time and the time-quantum is: TQ = √( mean\*highest burst-time) =

63

P1 P2 P3 P4 P5

13 45 102 168 249

Average Waiting Time = 65.6

Average Turnaround Time = 115.4

Context Switches = 4

5.1Comparison of algorithms:

Based on the experimental analysis done in the previous section,

we compare the algorithms with respect to following performance

measures:

1. Average waiting-time

2. Average Turnaround time

3. Number of Context Switches

Algorithm TQ AWT ATAT CS

RR 25 127.4 177.2 12

ADRR 25 80.6 130.4 8

IMRRSJF 63 78.2 128 6

SJFRRBTTQ 63 65.6 115.4 4

Graphical Comparison of average waiting-time in various

algorithms:

Fig. 1. Average Waiting Time Comparison

Graphical comparison of average turnaround time in various

algorithm:

Fig. 2. Average Turnaround Time Comparison

Graphical comparison of the number of Context Switches

Fig. 3. Number Of Context Switches Comparison

VI. Conclusion

The Proposed Algorithm provides improved performance as is

evident from the experiment results of previous section. The

number of context switches is lessened compared to RR

algorithm. It also reduces both Average Waiting-time and

Average Turnaround Time.

VII. Future Works

SJFRRBTTQ algorithm can be further enhanced by integrating

appropriate strategy to control starvation, and can also be

modified to consider arrival-time of the different processes.