A Hybrid Round Robin Scheduling Mechanism for Process Management

Khaji Faizan Ali  
SCOPE, VIT, Vellore  
TN, India

Abhijeet Marikal  
SCOPE, VIT, Vellore  
TN, India

Kakelli Anil Kumar, PhD  
Associate Professor  
SCOPE, VIT, Vellore  
TN, India

ABSTRACT
Every process which runs in an operating system is processed through multiprogramming, by sharing the processor. Multiprogramming is a capability of the operating system to execute one or a lot of programs in a single processing machine. Multiprogramming's main objective is to optimize the use of CPUs, throughput and minimize average waiting time (AWT), the average turnaround time (ATT), average response time (ART) and the number of context switches (NOC). Multi-programming can be performed by several CPU scheduling algorithms. Most preferable scheduling algorithm among all is Round robin (RR). RR algorithm is mainly dependent on time quantum which is stationary. Hence, the proposed work is new approach to the round-robin scheduling algorithm (RR algorithm) with an enhanced time quantum based algorithm. This enhancement of RR algorithm by using dynamic time quantum leads to minimize AWT, ATT, ART and NOC. This approach inherits the properties of Round robin, shortest job first (SJF) algorithm and first come first serve algorithm (FCFS). Therefore, the proposed algorithm is a hybrid round robin scheduling mechanism for process management (HYRR Mechanism). HYRR Mechanism is an innovative scheduling algorithm which reduces ATT, ART, AWT and NOC to the desired levels.

Keywords
Scheduling Algorithm, number of context switches, Process Management, Dynamic time Quantum.

1. INTRODUCTION
Operating system (OS) organizes the processes by allotting different tasks properly according to their arrival time [1]. To increase the CPU utilization, Operating system's primary role is to allow many processes concurrently [2, 3]. There are different types of CPU scheduling algorithm which perform differently according to the situation, mainly CPU-scheduling algorithms concentrate on increasing CPU performance by reducing turnaround time, the number of the context switch and waiting time [4, 5]. In this work, the proposed algorithm improves the capability of the round-robin-algorithm by introducing a new enhanced quantum time in place of the time quantum in round-robin.

The proposed work inherits the properties of RR Algorithm, SJF algorithm and other algorithms. In SJF Algorithm, length of the process is associated with CPU. Lengths are to schedule the process with the shortest time [6]. Once the Processor is free, it is given to the process which has the least next CPU. It is also called as shortest CPU [7]. FCFS scheduling is used for disrupting the knot when next two processes have same-length [8]. SJF stays ideal, so it gives least AWT for process [9]. In RR algorithm, a fixed time slot is cyclically allocated to each process [10]. Most widely used CPU scheduling strategies as a foundation. Because of time slots, processes don’t suffer starvation [11]. More number of context switches takes place which is a big disadvantage of RR Algorithm [12]. In this paper, the proposed mechanism tries to reduce NOC. Some of the basic terms used in this paper are, waiting time: the time that a process stays or waits in the ready queue [13].

Turnaround time: The time it takes for a CPU to execute a process. It is measured as the time between the request and completion of a phase [14].

CPU utilization: CPU is utilized to its extreme extents by keeping CPU busy.

Response time: Amount of time taken by process to get response from its initial request [15].

Throughput: Throughput is the rate of completion of processes per unit of time [14].

Context switch: switching of one process in CPU with another process which already is present in ready queue. While switching, the state of a currently executing process is preserved and CPU is loaded with a new process from its saved state is called context switch [16].

Therefore, to be a suitable scheduling algorithm, it must possess characteristics such as minimum waiting time, minimum turnaround time, maximum CPU utilization, minimum response time, maximum throughput and a minimum number of the context switch [17].

2. PROPOSED HYRR MECHANISM
This algorithm mainly focuses on improvement of Round robin scheduling algorithm. It maximizes CPU utilization, throughput and minimizes AWT, ATT, ART and NOC. This proposed algorithm works more effectively than the RR Algorithm. In HYRR Mechanism, time quantum is calculated dynamically by using the mean of burst time and the minimum of burst time. In phase-1, Enhanced quantum time (ETQ) is calculated using mean and minimum of burst time. After calculation, Process which has shortest burst time and which is not executed in CPU is given high priority and allocated in CPU for 1 Enhanced quantum time. Phase-1 is repeated until every process gets allocated in CPU for once. In phase-2, ready queue processes are ordered in ascending based on their remaining burst time. After arrangement, process which is first in ready queue is allocated in CPU for 1 enhanced quantum time. After allocation, if currently executing process remained burst time in CPU is less than or equivalent to 1ETQ then current process is reallocated in CPU for its execution. Phase-2 is repeated until the ready queue becomes empty.
2.1 Flow Chart

2.2 HYRR Algorithm
Step 1. Calculate average of CPU time of all process. Average = (P1+P2+P3+P4...... PN)/N

Step 2. Calculate enhanced quantum time by given formula. Enhanced quantum time= ceiling (average + minimum time)/2

Step 3. Select the process with minimum time, which is not executed even for 1 time in CPU.

Step 4. Allocate the selected process to CPU from ready queue up to 1 Enhanced quantum time and repeat STEP 3 until all processes in get executed for one time in CPU.

Step 5. After completion of step 4 order all the processes in ascending order of their remaining burst time with in the Ready queue state.

Step 6. A Process which is first in ready queue state is allocated to CPU for Enhanced Quantum time.

Step 7. The current process is reallocated in CPU if remaining process burst time is less than or equivalent to 1ETQ. If Process gets entirely executed in CPU, then it is discarded from ready queue state Else, Place the current process at the end of the ready queue for executing it in future.

Step 8. Allocate following process from the queue to CPU for the period of enhanced quantum time and recheck the step 7.

Step 9. Execute all processes until the queue is empty. Step 10. Calculate NOC, ATT, AWT and ART of processes.

3. EXPERIMENT AND RESULTS INVESTIGATION

Different processes are outlined with CPU Burst time and arrival time. These processes are scheduled in both HYRR Mechanism and RR Algorithm. After scheduling, ATT, AWT, ART, NOC and results are compared.

3.1 Experiment-1
In this experiment five process are defined only with Burst time as Pr0, Pr1, Pr2, Pr3, and Pr4. Time quantum of RR Algorithms=3ms.

<table>
<thead>
<tr>
<th>PROCESS-ID</th>
<th>TIME</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr0</td>
<td>15</td>
</tr>
<tr>
<td>Pr1</td>
<td>6</td>
</tr>
<tr>
<td>Pr2</td>
<td>1</td>
</tr>
<tr>
<td>Pr3</td>
<td>4</td>
</tr>
<tr>
<td>Pr4</td>
<td>11</td>
</tr>
</tbody>
</table>

3.1.1 RR-ALGORITHM:
Let time quantum be 5ms.

3.1.1.1 Gantt chart:

Fig-2: Gantt chart
Table 2. RR-Algorithm Results

<table>
<thead>
<tr>
<th>PROCESS ID</th>
<th>WAITING TIME (ms)</th>
<th>TURN AROUND TIME (ms)</th>
<th>RESPONSE TIME (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr0</td>
<td>21</td>
<td>36</td>
<td>0</td>
</tr>
<tr>
<td>Pr1</td>
<td>20</td>
<td>26</td>
<td>5</td>
</tr>
<tr>
<td>Pr2</td>
<td>10</td>
<td>11</td>
<td>10</td>
</tr>
<tr>
<td>Pr3</td>
<td>11</td>
<td>15</td>
<td>11</td>
</tr>
<tr>
<td>Pr4</td>
<td>26</td>
<td>37</td>
<td>15</td>
</tr>
</tbody>
</table>

Therefore,
NOC=9
AWT=17.6ms
ATT=25ms
ART=8.2ms

3.1.2 Proposed Hybrid Round Robin Scheduling Mechanism (HYRR):
Mean=7.4ms
Minimum burst time=1ms
ETQ=ceiling(mean+minimum)/2=ceil ((7.4+1)/2) =5ms
ETQ=5ms

3.1.2.1 Gantt chart:

![Gantt chart](image)

From Table 4 and Fig 4, the NOC, ART, ATT and AWT of proposed HYRR Mechanism has drastically decreased when compared with RR algorithm.

Table 4. Experiment-1 Result analysis based on processes arriving at same time.

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>RR ALGORITHM</th>
<th>HYRR MECHANISM (Proposed Algorithm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWT</td>
<td>17.6</td>
<td>10.8ms</td>
</tr>
<tr>
<td>ATT</td>
<td>25ms</td>
<td>18.2ms</td>
</tr>
<tr>
<td>ART</td>
<td>8.2ms</td>
<td>6.2ms</td>
</tr>
<tr>
<td>NOC</td>
<td>9</td>
<td>7</td>
</tr>
</tbody>
</table>

![Comparison Graph](image)
3.2 Experiment-2:
In this experiment six processes are defined with time and arrival time as Pr1, Pr2, Pr3, Pr4, Pr5 and Pr6.
Time quantum of RR algorithm=3ms

<table>
<thead>
<tr>
<th>PROCESS-ID</th>
<th>ARRIVAL TIME (ms)</th>
<th>BURST TIME (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>Pr2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Pr3</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>Pr4</td>
<td>0</td>
<td>8</td>
</tr>
<tr>
<td>Pr5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Pr6</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 5. Process Id with their burst time

3.2.1 RR-ALGORITHM:
Time quantum=3ms

3.2.1.1 Gantt Chart:

<table>
<thead>
<tr>
<th>Process</th>
<th>Waiting Time (ms)</th>
<th>Turn Around Time (ms)</th>
<th>Response Time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr5</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Pr6</td>
<td>16</td>
<td>13</td>
<td>16</td>
</tr>
</tbody>
</table>

Therefore, NOC=11
AWT= 15ms
ATT=18ms
ART=4.66ms

3.2.2 Proposed Algorithm:
Mean=4.5ms
Minimum time=1ms
ETQ=3ms

3.2.2.1 Gnatt chart:

Fig-6: GNATT chart

Table 7. Proposed Mechanism Results

<table>
<thead>
<tr>
<th>PROCESS ID</th>
<th>WAITING TIME (ms)</th>
<th>TURN AROUND TIME (ms)</th>
<th>RESPONSE TIME (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pr4</td>
<td>19</td>
<td>27</td>
<td>0</td>
</tr>
<tr>
<td>Pr5</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Pr6</td>
<td>2</td>
<td>4</td>
<td>7</td>
</tr>
</tbody>
</table>

Therefore,
NOC=9
AWT= 9ms
ATT=13.5ms ART=3.33ms
**Table 8. Experiment-1 Result analysis based on processes arrival time**

<table>
<thead>
<tr>
<th>ALGORITHM</th>
<th>RR Algorithm</th>
<th>HYRR Mechanism (Proposed Algorithm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>AWT</td>
<td>15ms</td>
<td>9ms</td>
</tr>
<tr>
<td>ATT</td>
<td>18ms</td>
<td>13.5ms</td>
</tr>
<tr>
<td>ART</td>
<td>4.66ms</td>
<td>3.33ms</td>
</tr>
<tr>
<td>NOC</td>
<td>11ms</td>
<td>9 ms</td>
</tr>
</tbody>
</table>

**Fig 5: comparison graph of Round-Robin and proposed Hybrid Round Robin Mechanism**

From above Table 7 and Fig 5, the ATT, AWT, ART and the NOC of proposed hybrid round-robin scheduling Mechanism has drastically decreased when compared with RR algorithm

5. CONCLUSION

In this paper the performance of traditional RR algorithm and proposed HYRR algorithm has been investigated and compared the results in terms of AWT, ATT, ART and NOC. The proposed mechanism is more effective in terms of maximum CPU utilization, throughput and minimized AWT, ATT, ART and NOC. The main reason is dynamic time quantum instead of static time quantum. The proposed mechanism can improve the performance of operating system in terms of better process management and optimization.

6. REFERENCES


