CPU Scheduling Algorithm Using Dynamic Time Quantum for Batch Systems

Nayana Kundargi
Department of Master of Computer Applications
Gogte Institute of Technology, Belgaum, Karnataka, India

Sheetal Bandekar
Faculty of Department of Master of Computer Applications
Gogte Institute of Technology, Belgaum, Karnataka, India

Abstract- Round-Robin is one of the scheduling algorithm which has been implemented by various operating systems having many advantages which include each process gets fair chance to execute its job and on the other hand this algorithm can be improvised in the area of time quantum. And according to this algorithm If time quantum is high it behaves as First Come First Serve basis, on the other hand if it is low the operating system has to concentrate more on context switching.

The aim of this paper is to improvise the Round-Robin algorithm and there by propose a new NK algorithm which computes the time quantum, by selecting the burst time depending on the set of available processes dynamically.

Keywords – Scheduling algorithm, Round Robin, Time Quantum, Operating system.

I. INTRODUCTION

Operating system has a task of allocating the set of arrived process to perform its task. As there are various algorithms which are implemented depending on its individual advantages and disadvantages. Round Robin is one of the CPU scheduling algorithm implemented in operating system such as Windows, UNIX, and Unix based systems etc.

CPU scheduling is the process of allocating CPU for a specific process by a time slice. Scheduling of the process is done on some criteria of the algorithm. Criteria include:

- CPU utilization: CPU should be kept as busy as possible. That is, it should involve a maximum usage.
- Throughput: Time in one measure of work that the numbers of processes are completed per unit time, is supposed to be maximum.
- Turn Around time: The time that is involved from the time of submission of a process till its completion, is supposed to be minimum.
- Waiting time: The addition of the time that the process is waiting for its execution, is supposed to be minimum.
- Response time: The time that the process is submitted for a request until the first response is produced, is supposed to be minimum.

It is desirable to optimize the maximum and minimum values depending on the average. For example, to guarantee that all users get good service we may minimize the maximum response time.

The rest of the paper is organized as follows. Existing algorithm and proposed algorithm are explained in section II. Experimental results are presented in section III. Concluding remarks are given in section IV.

II. EXISTING AND PROPOSED ALGORITHM

A. AN algorithm (Existing algorithm) –

As Round Robin is one of the widely accepted algorithm in operating system, measures have been taken to eliminate its disadvantages of the fixed time quantum. Researchers have found various solutions to eliminate this gap.

Abbas Noon and his colleagues have found a solution of fixed time quantum by finding the average of the burst time of processes in the ready queue and then they modified the time quantum by subtracting the time consumed by
this process and the remaining burst time of same process by again finding the average, and updating the new time quantum as the resulted average attained[4].

B. NK algorithm (Proposed algorithm) –

When a process arrives for the first time the time quantum is equivalent to the burst time of the process arrived. If there are more then one process arrived at same time then the time quantum is computed by taking the average burst time of the process being arrived. If more than one process arrives at the same time then the average of the burst time is computed and the time quantum is initialized. In this situation the processes are placed in the queue and the scheduling takes place depending on the first-come-first-serve basis. In this method we use two variables:

- Sum: Stores the sum of all the burst time of the processes.
- Average: Stores the average of the burst time.

Pseudo code

New process P arrives
P enters ready queue
Compute sum and average
Process P is loaded from ready queue into the CPU to be executed
If (ready queue is empty)
    Time Quantum <- Burst time(P)
    Compute Sum and Average
End if
If (ready queue is not empty)
    Time quantum <- Average of burst time
End if
Execute P by Time Quantum
If (P is not finished)
    Return P to the ready queue
End if.

II. EXPERIMENT AND RESULT

The efficiency of AN algorithm is improved by changing the method of calculating the time quantum.

Case 1: Assume processes arrived at time interval = 0 with burst time 20, 40, 60, and 80.

<table>
<thead>
<tr>
<th></th>
<th>AN Algorithm</th>
<th>NK Algorithm</th>
<th>Performance Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn-Around Time</td>
<td>100</td>
<td>75</td>
<td>25%</td>
</tr>
<tr>
<td>Waiting Time</td>
<td>50</td>
<td>25</td>
<td>55%</td>
</tr>
</tbody>
</table>

Case 2: Assume processes arrived at time interval = 0 with burst time 50, 100, 60, and 70.

<table>
<thead>
<tr>
<th></th>
<th>AN Algorithm</th>
<th>NK Algorithm</th>
<th>Performance Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn-Around Time</td>
<td>172</td>
<td>137</td>
<td>35%</td>
</tr>
<tr>
<td>Waiting Time</td>
<td>102</td>
<td>67</td>
<td>35%</td>
</tr>
</tbody>
</table>
Case 3: Assume processes arrived at time interval = 0 with burst time 0, 50, 199, and 72.

<table>
<thead>
<tr>
<th></th>
<th>AN Algorithm</th>
<th>NK Algorithm</th>
<th>Performance Increase</th>
</tr>
</thead>
<tbody>
<tr>
<td>Turn-Around Time</td>
<td>155</td>
<td>118</td>
<td>37%</td>
</tr>
<tr>
<td>Waiting Time</td>
<td>74</td>
<td>38</td>
<td>36%</td>
</tr>
</tbody>
</table>

An analysis of the above cases (Case 1, Case 2, Case 3) shows that the time quantum of NK algorithm is better than the time quantum of AN algorithm. Also the decrease in Average Turn around time is 32.33% and decrease in average waiting time is 42%.

IV. CONCLUSION

By comparing the existing algorithm (“AN” algorithm) and the proposed algorithm (NK algorithm) a better results has been obtained as the Turn around time, response time, CPU utilization and waiting time are minimum compared to AN algorithm. A solution has been implemented by NK algorithm that allows the operating system to select the time quantum dynamically.

To conclude, NK algorithm is implemented to obtain an optimal scheduling algorithm which is better than AN algorithm.

REFERENCES


