Dynamic Time Quantum in Shortest Job First Scheduling Algorithm (DTQSJF) for Unpredictable Burst Time of Processes

Dibyendu Barman
Department of Computer Science and Engineering
Government College of Engineering and Textile Technology
Berhampore, Mushidabad, WB, India
dibyendu.barman@gmail.com

Iti Sarkar
Department of Computer Application
BARLOW GIRLS' H.S (H.S), Malda
Malda, West Bengal, India
itisarkar@yahoo.in

Abstract- In multitasking and time sharing operating system the performance of the CPU depends on waiting time, response time, turnaround time, context switches from the process mainly depends on the scheduling algorithm. Shortest Job First(SJF) is one of the most widely used scheduling algorithm because it give less Average waiting Time compare to other Scheduling algorithm but this algorithm have some disadvantages. Here burst time of processes have to predict (i.e. how much time process will execute in CPU) before CPU start execution. But it is quite impossible to predict the Burst time of process before process complete execution and if processes did not finish their execution in predicted burst time then in which algorithm process will execute? If they execute in FCFS manner Average waiting time may increase and there is a possibility to occur starvation. In this paper based on the experiments and calculations a new algorithm is introduced. In this algorithm the main idea is if processes remain in ready queue after completion of their predicted Burst Time, a new quantum will generate dynamically and each non completed process will get that amount of time quantum to execute in CPU if any process does not completed after execution of that amount of time (Time Quantum) it will go to tail of the ready queue and waiting for it's turn.

Keywords- SJF, Turnaround Time, Waiting Time, CPU scheduling, Context Switch

I. INTRODUCTION

Operating System is a platform where user can execute programme in well and efficient manner. It is an interface between user and computer hardware. Modern operating system is more complex with the features of multiprocessing and multitasking. In this type of operating system more than one process in the job pool or ready queue waiting its turn to be assigned to the CPU. Process is allocated to CPU requires requires careful awareness that starvation not occur. Scheduling algorithm are mechanism to allocate resources or processes to CPU and execute different way to decrease turnaround time, waiting time, response time and number of context switches.

A. PRELIMINIARIES

Programme in execution call process represents by Process Control Block(PCB) contain different information of process such as process number, process state, register, list of open files, CPU scheduling information. When process enters into the main memory is kept in the Ready Queue. The processes which are waiting for I/O request are kept in Device Queue. The Long term scheduler or job scheduler select process from job pool and load them into main memory for execution. Short term scheduler or CPU scheduler select from among the processes that are ready to execute and allocates the CPU to one of them. Medium term scheduler is used in time sharing system.

B. PERFORMANCE CRITERIA

The Proposed algorithm is designed to meet all scheduling criteria such as maximum CPU utilization, maximum throughput, minimum turnaround time, minimum waiting time, context switches. Now what is arrival time, Burs Time, Waiting Time, Turnaround Time, Response Time, and Throughput?

Arrival Time: Arrival time is the time at which process arrive at main memory.

Burst Time: Burst time is the time for which process holds the CPU.

Waiting Time: Waiting time is amount of time process waiting in ready queue.

Turnaround Time: Turnaround time means time of arrival minus time of completion.

Response Time: Response time means time of arrival minus first response by CPU.

Throughput: Number of process completed per unit time

C. SCHEDULING ALGORITHM

There exists different CPU scheduling algorithm like FCFS (First Cum First Serve), Priority Scheduling, SJF (Shortest Job First), Round Robin (RR) scheduling with some advantages and disadvantages. FCFS is simplest scheduling algorithm where processes are allocated to CPU according to their arrival time. Its average waiting time quite long so it is not suitable for real time application. In priority Scheduling process is allocated to CPU according to their priority but here is a problem that high priority process may not allow low priority to access CPU which may cause starvation and solution to this problem is aging. Round Robin scheduling algorithm is simple and most widely used algorithm especially for time-sharing system. All process are kept in circular queue and CPU scheduler goes around the queue allocating the CPU to each process small unit of time call time slices or time quantum. New arriving processes are placed in tail of the queue. The Scheduler picks up first process of the queue to and sets a timer to interrupt after one time quantum and dispatches the process. If the process still running after completion of one Time Quantum the CPU pre-empted and place the process at tail of the queue and second process of the queue allocate to CPU with one time quantum and so on If the time quantum is too small then large number of context switching may occur and if time quantum is too large then response time will increase causes degrade the performance of CPU

D. WHAT IS SJF?

This algorithm associates with each process the length of the process's next CPU burst. When the CPU is available, it is assigned to the process that has the smallest next CPU burst. If the next CPU bursts of two processes are the same, FCFS scheduling is used. In this Shortest-Job-First (SJF) scheduling algorithm waiting job (or process) with the smallest estimated run-time-to-completion is run next. In other words, when CPU is available, it is assigned to the process that has smallest next CPU burst. The SJF scheduling is especially appropriate for batch jobs for which the run times are known in advance. Since the SJF scheduling algorithm gives the minimum average time for a given set of processes, it is probably optimal. The SJF algorithm favours short jobs (or processors) at the expense of longer ones. The obvious problem with SJF scheme is that it requires precise knowledge of how long a job or process will run, and this information is not usually available. The best SJF algorithm can do is to rely on user estimates of run times.

II.PROPOSED APPROACH

So from above study we know that SJF algorithm require precise knowledge of how long process will execute in CPU but this is quite impossible to know the exact execution time of process before CPU start execution. It may happen that process remain in the Ready Queue (RQ) after completion of predicted burst time of processes that in which way they will execute? If they run in FCFS manner then average waiting time, Average Turnaround time may increase and there is a possibility of starvation. Here we design an algorithm that starvation will not occur & waiting and Turnaround time remain in control.

A. FLOW CHART OF THE ALGORITHM

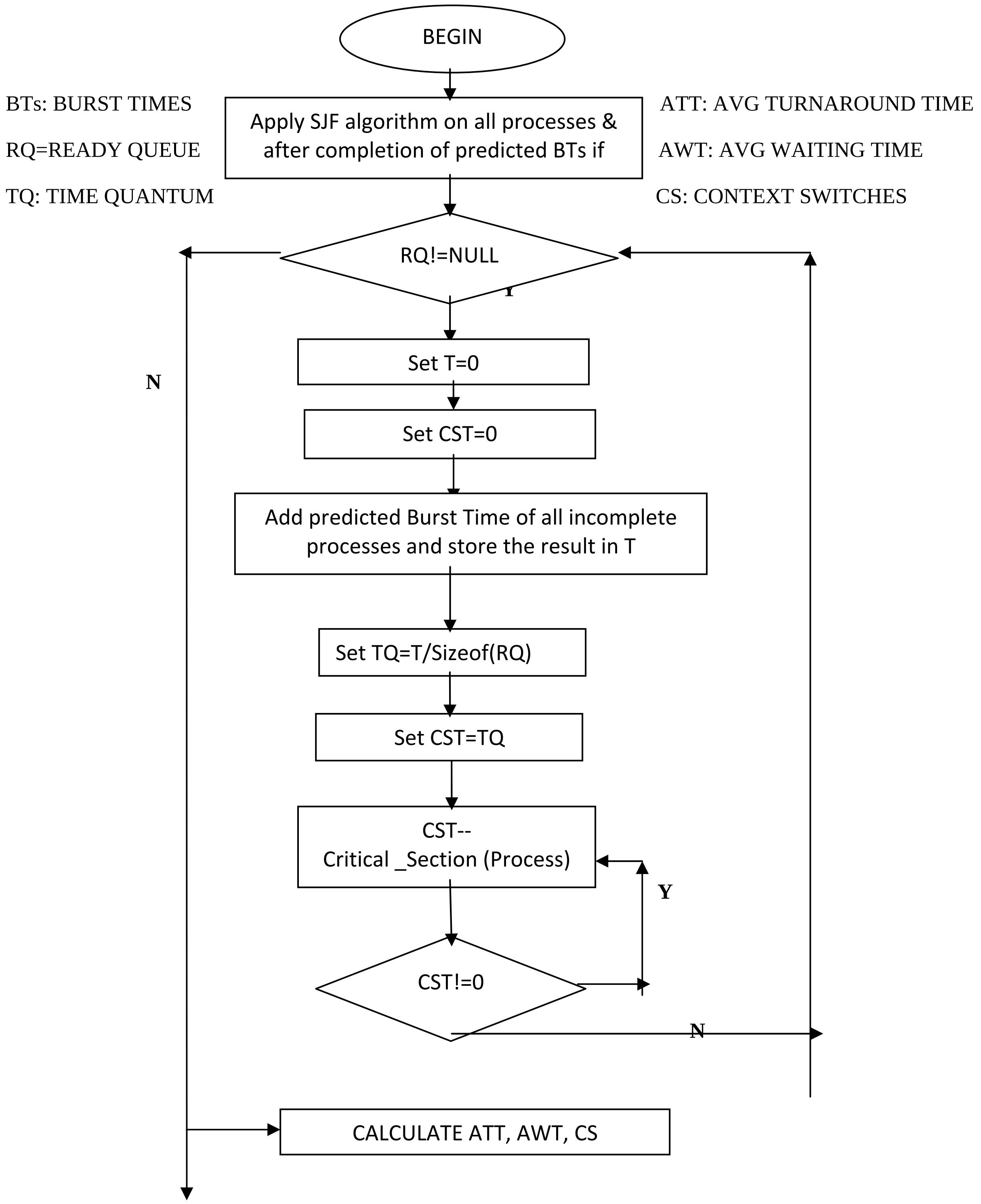

FIG 1: Flow chart of proposed DTQSJF algorithm

B. PROPOSED ALGORITHM

//RQ is an integer variable count size of Ready Queue

//TQ is time quantum calculate total

//Sizeof(RQ) is a function calculate number of process still remain in Ready Queue

Apply SJF Scheduling Algorithm to all the processes on the basis of predicted Burst Time.

//N is the number of processes

//i is loop variable

//BT is the Burst Time of the process

While (RQ! =NULL) //i.e. all process not completed in SJF algorithm

//RQ is the ready queue of the process

//T is integer type variable store predicted Burst time of all process not completed if SJF

//CST is integer type variable count amount of time process remain in Critical Section

Do

//initialize variable T & CST to 0

T=0;

CST=0;

// add predicted Burst time of all process not completed if SJF store in T

// $PRD\_BURST\_TIME_i$ means predicted Burst Time of i^{th} process

For i= 1 to sizeof(ReadyQueue)

do

$T=T+PRD\_BURST\_TIME_i$;

done

//Calculate average of T and store in TQ

//Copy TQ into variable CST

TQ=T/Sizeof(RQ)

CST=T;

//decease CST first

//process enter into Critical Section to execute

//start do while

do

CST=CST-1;

CRITICAL SECTION (P_i)

While (CST! =0)

//end do while

Calculate ATT, AWT, CS

//ATT is Average Turnaround Time

//AWT is Average Waiting Time

//CS is number of Context Switch

End

IV Acknowledgment

So it is conclude that if any process does not finish it's execution in predicted execution by Shortest Job First (SJF) scheduling then there will be an ambiguity of how the remaining uncompleted process will finish but in the proposed approach (DTQSJF) processes will first by SJF algorithm depending on predicted burst time then if the prediction is not correct i.e. actual burst time is more than the predicted then a new time quantum will generate depending on predicted burst time and remaining processes will finish it's execution on the basis of that Time Quantum which turn decrees the possibility of occur starvation.

V REFERENCE

[1] "Silberschatz, A., P.B. Galvin and G. Gagne, 2008" Operating Systems Concepts. 7th Edn., John Wiley and Sons, USA., ISBN: 13 9780471694663, pp.944.

[2] Tanenbaun, A.S., 2008" Modern Operating Systems. 3rd Edn., Prentice Hall, ISBN: 13: 9780136006633, pp. 1104.

[3] Sarojhiranwal and D.r. K.C.Roy"Adaptive Round Robin Scheduling using Shortest Burst Approach Based on Smart Time Slice". volume 2,No. 2,July-Dec 2011,pp. 319-32

[4] 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), 2009, pp.1831-1837.

[5] Stallings, W.: Operating Systems Internals and Design Principles, 5th edition, Prentice Hall, (2004).

[6] M.Kaladevi, M.C.A.,M.Phil., and Dr.S.Sathiyabama, M.Sc.,M.Phil.,Ph.D, “A Comparative Study of Scheduling Algorithms for Real Time Task”, International Journal of Advances in Science and Technology, Vol. 1, No. 4, 2010.

[7] Houssine Chetto and Maryline Chetto (1989) “ Some Results of Earliest Deadline Scheduling Algorithm” journal on IEEE TRANSACTION ON SOFTWARE ENGINEERING. VOL 15. NO.10, pg 1261-1269.

[8] Pallab Banerjee , Probal Banerjee , Shweta Sonali Dhal “Performance Evaluation of a New Proposed Average Mid Max Round Robin(AMMRR) Scheduling Algorithm with Round Robin Scheduling Algorithm”, IJARCSSE,ISSN:2277-128X, Volume-2,Issue 8,August-2012.

[9] “Prof. Rakesh Mohanty, Prof. H. S. Behera, Khusbu Patwari, Manas Ranjan Das, Monisha Dash, Sudhashree” Design and Performance Evaluation of a New Proposed Shortest Remaining Burst Round Robin (SRBRR) Scheduling Algorithm, Am. J. Applied Sci., 6 (10), 2009, pp. 1831-1837. ISSN 2277-1956/V1N4-2026-2034

[10] “Tarek Helmy, Abdelkader Dekdouk” Burst Round Robin: As a Proportional-Share Scheduling Algorithm, IEEE Proceedings of the fourth IEEE-GCC Conference on towards Techno-Industrial Innovations,11-14 November 2007, pp.424-428.

[11] ”Shreedhar, M.; Varghese,G. (October 1995)” Efficient fair queueing using deficit round robin .ACM SIGCOMM Computer communication Review 25 (4): 231. Doi:10.1145/21739.

[12] “Tong, W. and J. Zhao, 2007” Quantum varying deficit round robin scheduling over priority queues.Proceedings of the International Conference on Computational Intelligence and Security, Dec. 15-19, Computer Society, Harbin, China, pp: 252-256.

[13] “Back, D.S., K. Pyun, S.M. Lee, J. Cho and N. Kim, 2007” A hierarchical deficit round-robin scheduling algorithm for a high level of fair service. Proceedings of the International Symposium on Information Technology Convergence, Nov. 23-24, IEEE Computer Society, Washington DC., USA., pp: 115-119. International Journal of Data Engineering (IJDE), Volume 2, Issue 3