ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

Dynamic Quantum based Genetic Round Robin Algorithm

Ms. Rashmi A. Dhumal[1], Ms. Tabassum A. Maktum[2], Ms. Lata Ragha[3]

Assistant Professor, Computer Engineering, TEC, Navi Mumbai, India[1,2]

Professor, Computer Engineering, TEC, Navi Mumbai, India[3]

Abstract: The performance of CPU is essentially depends on the scheduling algorithms. These algorithms provide a schedule for execution of processes waiting in ready queue. There are various scheduling algorithms; Round Robin is one of them. The performance of Round Robin algorithm is majorly depends on the quantum, generally which is static for all processes. There is not any standard way to decide the quantum. The larger quantum maximizes the waiting time and the smaller one increases the overhead of context switching among the processes. In order to improve the performance, the quantum can be dynamically selected for all iterations, instead of keeping it static. We implement Genetic Approach based Round Robin Algorithm with dynamic quantum. Genetic algorithm is evolutionary technique which finds optimal solutions for NP hard problems. We present comparison of three algorithms: Round Robin with static quantum, Round Robin with dynamic quantum and Genetic algorithm based Round Robin with dynamic quantum.

Keywords: GA, FCFS, SJF, RR

I. INTRODUCTION

CPU scheduling is one of the most important research areas of operating system [1]. CPU scheduling deals with allocation of CPU time to the processes waiting in ready queue. Scheduling should be done fairly and correctly, so that every process get chance to execute on processor. The efficiency of scheduling algorithm is based on different criteria's such as waiting time, average waiting time, turnaround time and average turnaround time etc. The different CPU scheduling algorithms available are: First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR) etc.

A. Scheduling Algorithm

The simplest CPU scheduling algorithms is First Come First Serve (FCFS) which selects the first arrived process for execution from the ready queue [2]. The drawback of FCFS is that, process with smaller burst time unnecessary waits for long duration if the process with longest burst time selected first. Another CPU scheduling algorithm is Shortest Job First (SJF), each time which selects a process with shortest burst time from the ready queue [2]. This algorithm is more efficient than FCFS as it ensure minimum average waiting time. This algorithm favours the processes with smaller burst time but the processes with larger burst time will wait for long duration. The drawback of SJF is that it requires knowledge about CPU burst time in advance [2, 3]. Thus, FCFS always select first process where as SJF selects a process with shortest burst time. The Round Robin algorithm uses the concept of time slicing [2]. Each process gets a chance to execute for same amount of time, known as quantum. When the quantum for a current process expires its execution is temporarily halted and it is placed at the end of ready queue. The next ready process is chosen for execution. This process is repeated till all processes complete their execution. The efficiency of Round Robin is totally depends on the quantum. If the quantum is too small, every process executes for a small amount of time which causes frequent context switching. This puts overhead on CPU. Otherwise, if the quantum is too large, the process executes for long duration which maximizes the waiting time.

B. Genetic Algorithm

Genetic algorithm is an evolutionary approach to solve optimization problems like, sequencing, travelling salesman problem, scheduling etc [4]. In general optimization is a technique of finding best solution for a problem from a set of available solutions. Thus it is a search technique to find near optimal solution for a problem. Genetic algorithm uses randomly selected set of possible solution. It starts with a set of individuals called as initial population. Each solution is represented using some abstract genetic representation which is called as a chromosome. Each chromosome is a collection of genes. This process of representation of solution in some form is called as encoding. There are various techniques used to encode these chromosomes such as binary encoding, permutation encoding, value encoding etc [4]. The next step after encoding is to apply different reproduction operations in order to evolve new chromosomes. These operators are applied on the initial population and a new generation is created. The operators used for this purpose are crossover and mutation. Another important operation performed is selection. During selection process some individuals are selected on basis of their fitness for the further process. In order to decide which individual is best some value is associated with it which is called as fitness of an individual. The fitness function is used to calculate the fitness. The fitness function will be decided according to the problem. In our case the fitness of individual is based on the average waiting time. Thus the selection operation uses fitness function to select good individuals. There are various selection methods like Roulette wheel, Tournament, Elitism, Rank, Steady state **[**5**].** Once the selection method and the fitness function have been appropriately defined, the next step is to apply and then the reproduction operators on selected individuals to generate the new population. The crossover operator exchanges portions of chromosome according to some criteria. There are several crossover operators described as Single point, Multi point and Uniform [6]. Using the crossover operator we generate the set of new individuals but there is a chance of duplication of individuals in this new. In order to avoid this, mutation operator is applied. Mutation operator does the random change in some values of selected chromosome. The various mutation criteria are: Flipping, Interchanging, Reversing etc. [6]. While performing all these operations it may happen that the fittest individual is not included in the new population. Therefore, genetic algorithm uses the concept of elitism. According to elitism each time the fittest individual is carried forward in the new population. Along with these operations there are some basic parameters which must be decided before applying the algorithm. These include the size of initial population, mutation rate, crossover rate and stopping criteria [7]. The size of initial population has a major impact on performance of algorithm. It represents how many individuals will be created initially from which the next generation will evolve. The crossover rate denotes the probability of applying this operator. Generally the crossover rate is kept maximum. Similarly the mutation rate means the probability of applying the mutation operation and it is kept quite small. The stopping criterion decides when the algorithm will stop. The possible stopping criteria are maximum number of generations, some time period or some predefined fitness value. Now we are ready with everything, the encoding scheme, the fitness function, selection method, crossover operator, mutation operator and all basic parameters. Thus the genetic algorithm starts with generation of initial population and then repeatedly evolves the new generations using the decided operators and finds the optimal solution. The idea proposed in this paper is to combine Genetic algorithm and traditional Round Robin algorithm. We have developed a genetic algorithm based Round Robin which uses the dynamic time quantum for scheduling processes on single processor. We have compared average waiting time of our algorithm with RR with static quantum and RR with dynamic quantum.

II. RELATED WORK

The major issue in Round Robin algorithm is how to decide the time quantum. A huge amount of research has been done on this topic; some of it is listed below. The AN algorithm is proposed in [8] which uses the dynamic time quantum and the operating system itself finds the optimal time quantum without user intervention. The results of this paper shows that the AN algorithm performs better than the simple RR algorithm. A new RR algorithm based on the non-linear programming mathematical model is proposed in [9] . The method used to calculate the value of time quantum is SRBRR and the results show that the proposed model obtains better values. The Dynamic Quantum Readjusted Round Robin (DQRRR) is presented in [10]. Here, the processes are arranged in ascending order according to their burst time and the optimal time quantum is found using the median method.

DQRRR algorithm reduces the average waiting time, average turnaround time and hence improves the performance. The median approach to find an optimal time quantum is proposed in [10, 11], has obtained good results. An optimized Round Robin algorithm based on the concept of fuzzy rules (FRRCS) is presented in [12]. The researchers have showed that the new algorithm reduces the average waiting time. In [13], an algorithm based on integer programming method to calculate the value of the time quantum is introduced. The concept presented by us is to integrate Genetic algorithm with Round Robin algorithm in order to improve the performance. One such approach is presented in [14], where Round Robin algorithm is implemented using Genetic approach and better results are achieved. A similar concept is implemented in [15] where a hybrid Scheduling Algorithm using genetic approach is presented. This work shows the GA based approach outperforms the traditional algorithms.

The paper is organized as follows:
In section 3 the Round Robin algorithm is explained in detail. Section 4 gives details about our proposed methodology. In Section 5 the comparison of the three algorithms is presented. The conclusion and future work is discussed in section 6.

III. ROUND ROBIN (RR) ALGORITHM

Round Robin is one of the efficient algorithms for CPU scheduling. The basic idea in RR is it gives equal CPU time to each process. This CPU time is known as a quantum. The performance of RR is basically depends upon the quantum value [1, 2]. There is no specific method is available to select the quantum. If quantum is too large it maximizes the waiting time whereas if it is too small it increases the context switching. The quantum can be fixed during all iterations or change dynamically based on some optimization technique. The simple RR is primarily selects a fixed time quantum. All processes executes for fixed amount of time. When quantum expires process is attached at the end of ready queue and a new process executes on CPU for next quantum. This procedure continues till all processes complete their execution on CPU.

Consider five processes (P1, P2, P3, P4, P5) with their respective burst times (10, 12, 5, 15, 11) and quantum=5

P1	P2	P3	P4	P5	P1	P2	P4	P5	P2	P4	P5	
0	5	10	15	20	25	30	35	40	45	47	52	53

Figure 1. Giant Chart

Table 1. Process waiting time

Process	Waiting Time
P1	25
P2	35
P3	10
P4	37
P5	42

Average waiting time=(25+35+10+37+42)/5=29.8

ISSN (Online) : 2278-1021
ISSN (Print) : 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 3, March 2014

The figure 1 gives the giant chart for execution of processes. The table 1 shows the waiting time of each process. The average time of these processes is calculated as average of waiting time of all the processes. As mentioned above the main question is how to select static quantum. Hence the next approach is select quantum dynamically. The Round Robin algorithm with dynamic quantum selects a quantum dynamically for all iteration. The selection is based on different criteria's: mean or median of burst times of all the processes.

Consider the above example and quantum is the median of all the processes burst time. Thus initial quantum is 5 for first iteration. The quantum is selected for next iterations from the remaining burst time of processes. The completed processes are not considered for this. The remaining burst times of processes are arranged so that the median of it is selected as a next quantum value. Hence the quantum is 7 for second iteration. This procedure continues till all processes complete their execution.

iteration 1 quantum=5					iteration2 quantum=7				iteration 1 quantum=3	
P1	P2	P3	P4	P5	P1	P2	P4	P5	P4	
0	5	10	15	20	25	30	37	44	50	53

Figure 2. Giant Chart

Table 2. Process waiting time

Process	Waiting Time
P1	20
P2	25
P3	10
P4	38
P5	39

Average waiting time= (20+25+10+38+39)/5=26.4

The figure 2 gives the giant chart for execution of processes. The table 2 shows the waiting time of each process. The performance of Round Robin with dynamic quantum is better than the Round Robin with static quantum.

IV. PROPOSED METHODOLOGY

In this proposed study we have applied Genetic Approach based Round Robin algorithm for CPU scheduling. In Genetic algorithm, first we have generated initial population with set of ten chromosomes randomly, each of which is a sequence of integers that represents distinct process number. The size of each chromosome is five that remain same for every generation. As stated above in CPU scheduling, waiting time, average waiting time, turnaround time average turnaround time etc are various criteria's for scheduling. We have selected average waiting time as a fitness of the chromosome. The fittest chromosome is one with minimum average waiting time. Round Robin algorithm with dynamic quantum is used to calculate fitness of the chromosome. Initially, the quantum is the median of all the processes burst time. Consider the processes (P1, P2, P3, P4, P5) with their respective burst times (11, 12, 5, 15, 10).

The selected chromosome is,[13524] and respective burst times are (11, 5, 10, 12, 15) then the median 10 is a quantum for first iteration. After first iteration, remaining burst times of processes are arranged so that the median is selected among them. The completed processes are not considered during subsequent iterations. The remaining burst times of processes are (1, 0, 0, 2, 5). Hence the median 2 is selected as next quantum. This procedure continues till all processes complete their execution. We have used tournament selection method with size of five. Randomly five individuals are selected from current population and among these fittest one is selected as a parent. Two parents are selected to reproduce new offspring's. The single point crossover operation with 100 percent probability is applied over these parents to generate new offspring's. The elitism criteria of Genetic algorithm survive the fittest individual to the next generation. Thus the fittest individual with minimum waiting time is included in all next generations. The mutation operation is applied to some individuals by interchanging values of two random positions. The probability of mutation is 0.005. We have applied above procedure till the tenth generation.

V. RESULT AND DISSCUSSION

We have implemented and compared three approaches to implement Round Robin algorithm: Simple RR with static quantum, Simple RR with dynamic quantum and GA based RR with dynamic quantum. All these algorithms are implemented using JAVA language. The figure 3 shows the comparisons of these algorithms on the basis of average waiting time. Table3 shows the sequences of five processes along with burst time and the respective average waiting time for these algorithms. The result shows that the performance of Round Robin is improved with the dynamic quantum. The Genetic Approach based RR gives drastically enhanced performance over simple RR

SR. NO .	Processes burst time [P1,P2,P3,P4,P5]	RR Static	RR dynamic	GA RR dynamic
1	[15,12,5,18,16]	36.0	29.4	20.6
2	[12,14,10,12,11]	39.2	39	22.0
3	[80,70,20,15,75]	123	117	71
4	[40,45,50,30,35]	85	85	72
5	[19,40,7,25,18]	54.4	49	35
6	[47,37,17,15,8]	62.4	61	37.6
7	[120,118,55,84,77]	292.2	272.6	155
8	[70,72,46,67,58]	194.6	208.8	119.2
9	[32,26,17,45,38]	82.8	79.4	53.2
10	[78,57,18,28,33]	111	109.8	62.6

Table 3. Comparison of three algorithms

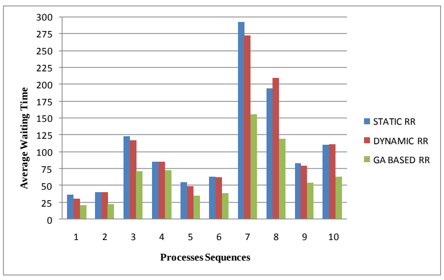

Figure 3. Comparison Of Three Different Round Robin Algorithms

VI. CONCLUSION

The purpose of this work is to minimize average waiting time by selecting different quantum values for Round Robin algorithm of CPU scheduling. We have implemented and compared simple RR, dynamic RR and Genetic Approach based Round Robin algorithm. We have concluded that Round Robin with dynamic quantum is better than RR with static quantum. GA based RR with dynamic quantum majorly improves the performance over simple RR. This work can be extended by using various other genetic algorithm operators and changing the basic parameter of it. Also we can use some other criteria for selecting the quantum value dynamically.

REFERENCES

[1] Abraham S. & Peter B. Galvin & Greg Gagne," Operating System Concepts", John Wiley & Sons,Inc. 2007.
[2] Silberschatzs, A., Galvin, P.B., Gagne, G. 2009, "Operating System Concepts. Massachusetts: Addison Wesley".
[3] Lingyun Yang, Jennifer M. Schopf and Ian Foster, "Conservative Scheduling: Using predictive variance to improve scheduling decisions in Dynamic Environments", Supercomputing 2003, November 15-21, Phoenix, AZ, USA.
[4] L.M. Schmitt, "Fundamental Study Theory of Genetic Algorithms", International Journal of Modelling and Simulations Theoretical Computer Science, 2001.
[5] Vikas Gaba et al.," Comparison of processor scheduling algorithms using Genetic Approach" International Journal of Advanced Research in Computer Science and Software Engineering 2 (8), August- 2012.
[6] Davis, L. "Handbook of Genetic Algorithm, Von Nostrand Reinhold, Newyork, 1991.
[7] Lobo, Fernando Miguel, "The parameter-less genetic algorithm: rational and automated parameter selection for simplified genetic algorithm operation.", International Conference on Genetic Algorithms, in Lisboa, 2000.
[8] Abbas Noon et al.," A new Round Robin Based Scheduling Algorithm for Operating System: Dynamic Quantum Using the Mean Average", International Journal of Computer Science, Issue, Vol. 8,Issue 3, No. 1, May 2011
[9] Shahram Saeidi et al.,"Determining the optimum time Quantum Value in Round Robin Process Scheduling Method", I. J. Information Technology and Computer Science, September 2012,
[10] H.S. Behera et al, "A new Proposed Dynamic Quantum with Re-adjusted Round Robin Scheduling Algorithm and its Performance Analysis" International Journal of Computer Applications, Issue, Vo5, No. 5, August 2010 [11] M. Fahimi, Operating Systems, Vol. 1, 1st Edition, Tehran:Jelve Publishing, 1992.
[11] M.H. Zahedi, M. Ghazizadeh¸ and M. Naghibzadeh, "Fuzzy Round Robin CPU Scheduling (FRRCS) Algorithm", Advances in Computer and Information Sciences and Engineering. 348-353, 2008.
[12] S.M. Mostafa, , S.Z. Rida, and S.H. Hamad, "Finding time quantum of Round Robin CPU scheduling in general computing systems using integer programming", International Journal of Research and Review in Applied Science. 5(1), 64-71. 2010.
[13] Maria Ulfah Siregar,"A new approach to CPU scheduling: genetic round robin", International Journal of Computer Applications,Vol. 47,issue no. 19,June 2012.
[14] Jyotirmay Patel, A.K. Solanki, "Performance Enhancement of CPU Scheduling by Hybrid Algorithms Using Genetic Approach", International Journal of Advanced Research in Computer Engineering &Technology", Vol. 1, Issue 4, June 2012.