GENETIC ALGORITHM BASED ADAPTIVE SCHEDULING ALGORITHM FOR REAL TIME OPERATING SYSTEMS

Sachin R. Sakhare[1] and Dr. M.S. Ali[2]
[1]Department of Information Technology, VIIT,Pune,M.S. India
sakharesachin7@gmail.com
[2]PRMCOEM , Badnera, Amravati , M.S. India
softalis@gmail.com

ABSTRACT

In this paper novel technique for CPU scheduling in real time operating systems by using genetic algorithm (GA) is proposed. Proposed adaptive algorithm is a combination of existing dynamic priority driven algorithm i.e. Earliest Deadline First (EDF) and new genetic algorithm (GA) based scheduling algorithm. First we have developed GA based scheduling algorithm and tested it during both under loaded and overloaded condition.

Initially, in underloaded condition EDF is used for scheduling and in overloaded condition system will change to a GA based scheduling algorithm .Thus our Adaptive algorithm uses the strong features of both algorithms and overcome their drawbacks.

We have simulated, proposed adaptive algorithm along with both EDF and GA based algorithms for real time systems. %Success Rate and %Effective CPU Utilization are used as performance measuring criteria for all these 3 algorithms. The evaluation of results and comparison of our proposed adaptive CPU scheduling algorithm with EDF algorithm shows that the proposed adaptive algorithm is optimal and efficient during underloaded as well as overloaded situations compared to EDF.

KEYWORDS

Real time Scheduling, Genetic Algorithms, Real Time Operating Systems, Earliest deadline first, scheduling algorithm.

1. INTRODUCTION

Real-time task scheduling is one of the interesting topics in the context of real time operating systems (RTOS). The interest in the topic started with the seminal work of Liu and Layland in 1973. Since then many algorithms have been proposed for real time scheduling. Schedulability analysis is a fundamental aspect of real-time scheduling. A set of task is said to be schedulable if enough CPU time is available to execute all these tasks before their deadlines. Each real time task is assigned a priority and a deadline.

The real time tasks are of 3 types [1]. If a task needs to be executed after regular time interval then it is called periodic task. If a task's relative activation time is not known then it is non-periodic task. A non-periodic task with a hard deadline is called sporadic task. In RTOS both periodic and sporadic tasks must be scheduled by satisfying their timing constraints. Scheduling algorithms can be divided into Static-priority based algorithms and Dynamic-priority based algorithms. Scheduling can be done for either preemptive or non-preemptive tasks. Two common

DOI : 10.5121/ijesa.2012.2310

constraints in scheduling are the resource requirements and the preference of execution of the tasks. Typical parameters associated with tasks are:

- Average execution time
- Worst case execution time
- Dispatch costs
- Arrival time
- Period (for periodic tasks).

The objective of scheduling is to minimize schedule-length, average tardiness or laxity, and to maximize average earliness and number of arrivals that meet deadlines. Real-time scheduling algorithms can be classified into two categories: static priority algorithms and dynamic priority algorithms.

Rate Monotonic scheduling (RMS) is a static priority preemptive scheduling scheme for uniprocessor systems [2].RMS assigns priorities to tasks on the basis of their periods. For RMS, the highest-priority task is one with shortest period. When multiple tasks are available for the execution then the one with shortest period is serviced first.

Another CPU scheduling algorithm for real time system is Earliest deadline first (EDF). EDF can be used for both static and dynamic type of scheduling. When n number of tasks are to be scheduled by EDF, then it's complexity is $O(n^2)$ and percentage CPU utilization is 100%.
Least Laxity First (LLF) scheduling is a variant of EDF. In LLF tasks having least laxity are executed first. Unlike EDF execution time of the task is considered for scheduling in LLF [3]. Scheduling problems with deadlines which characterize real-time systems are almost always shown to be NP-hard in either single processors or multi-processors. Many researches have o focused on searching for heuristic scheduling algorithms whose results are compared to the optimal results. In some studies use of genetic algorithm (GA) for real-time task scheduling is found significant. The advantage of using GA is that it relieves the designer from the overhead of constructing a solution and the designer has to assess a given solution [4]. Genetic algorithm has been utilized to optimize the total execution time. The simulation studies presented in this paper shows the efficiency of the GA based adaptive scheduler compared to other schedulers.

The whole paper is organized as follows: In Section 2, schedulability analysis for RM, EDF as well as proposed adaptive algorithm is discussed. Section 3 portrays the proposed adaptive algorithm. Section 4 contains simulation method and performance measuring parameters. Section 5 contains the results obtained and the paper ends with a brief conclusion in Section 6.

2. SCHEDULABILITY ANALYSIS

The basic schedulability conditions for RM and EDF proposed by Liu and Layland (1973) were derived for a set of n periodic tasks under the assumptions that all tasks start simultaneously at time t = 0 (that is, $A_i = 0$ for all i =1, . . . , n), relative deadlines are equal to periods (that is, di,k =k Ti) and tasks are independent (that is, they do not have resource constraints, nor precedence relations). Under such assumptions, a set of n periodic tasks is schedulable by the RM algorithm

if

$$\sum_{i=1}^{n} U_i \leq n\,(2^{1/n} - 1)\ [5]$$

Under same assumptions, a set of n periodic tasks is schedulable by the EDF algorithm if and only if

$$\sum_{i=1}^{n} U_i \leq 1$$

The schedulability bound of RM is a function of the number of tasks, and it decreases with n. We recall that

$$\lim_{n \to \infty} n\,(2^{1/n} - 1) = \ln 2 \sim 0.69$$

meaning that any task set can be scheduled by RM if $U \leq 0.69$, but not all task sets can be scheduled if $0.69 < U \leq 1$[6]

While, proposed GA based adaptive algorithm can schedule tasks even after system is overloaded i.e.

$$\sum_{i=1}^{n} U_i \geq 1$$

3. PROPOSED ADAPTIVE ALGORITHM

The Adaptive algorithm is combination of two scheduling algorithms: EDF algorithm and GA based Scheduling algorithm.

3.1 EDF Algorithm:

The EDF scheduling algorithm is a priority driven algorithm. The task with nearest deadline is given highest priority and it is selected for execution. This algorithm is simple and proved to be optimal when the system is preemptive, underloaded and there is only one processor [7].

3.2 GA Based Scheduling Algorithm:

The first step of GA is to encode any possible solution of the problem as a set of strings called as chromosomes. The genetic algorithm used in our system is given below.

Genetic Based Scheduling Algorithm

1. Randomly initialize population (n)

2. Determine fitness of population (n)

3. Repeat

   a. Select parents from population (n)
   b. Perform 2 point partially matched crossover on parents creating population (n+1)
   c. Perform 1 bit mutation of population (n+1)

d. Determine fitness of population (n+1)

5. Until best individual is good enough.

3.3 Adaptive Scheduling Algorithm

Proposed Adaptive algorithm combines both of these algorithms and it works as per following:

- During underloaded condition, the algorithm uses EDF algorithm i.e. priority of the job will be decided dynamically depending on its deadline.

- During overloaded condition, it uses GA based Scheduling algorithm i.e., priority of the jobs will be decided depending on the fitness value of each chromosome. Fitness value of chromosome is decided by the percentage Success Rate(%SR) which is defined as :

$$\%SR = \frac{\text{Number of tasks successfully scheduled}}{\text{Total number of tasks arrived}} \times 100$$

Switching Criteria:

- Initially the proposed algorithm uses EDF algorithm considering that the condition is not overloaded. But when a job has missed the deadline, it will be identified as overloaded condition and the algorithm will switch to GA based scheduling algorithm. After 10 jobs have continuously achieved the deadline, again the algorithm will shift to EDF algorithm considering that overloaded condition has been disappeared.

- During underloaded condition, EDF algorithm is used for reducing execution time and during overloaded condition GA based scheduling algorithm is used for achieving better performance. By this way, adaptive algorithm has taken advantage of both algorithms and overcome their limitations.

4. SIMULATION METHOD

We have developed simulator using JAVA platform to simulate EDF, GA based & the adaptive algorithms. We have run simulations to accumulate empirical data. For simulation only periodic tasks running on uniprocessor systems are considered. We have tested the simulation program for 500 different task schedules. In each task schedule, 200 tasks are generated. The results for 20 different values of load are taken and tested on more than 10,000 tasks. The simulator is executed on Intel® Core™ 2 Duo CPU @ 1.50 GHz with 2 GB RAM and Microsoft® Windows7 Operating System. We have implemented EDF, GA based & the adaptive algorithms and have run simulations to accumulate empirical data. Initially overall load of the system is calculated as follows

$$U = \sum_{i=1}^{n} \frac{C_i}{D_i},$$

Where Ci is the execution time of ith task and Di is the deadline of ith task.

The system is said to be overloaded when, value of U ≥1. We have considered following three quantities as our main performance measuring criteria for the real time scheduling algorithms:
We can measure the performance of any real time scheduling algorithm by using Percentage

1) Success Rate (%SR). It is a percentage of tasks successfully completed before their respective deadlines. It is defined as follows:

$$\%\ SR = \frac{\text{Number of tasks successfully scheduled}}{\text{Total number of tasks arrived}} \text{ x } 100$$

2) Missing rate (MR) is the number of tasks which are not scheduled during their respective deadlines. It can be defined as,

$$MR = \frac{Number\ of\ tasks\ which\ missed\ their\ deadlines}{Total\ number\ of\ tasks\ arrived}$$

3) CPU Utilization (CU) is the amount of time processor is busy in scheduling the tasks. But here we measure percentage effective CPU utilization (%ECU) which gives information of how effectively the processor time is utilized.

$$\%ECU = \sum_{i \in S} \frac{V_i}{T} x\ 100$$

Where,
V is value of a job and,
– Value of a job = Execution time of a job, if the job completes within its deadline.
– Value of a job = 0, if the job fails to meet the deadline.
S is a set of all the jobs which are executed by the CPU.
T is total time of scheduling.

5. RESULTS

Following figure 1 & figure 2 represents the results obtained from simulation study.. Fig. 1 shows the results obtained in terms of %SR and %ECU during overloaded conditions, using the same algorithms. Fig. 2 shows the comparison of the execution time taken by each algorithm. From the results of Fig. 1 and Fig. 2,

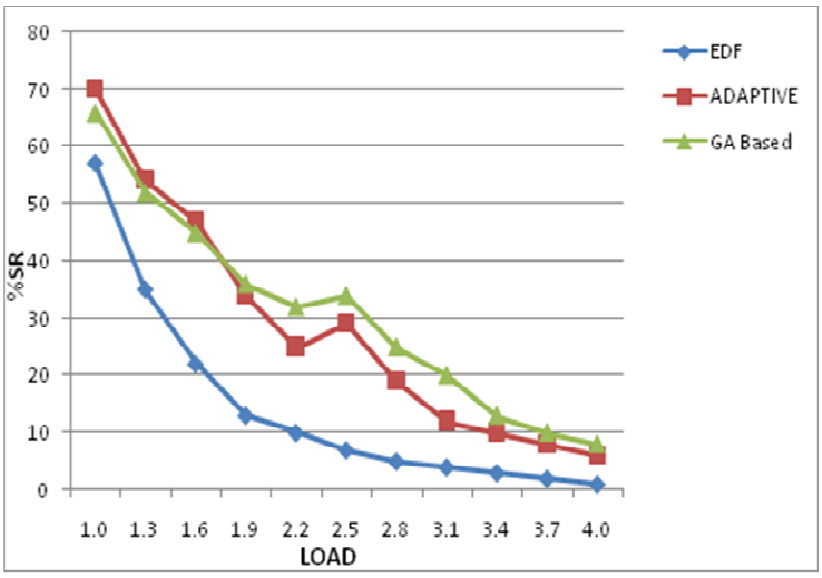

Figure 1: Load Vs %SR

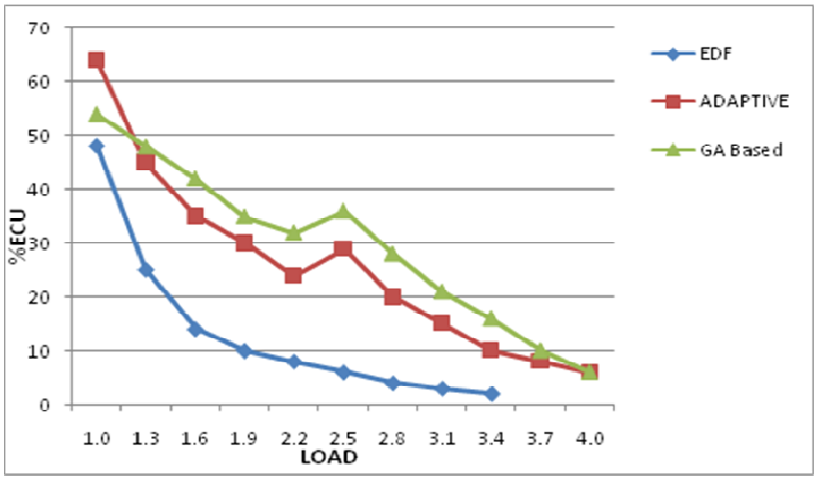

Figure 2: Load Vs %ECU

Figure 1 shows the results obtained in terms of %SRVs Load & Figure 2 shows the results obtained in terms of %ECU Vs Load. From the results we can observe that the adaptive algorithm performs better than EDF algorithm during overloaded conditions even GA based algorithm performs better than EDF algorithm during overloaded conditions.

5. CONCLUSION

In this paper, we have proposed GA based adaptive scheduling algorithm for scheduling periodic tasks on single processor environment when the tasks are preemptive. The results achieved during simulation prove the following:

- The proposed adaptive algorithm is more efficient than EDF for single processor, preemptive environment when the system is overloaded.
- EDF algorithm does not perform well when the system is overloaded and GA based scheduling algorithm takes more execution time in that type of condition. These are the main limitations of both algorithms.
- During underloaded condition, the execution time taken by the proposed algorithm is almost same as EDF algorithm (i.e. less time).
- The algorithm can switch automatically between EDF algorithm and GA based scheduling algorithm. Therefore, the proposed adaptive algorithm is very useful when future workload of the system is unpredictable.

REFERENCES

[1] A. Silberschatz, P.B. Galvin and G. Gagne, (2001) Operating Systems Concepts, Sixth edition, John Wiley Publishers.

[2] C.L.Liu and L. Layland,(1973) “Scheduling algorithms for multiprogramming in a hard real-time environment,” Journal of ACM, vol. 20(1), pp. 46-61.

[3] S. Baskiyar and N. Meghanathan,(2005) “A Survey Of Contemporary Real-time Operating Systems,” Informatica 29 pp. 233-240.

[4] David E. Goldberg,(2011) Genetic Algorithms, in search, Optimization and Machine Learning, Sixth edition, Pearson Education Publishers.

[5] Giorgio C. Buttazzo,(2005) "Rate Monotonic vs. EDF: Judgment Day" Real-Time Systems, Springer Science + Business Media, Inc. pp. 5-26.

[6] Stankovic J. A., Ramritham K.,(1994) "Scheduling algorithm and Operating System support for real-time system", proceedings of the IEEE, Vol.82(1), pp55-67 .

[7] Ketan Kotecha and Apurva Shah,(2008) "Adaptive Scheduling Algorithm for Real-Time Operating System," IEEE Congress on Evolutionary Computation (CEC 2008),pp 2109-2112.

[8] V. J. Mooney and D. Blough, (2002) "A hardware-software real time operating system framework for SoCs," IEEE Design & Test of Computers, vol. 19, Volume no. 6, pp. 44–51.

[9] G.Saini, (2005) "Application of fuzzy logic to real-time scheduling", Real Time Conference, 14th IEEE-NPSS.

[10] G. Koren and D. Shasha, (1995) "Dover: An optimal on-line scheduling algorithm for overloaded real-time systems," SIAM J of Computing, vol. 24(2), pp. 318-339.

[11] Jane W.S. Liu, (2001) Real-Time Systems, Pearson Education, India, pp. 121 & 26.

Authors

Sachin R. Sakhare is currently is working as Associate Professor at Vishwakarma Institute of Information Technology, Pune. He did his B.E.(Comp. Sci. & Engg.) and M.E.(Comp. Sci. & Engg.) from S.G.B. Amravati University. He is currently pursuing Ph.D. in the faculty of Engineering & Technology in the area of Operating Systems from S.G.B. Amravati University. He is a life member of ISTE, New Delhi, and Member of IAENG Hong Kong.

Dr. M. S. Ali is currently working as Principal at Prof Ram Meghe College of Engineering & Management, Badnera-Amravati. He did his B.E (Electrical) from Government College of Engineering, Amravati in 1981, M. Tech. from IIT Bombay in 1984 and Ph.D. from S.G.B. Amravati University in 2006 in the faculty of Engineering & Technology in the area of e-Learning. He is life member of ISTE, New Delhi, Fellow of IETE, New Delhi and Fellow of IE (India). He is Chairman of IETE Amravati Local Center.