Journal of Engineering Sciences, Assiut University, Vol. 36, No. 1, pp. 105-114, January 2008

A BASELINE MODEL FOR SCHEDULING SOFT REAL TIME TASKS ON UNI-PROCESSOR SYSTEMS

Ahmed M. Mohamed
Electrical Engineering Department
Aswan Faculty of Engineering, Aswan, Egypt
ahmed@engr.uconn.edu

(Received August 22, 2007 Accepted October 27, 2007)

In this paper, we consider the different performance parameters that influence the scheduling of soft real time tasks on uni-processor systems. This environment is rich of parameters that affect the rejection ratio of the system. This work models such stochastic environment and proposes a baseline model that estimates the rejection ratio of the system before applying any special scheduling algorithm. The model presents many cases where the system produces a small rejection ratio without the need for any special scheduling algorithm. We validate our analytical model with simulations that represent the real computing environment. The results of our simulations show that our mathematical model can predict the expected percentage of tasks that miss their deadline accurately.

1. INTRODUCTION

The distinguishing feature of real time systems is their attempt to achieve both logical and temporal correctness of computation. A computation is temporally correct if it finishes within a specified time frame. In this sense, all time constrained computer applications require a real time computer system. There are two types of real time computer systems, the hard real time and soft real time. Hard real time systems consider tasks with stringent time constraints. Task deadlines are guaranteed by careful but often complex system design and in many cases by using expensive specialized hardware. On the other hand, there are the soft real time applications for which meeting every task deadline causes poor utilization of the system resources. These applications include digitized voice packets and banking transaction processing systems. A typical requirement for such systems is to complete most of the tasks in the specified deadline. Even in cases where end users have no deadlines, it is still highly desirable to make applications more responsive to user timing requirements.

To maximize the number of real time tasks that can be processed without timing violation, real time computer systems use sophisticated scheduling algorithms to decide the order in which tasks are executed. A scheduling algorithm is said to be feasible if the execution of each task can be completed before its deadline. In the literature there are many scheduling algorithms for both hard and soft real time applications. These algorithms are assumed to enhance the performance of the real time computer systems. We have not seen any work that analyze such computing environment and determines the performance parameters that influence the behavior of such systems. We believe that those systems contain different application and system

parameters that have a great impact on the number of tasks that meet their deadlines. We discovered that there are many cases where we do not need any specialized scheduling algorithms. The reminder of this paper is structured as follows. We first review the related work. Next, we present our baseline model. The simulation results are presented in Section 4. Finally, conclusions are given.

2. RELATED WORK

Given the enormous amount of literature available on scheduling, any survey can only scratch the surface. Moreover, a large number of scheduling approaches are used upon radically different assumptions making their comparison on a unified basis a rather difficult task. In [3,14] there is a survey on the scheduling algorithms for hard real time systems. In soft real-time processing, applications are allowed to miss deadlines, particularly in situations of system overload. With respect to timing behavior, an impressive amount of work has been carried out in the area of schedulability analysis (e.g. [11, 1, 17]) focusing on worst case analysis. However, less work addresses the analysis of systems with probabilistic behaviors. For soft real-time systems, it is important to analyze the variations in the runtime behavior to determine the likelihood of occurrence of certain undesired situations and, based on that, to dimension the system. In [12] and [17] it is showed that the techniques proposed in this area are quite restrictive. Some of them target certain application classes. Other approaches address specific scheduling policies or assume highly-loaded systems.

Also, there is a number of soft real-time (SRT) scheduling mechanisms have been proposed to support QoS requirements of multimedia applications. These approaches typically integrate predictable CPU allocation (such as proportional sharing [4, 5, 7, 8] and reservation [9, 15]) and real-time scheduling algorithms. Our model distinguishes itself from the above approaches for two reasons: first our model assumes that each task's CPU demands is unknown, while others typically assume that the CPU demands are known in advance. Second, our model considers the CPU speed as a parameter in the model, while others implicitly assume a constant CPU speed. The variable speed context brings challenges to SRT scheduling.

3. THE MODEL

Our analytical model is based on the following assumptions, Tasks interarrival times are random variables that follows a Poisson distribution with mean interarrival time $1 / \lambda$. Tasks execution times are random variables that are exponentially distributed with mean service time $1 / \mu$. Associated with each task is its deadline. Task deadline is assigned according to uniform distribution. The uniform distribution over the range is $[D_{min}, D_{max}]$. That is the task deadline falls in between the minimum value D_{min} and the maximum value D_{max}. The probability density function of the uniform distribution can be written as follows,

$$D(x) = \begin{cases} 0 & x < D\min \\ \dfrac{1}{k} & D\min < x < D\max \\ 0 & x > D\max \end{cases}$$

Where, $k = D_{max} - D_{min}$

The queueing discipline is First Come First Serve (FCFS). Tasks are independent.

In our study we focus on soft real time systems. In such systems, a primary performance metric is to meet as many deadlines as possible. We assume that all the tasks in the system are of equal importance. When the system starts working, it receives a stream of tasks. The service time of a task is a random variable (unknown value). Also, each task is associated with its deadline. When a task arrives to the system, it joins waiting queue if the server is busy. If the server is not busy, it will start executing the task. Assume that task i spends T_i units of time in the system. Then T_i can be define as follows,

$$T_i = t_i^w + t_i^s \quad (1)$$

Where t_i^w is the time spent by task i waiting in the queue and t_i^s is the time spent by task i in the processor (server). Based in our assumptions, we will use the M/M/1 model. Consider a task with deadline d. The probability that it misses its deadline is equal to the probability that it must wait in the server queue more than d units of time. Assume that the waiting time reliability function is W(t),

W(t) := Probability that (t_i^w > t) = Pr (t_i^w > t)

Then the probability of a task i to miss its deadline d_i is W(d_i). To compute the fraction of tasks that misses their deadlines MD (i.e., the expected probability of missing a deadline), we must consider the distribution of the task deadlines and can be expressed as follows,

$$MD = \int_0^\infty D(x)W(x)dx \quad (2)$$

The probability density function of the tasks deadline is assumed to be uniform. Then we derive the reliability function of the tasks waiting time W(t). The reliability function R(t) of the system time (the waiting time in the queue plus the service time) is already known [10]. The reliability function R(t)$_i$ is the probability that a task will spend in the system time greater than t. R(t)$_i$ is define as follows,

$$R(t)_i = \Pr(T_i > t) = e^{-\mu(1-\rho)t} \quad (3)$$

Where $\rho = \dfrac{\mu}{\lambda}$ and is called the system utilization (the percentage of time the server is busy). From equation (1) we can redefine R(t)$_i$ as,

$$R(t)_i = \Pr(T_i > t) = \Pr(t_i^w + t_i^s) > t) \quad (4)$$

Since t_i^w and t_i^s are independent, then we can say that, R(t) is the convolution of the reliability function of the waiting time the reliability function of service time.

$$R(t) = W(t) \otimes B(t) \quad (5)$$

where W(t) is the reliability function of the waiting time and B(t) is the reliability function of service time. B(t) is known from our assumption and it is exponential,

$B(t) = e^{-\mu t}$

Taking the Laplace transform for both sides of equation (5),

$$R(s) = W(s) * B(s)$$

$$\frac{\mu(1-\rho)}{\mu(1-\rho)+s} = W(s) * \frac{\mu}{\mu+s}$$

$$W(s) = \frac{(1-\rho)(\mu+s)}{\mu(1-\rho)+s} = \frac{\mu(1-\rho)+s(1-\rho)}{\mu(1-\rho)+s}$$

$$W(s) = \frac{[\mu(1-\rho)+s]-\rho s}{\mu(1-\rho)+s}$$

add and subtract $\rho\mu(1-\rho)$ to the above equation,

$$W(s) = \frac{(\mu(1-\rho)+s)-\rho s-\rho\mu(1-\rho)+\rho\mu(1-\rho)}{\mu(1-\rho)+s}$$

$$W(s) = \frac{(\mu(1-\rho)+s)-\rho(\mu(1-\rho)+s)+\rho\mu(1-\rho)}{\mu(1-\rho)+s}$$

$$W(s) = \frac{(\mu(1-\rho)+s)(1-\rho)+\rho\mu(1-\rho)}{\mu(1-\rho)+s}$$

$$W(s) = (1-\rho) + \frac{\rho\mu(1-\rho)}{\mu(1-\rho)+s} \qquad (6)$$

From the transform of W(s) we conclude that W is with probability (1−ρ) equal to zero, and with probability ρ equal to an exponential random variable with parameter μ(1−ρ). Taking the inverse Laplace transform to (6)

$$W(t) = \rho e^{-\mu(1-\rho)t} \qquad (7)$$

Now we can substitute the above result in equation 2.

$$MD = \int_0^{\infty} D(x)W(x)dx = \int_0^{\infty} \frac{1}{k}\rho e^{-\mu(1-\rho)x}dx \qquad (8)$$

$$MD = \frac{\rho}{k}\int_{D_{\min}}^{D_{\max}} e^{-\mu(1-\rho)x}dx$$

$$MD = \frac{\rho}{k\mu(1-\rho)}\left[e^{-\mu(1-\rho)D_{\min}} - e^{-\mu(1-\rho)D_{\max}}\right] \qquad (9)$$

Equation 9 represents our mathematical model to compute the expected fraction of tasks that miss their deadlines. This model includes the application

parameters (the deadline distribution and the interarrival distribution) and the system distribution (service distribution and the system utilization). Thus the model can be tuned to the type of the system and the application that we examine.

One might say that this model is based only on the assumptions that we stated. This model can be used even if some of the assumptions are not fulfilled. If the deadline distribution is changed then we can recalculate the integration in equation 8. Also, if the interarrival distribution or the service distribution is not exponential, we can use the G/M/1, M/G/1 or G/G/1 model [10, 18]. In all cases our approach still correct and can be applied. In Figure 1 we use our model to represent a specific system with the following parameters,

Table 1

μ	ρ	D_{min}	D_{max}
1	0.1 to 0.9	0.5	10.5

From Figure 1, the expected fraction of missed deadlines is less than 10% when the system is 60% of the time busy. So, 90% of the tasks will be completed in the specified deadline if the system is 60% of the time busy without using any special scheduling algorithm. Also, 98% of the tasks will be completed in the specified deadline if the system is 20% of the time busy without using any special scheduling algorithm.

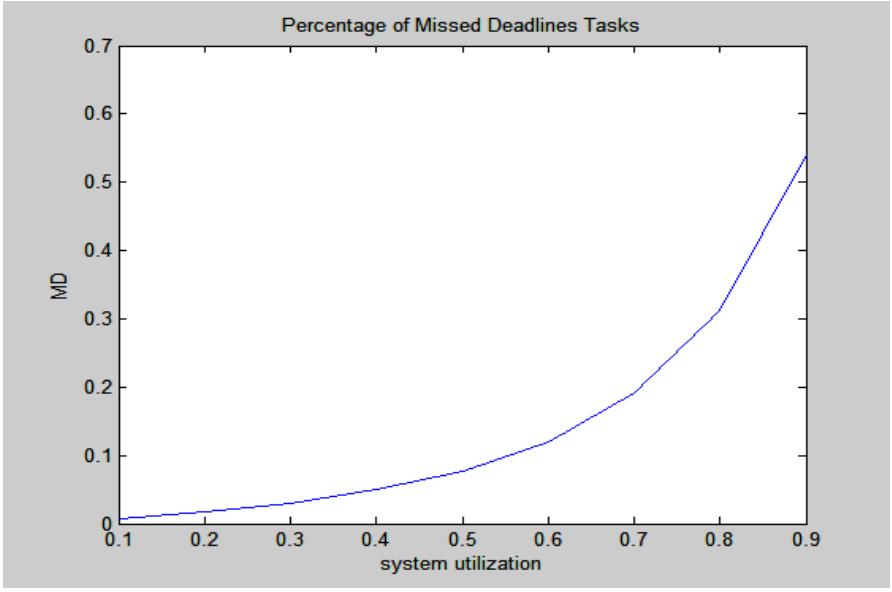

Fig 1. The expected fraction of missed deadline tasks for the values in Table 1

We should know that all of the measurements made in many research centers and universities indicate that in the worst case computing systems do not exceed 30% busy (system utilization is less than 30%) [2, 6]. In next section, we validate our analytical model with simulations that represent the real computing environment.

4. SIMULATION

Now that we have derived the analytical model, we are ready to validate our model using simulation of the real computing environment. The simulation we wrote is an event driven simulation where each event corresponds to an *epoch*. In each epoch there will be either a task arrival or a completed task leaves the system. We have carried out an exhaustive set of simulation calculations over several parameters. Since the results are consistent with each other we only present few here. Each simulation run includes servicing 10,000 tasks. Each task is associated with its deadline. The deadline is a random variable that is randomly chosen from a uniform distribution. The task service time is also a random variable that is chosen from exponential distribution. Each simulation run calculates the percentage of the missed deadline tasks for system utilization (ρ) from 0.1 to 0.9. To have stable results, we use the average of 100 simulation runs.

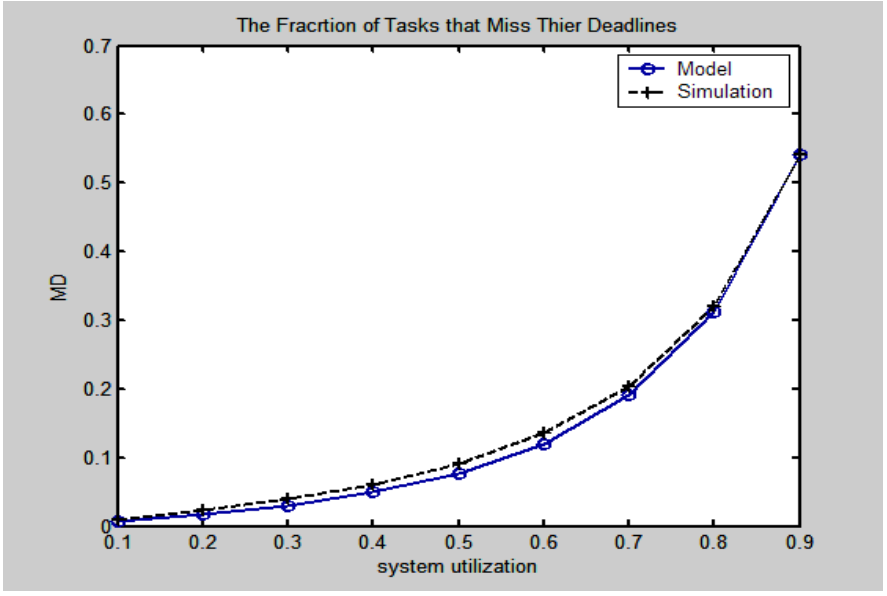

Figure 2. The percentage of missed deadline tasks when the deadlines are chosen from a uniform distribution with minimum value 0.5 and maximum value 10.5.

In Figure 2 we plot the percentage of missed deadline tasks for both the mathematical model and simulation. The deadlines are chosen from a uniform distribution with minimum value 0.5 and maximum value 10.5. These application parameters are considered as moderate deadline tasks since the average deadline is five units of time and the average interarrival times between tasks ranges from 10 time units to 1.1 time units. As we mentioned earlier each point in the figure is the average of 100 simulation runs and each simulation run includes servicing 10,000 tasks. The worst error between our model and the simulation is 1.47 % that occurred at (ρ = 0.6). At this point our model predicted a missed deadline of 12.06% of the tasks where the simulation results showed a missed deadline of 13.53 % of the tasks. The average error between our model and the simulation is 0.859%.

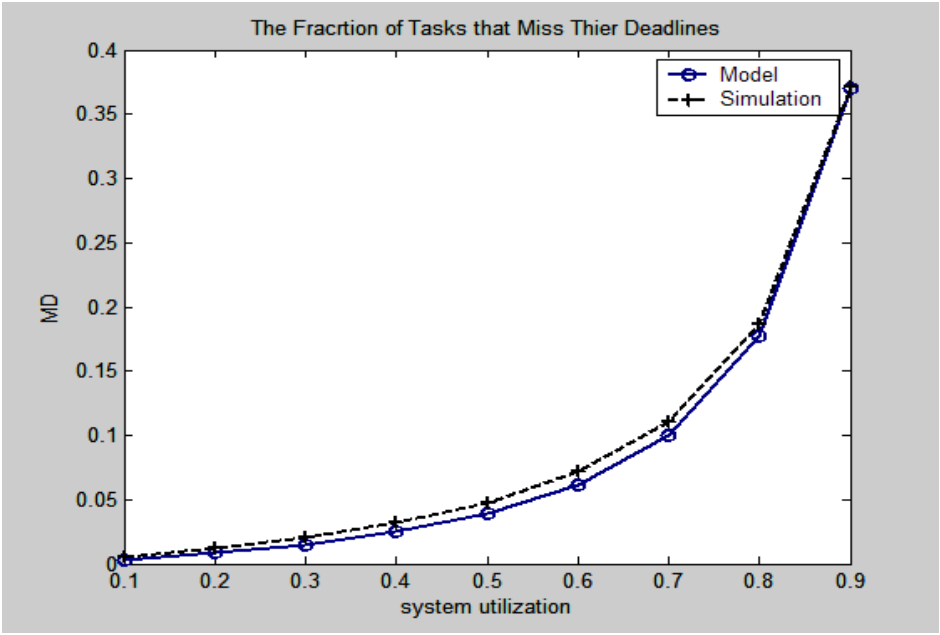

Figure 3. The fraction of missed deadline tasks when the deadlines are chosen from a uniform distribution with minimum value 0.5 and maximum value 20.5.

In Figure 3, the deadlines are chosen from a uniform distribution with minimum value 0.5 and maximum value 20.5. This system is considered as relaxed deadline system since we fixed the average interarrival and average service times and increased the average deadline to 11 time units. The worst error between our model and the simulation is 1.04 % that occurred at (ρ = 0.7). At this point our model predicted a missed deadline of 10.02% of the tasks where the simulation results showed a missed deadline of 11.06 % of the tasks. The average error between our model and the simulation is 0.633%.

In Figure 4, the deadlines are chosen from a uniform distribution with minimum value 0.5 and maximum value 5.5. This system is considered as tight deadline system since we fixed the average interarrival and average service times and reduced the average deadline to 3 time units. The worst error between our model and the simulation is 1.62 % that occurred at (ρ = 0.5). At this point our model predicted a missed deadline of 14.3% of the tasks where the simulation results showed a missed deadline of 15.92 % of the tasks. The average error between our model and the simulation is 0.9789%.

In our simulation we considered three cases the tight deadline tasks, the moderate deadline tasks and the relaxed deadline tasks. Also, our results show that it is very important to study the computing system carefully before making the choice to employ a complex scheduling algorithm for soft real time tasks. We showed that in most of the cases the computing system can guarantee more than 90% of the tasks to be executed in time less than their deadline.

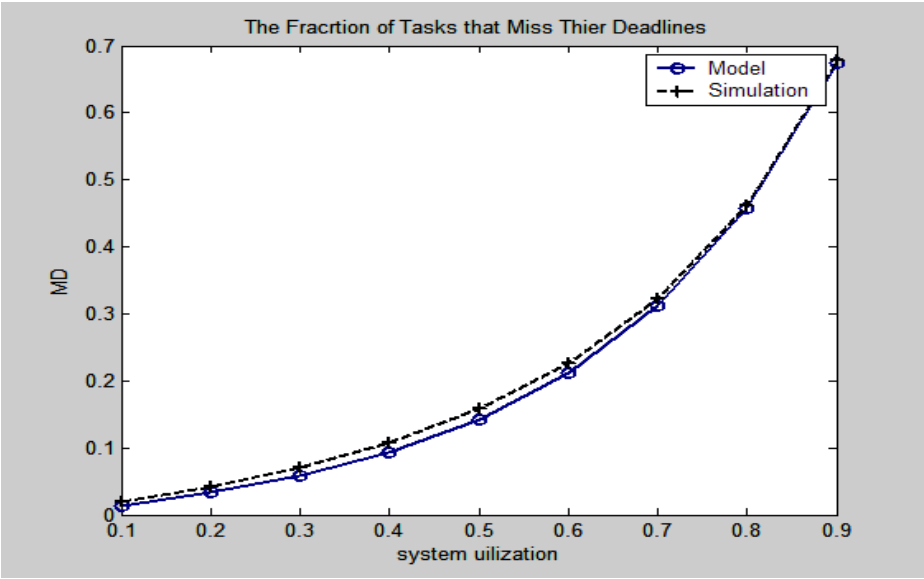

Figure 4. The fraction of missed deadline tasks when the deadlines are chosen from a uniform distribution with minimum value 0.5 and maximum value 5.5.

5. CONCLUSION

The scheduling of soft real time tasks is an important problem these days especially with the existence of many online applications (i.e., online banking, e-learning ,..,etc). We have developed a mathematical baseline model for the scheduling of soft real time tasks. The model includes the significant performance factors that affect the behavior of the system. The model can calculate the average number of tasks that miss their deadlines under a specific set of application and system parameters. We validated our results with extensive simulations to the real computing environment. Different application demands are considered, the tight deadline applications, the moderate deadline applications and the relaxed deadline applications. The results that we presented show two important results. First, the mathematical model can predict the fraction of missed deadline tasks accurately. Second, it is important to examine the system and application parameters carefully before applying any complex scheduling algorithm or using very expensive resources.

REFERENCES

1. G. Buttazzo, "Hard real-time computing systems: predictable scheduling algorithms and applications". Kluwer Academic Publishers, 1997
2. R. Buyya, "High Performance Cluster Computing: Architecture and Systems," Prentice Hall PTR, NJ, 1999.
3. T. Casavant, J. Kuhl, ''A Taxonomy of Scheduling in General Purpose Distributed Computing Systems'', IEEE Transactions on Software Engineering Vol. 14, pp. 141-154, Feb. 1988.

4. Chandra, Adler, P. Goyal, and P. Shenoy, "Surplus fair scheduling: Aprop ortional-share CPU scheduling algorithm for symmetric multiprocessors", In Proc. of 4th Symposium on Operating System Design and Implementation, Oct. 2000.
5. K. Duda and D. Cheriton. "Borrowed-virtual-time (BVT) scheduling: Supporting latency-sensitive threads in a general purpose scheduler". In Proc. of 17th Symposium on Operating Systems Principles, Dec. 1999.
6. I. Foster and Kesselman, "The Grid: Blueprint for a New Computing Infrastructure," Morgan-Kaufmann, 1998.
7. P. Goyal, X. Guo, and H. Vin. "A hierarchical CPU scheduler for multimedia operating systems" In Proc. of Symposium on Operating System Design and Implementation, Oct. 1996.
8. K. Jeffay, F. D. Smith, A. Moorthy, and J. Anderson. "Proportional share scheduling of operating system services for real-time applications." In Proc. of the 19th IEEE Real-Time Systems Symposium, Dec. 1998.
9. M. Jones, D. Rosu, and M. Rosu. "CPU reservations & time constraints: Efficient, predictable scheduling of independent activities" In Proc. of 16th Symposium on Operating Systems Principles, Oct. 1997.
10. L. Lipsky, "QUEUEING THEORY: A Linear Algebraic Approach", McMillan, New York, 1992.
11. C. Liu, W. Layland, "Scheduling algorithms for multiprogramming in a hard real time environment" Journal of ACM, Vol., 20(1), pp. 46:61, 1973.
12. S. Manolache, "Analysis and optimization of real-time systems with stochastic behaviour" PhD thesis, Linkpings University, 2005.
13. Nieh, Lam. "The design, implementation and evaluation of SMART", In Proc. of 16th Symposium on Operating Systems Principles, Oct. 1997.
14. Ed Overton, T. Brylawski and J. Anderso "A Foray into Uniprocessor Real-Time Scheduling Algorithms and Intractability" Dec. 1997-UNC CH
15. R. Rajkumar, K. Juvva, A. Molano, and S. Oikawa. "Resource kernels: A resource centric approach to real-time systems" In Proc. of SPIE Multimedia Computing and Networking Conference, Jan. 1998.
16. J. Regehr, Stankovic, "HLS: A Framework for Composing Soft Real-Time Schedulers" Proc. of the 22nd IEEE Real-Time Systems Symposium, Dec. 2001.
17. B. Theelen, "Performance Modeling for System-Level Design" PhD thesis, Eindhoven University of Technology, 2004.
18. K. Trividi, "Probability & Statistics with Reliability, Queueing and Computer Science Applications", Prentice-Hall, Inc., New Jersey, 1982.
19. Yuan, Nahrstedt, "Energy-efficient CPU scheduling for multimedia applications", ACM Transactions on Computer Systems, Vol. 24, No. 3, p.292-331, Aug. 2006.

نموذج تقييمي لجدولة التطبيقات ذات الوقت الحقيقى الرقيق على الأنظمة ذات المعالج الواحد

تقوم هذه الورقة البحثية بدراسة العوامل المختلفة التى تؤثر على كفاءة أداء جدولة التطبيقات ذات الوقت الحقيقى الرقيق على الأنظمة ذات المعالج الواحد. هذة البيئة غنية بالعوامل التى تؤثر على كفاءة أداء هذة الأنظمة. يهدف هذا العمل الى نمذجة هذة البيئة وتقديم نموذج لتقدير نسبة التطبيقات التى لا تستطيع اللحاق بزمنها التنفيذى قبل تطبيق أى خوارزم خاص بالجدولة. يقوم النموذج بتقديم العديد من الحالات التى يستطيع فيها النظام التنفيذى تقديم نسبة قليلة من التطبيقات التى لا تستطيع اللحاق بزمنها التنفيذى دون الحاجة الى أى خوارزم خاص بالجدولة. و نقوم أيضا بتقييم هذى النموذج باستخدام محاكاة للبيئة التنفيذية الفعلية. أظهرت نتائج المحاكاة أن نموذجنا الرياضى يستطيع تقديرنسبة التطبيقات التى لا تستطيع اللحاق بزمنها التنفيذى بدقة .