VALIDATION OF SCHEDULING ALGORITHMS FOR APERIODIC TASKS IN THE REAL TIME SYSTEMS

Goran Martinović*, Željko Hocenski*, Leo Budin**,
* Faculty of Electrical Engineering, Josip Juraj Strossmayer University of Osijek
Kneza Trpimira 2b, 31000 Osijek, Croatia
E-mail: goran.martinovic@etfos.hr, zeljko.hocenski@etfos.hr
** Faculty of Electrical Engineering and Computing, University of Zagreb
Unska 3, 10000 Zagreb, Croatia
E-mail: leo.budin@fer.hr

Abstract: *Mixture of periodic and aperiodic task sets is very often in the scheduling. This mixture is scheduled by the combination of the periodic and aperiodic tasks scheduling algorithms. Validation of these algorithms could be made by the developed visual software support. Timing properties of the scheduling algorithms are in opposition to the resource utilization and workload. Diagrams of the aperiodic response time, number of missed deadlines and resource utilization gives possibility for the mutual comparison of the tested algorithms. The validation is performed through the proposed criteria. Algorithms with the server tasks and bandwidth preservation give better results.*

Keywords: validation, task, scheduling algorithm, response time, utilization, deadline

1. Introduction

In the given uniprocessor system periodic tasks have hard, while aperiodic tasks have soft deadlines. The tasks are independent and could be mutually preempted. The periodic tasks are presented by the execution time (c) and period (p) and the aperiodic tasks by the execution time (c) and interarrival time (t_{ia}). The input task characteristics are determined by measurement or by estimation. The ready time of the aperiodic task (t_r) is approximated by the selected statistical distribution, while the ready time of the periodic task is determined by the period.

A generalized model of tasks based on the arrival characteristics is the mixture of periodic and aperiodic tasks. This mixture is a union of periodic and aperiodic task sets in set theory. The scheduling algorithm activity results with the time determined execution vector (Martinović, 2000).

Simulation of the scheduling algorithm execution by a sufficient number of experiments and defined execution conditions performs good presentation of the algorithm properties. Two algorithms for the periodic tasks scheduling and five algorithms for the aperiodic tasks scheduling are validated. The validation procedure is simplified through the mutual comparison of scheduling results. These results are presented by diagrams of important validation parameters as functions of the aperiodic workload.

2. Validation Software Support and Scheduling Algorithms

The validation software support block diagram used for the algorithm validation is presented in Fig.1. It is realized as a fully visual user interface, which supports an easy definition of scheduling conditions, task sets generation, algorithm characteristics, simulation execution, analysis, visualization and saving of simulation results. The validation procedure is based on the C++ source code of the scheduling algorithm which is incorporated into the visual environment (Martinović, 2000).

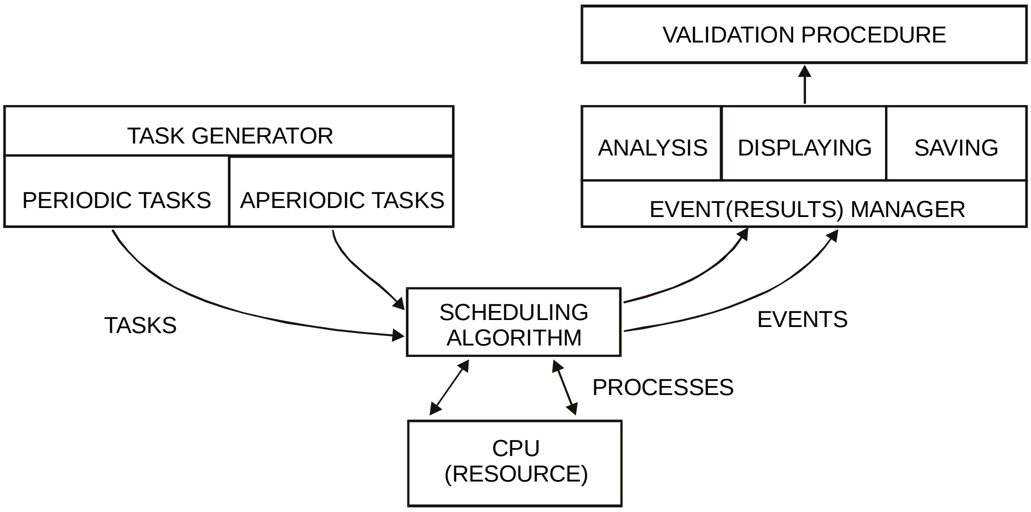

Fig.1. The validation software support block diagram

It is necessary to define tasks behavior based on the application before the task sets generation (Panzieri et al, 1993). After that the task generator performs periodic and aperiodic task sets. All of the tasks parameters are saved in the database, but only the ready time of aperiodic tasks is generated by one of the implemented distributions like normal, constant or Poisson distribution (Martinović, 2000).

The algorithms dedicated to the periodic tasks scheduling are based on the tasks sorting by the significant parameter. With the RMS algorithm tasks are sorted by the period, but with the EDF algorithm by the deadline. The implemented algorithms for the aperiodic tasks scheduling are based on the conversion of the aperiodic to periodic tasks (Kopetz, 1997). The simplest is the background processing algorithm (BP), because it processes aperiodic tasks only if the periodic tasks are not executed. The BP does not generate server tasks. The polling server (PS) generates server tasks periodically, but does not protect the aperiodic execution time. The deferrable server (DS) and the priority exchange server (PES) generate server tasks and protect aperiodic time. While the DS has one priority level, PES has more levels, but it has a greater server size. The synthesis of good characteristics of the DS and PES algorithms is implemented in the sporadic server algorithm (SS). It protects the aperiodic execution time and replenishes only the consumed aperiodic servicing time. The basic characteristics of the aperiodic scheduling algorithms are the capacity and the period of the server task (Sha et al, 1989).

The execution vector is generated by the combined action of the periodic and aperiodic scheduling algorithms. It contains the execution order of both task types and by this order tasks are assigned to the processor. The task execution results as events (Goossens et al, 1998). The event manager collects events and selects significant events. Significant events are simulation results, which are analyzed, processed and presented by the 2D diagrams and tables. The results of the analysis are saved in the database.

The algorithm validation is done based on the simulation results and by the proposed validation criteria. A general description of the validation is described in (Levi et al, 1995).

3. Validation Parameters

Some parameters important for the validation are explained in the following text (Goossens et al, 1998, Martinović, 2000). Some of these parameters are produced by the simulation, some by the analysis after the results presentation.

The response time ($t_{response}$) is the time interval between the task execution request (used equal to the ready time) and the end of task execution. It can be calculated for the periodic and aperiodic tasks as the difference between the end time t_e and the ready time t_r, according to the expression:

$$t_{response} = t_e - t_r \tag{1}$$

The mean response time ($\bar{t}_{response}$) is the mean time between the task set request and the end of task set execution for any of n tasks, as in expression:

$$\bar{t}_{response} = \sum_{i=1}^{n} \frac{t_{response}}{n} \tag{2}$$

The resource utilization (U) is equal to the ratio of the task execution time and the task period for any task i, as shown in:

$$U_i = \frac{c_i}{p_i} \tag{3}$$

The resource utilization of the task set is calculated as the mean resource utilization. It is possible to separately express periodic and aperiodic task set utilization. The resource utilization of the mixture of both task sets U_M can be calculated according to (4) or as the sum of all ratios of the processor busy time (t_w) and total simulation time (t_{sim}) for all n iterations, as in the following expression:

$$U_M = \sum_{i=1}^{n} \frac{t_w}{t_{sim}} \tag{4}$$

The server task utilization (U_{T_s}) is the ratio of the sums of all server task execution times, which service aperiodic requests ($C_{T_s}^{ef}$) and sum of total execution times of all generated server tasks ($C_{T_s}^{tot}$). The server task utilization is an important parameter for the algorithms, which create server tasks and can be calculated by this expression:

$$U_{T_s} = \frac{C_{T_s}^{ef}}{C_{T_S}^{tot}} = \frac{\sum_{i=1}^{n_{ef}} C_{Ti_s}^{ef}}{\sum_{j=1}^{m_{ef}} C_{Ti_s}^{tot}} \tag{5}$$

The number of tasks with missed deadlines (n_{md}) is the number of tasks from the input task sets whose execution times are greater than deadline ($c > t_d$).

The resource breakdown utilization (Martinović, 2000) is the value of resource utilization for which the system can guarantee satisfaction of the requested deadline level. It can be determined from the utilization diagram and from the diagram of the numbers of the missed deadlines. This is the utilization value (for the single algorithm) for which the number of missed deadlines is less than required.

The aperiodic breakdown workload (Martinović, 2000) is the value of the aperiodic workload for the constant periodic workload at which resource breakdown utilization occurred. It shows a boundary value of the aperiodic workload.

4. Experimental Results

An example of the scheduling algorithms validation is performed by the developed validation software support. The periodic and aperiodic tasks sets contain 11 tasks, which are presented by the execution time, by the period and the interarrival time. The simulation duration is 10,000 time units. The RMS algorithm for the periodic task scheduling and all of the implemented aperiodic scheduling algorithms are validated. The Poisson distribution determines ready time of the aperiodic tasks and the distribution parameter is 0.5. The constant periodic workload is 69%. The display range of the workload area is 0-100% with the step of 10%, the server task capacity is 3 and period 10. The mean aperiodic response time is presented in Fig.2. The number of the tasks with the missed deadlines are presented in Fig.3 and the resource utilization is shown in Fig.4. The mean resource utilization and the effective utilization of the server tasks are calculated by simulation.

5. Analysis of the Results and Validation

The mutual comparison procedure simplifies the validation as the comparison is made using the graphic form of the results. The periodic workload is set to the periodic workload boundary value (at 69%) for the RMS algorithm. The high schedulability and high resource utilization are guaranteed at this workload value.

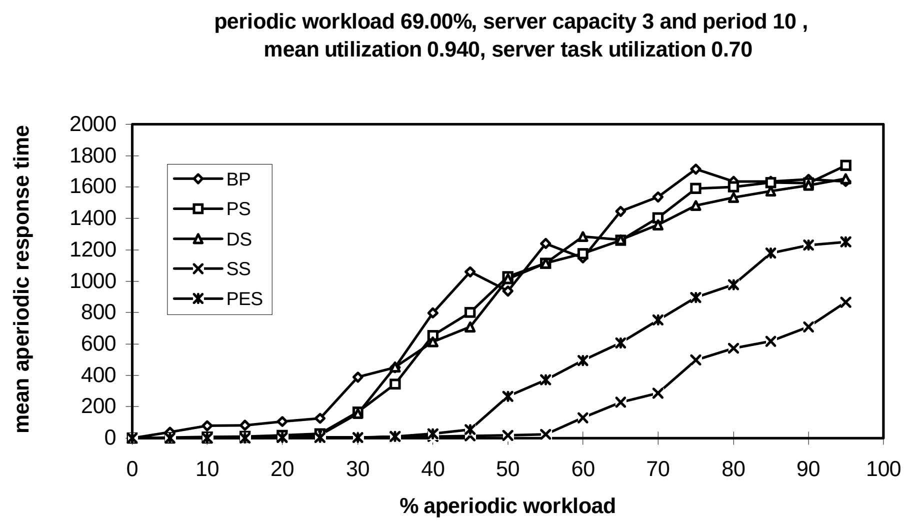

Fig.2. The diagram of mean aperiodic response time.

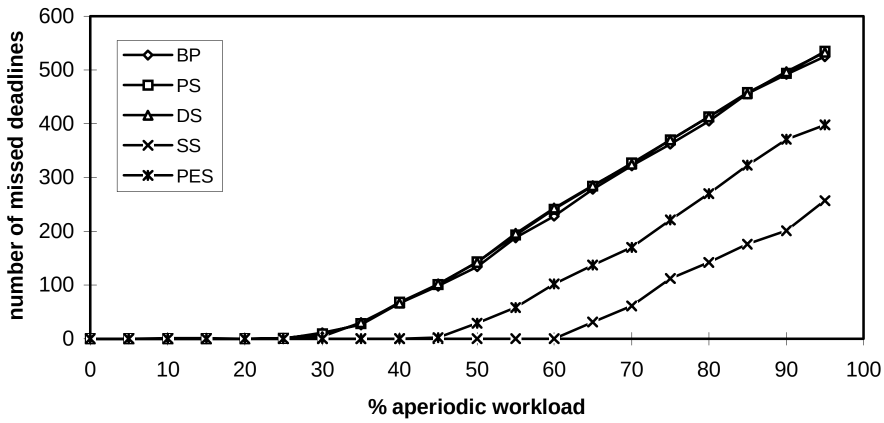

Fig.3. The diagram of number of missed deadlines

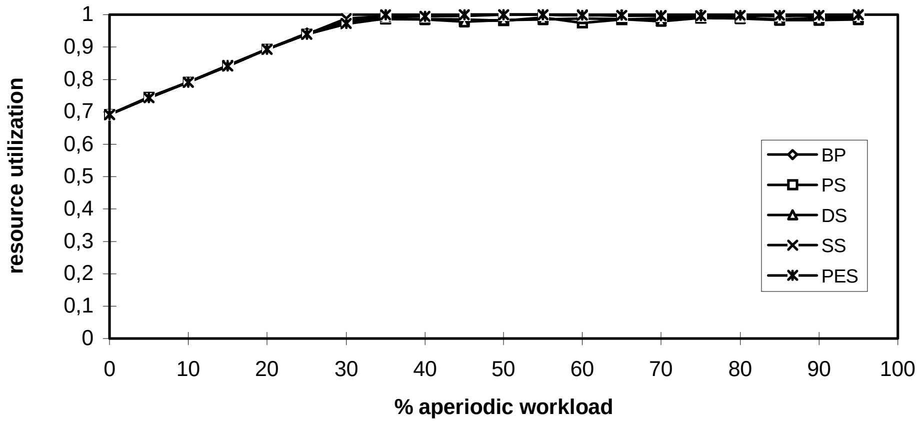

Fig.4. The diagram of resource utilization

The mean aperiodic response time (Fig.2) is very good for all of the algorithms under the aperiodic workload of 30%. Over this workload the mean aperiodic response time becomes greater first for the BP and after for the PS and the DS algorithm. Over the workload of 40% this parameter gives poor values for the PES algorithm and over 55% for the SS algorithm. Through the complete range of the workload the BP algorithm has the worst value of the mean aperiodic response time, somewhat better are PS and DS algorithms. The best results are obtained for the SS and the PES algorithm. The PES and the SS algorithms are also suitable for a high aperiodic workload.

The diagram of the number of tasks with the missed deadlines (Fig.3) has a form similar to the previous diagram (Fig.2). The best results are obtained for the SS and the PES algorithm.

The resource utilization presented in Fig.4 is very high and close to the value of 100% for the aperiodic workload of 35% and more regarding to the relative high periodic workload. It confirms the calculated mean resource utilization of 94%. The mean effective utilization is very good (70%).

The resource breakdown utilization is possible to be determined for a single selected algorithm. If the number of missed deadlines for the SS algorithm is 200, for example, this results in resource utilization less than 98%, the aperiodic breakdown workload is 73% by the periodic workload 69%.

6. Conclusion

The validation of the scheduling algorithms is a usual procedure, which proceeds to the application of algorithms in the real-time systems. The validation software support with visual interface is developed for this purpose. This support performs the validation based on the sufficient number of simulations. The increase of the aperiodic workload by the constant periodic workload influences the scheduling algorithm characteristics. By increasing the aperiodic workload the aperiodic response time and the number of missed deadlines are increased. The resource utilization, the mean resource utilization and the effective utilization of the server tasks are also increased. The resource breakdown utilization and aperiodic breakdown workload are boundary values, which are determined from the diagrams. As expected, the best results are performed by the SS and PES algorithms, which generate server tasks and preserve the aperiodic bandwidth. The PS and DS algorithms without bandwidth preservation give poor results. The worst is the simplest algorithm, the BP algorithm. The selection of the best algorithm is often based on the best resulting parameters in the required aperiodic workload range. The algorithm for the aperiodic tasks scheduling could not be optimal for scheduling of the mixture of periodic and aperiodic tasks if it makes timing properties of the periodic tasks worse.

7. References

Goossens, J. and Hernalsteen, C. (1998) "A Tool for Statistical Analysis of Hard Real-Time Scheduling Algorithms", *31st Annual Simulation Symposium (SS '98)*, Boston, MA, USA, Apr. 5-9, 1998, http://www.computer.org/CSPRESS/CATALOG/pr08418.

Kopetz, H. (1997) *Real-Time Systems Design Principles for Distributed Embedded Applications*, Kluwer Academic Publishers.

Levi, S.T., Agrawala, A. K. (1995) *Real Time System Design*, McGraw – Hill International Editions, Computer Science Series.

Martinović, G. (2000), *Radno okruženje za vrednovanje algoritama raspoređivanja u sustavima stvarnog vremena*, Master thesis, Faculty of Electrical Engineering and Computing, University of Zagreb.

Martinović, G., Hocenski, Ž. and Budin, L. (1999), "Scheduling Algorithms for the Aperiodic Tasks in Embedded Real-Time Computing Systems", *Proc. Computer in Technical Systems,* MIPRO '99, Opatija, May 17-21, 1999, pp. 21-24.

Panzieri, F., Donatiello, L. and Poretti, L. (1993) "Scheduling Real-Time Tasks: A Performance Study", Technical report, Laboratory for Computer Science, University of Bologna, Italy, May 1993, http://www4.ncsu.edu/~alexe/class/ece292a/RTsched/index.html.

Sha, L., Sprunt, B. and Lehoczky, J. (1989), "Scheduling Sporadic and Aperiodic Events in a Hard Real –Time System", The Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, USA, Apr 1989, http://www.sei.cmu.edu/website/siteIndex/siteIndexTR.html.