Towards Exact Thresholds for Scheduling n Tasks on m Processors Based on Partitioned EDF

Dirk Müller and Matthias Werner
Chemnitz University of Technology
Operating Systems Group
D-09107 Chemnitz, Germany
{dirkm|mwerner}@cs.tu-chemnitz.de

Abstract

Partitioned preemptive EDF scheduling is very similar to bin packing, but there is a subtle difference. Estimating the probability of schedulability under a given total utilization has been studied empirically before. Here, we show an approach for closed-form formulae for the problem, starting with $n = 3$ tasks on $m = 2$ processors.

1. Introduction

We consider periodic, preemptive, synchronous, independent tasks with implicit deadlines on an identical multiprocessor with m processors or cores. Say n tasks consisting of a potentially infinite number of jobs to be scheduled. A task is then fully characterized by only two parameters: its worst-case execution time (WCET) $e_i > 0$ and its period $p_i > 0$ which coincides with its deadline $d_i = p_i$. A useful derived property is the task's utilization $u_i := \frac{e_i}{p_i}$.

All deadlines should be met. It is tolerable that some deadlines are exceeded. Thus, we consider soft real-time scheduling contrary to hard real-time where all deadlines must be met. Preemptions are always possible and their costs (in terms of time) are assumed to be zero.

Global scheduling is characterized by a common queue of ready jobs and a migration of jobs from one processor to another. We consider here only *partitioned* scheduling where each processor has its own queue. Then, the main problem is to map the tasks to the processing units. Partitioned scheduling is better understood since it relies on well-studied uniprocessor scheduling.

Earliest Deadline First (EDF) [12] is an optimal dynamic scheduling policy based priorities according to increasing absolute deadlines. It is optimal since it achieves the maximal possible total utilization $u = \sum_{i=1}^{n} u_i$ which is 1.

By combining partitioned scheduling with EDF, we obtain partitioned EDF which is closely related to the classical problem of *bin packing*. The bins have a constant capacity of 1 resulting from the uniprocessor EDF utilization bound. Following the classical approach of bin packing, one has to minimize the number of bins (processors) given a set of items (tasks). This is not typical in practice since the number of cores or processors, the hardware, is fixed in a characteristic setting. Thus, it is more useful to indicate tolerable total utilization u for a fixed number of bins m and a given number of tasks n.

In hard real-time systems, the miss of a single deadline is critical. Hence, utilization bounds taking the *worst* case into account are appropriate. On the other hand, the probability of failing to meet a deadline might be very low at utilizations beyond such a bound. It turns out that there is such a region in many cases. The typical behavior in scheduling is a phase transition with a sharp threshold. Then, for soft real-time systems, it is justified to include the interval from the utilization to the (sharp) utilization threshold in the set of tolerable total utilization values.

1.1. Motivation

It is the goal of this article to pioneer the calculation of exact threshold values by analytical formulae. The great advantage for soft real-time systems is the better exploitation of available hardware by substituting conservative utilization bounds by utilization thresholds. An important point is to ensure the boundedness of the probability of a deadline miss. For the particular scheduling problem here, it means to bound the likelihood of a task-processor matching where a per-processor utilization exceeds the uniprocessor EDF bound of 1. Thus, a good characterization is the function $Pr(u, n, m)$ which should take course of a sharp threshold for bounding processor overflow probability.

The rest of the article is structured in the following way. In Section 2, a utilization bound is recapitulated. The general concept of thresholds and sharpness of them is given in Section 3. Next, the concept is discussed when applied to RMS in Section 4. Section 5 sheds light on the relationship between partitioned EDF and bin packing. In Section 6, empirical and analytical approaches for partitioned EDF are compared. Finally, a conclusion is given in Section 7.

2. Utilization Bound for Partitioned EDF

The intention of a utilization bound u_B is to provide us a linear-time sufficient test for schedulability. The summation of n task utilization values u_i takes linear time. Bounds related to utilization are typically *upper* bounds. All task sets with a u value lower than or equal to u_B are schedulable. Clearly, the most useful upper bound is the highest one possible. Then, there is a task set and an arbitrarily small $\varepsilon > 0$ such that its utilization is $u_B + \varepsilon$ and it is not schedulable. Such a bound is called a *tight* bound. The most famous example for uniprocessor scheduling is the Liu/Layland bound for Rate-Monotonic Scheduling (RMS) $u_{LL}(n) = n(2^{1/n} - 1)$ with its limit case $u_{LL}(\infty) = \ln 2$, cf. [12]. Subsequently, the term bound refers to an abbreviation of tight upper utilization bound.

Partitioned scheduling is characterized by a static mapping of tasks to processors. Thus, a single task can block a processing unit when having a utilization of $0.5 + \varepsilon$ with $\varepsilon > 0, \varepsilon \to 0$ and all other tasks having the same utilization value. In order to block m processors, m such tasks are required. Then, the $(m+1)$-th task can not be attached. An infinitesimally small decrease in its u_i value by 2ε would render the task set schedulable since, then, it would exactly fit to an arbitrary processor. So, the utilization bound for partitioned scheduling is the total utilization in this worst-case situation: $\frac{m+1}{2}$. Asymptotically, for $m \to \infty$, it becomes normalized[1] 0.5. Just a half of the potential processing capacity can be used, see (1) and (2), formulae for utilization bound and *normalized* utilization bound.

$$u_B = \frac{m+1}{2} \tag{1}$$

$$u_{BN} = \frac{m+1}{2m} \tag{2}$$

This bound was shown to be valid in the popular special cases of partitioned EDF [14] [13] and partitioned RMS [1]. Note that for partitioned scheduling, the bounds for static and dynamic priority scheduling coincide.

Contrary to that, for global scheduling, the normalized bounds are $\frac{2}{3+\sqrt{5}} \approx 0.382$ (with Slack-Monotonic Utilization Separation, SM-US [2]) in the static case and $\frac{m+1}{2m} \to 0.5$ (with EDF-Utility Separation, EDF-US[0.5] [4]) in the dynamic case. Note that the non-constructive normalized bound for global static-priority scheduling is $\sqrt{2} - 1 \approx 0.414$ under the constraint of only using e_i and p_i values [3]. Thus, there still remains a normalized gap of more than 0.03 to improve on a global static priority scheme. Relaxing this restriction by allowing more complex priority assignments leads to an even larger gap.

Both EDF and RMS without an improvement by the US modification suffer in the global case from the well-known Dhall's effect [6] where guaranteed utilization drops to 1 corresponding to a normalized u value of zero[2]. This shows clearly that a straight-forward lifting of approved uniprocessor algorithms can result in a disaster; more sophisticated approaches are required.

3. Thresholds and Sharpness

Utilization thresholds are inherent to many scheduling problems. This becomes obvious when noting that small total utilizations u imply small average utilizations u_i often resulting in utilization bounds u_B discussed in Section 2. This guarantee for bounded utilizations corresponds to a probability of 1 for schedulability. There is always such a guarantee on uniprocessors, and often on multiprocessors. An exception are the policies affected by Dhall's effect, global EDF and global RMS, see Section 2. But even there, a degenerated utilization bound equal to 1 can be regarded.

At the other end of the spectrum, at high utilizations, the necessary condition[3] of $u \leq m$ is general. Thus, its violation implies always a likelihood of schedulability of 0.

Relating these two fixed data points and regarding the interior of the $[u_B, m]$ interval makes clear that there must occur a changeover from probability 1 to 0. This knowledge becomes useful when one can show that it is a *sharp* threshold u_T. Then, in the interval (u_B, u_T), there is a high probability of schedulability, and in (u_T, m), there is a high probability of unschedulability. Exactly at u_T, the odds are fifty-fifty. The terms threshold and sharp threshold are mathematically defined by Gopalakrishnan in [10] who first tackled real-time scheduling problems using sharp utilization thresholds in chapter 6 of his PhD thesis [9]. Further, there is shown that the width of the threshold interval will narrow. When the number of tasks approaches infinity, $n \to \infty$, it even converges to a binary switch from 1 to 0. More precisely, the width is $\mathcal{O}(1/\sqrt{n})$, cf. [10].

Real-time scheduling can profit from sharp thresholds by accepting task sets with $u \in (u_B, u_T]$. Due to the sharpness of the threshold, the probability of obtaining a false

[1]This means a scaling with the reciprocal of the number of processors m. Such a normalization is useful for a comparison with uniprocessor bounds since the result is independent of the number of processors.

[2]But, we have to admit that tardiness under global EDF is bounded. Thus, it might be an option for soft real-time systems.

[3]There is a second necessary condition which is frequently no more mentioned and implicitly assumed: $\forall i : u_i \leq 1$.

positive decision is small. Symmetrically, task sets with $u \in (u_T, m]$ should be rejected. Again, the probability to obtain a false negative result is small.

4. Thresholds for RMS

In [10], RMS on a uniprocessor is analyzed. By resorting to a task set graph abstraction, it can be shown that RMS is provided with a sharp utilization threshold. The threshold value can be estimated between 0.85 and 0.9. Since the RMS utilization bound $u_{LL}(\infty) = \ln 2 \approx 0.69$, see Section 2, is much less than the estimated threshold, it makes sense to use the threshold obtained for RMS. Similar measures going beyond the pessimistic utilization bound concept, see Section 2, are *Breakdown Utilization, BU* [11] and *Numerical Optimality Degree, NOD* [5]. These metrics are compared and discussed in [5]. They are obtained empirically by probabilistic simulation. Such a method bears always the danger of a bias, cf. [5]. There, it turned out that NOD is more fair than BU.

5. Partitioned EDF and Bin Packing

In [9], partitioned EDF on multiprocessors is considered. Again, a sharp threshold could be proved. Partitioned EDF is closely related to bin packing, but in some respect a dual problem to it. Bin packing is (3), but partitioned EDF is (4).

$$m \to \min. \qquad \text{subject to} \qquad u = \text{const.} \tag{3}$$

$$u \to \max. \qquad \text{subject to} \qquad m = \text{const.} \tag{4}$$

Partitioned EDF and bin packing are NP-Hard [8]. Thus, heuristics and tests of polynomial complexity are required. In [9], the approved heuristic First-Fit Decreasing (FFD) was applied. It is known to require in the worst case $\frac{11o+6}{9}$ bins where o is the optimal number of bins. The bound is tight and was proven in [7]. Note that this approximation ratio result is lacking a link to utilization. Thus, rearranging it can not give a result of the form $u = f(m)$ as required for solving the scheduling problem.

6. Thresholds for Partitioned EDF

6.1. Experimental Evaluation

The study in chapter 6 in [9] considers $m = 2, 4, 8, 16, 32, 64$ processors and $n = 2^i m + 1; i = 0, 1, 2, 3, 4$ tasks respectively except for $m = 64$ where only the cases $n = 65, 129, 257$ are regarded. In all cases, a sharp threshold for larger values of n - as predicted by theory - was observed. These thresholds are located at about $0.95m$. It is a huge difference to the bound (2) which quickly drops from 0.75 for $m = 2$ to ca. 0.51 for $m = 64$, indicating the convergence to 0.5. Thus, the method of sharp thresholds succeeds for partitioned EDF.

But the experimental method has the following weak points. First, the application of FFD gives only an approximation from below of the achievable utilization. Second, the Monte Carlo method used here comes at the risk of biases which are sometimes hard to notice, cf. Section 4. We favor to ensure uniformity in the space of task sets with a *lattice* as proposed in the next section.

6.2. Towards Exact Thresholds by Analysis

In order to overcome the shortcomings of the experimental method, we want to try to analytically derive formulae for partitioned EDF.

For determining the schedulability probability, we first use a generalized midpoint rule for integration. An equidistant ($d > 0$) lattice of u_i values is generated in the space $(0, 1]^n$. There are $(1/d)^n$ points. Then, the ratio of instances meeting the schedulability constraints to the number of them meeting the total-utilization condition is an estimator of the probability. The number of instances under investigation is controlled by d. According to frequentists' probability, the ratio converges to $Pr(u, n, m)$ for $d \to 0$.

We start with the simplest reasonable case of $m = 2$ and $n = 3$. The interval of interest is $(u_B, m) = (1.5, 2.0)$. Here, exact constraints for schedulability can be derived when considering the EDF criterion $u \leq 1$ and that at least two of the three tasks must be attached to one and the same processor. Thus, the schedulability constraint is (5).

$$u_1 + u_2 \leq 1 \vee u_1 + u_3 \leq 1 \vee u_2 + u_3 \leq 1 \tag{5}$$

The total-utilization condition can be obtained when recognizing that there are only two degrees of freedom when u is fixed. The third utilization is just $u_3 = u - u_1 - u_2$. Then, in the $u_1 u_2$ square, it has to be checked whether a $0 < u_3 \leq 1$ is obtained.

Using a *C* program, our empirical method is just counting the points meeting the constraints and then to print out the ratio for the respective u value.

The analytical method is to calculate the areas of the appropriate shapes in the $u_1 u_2$ plane, see Fig. 1. The ratio of these areas is the exact value of probability $Pr(u, 3, 2)$, derived using geometrical method. The first total-utilization condition $u_3 \leq 1$ cuts the lower left triangle from the $u_1 u_2$ square. And the second condition $u_3 > 0$ cuts the upper right triangle. Thus, the residual shape representing all task sets meeting the total-utilization condition is a hexagon plotted with a thick line. Next, the schedulability constraint (5) cuts an inner triangle (filled with black) from this hexagon. Hence, the ratio is the area of the hatched region to the area of the hexagon. This evaluates to (6).

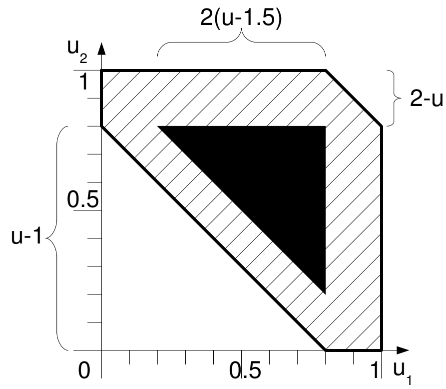

Figure 1. Derivation of the closed-form formula for $Pr(u, 3, 2)$; here: $u = 1.8$

$$Pr(u,3,2) = 1 - \frac{\frac{(2(u-1.5))^2}{2}}{1 - \frac{(u-1)^2+(2-u)^2}{2}}$$
$$= 3 + \frac{3}{2u^2 - 6u + 3}; \qquad 1.5 \le u \le 2.0 \tag{6}$$

Both empirical (for $d = 0.01$) and analytical result are given in Fig. 2. The two graphs lie close to each other confirming the analytical result. As expected, the probabilities are slightly greater than the ones obtained in [9] due to the inherent pessimism in the FFD approximation there. The threshold is $u_T \approx 1.89$. We hope to apply this approach for general n and m values in the future.

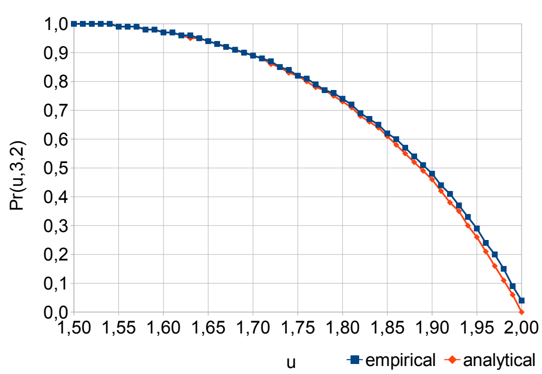

Figure 2. Probability of schedulability $Pr(u, 3, 2)$ for partitioned EDF of 3 tasks onto 2 processors; step $d = 0.01$

7. Conclusion

A formula for the partitioned-EDF scheduling of 3 tasks onto 2 processors has been derived and compared to former empirical results. The threshold is $u_T \approx 1.89$. This probabilistic result can be applied in soft real-time scheduling or as a pre-test in hard-real time scheduling. In the future, this method shall be used in order to obtain $Pr(u, n, m)$ formulae and thresholds for greater n and m values.

References

[1] B. Andersson. *Static-priority scheduling on multiprocessors*. PhD thesis, Chalmers University of Technology, 2003.

[2] B. Andersson. Global static-priority preemptive multiprocessor scheduling with utilization bound 38%. In *Proceedings of the 12th International Conference on Principles of Distributed Systems*, OPODIS '08, pages 73–88, Berlin, Heidelberg, 2008. Springer-Verlag.

[3] B. Andersson and J. Jonsson. The utilization bounds of partitioned and pfair static-priority scheduling on multiprocessors are 50%. In *Real-Time Systems, 2003. Proceedings. 15th Euromicro Conference on*, pages 33 – 40, July 2003.

[4] T. P. Baker. An analysis of EDF schedulability on a multiprocessor. *IEEE Trans. Parallel Distrib. Syst.*, 16:760–768, August 2005.

[5] E. Bini and G. C. Buttazzo. Measuring the performance of schedulability tests. *Real-Time Syst.*, 30:129–154, May 2005.

[6] S. K. Dhall and C. Liu. On a real-time scheduling problem. *Operations Research*, 26(1):127–140, 1978.

[7] G. Dósa. The tight bound of first fit decreasing bin-packing algorithm is FFD(I)=(11/9)OPT(I)+6/9. In B. Chen, M. Paterson, and G. Zhang, editors, *Combinatorics, Algorithms, Probabilistic and Experimental Methodologies*, volume 4614 of *LNCS*, pages 1–11. Springer, 2007.

[8] M. R. Garey and D. S. Johnson. *Computers and Intractability; A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1990.

[9] S. Gopalakrishnan. *Resource Management for Real Time Environments*. PhD thesis, University of Illinois at Urbana-Champaign, 2006.

[10] S. Gopalakrishnan. Sharp utilization thresholds for some real-time scheduling problems. *CoRR*, abs/0912.3852, 2009.

[11] J. P. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *RTSS*, pages 166–171, 1989.

[12] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *J. ACM*, 20(1):46–61, January 1973.

[13] J. M. López, J. L. Díaz, and D. F. García. Utilization bounds for EDF scheduling on real-time multiprocessor systems. *Real-Time Syst.*, 28:39–68, October 2004.

[14] J. M. López, M. García, J. L. Díaz, and D. F. García. Worst-case utilization bound for EDF scheduling on real-time multiprocessor systems. In *Real-Time Systems, 2000. Euromicro RTS 2000. 12th Euromicro Conference on*, pages 25 –33, 2000.