What is the Exact Speedup Factor for Fixed Priority Pre-emptive versus Fixed Priority Non-pre-emptive Scheduling?

Robert I. Davis[1] , Oliver Gettings[1], Abhilash Thekkilakattil[2] Radu Dobrin[2], Sasikumar Punnekkat[2]
[1]Real-Time Systems Research Group, Department of Computer Science, University of York, York, UK.
[2]Mälardalen Real-Time Research Center, Mälardalen University, Sweden.
rob.davis@york.ac.uk, og501@york.ac.uk, abhilash.thekkilakattil, radu.dobrin, sasikumar.punnekkat {@mdh.se}

Background

The performance of real-time scheduling algorithms can be compared in a number of different ways. *Empirical techniques* typically rely on generating a large number of task sets with parameters chosen from some appropriate distributions. The performance of the scheduling algorithms is then compared by determining task set schedulability according to exact or sometimes sufficient schedulability tests and plotting a graph of the success ratio (i.e. the proportion of task sets that are deemed schedulable) at different utilisation levels. An alternative *theoretical method* of comparing real-time scheduling algorithms is to determine the *resource augmentation* bound or *speedup factor* [6] required. This approach focuses on those task sets that are particularly difficult to schedule using one algorithm but easy to schedule using another.

The *Speedup Factor* $S(A,B)$ comparing two real-time scheduling algorithms *A* and *B* is given by the minimum factor by which the speed of the processor needs to be increased to ensure that any task set that is schedulable according to algorithm *B* is guaranteed to be schedulable by algorithm *A*. When comparison is made against an optimal algorithm (*OPT*), then $S(A,OPT)$ is referred to as the *sub-optimality* of algorithm *A*.

Combining the utilisation bounds for fixed priority pre-emptive (FP-P) and EDF pre-emptive (EDF-P) scheduling from the seminal paper of Liu and Layland [8] shows that the speedup factor $S(\textit{FP-P, EDF-P}) = 1/\ln(2) \approx 1.44270$ for implicit deadline task sets. Since EDF-P is an optimal uniprocessor scheduling algorithm [4] this result also determines the sub-optimality of FP-P for implicit deadline task sets. In 2009, Davis et al. [1] derived the exact sub-optimality of FP-P for constrained-deadline task sets; $S(\textit{FP-P, EDF-P}) = 1/\Omega \approx 1.76322$ (where Ω is the mathematical constant defined by $\ln(1/\Omega) = \Omega$, hence, $\Omega \approx 0.567143$). In 2009, Davis et al. [2] gave upper and lower bounds of *S*(*FP-P, EDF-P*) = 2 and $1/\Omega \approx 1.76322$ for the case of arbitrary deadline task sets. In 2010 [3] they gave upper and lower bounds for the non-pre-emptive case of *S*(*FP-NP, EDF-NP*) = 2 and $1/\Omega \approx 1.76322$ for implicit, constrained, and arbitrary deadline task sets. In 2013, Thekkilakattil et al. [9] provided an upper bound on the sub-optimality of non-pre-emptive EDF (i.e. *S*(*EDF-NP, EDF-P*) parameterised by the shortest task deadline and the longest task execution time.

Open Problem: Speedup Factor for FP-P v. FP-NP

EDF-P is an optimal uniprocessor scheduling algorithm [4] and so dominates EDF-NP, FP-P, and FP-NP. Further, EDF-NP dominates FP-NP [5]; however, there are no such dominance relationships between pre-emptive and non-pre-emptive fixed priority scheduling. Hence there are non-trivial speedup factors in both directions: *S*(*FP-P, FP-NP*) and *S*(*FP-NP, FP-P*). An exact speedup factor for the latter problem has recently been derived by the authors and is under submission. The former problem, determining the exact speedup factor needed so any task set that is schedulable under fixed priority non-pre-emptive scheduling is guaranteed to be schedulable under fixed priority pre-emptive scheduling is to the best of our knowledge open and forms the focus of this abstract.

Upper bounds on *S*(*FP-P, FP-NP*): since EDF-P dominates FP-NP, then the exact values of $1/\ln(2) \approx 1.44270$, $1/\Omega \approx 1.76322$, and *2* for *S*(*FP-P, EDF-P*) provide upper bounds on *S*(*FP-P, FP-NP*) for implicit, constrained and arbitrary deadline task sets respectively.

Theorem 1: A *lower bound* on *S*(*FP-P, FP-NP*) for constrained or arbitrary-deadline task sets is $S = \sqrt{2}$.

Proof: Consider the following task set scheduled on a processor of speed 1.

$$\tau_A : C_A = \sqrt{2} - 1,\ D_A = 1,\ T_A = 1,$$
$$\tau_B : C_B = (2 - \sqrt{2})/2,\ D_B = \sqrt{2},\ T_B = \infty$$
$$\tau_C : C_C = (2 - \sqrt{2})/2,\ D_C = \sqrt{2},\ T_C = \infty$$

As the task set has constrained deadlines, then under FP-P scheduling, Deadline Monotonic Priority Order (DMPO) [7] is optimal. It is easy to see that the task set is only just schedulable with this priority ordering, since any increase in execution times would cause task τ_C to miss its deadline – see Figure 1. Next consider the same task set scheduled under FP-NP on a processor of lower speed: $f = \sqrt{2}/(2-\varepsilon)$ where ε takes an infinitesimally small value. Again assume the task priorities are in DMPO. Now the sum of the execution times of task τ_A and task τ_B (or task τ_A and τ_C) can be expressed as follows:

$$2(\sqrt{2} - 1)(2 - \varepsilon)/2\sqrt{2} + (2 - \sqrt{2})(2 - \varepsilon)/2\sqrt{2} = (2 - \varepsilon)/2 < 1 \quad (1)$$

Hence task τ_A is schedulable when blocked by either of tasks τ_B or τ_C. Further, once the first jobs of tasks τ_A and τ_B have executed, the first job of task τ_C is able to start executing before the second job of task τ_A is released. Thus the first job of task τ_C has a response time of:

$$(\sqrt{2}-1)(2-\varepsilon)/\sqrt{2}+(2-\sqrt{2})(2-\varepsilon)/\sqrt{2}=(2-\varepsilon)/\sqrt{2}<\sqrt{2} \tag{2}$$

and so meets its deadline at $\sqrt{2}$. Continuing on through the busy period, the second job of task τ_A completes at: $2+\varepsilon<2$ meeting its deadline. At this point the busy period ends. Thus we have shown that all of the tasks are schedulable. Task τ_A has a worst-case response time of $(1-\varepsilon)/2<1$ and tasks τ_B and τ_C have worst-case response times of $(2-\varepsilon)/\sqrt{2}<\sqrt{2}$. Hence the task set is schedulable at speed f under FP-NP scheduling. The speedup factor required such that this task set is schedulable under FP-P therefore tends to $\sqrt{2}$ as ε tends to zero □

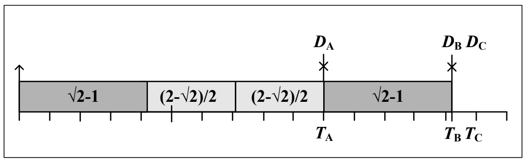

Figure 1: FP-P schedule

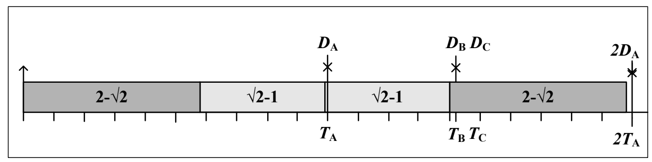

Figure 2: FP-NP schedule

Figure 3 shows the results of an empirical investigation into the speedup factor *S*(*FP-P, FP-NP*). These results were produced using a genetic algorithm which explored a wide range of different values for the task parameters (execution time, period, and deadline). The results are for 400 generations of a population of 20,000 task sets, i.e. 8 million task sets for each task set cardinality and deadline type.

For three or more tasks, then with constrained or arbitrary deadlines, the maximum speedup factor found by the genetic algorithm is very close to $\sqrt{2}\approx 1.414213562$. In fact the values range from 1.4118 to 1.4139 (constrained deadlines) and from 1.4089 to 1.4128 (arbitrary deadlines) for task sets of cardinality 3 to 10. With implicit deadline task sets, the largest speedup factor found was somewhat lower at 1.3405. The fact that the maximum value found empirically (1. 4139) is very close to but does not exceed $\sqrt{2}\approx 1.414213562$ gives credence to the hypothesis that the theoretical lower bound (of $\sqrt{2}$) on the speedup factor is the exact value. It remains an interesting open question whether or not this is the case.

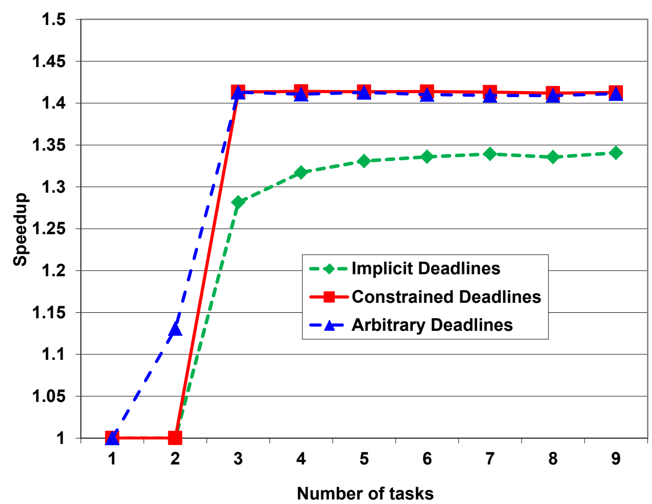

Figure 3: Empirical results for *S*(*FP-P, FP-NP*)

References

[1] Davis R.I., Rothvoß T., Baruah S.K., Burns A., "Exact Quantification of the Sub-optimality of Uniprocessor Fixed Priority Pre-emptive Scheduling." *Real-Time Systems*, Volume 43, Number 3, pages 211-258, November 2009.

[2] Davis, R.I., Rothvoß, T., Baruah, S.K., Burns, A., "Quantifying the Sub-optimality of Uniprocessor Fixed Priority Pre-emptive Scheduling for Sporadic Task sets with Arbitrary Deadlines". *In proceedings of Real-Time and Network Systems (RTNS'09)*, pages 23-31, October 26-27th, 2009.

[3] Davis R.I., George L., Courbin P., "Quantifying the Sub-optimality of Uniprocessor Fixed Priority Non-Pre-emptive Scheduling". In proceedings 18th International Conference on Real-Time and Network Systems (RTNS'10), pages 1-10, November 4-5th, 2010.

[4] Dertouzos M.L., "Control Robotics: The Procedural Control of Physical Processes". *In Proc. of the IFIP congress*, pages 807-813, 1974.

[5] George, L., Muhlethaler, P., Rivierre, N., "Optimality and Non-Preemptive Real-Time Scheduling Revisited," Rapport de Recherche RR-2516, INRIA, Le Chesnay Cedex, France, 1995.

[6] Kalyanasundaram B., Pruhs K., "Speed is as powerful as clairvoyance". *In Proceedings of the 36th Symposium on Foundations of Computer Science*, pages 214-221, 1995.

[7] Leung J.Y.-T., Whitehead J., "On the complexity of fixed-priority scheduling of periodic real-time tasks". *Performance Evaluation*, 2(4), pages 237-250, 1982.

[8] Liu C.L., Layland J.W., "Scheduling algorithms for multiprogramming in a hard-real-time environment", *Journal of the ACM*, 20(1) pages 46-61, 1973.

[9] Thekkilakattil, A.; Dobrin, R.; Punnekkat, S., "Quantifying the Sub-optimality of Non-preemptive Real-Time Scheduling," *25th Euromicro Conference on Real-Time Systems (ECRTS),* pp.113,122, 9-12 July 2013