Global Best-Case Response Time for Improving the Worst-Case Response Times in Distributed Real-Time Systems

Steffen Kollmann, Victor Pollex, Frank Slomka

▶ **To cite this version:**

Steffen Kollmann, Victor Pollex, Frank Slomka. Global Best-Case Response Time for Improving the Worst-Case Response Times in Distributed Real-Time Systems. 7th IFIP TC 10 Working Conference on Distributed, Parallel and Biologically Inspired Systems (DIPES) / 3rd IFIP TC 10 International Conference on Biologically-Inspired Collaborative Computing (BICC) / Held as Part of World Computer Congress (WCC) , Sep 2010, Brisbane, Australia. pp.157-168, 10.1007/978-3-642-15234-4_16 . hal-01054478

HAL Id: hal-01054478

https://inria.hal.science/hal-01054478

Submitted on 7 Aug 2014

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution 4.0 International License

Global Best-Case Response Time for Improving the Worst-Case Response Times in Distributed Real-Time Systems

Steffen Kollmann, Victor Pollex, and Frank Slomka

Ulm University
Institute of Embedded Systems/Real-Time Systems
{firstname.lastname}@uni-ulm.de

Abstract. In this paper an improvement of the schedulability analysis for fixed-priority distributed hard real-time systems is presented. During the analysis it is not sufficient to include the tasks' worst-case execution time, but also the best-case execution time has to be considered, because the lower bound of the execution has a direct impact on the event densities in the system. The presented approach improves the best-case response time analysis introduced by Redell et al. The paper shows how it is possible to calculate a lower bound for the best-case response time using an expressive event model. This new lower bound of the response time will relax the event densities in a distributed system and will therefore lead to more relaxed worst-case response times.

1 Introduction

In our daily life we are surrounded by many different computer systems. Most of them are hidden in a technical context, like an airbag control or an anti-lock system, and are called embedded systems. Some of these systems have to satisfy time constraints. In such cases we talk about embedded real-time systems which means that the correctness of the systems depends on correctly computed values as well as on the time intervals in which these values are computed.

In modern systems several CPUs are connected by several buses. Especially in the automotive industry we have large distributed systems with many different time constraints. During the design process of such systems it is desirable to prove the correctness of time constraints by a schedulability analysis. To achieve realistic results it is necessary to have tight bounds for the minimum and maximum occurrence of events in a system.

For instance, assume a sensor triggered every 5 ms. The sensor has an execution time between 1 ms and 3 ms and triggers a successive task. It is obvious that the trigger of the task depends directly on the execution time of the sensor. In the worst-case two events can occur in a time interval of 3 ms and in best-case in a time interval of 7 ms. Therefore it is not sufficient to include the worst-case execution time of tasks into the analysis, but also the best-case execution time, because the lower bound of the execution has a direct impact on the maximum event densities in the system and thus on the worst-case response times.

This lower bound of execution time can be improved by considering higher priority tasks as shown in [11] where a best-case response time for tasks is introduced and the impact on the worst-case response times is shown. We will show that this best-case response time can be improved when an expressive event model is used.

2 Related Work and Contribution

In order to improve the calculation of the event densities and thereby the worst-case response times in a system it is possible to include the lower bounds of the stimulations into the real-time analysis. Some models considering these lower bounds of event densities are, for example, the periodic task model with jitter based on the busy-window approach [12] or the real-time calculus (RTC) [14].

The latter uses curves describing the arrival of events and the capacities of resources. Based on the network calculus [2] the curves are used to calculate the response times in the system. During the calculation of the outgoing event curves, the RTC considers the lower bounds of the incoming stimulations of the tasks. But the technique used cannot be applied to the busy-window approach.

The busy-window approach is very popular and many research has been done with it like a response-time analysis for Round-Robin [10] or considering offsets between tasks [9]. To calculate a best-case response time of a task was also an aim in the past. Redell et al. show in [11] how the calculation of a best-case response time can be obtained by the periodic model with jitter when lower bounds of stimulations are considered. The SymTA/S approach [12] uses this best-case response time analysis in order to relax the event densities in distributed systems. Palencia and Harbour show in [4] how a lower bound for the best-case response time can be determined. But this bound is not exact. Henderson et al. [5] improved this by a search through all possible orderings of higher priority tasks executions, but according to Redell [11], this solution leads to a numerically intractable search.

Redell's approach [11] is for some cases not exact, because it does not consider the occurrence of each single event exactly. This is founded by the fact that the periodic model with jitter is not expressive enough. For this reason, we use the event stream model from Gresser [3] which allows us to describe a wider range of task's stimulation. First we will exploit the lower bound of the stimulations in order to relax the maximum density of events in a distributed system and calculate the occurrence of each single event as accurately as possible. So we will adapt Redell's approach to the event stream model [3]. We call this approach local-best case response time.

Based on the local-best case response time we will improve the idea by means of a global context of jobs. We use the intervals between successive jobs in order to determine whether more interrupts from higher priority tasks have to be considered. We call this approach global best-case response time.

3 System Model

3.1 Task Model

Γ is the set of tasks on one resource $\Gamma := \{\tau_1, ..., \tau_n\}$. A task $\tau := (c^+, c^-, d, \phi, \Theta^+, \Theta^-)$ consists of c^+ the worst-case execution time, c^- the best-case execution time, d the deadline, ϕ the priority for the scheduling (the lower the number the higher the priority), Θ^+ defines the maximum stimulation (maximum number of events in an interval) and Θ^- the minimum stimulation (minimum number of events in an interval). An interval denotes the length of an interval. Let $\tau_{i,j}$ be the j-th job/execution of task τ_i.

In our model we assume that a task can only generate an event at the end of its execution to notify other tasks. In the following, incoming events are events triggering tasks and outgoing events are events generated by tasks. Furthermore we assume a pre-emptive fixed-priority scheduling.

3.2 Maximum Event Streams

Event streams have been first defined in [3]. The purpose was to give a generalized description for every kind of stimulation. The basic idea is to define an event function $\eta(\Delta t, \Theta^+)$ which can calculate for every interval Δt the maximum amount of events occurring within Δt. In the following, when speaking of intervals we mean the length of the interval. The idea is to describe for each number of events the minimum interval which can include this number of events. Therefore we get an interval for one event two events and so on. The interval for one event is infinitely small and therefore considered to be zero. The result is a sequence of intervals showing a non-decreasing behavior. The reason for this behavior is, that the minimum interval for n events cannot be smaller than the minimum interval for n-1 events since the first interval also includes n-1 events.

Definition 1 ***(Maximum Event Stream*** Θ^+***)*** *A maximum event stream is a set of event stream elements* $\theta : \Theta^+ = \{\theta_1, \theta_2, ..., \theta_n\}$ *and each event stream element* $\theta = (p, a)$ *consists of an offset-interval* a *and a period* p. *The maximum event stream complies the characteristic of sub-additivity.:* $\eta(\Delta t_1 + \Delta t_2, \Theta^+) \leq \eta(\Delta t_1, \Theta^+) + \eta(\Delta t_2, \Theta^+)$.

This means that the maximum number of events of an interval cannot exceed the cumulated maximum number of events of its subintervals.

Each event stream element θ describes a set of intervals $\{a_\theta + k \cdot p_\theta | k \in \mathbb{N}\}$ of the sequence. With an infinite (∞) period it is possible to model irregular behavior. The event function is defined as follows:

Definition 2 ***(Event Function*** $\eta(\Delta t, \Theta)$***)*** *The event function calculates an upper bound of events for a given event stream* Θ *and a given length of the interval* Δt*:*

$$\eta(\Delta t, \Theta) = \sum_{\substack{\theta \in \Theta \\ a_\theta \leq \Delta t}} \left\lceil \frac{\Delta t - a_\theta}{p_\theta} \right\rceil \tag{1}$$

As inverse function we define the interval function which denotes the minimum interval in which a given number of events can occur:

Definition 3 ***(Interval Function*** $\Delta t(n, \Theta)$***)*** *The interval function gives for an event stream* Θ *and a number of* n *events the corresponding minimum interval in which these events can occur:*

$$\Delta t(n, \Theta) = inf\{\Delta t | \eta(\Delta t, \Theta) \geq n\} \tag{2}$$

A detailed definition of the concept and the mathematical foundation of the event streams can be found in [1].

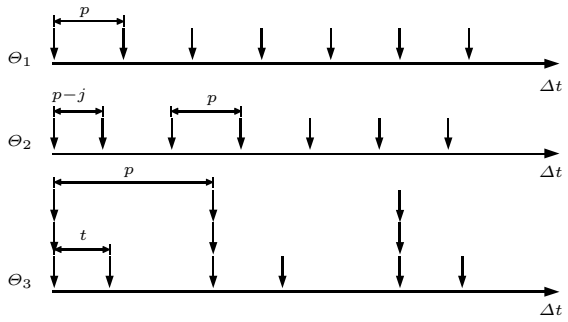


Fig. 1: Three different event streams

Example 1. In figure 1 some examples for event streams can be found. The first one $\Theta_1^+ = \{(\text{p},0)\}$ has a strictly periodic stimulus with a period p. The second example $\Theta_2^+ = \{(\infty,0),\ (\text{p},\text{p-j})\}$ shows a periodic stimulus in which the single events can jitter within a jitter interval of size j. Since we derive the maximum occurrence of the events in an interval the worst-case is the following: The first event is delayed by $j/2$ and the following events are delayed by $-j/2$. Therefore two events can occur in a time interval of $p-j$, three events in $2p-j$ and so on. In the third example $\Theta_3^+ = \{(p,0),(p,0),(p,0),(p,t)\}$ three events occur at the same time and the fourth occurs after a time t. This pattern is repeated with a period of p.

3.3 Minimum Event Streams

Analogously we define the minimum event stream which describes for every interval Δt the minimum stimulation in such an interval.

Definition 4 ***(Minimum Event Stream*** Θ^-***)*** *A minimum event stream is a set of event stream elements* $\theta : \Theta^- = \{\theta_1, \theta_2, ..., \theta_n\}$ *and each event stream element* $\theta = (p, a)$ *consists of an offset-interval* a *and a period* p. *The minimum event stream complies the characteristic of super-additivity:* $\eta(\Delta t_1 + \Delta t_2, \Theta^-) \geq \eta(\Delta t_1, \Theta^-) + \eta(\Delta t_2, \Theta^-)$.

This means that the minimum number of events of an interval can exceed the cumulated minimum number of events of its subintervals.

The event function (1) and the interval function (2) apply also for the minimum event streams.

Example 2. The corresponding minimum event streams for the examples shown in figure 1 can be described as follows: The first one $\Theta_1^- = \{(\text{p,p})\}$. The second example $\Theta_2^- = \{(\text{p,p+j})\}$. In the third example $\Theta_3^- = \{(\text{p,p-t}), (\text{p,p}), (\text{p,p}), (\text{p,p})\}$.

3.4 Real-Time Analysis

Based on previous work we define the real-time analysis with event streams. As described in [13] in each global iteration step of the real-time analysis the worst/best-case response time and the outgoing maximum/minimum stimulation for each task in the system are computed until a fix-point is found. How to perform a real-time analysis with event streams is described in [7]. For the next section we will repeat how the necessary parameters to perform a real-time analysis can be computed for the event stream model.

Worst-Case Response Time The worst-case response time of a task with event streams is bounded by the following equation:

Lemma 1. *The worst-case response time with event streams is calculated by:*

$$r^+(\tau) = \max_{k=1,\ldots,n} \{r^+(k,\tau) - \Delta t(k, \Theta_\tau^+) | r^+(k-1,\tau) > \Delta t(k, \Theta_\tau^+)\} \tag{3}$$

$$r^+(k,\tau) = \begin{cases} c_\tau^+ & k = 0 \\ \min\{\Delta t | \Delta t = k \cdot c_\tau^+ + \sum\limits_{\tau' \in HP} \eta(\Delta t, \Theta_{\tau'}^+) \cdot c_{\tau'}^+\} & k \geq 1 \end{cases} \tag{4}$$

Proof. The proof is given in [8]

Equation (3) determines the maximum of the response times of each job $(r^+(k,\tau) - \Delta t(k, \Theta_\tau^+))$ in the busy window $(r^+(k-1,\tau) > \Delta t(k, \Theta_\tau^+))$. Equation (4) delivers the completion time of each job measured from the critical instance up to its finishing time. Since the calculation of the worst-case response time has not changed but only the model describing it, the proof in [8] is still valid.

Best-Case Response Time Additionally to the worst-case response-time it is possible to determine a best-case response time, since we have minimal event streams. For this we have adapted the best-case response time from Redell [11]:

Lemma 2. *Best-Case Response Time*

$$r^-(\tau) = max\{\Delta t | \Delta t = c_\tau^- + \sum_{\tau' \in HP} \eta(\Delta t, \Theta_{\tau'}^-) \cdot c_{\tau'}^-\} \tag{5}$$

Proof. The proof is given in [11].

Since only the model has changed to calculate the best-case response time and not the calculation itself, the proof in [11] is still valid. The equation adds to the best-case execution time of task τ the best-case execution time of higher priority tasks. How many execution times are added depends on the minimal event streams of the higher priority tasks. It is possible to find the best-case response time as well as the worst-case response time by a fix-point iteration. Since the computation of the best-case response time is done only once for all jobs, we call Redell's approach in conjunction with event streams local best-case response time.

Maximum Outgoing Event Density To derive the outgoing event densities we give the following definition:

Definition 5 *The completion time $r^{\pm}(n,\tau)$ of the n-th job is the interval from the request of the first job up to the point in time where the n-th job has finished its execution. The response time of a job is the completion time minus the request time $\Delta t(n,\Theta_\tau)$.*

Lemma 3. *A number of outgoing events occurs in the maximum density when the first event is delayed as much as possible and all further events occur as early as possible whereas a job can only be executed when the previous jobs have been finished. So the minimum interval between n outgoing events of a task is bounded by:*

$$\Delta t_{min}(n,\tau) = r^{\pm}(n,\tau) - r^{+}(\tau) \tag{6}$$

$$r^{\pm}(n,\tau) = \begin{cases} r^{+}(\tau) & n = 1 \\ max(\Delta t(n,\Theta_\tau^{+}), r^{\pm}(n-1,\tau)) + r^{-}(\tau) & n > 1 \end{cases} \tag{7}$$

Proof. The proof is given in [7].

Minimum Outgoing Event Density

Lemma 4. *A number of outgoing events occurs in the minimum density when the first event occurs as early as possible and all further events occur as late as possible. So the maximum interval between n outgoing events of a task is bounded by:*

$$\Delta t_{max}(n,\tau) = \Delta t(n,\Theta_\tau^{-}) + (r^{+}(\tau) - r^{-}(\tau)) \tag{8}$$

Proof. The proof is analogous to the proof of the maximum event density.

To calculate the outgoing event streams concretely, see [6] where a normalization for event streams is proposed.

4 Improved Maximum Event Density

Up to this point we have shown how to adapt the best-case response time analysis of Redell et al. [11] to the event stream model and how to conduct a real-time analysis for distributed systems. We will now introduce a methodology in order to calculate the best-case response times of each job in a global context and show how this improves the outgoing maximum event densities of tasks. Global context means here, that the order of the job execution and the time of incoming events are considered.

The idea of the approach is that between successive jobs more interference from higher priority tasks can occur than the local best-case response time ascertains. So we determine whether higher priority tasks produce more load than the interval between n outgoing events can provide. In case that the load is greater than the interval, we are able to relax the best-case response time of a job $r^-(n,\tau)$ and improve $\Delta t_{min}(n,\tau)$.

With figure 2 and equation (9), (10) and (11) we develop our new methodology.

Lemma 5. *A number of outgoing events occurs in the maximum density when the first event is delayed as much as possible and all further events occur as early as possible whereas a job can only be executed when the previous jobs have been finished and the best-case response time of the n-th job exploits the intervals between successive jobs. So the minimum interval between n outgoing events of a task is bounded by:*

$$\Delta t_{min}(n,\tau) = r^{\pm}(n,\tau) - r^{+}(\tau) \tag{9}$$

$$r^{\pm}(n,\tau) = \begin{cases} r^{+}(\tau) & n = 1 \\ \max\limits_{l \in \mathbb{N}_0}\{r^{\pm}(n,\tau,l)\} + r^{+}(\tau) & n > 1 \end{cases} \tag{10}$$

$$r^{\pm}(n,\tau,l) = \begin{cases} max(\Delta t(n,\Theta^{+}_{\tau}), r^{\pm}(n-1,\tau)) + r^{-}(\tau) - r^{+}(\tau) & l = 0 \\ (n-1)\cdot c^{-}_{\tau} + \sum\limits_{\tau' \in HP(\tau)} \eta(r^{\pm}(n,\tau,l-1) + c^{+}_{\tau'}, \Theta^{-}_{\tau'}) \cdot c^{-}_{\tau'} & l > 0 \end{cases} \tag{11}$$

Proof. We have to show that there exists no interval smaller than $\Delta t_{min}(n,\tau)$

Case 1 ($n = 1$): According to the lemma the first event $n = 1$ is delayed maximal. This is the worst-case response time $r^+(\tau)$ by definition and therefore the interval for one event is zero.

Case 2: ($n > 1$): For this case we have to show that the two cases in equation (11) are bounds for the completion times. The first case $l = 0$ is the lower bound shown in lemma 3. This can be obtained by inserting this case in equation (10) and we get:

$$\begin{aligned} &max(\Delta t(n,\Theta^{+}_{\tau}), r^{\pm}(n-1,\tau)) + r^{-}(\tau) - r^{+}(\tau) + r^{+}(\tau) \\ &\quad = max(\Delta t(n,\Theta^{+}_{\tau}), r^{\pm}(n-1,\tau)) + r^{-}(\tau) \end{aligned}$$

Second case of equation 11 assumes an interval Δt which is the earliest completion time that fulfills the lemma. So we get: $\Delta t = r^+(\tau) + \Delta t'$. If the condition

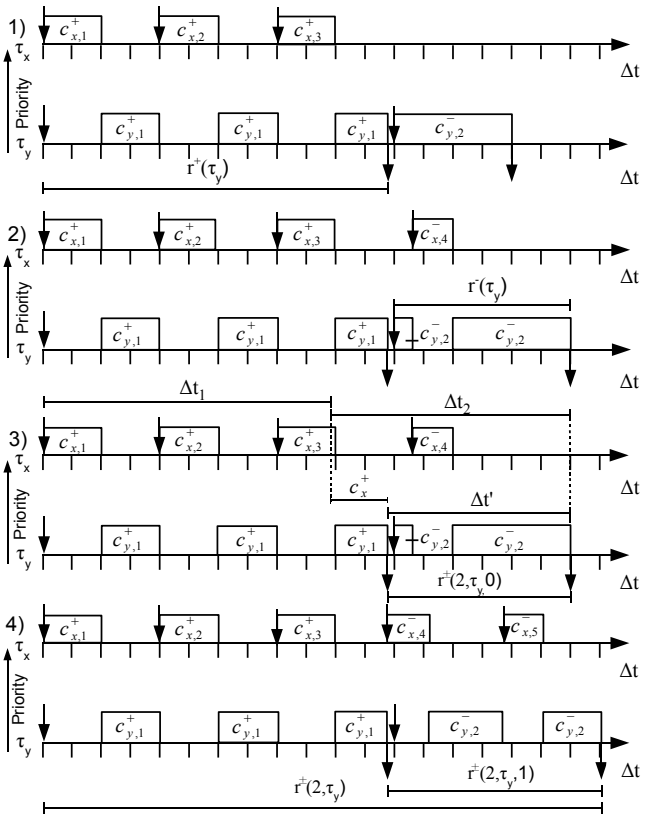


Fig. 2: BCRT of the second job

$r^{\pm}(n,\tau,l+1) > r^{\pm}(n,\tau,l)$ holds, the processor is always busy in $\Delta t'$ and we get:

$$\Delta t = r^{+}(\tau) + (n-1)\cdot c_{\tau}^{-} + \sum_{\tau' \in \Gamma_{HP}} m_{\tau'} \cdot c_{\tau'}^{-} \tag{12}$$

The number of events occurring in Δt are divided in the interval Δt_1 which considers all the occurrences of the events during the worst-case response time $\eta(\Delta t_1, \Theta_{\tau'}^{+})$ and the rest interval Δt_2 considering the minimal occurrence of events from a task $\eta(\Delta t_2, \Theta_{\tau'}^{-})$. The last possible occurrence of an event from a higher priority task during the worst-case response time is $r^{+}(\tau) - c_{\tau'}^{+}$. So it follows:

$r^{+}(\tau) - c_{\tau'}^{+} + \Delta t_2 = \Delta t \Leftrightarrow \Delta t_2 = \Delta t - r^{+}(\tau) + c_{\tau'}^{+} \Leftrightarrow \Delta t_2 = \Delta t' + c_{\tau'}^{+}$

This interval can be inserted into equation (12):

$$\Delta t = r^{+}(\tau) + (n-1)\cdot c_{\tau}^{-} + \sum_{\tau' \in \Gamma_{HP}} \eta(\Delta t' + c_{\tau'}^{+}, \Theta_{\tau'}^{-}) \cdot c_{\tau'}^{-}$$

The maximum interval Δt which fulfills $r^{\pm}(n, \tau, l+1) > r^{\pm}(n, \tau, l)$ can be found by a fix-point iteration and we get:

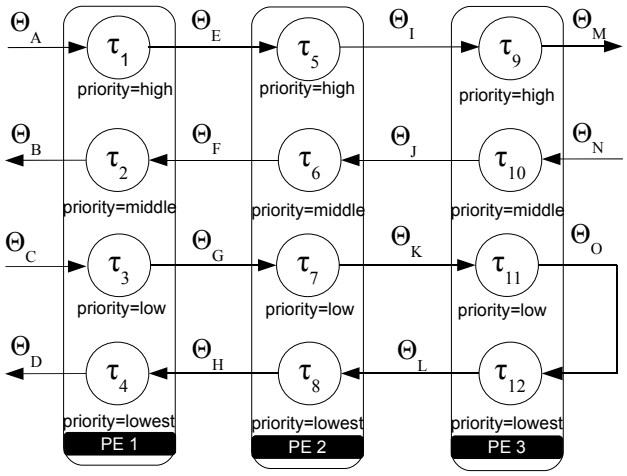
$$r^{+}(\tau) + (n-1) \cdot c_{\tau}^{-} + \sum_{\tau' \in \Gamma_{HP}} \eta(r^{\pm}(n, \tau, l) + c_{\tau'}^{+}, \Theta_{\tau'}^{-}) \cdot c_{\tau'}^{-}$$

So the minimal interval between n outgoing events can be calculated by lemma 5: $\Delta t_{min}(n, \tau) = r^{\pm}(n, \tau) - r^{+}(\tau)$. □

In figure 2 it is exemplarily shown how the approach works. Part one shows the event density if only the best-case execution time is considered. The second part shows the idea from Redell which is the initial value for our approach. Part three and four of figure 2 depicts the new developed fixed-point iteration. It can be seen that the load produced in interval Δt_2 is greater than the interval itself and therefore the two outgoing events have a relaxed event density.

5 Experiments and Results

To consider the improvement of the new algorithm we have used the synthetical distributed system depicted in figure 3. Due to the reversed paths we have chosen this example. The distributed system has three processing elements. All the tasks are scheduled by a fixed-priority schedule. The system has been evaluated with different utilization. The event density of the three inputs Θ_A, Θ_C, Θ_N has been



τ	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9	τ_{10}	τ_{11}	τ_{12}
$c^{-}[ms]$	200	20	1500	300	50	8	1000	500	50	4	900	1000
$c^{+}[ms]$	200	20	1500	300	50	16	1000	1000	150	4	1200	1100

Fig. 3: Distributed System

varied to achieve this. In order to vary the input stimulation only the period and the jitter have been modified and mapped to the different event models. The jitter was up to five times the period. For each utilization step the average of 100 variations has been taken. The assumed execution demands are described in the table of figure 3.

Figure 4 depicts the average utilization of the system versus the cumulated average worst-case response time in the system. In this figure the absolute improvement of the global best-case response time versus the local best-case response time, Redell's [11] approach and the SymTA/S approach [12] can be observed. The SymTA/S approach extends Redell's methodology by a minimal distance between events. We have implemented all techniques in one framework. The SymTA/S approach is implemented as described in [12]. The improvement is especially huge for high utilization. When the utilization is low (50%-70%) it is improbable that the execution of the tasks are interrupted very often by higher priority tasks. Therefore the improvement in this range is smaller. Between the local best-case response time and global best-case response time we have also no significant improvement for high utilization (95%-99%). This is founded by the fact that we have a very high utilization and the gaps for the possible improvements are very small, because the utilization is near 100%. So in this case the local best-case response time and the global best-case response time converge. In the range (70%-95%) where the utilization is high enough for many interrupts from higher priority tasks and when enough gaps are available between successive jobs, we have good improvements concerning the worst-case response times. The relative improvement in percent is also depicted in figure 5 (left) and underlines the states above.

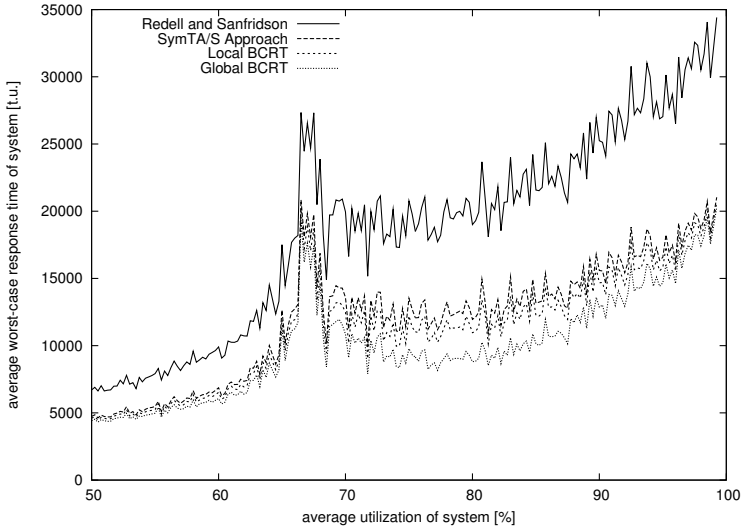


Fig. 4: Utilization vs. WCRT

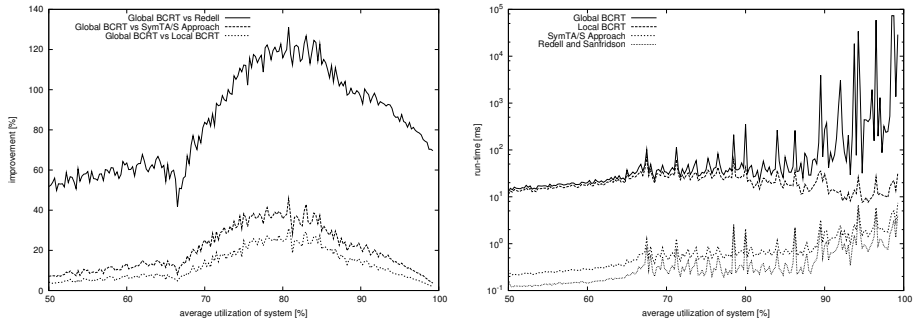


Fig. 5: Relative Improvement of the WCRT (left), Utilization vs. Runtime (right)

Figure 5 (right) gives an overview about the runtime of the implemented approaches. It is obvious that the global best-case response time approach is slower than the local best-case response time calculation. This is founded by the fact, that we have to calculate the best-case response time for each job. It is also obvious that Redell's approach and the SymTA/S approach are faster than the approaches with event streams, because not every event is calculated exactly.

Sometimes we have outliers in the runtime of the global best-case response time but not a significant improvement. In these cases we have to calculate many global best-case response times, but the effect on the worst-case response times is marginal. This occurs when the improvement over many instances is small and the relaxation has no influence on the interruption for the worst-case response times.

The experiments show that we are able to perform a real-time analysis with the global best-case response times and get tighter results than Redell's best-case response time analysis. We get up to 24% of improvement of the average worst-case response time in the system versus the local best-case response time, 41% versus the SymTA/S approach and up to 135% versus Redell's best-case response time analysis. The runtime of the global best-case response times is for lower utilization almost identical to the local best-case response time analysis. Only where the improvement is high we have higher runtime.

6 Conclusion

In this paper we have shown how to use lower bounds of stimulation in order to improve the real-time analysis of distributed systems. Two contributions are presented in this paper. The first one is the adaption of Redell's methodology [11] of the best-case response time analysis to the event stream model called local best-case response time. This technique has been improved using the intervals between successive jobs in order to determine the interrupts from higher priority task in a global context leading to best-case response times on job-level. Furthermore we have shown that this leads directly to more realistic response

times of the system. Since we have only considered fixed priority pre-emptive scheduling, it would be interested in the future to consider this approach for other scheduling policies like round-robin.

References

1. Albers, K., Slomka, F.: An event stream driven approximation for the analysis of real- time systems. In: ECRTS '04: Proceedings of the 16th Euromicro Conference on Real-Time Systems. pp. 187–195. IEEE (July 2004)
2. Boudec, J.Y.L., Thiran, P.: Network calculus: a theory of deterministic queuing systems for the internet. Springer-Verlag New York, Inc., New York, NY, USA (2001)
3. Gresser, K.: An event model for deadline verification of hard real-time systems. In: Proceedings of the 5th Euromicro Workshop on Real-Time Systems. pp. 118–123 (1993)
4. Gutirrez, J., Garca, J., Harbour, M.: Best-case analysis for improving the worst-case schedulability test for distributed hard real-time systems. In: Proceedings of the 10th Euromicro Workshop on Real-Time Systems. pp. 35–44. IEEE Computer Society (1998)
5. Henderson, W., Kendall, D., Robson, A.: Improving the accuracy of scheduling analysis applied to distributed systems computing minimal response times and reducing jitter. Real-Time Syst. 20(1), 5–25 (2001)
6. Kollmann, S., Albers, K., Slomka, F.: Effects of simultaneous stimulation on the event stream densities of fixed-priority systems. In: Spects'08: Proceedings of the International Simulation Multi-Conference. pp. 353–360. IEEE (June 2008)
7. Kollmann, S., Pollex, V., Slomka, F.: Holisitc real-time analysis with an expressive event model. In: proceedings of the 13th Workshop of Methoden und Beschreibungssprachen zur Modellierung und Verifikation von Schaltungen und Systemen (2010)
8. Lehoczky, J.P.: Fixed priority scheduling of periodic task sets with arbitrary deadlines. In: Proceedings of the 11th IEEE Real-Time Systems Symposium. pp. 201–209 (December 1990)
9. Palencia, J.C., Harbour, M.G.: Schedulability analysis for tasks with static and dynamic offsets. In: RTSS. p. 26 ff (1998)
10. Racu, R., Li, L., Henia, R., Hamann, A., Ernst, R.: Improved response time analysis of tasks scheduled under preemptive round-robin. In: CODES+ISSS '07: Proceedings of the 5th IEEE/ACM international conference on Hardware/software codesign and system synthesis. pp. 179–184. ACM, New York, NY, USA (2007)
11. Redell, O., Sanfridson, M.: Exact best-case response time analysis of fixed priority scheduled tasks. In: ECRTS '02: Proceedings of the 14th Euromicro Conference on Real-Time Systems. p. 165. IEEE Computer Society, Washington, DC, USA (2002)
12. Richter, K.: Compositional Scheduling Analysis Using Standard Event Models - The SymTA/S Approach. Ph.D. thesis, University of Braunschweig (2005)
13. Tindell, K., Clark, J.: Holistic schedulability analysis for distributed hard real-time systems. Microprocessing and Microprogramming 40, 117–134 (April 1994)
14. Wandeler, E.: Modular Performance Analysis and Interface-Based Design for Embedded Real-Time Systems. Ph.D. thesis, ETH Zurich (September 2006)