On Using Extreme Value Theory in Response-Time Analysis of Priority-Driven Periodic Real-Time Systems

Yue Lu, Thomas Nolte
Mälardalen Real-Time Research Centre
Mälardalen University, Västerås, Sweden
{yue.lu, thomas.nolte}@mdh.se

Abstract

In this paper we present work toward using our previously proposed method RapidRT to perform response-time analysis of periodic real-time systems, where the execution time of the adhering tasks is a random variable from a known distribution. In effect, we not only aim at validating the potential of considering the results given by RapidRT as upper bounds on tasks' worst-case response time estimates, but also investigate the possibility of using RapidRT as a good substitute for the referenced exact stochastic analysis method which is generally intractable for large systems.

1 Introduction

Traditional schedulability analysis methods for hard real-time systems are often based on a periodic task model, where a simplifying assumption on the existence of tasks' single value Worst-Case Execution Time (WCET) is made in order to provide a deterministic and stringent guarantee that all the tasks in the system can meet their deadlines. If deadlines of all tasks are met, then the system is deemed schedulable. However, in the context of timing analysis for soft real-time systems where a failure in meeting timing requirements will not result in the failure of systems that potentially leads to catastrophic human consequences, such a stringent guarantee is not required. Moreover, it is often better off providing a probabilistic guarantee that the deadline miss ratio of a task is below a certain threshold. Consequently, the assumption on tasks' WCET has to be relaxed.

A stochastic analysis framework is presented in [1], which does not introduce any worst-case or restrictive assumptions into the analysis, and is applicable to general priority-based real-time systems including both fixed-priority scheduling systems and dynamic-priority scheduling systems. The analysis method can handle any periodic task set consisting of tasks, each of which the execution time of the adhering jobs is specified as a random variable with a known discrete distribution. Furthermore, the analysis can give the exact *Probability Mass Function* (PMF) of response time of the tasks in the system, and a probability of deadline miss of tasks.

In [2], we present the timing analysis method namely *RapidRT*, which combines Extreme Value Theory (EVT) [3] with other statistical methods in order to produce a Worst-Case Response Time (WCRT) estimate of tasks, under a certain statistic constraint, i.e., a certain probability of being exceeded. Moreover, RapidRT performs WCRT analysis of the target system based on a number of traces[1] containing response time data of tasks. In this work, we are interested in validating RapidRT, by examining if the results given by RapidRT can be considered as safe upper bounds on the WCRT estimates of tasks in the context of priority-driven periodic real-time systems. In such systems, exact WCRT distributions can be obtained through using techniques such as the stochastic analysis framework in [1]. In addition, such work is meaningful and valuable in the sense that it can also help us to validate if RapidRT is a suitable alternative for the referenced stochastic analysis framework, which by its authors is argued intractable for large systems.

2 System Model and Notations

The system model S consists of a set of n independent periodic tasks running on a uniprocessor, i.e., $S \leftarrow \tau_1, ..., \tau_n$, where $n \in \mathbb{N}$. A task τ_i is characterized by a set of parameters $\langle T_i, \Phi_i, e_i, D_i, M_i \rangle$, where T_i is the task period, Φ_i is the initial phase, e_i is execution time, D_i is the relative deadline or the temporal constraint, M_i is the maximum allowed probability of missing the deadline. The execution time e_i is a discrete random variable with a known distribution. The PMF is denoted by $f_{e_i}(\cdot)$, where $f_{e_i}(e) = P\{e_i = e\}$. Each periodic task results in an in-

[1] Such traces are either from simulation models or from the execution of the real system.

finite number of jobs. $\Gamma_{i,j}$ denotes the *j*th job of the task τ_i. Each job $\Gamma_{i,j}$ is released at the deterministic time $\lambda_{i,j}$, which is computed via Equation 1.

$$\lambda_{i,j} = \phi_i + (j-1)T_i \tag{1}$$

The response time of a job $\Gamma_{i,j}$ is a discrete random variable denoted by $R_{i,j}$, and the response time of the task τ_i, i.e., R_i is computed by averaging the response time of its jobs as shown in Equation 2:

$$f_{R_i}(r) = \frac{1}{m_i}\sum_{j=1}^{m_i} f_{R_{i,j}}(r) \tag{2}$$

where $m_i = T/T_i$, which is the number of jobs from the task τ_i released in a hyper-period of the length T. In addition, a task τ_i is said to be schedulable if $P\{R_i > D_i\} \leq M_i$.

3 The Stochastic Analysis Framework

The referenced stochastic analysis framework in this work, is proposed in [1], which will be introduced briefly as follows. For the sake of simplicity, the task to which a job belongs is not tracked, thus a job has a single index, e.g., Γ_j. The index of a job refers to its order in the infinite sequence of jobs, i.e., Γ_k is released before Γ_{k+1}, that is $\forall k$, $\lambda_k \leq \lambda_{k+1}$. The response time of a job Γ_j is computed by using Equation 3:

$$R_j = W(\lambda_j) + e_j + J_j \tag{3}$$

where R_j denotes the response time distribution of an arbitrary job Γ_j; $W(\lambda_j)$ denotes the backlog at time λ_j, i.e., the sum of the remaining execution times of all the jobs (with higher priorities than the job under analysis) that do not finish up to the time λ_j; J_j denotes the interference of all higher priority jobs released after the job Γ_j.

The backlog at the release time of any job Γ_j, denoted by W_{λ_j}, can be computed by following the iterative procedure introduced in [1]:

$$W(\lambda_{k_0}) = 0 \tag{4}$$

$$W(\lambda_k) = shrink(W(\lambda_{k-1}) + e_{k-1}, \lambda_k - \lambda_{k-1}) \tag{5}$$

where λ_{k_0} denotes the release time of the first job released before Γ_j and has a higher priority. The *shrink function* is shown by Equation 6:

$$f_{shrink}(W,\Delta)(x) = \begin{cases} 0 & \text{if } x < 0, \\ \sum_{z=-\infty}^{0} f_W(z+\Delta) & \text{if } x = 0, \\ f_W(x+\Delta) & \text{if } x > 0. \end{cases} \tag{6}$$

The iterative procedure starts with a zero backlog as shown by Equation 4 and iterates on all higher priority jobs released before Γ_j. After computing the backlog at the release time of Γ_j, the backlog distribution is convolved with the execution time distribution. Such convolution results in a partial response time which is valid only if there is no interference given by other higher priority jobs. In case of the existence of higher priority jobs which are released after λ_j, this partial response time will be valid only in the range from λ_j to λ_{j+1}. A validity range is indicated as a super index for the response time $R^{[0,\lambda_{j+1}-\lambda_j]}$, which is computed by Equation 7:

$$R^{[0,\lambda_{j+1}-\lambda_j]} = W(\lambda_j) + e_j \tag{7}$$

In addition, Equation 8 can be used to increase the range of the validity.

$$R^{[0,\lambda_{k+1}-\lambda_j]} = AF(R^{[0,\lambda_k-\lambda_j]}, \lambda_k - \lambda_j, e_k), k > j \tag{8}$$

where the job Γ_k has a higher priority than Γ_j, and AF is the stochastic function shown by Equation 9:

$$f_{AF(R,\Delta,e)}(x) = \begin{cases} f_R(x) & \text{if } x \leq \Delta, \\ \sum_{i=\Delta+1}^{\infty} f_R(i)\cdot f_e(x-i) & \text{if } x > \Delta. \end{cases} \tag{9}$$

Each iteration using Equation 8 increases the interval of the validity of the partial response time, until the deadline is included in the validity range. Correspondingly, the probability of missing the deadline for a certain job can be computed by using Equation 10:

$$P(R_j > D_j) = 1 - \sum_{k=0}^{D_j} P(R_j^{[0,\Delta]} = k) \tag{10}$$

After completing the analysis, the probability of missing the deadline of a certain task is computed by averaging the probabilities of missing the deadlines of all its jobs, as shown by Equation 2.

4 RapidRT Using Extreme Value Theory

Our proposed method RapidRT is based on Extreme Value Thoery (EVT) [3], which is a separate branch of statistics for dealing with the tail behavior of a distribution. EVT is used to model the risk of the extreme, rare events, without the vast amount of sample data required by a brute-force approach. Example applications of EVT include risk management, insurance, telecommunications and so on.

RapidRT is a recursive procedure which, as the first two arguments, takes n reference data sets each of which contains m simulation traces containing tasks' response times. For each reference data set, the algorithm returns the WCRT

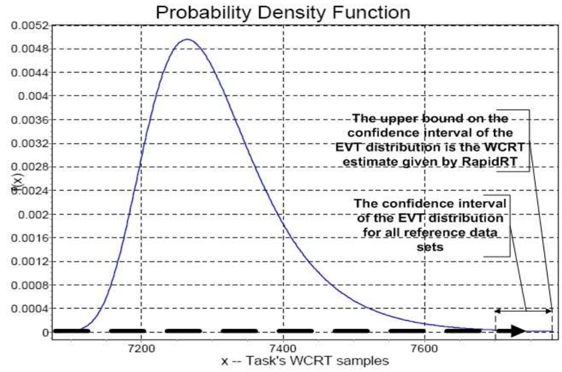

Figure 1. The point, at which the bold dash line intersects with the Gumbel Max curve, is the WCRT estimate given by RapidRT for each reference data set. The EVT distribution is constructed on these points for all reference data sets.

estimate of the task under analysis with a probability of being exceeded, e.g., 10^{-9}, which is the third algorithm argument. Next, RapidRT will verify if the sampling distribution consisting of n WCRT estimates given by EVT for all n reference data sets (we refer to such a sampling distribution as the EVT distribution hereafter) conforms to a normal distribution or not, according to the result given by the non-parametric Kolmogorov-Smirnov test (the KS test hereafter). If it is, then RapidRT will calculate the confidence interval (i.e., CI hereafter) of the EVT distribution, at the given confidence level 99.7%, and choose the upper bound on the CI as the final WCRT estimate, as shown in Figure 1. This invents a new hard statistic constraint, i.e., from the statistical perspective, given the modeled system, the possibility of the existence of a higher WCRT estimate (i.e., the actual WCRT of the task on focus) than the WCRT estimate given by RapidRT is no more than 1.5×10^{-12} (i.e., $(100\% - 99.7\%)/2 \times 10^{-9}$). Otherwise, if the EVT distribution cannot be fitted to a normal distribution, a *re-sampling* statistic *bootstrap* will be adopted to obtain the upper bound on the CI of the EVT distribution.

RapidRT consists of the following three steps: 1) construction of the referenced data sets, 2) WCRT estimation of each referenced data set using EVT, and 3) derivation of a final WCRT estimate that is given by the algorithm. For more details and thorough explanations about each step in RapidRT, the interested readers can refer to [2]. In addition, the outline of the algorithm is as follows:

1. Construct n reference data sets for the WCRT estimates by running m Monte Carlo simulations for each reference data at first, and then choosing the highest response time value of the task under analysis in each simulation. Consequently, the sampling distribution of Response Time (RT) data per reference data set consists of the m highest RT data collected from the m simulations, respectively.
2. Perform the WCRT estimates on the task under analysis per each reference data set, i.e., est_i where $1 \leq i \leq n$.
3. After verifying if the EVT distribution (i.e., $EST \leftarrow est_1, ..., est_i, ..., est_n$) can successfully be fitted to a normal distribution by using the KS test, RapidRT will return a result, i.e., $\overline{EST} + 3\sigma_{EST}$ (the sum of the *mean* value and 3 *standard deviation* of EST at the confidence level 99.7%). Otherwise, the bootstrap test will be used in the context.

5 Evaluation

The target priority-based periodic real-time systems (introduced in Section 2) will be modeled and analyzed by using our RTSSim simulation framework [4]. RTSSim is quite similar to *ARTISST* [5] and *VirtualTime* [6], and allows for simulating job-level system models on a single processor. Further, RTSSim provides typical RTOS services to its simulation model, such as Fixed-Priority Preemptive Scheduling (FPPS), intricate task execution dependencies on job-level including Inter-Process Communication (IPC) via message queues and synchronization (semaphores). The execution time of jobs can be modeled as a random variable with a specific type of distribution. All time-related operations in RTSSim, such as timeouts and activation of time-triggered tasks, are driven by the simulation clock, which makes the simulation result independent of process scheduling and performance of the analysis PC. The response time and execution time of tasks or jobs are measured whenever the scheduler is invoked, which happens for example at IPC, task or job switches, *EXECUTE* statements, operations on semaphores, task or job activations and when tasks or jobs end. This, together with the simulation clock behavior, guarantees that the measured response time and execution time are exact. In RTSSim, a task may not be released for execution until a certain non-negative time (i.e., the offset) has elapsed after the arrival of the activating event. Each task also has a period, a maximum arrival jitter, and a priority. Tasks with equal priorities are served on the first come first serve basis.

In addition, we will propose a number of evaluation frameworks from the following perspectives:

1. **Different statistical constraints in RapidRT:** In our evaluation, the probabilities of being exceeded in RapidRT can be set either low or high, when compared to what we used in previous research, i.e., 10^{-9}. For example, such probabilities can be $10^{-3}, 10^{-6}, 10^{-12}, 10^{-20}$ etc. The intention is to evaluate that if the results given by RapidRT can successfully cover the exact value of WCRT of tasks,

when different levels of statistical constraints are applied. We intend to compare against exact data provided by the referenced stochastic analysis framework, using the same system model when applying both approaches.

2. **Different confidence levels of the EVT distribution:** We also consider to use different confidence levels in the EVT distribution in RapidRT, such as 95%, which is a typical value that based on preliminary assessments provides appropriate results.
3. **Scalability of RapidRT:** This can be done by creating independent "subsystems" where each subsystem represents a complete model, i.e., a priority-driven periodic real-time system (as introduced in Section 2). More details about using "subsystems" for scalability evaluation can be found in [7].
4. **Optimization on the number of samples in RapidRT:** The KS test will be used in this context with the purpose of optimizing the number of samples in RapidRT, while keeping the accuracy of results. This will reduce the computation time required by RapidRT, which is especially meaningful and necessary for the cases about timing analysis of large systems.

6 Related Work

As introduced in [1], the exact stochastic analysis of most real-time systems under preemptive priority driven scheduling is not affordable in practice currently. Some approaches about performing stochastic analysis with a specific scheduling model that isolates tasks so that each task can be analyzed independently are proposed [8, 9]. In addition, in order to simply the stochastic analysis in such context, the worst-case assumptions are introduced. Manolache presents the way of restricting tasks preemption, and some others [10, 11] introduce the assumption on the critical instance. Díaz et al. [12] further their previous study by introducing an approximate analysis, in order to decrease the memory demand on the computation of backlog and response time distributions. Recently, Refaat [13] proposes a method for efficient stochastic analysis by simplifying the exact distributions of jobs through random sampling.

7 Future Work

This work-in-progress paper has presented ongoing work on using our previously proposed method RapidRT in response-time analysis of priority-driven periodic real-time systems. We are also interested in using RapidRT as a good substitute for the referenced exact stochastic analysis method which is generally intractable for large systems. In particular, we have expressed the idea of comparing the results given by RapidRT to the ones obtained through the referenced stochastic analysis framework, which provides us with exact PMFs for task response times. Future work will mainly lie in implementation and evaluation.

Acknowledgment

This work was partially supported by the Swedish Foundation for Strategic Research (SSF) and the Swedish Research Council (VR).

References

[1] J. M. López, J. L. Díaz, J. Entrialgo, and D. García, "Stochastic analysis of real-time systems under preemptive priority-driven scheduling," *Real-Time Syst.*, vol. 40, no. 2, pp. 180–207, 2008.

[2] Y. Lu, T. Nolte, J. Kraft, and C. Norström, "A Statistical Approach to Response-Time Analysis of Complex Embedded Real-Time Systems," in *RTCSA' 10*, Aug 2010, pp. 153–160.

[3] J. Beirlant, Y. Goegebeur, J. Segers, and J. Teugels, *Statistics of Extremes: Theory and Applications*. Wiley Press, 2004.

[4] J. Kraft, "RTSSim – A Simulation Framework for Complex Embedded Systems," Mälardalen University, Tech. Report., Mar 2009.

[5] D. Decotigny and I. Puaut, "ARTISST: An Extensible and Modular Simulation Tool for Real-Time Systems," in *ISORC' 02*, 2002, pp. 365–372.

[6] "Website of Rapita systems," 2008. [Online]. Available: www.rapitasystems.com

[7] M. Bohlin, Y. Lu, J. Kraft, P. Kreuger, and T. Nolte, "Simulation-Based Timing Analysis of Complex Real-Time Systems," in *RTCSA' 09*, Aug 2009, pp. 321–328.

[8] A. Atlas and A. Bestavros, "Statistical rate monotonic scheduling," in *RTSS' 98*, 1998, pp. 123–132.

[9] G. A. Kaczynski, L. Lo Bello, and T. Nolte, "Deriving Exact Stochastic Response Times of Periodic Tasks in Hybrid Priority-driven Soft Real-time Systems," in *ETFA' 07*, Sep 2007, pp. 101–110.

[10] T.-S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.-C. Wu, and J. W.-S. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times," in *RTAS' 95*, 1995, pp. 164–173.

[11] M. K. Gardner and J. W.-S. Liu, "Analyzing stochastic fixed-priority real-time systems," in *TACAS' 99*, 1999, pp. 44–58.

[12] J. L. Diaz, J. M. Lopez, M. Garcia, A. M. Campos, K. Kim, and L. Lo Bello, "Pessimism in the stochastic analysis of real-time systems: Concept and applications," in *RTSS' 04*, 2004, pp. 197–207.

[13] K. S. Refaat and P.-E. Hladik, "Efficient stochastic analysis of real-time systems via random sampling," in *ECRTS' 10*, 2010, pp. 175–183.