Approximate Timing Analysis of Complex Legacy Real-Time Systems using Simulation Optimization

Yue Lu, Johan Kraft, Thomas Nolte and Christer Norström
Mälardalen Real-Time Research Centre
Mälardalen University, Västerås, Sweden
{yue.lu, johan.kraft, thomas.nolte, christer.norstrom}@mdh.se

Markus Bohlin and Per Kreuger
Swedish Institute of Computer Science
Kista, Sweden
{markus.bohlin, piak}@sics.se

***Abstract*—In this paper, we present an approach towards guiding the probabilistic simulation of the model by using a meta-heuristic algorithm in order to find extreme response times more efficiently than by traditional simulation. The proposed approach should scale to industrial-size complex systems and should be regarded as complementary to testing, since it does not guarantee that the true worst case response time will be identified.**

I. Introduction

Most existing embedded real-time software systems today have been developed in a traditional code-oriented manner. When such systems grow over time, due to new features and other changes, they become large and complex. We refer to such systems as legacy systems.

The existing timing analysis proposed so far can be broadly divided into two main classes of analytic techniques and simulation based methods [1]. Classical response time analysis, for instance Fixed-Priority Analysis (FPA [2]), often gives too pessimistic results for many real industrial systems. The temporal model grounded on task level abstraction assumed by the method cannot capture the complex dependencies between tasks that exist in many legacy systems. Moreover, FPA assumes a single WCET per task. Whereas a task may have different WCETs in different situations (different states). On top of this, although more optimistic (but not safe) WCET of tasks can be obtained for instance by using hybrid WCET analysis in [3], the exact WCET of tasks is often practically impossible to find due to high complexity. Similar problems arise for compositional analysis, e.g. Sym TA/S [4] and Real-Time Calculus [5].

To use a more detailed system model is another approach, where the model describes the tasks' behavior with respect to inter-process communication, usage of CPU time and usage of logical resources. Further, abstractions are necessary since not all the aspects of the system can be taken into account. The relevant work about extracting such model from the industrial system, is proceeding in our parallel research, and will be briefly introduced in Section II.

Without a very carefully chosen level of abstraction, the state space of such models are too large for practical exhaustive analysis, e.g. UPPAAL [6], a timed automata (TA) based analysis. Another approach is simulation based analysis, for example probabilistic discrete event simulation. Simulation based analysis methods have large modeling scope and are widely used in industry. Moreover, they are less dependent on the size of the system state space since only a subset of the state space is explored.

The obvious pitfall of simulation based analysis is insufficient test case coverage, which makes it impossible to give any guarantees regarding the behavior of the simulated model, e.g. the WCRT of a particular task. Several frameworks exist for probabilistic simulation of real-time system models, e.g. the commercial tool *VirtualTime* [3] and the academic tool *ARTISST* [7].

Our previous work [8] presented a promising meta-heuristic approach called MABERA, where a probabilistic discrete event simulator is guided by a meta-heuristic search algorithm in order to find extreme response times. We showed in [8] that MABERA can find large response times in a more efficient manner compared to traditional probabilistic simulation. However, MABERA is far from optimal, as not only it did not find the true WCRT of the task in focus, but also it obtained very unstable results and required many replications, large number of simulations and long run time to achieve the best results. The reason for this is a too abstract representation of simulation state in the simulation optimization algorithm. We therefore propose a new approach which addresses this issue and present preliminary results from a prototype implementation in this paper. In addition, we intend to compare the results of our new approach with classical response time analysis for Fixed-Priority Preemptive Scheduling (FPPS), Monte Carlo simulation and the earlier MABERA approach [8] with respect to a model of a real industrial legacy system, a control system for industrial robots developed by our industrial partner ABB.

II. Simulation Model Extraction

Approximate timing analysis based on timing-accurate simulation requires an analyzable model of the system that describes both functional and temporal behavior of the individual tasks on a proper level of abstraction. The model should focus on task behavior which has a significant impact on the task scheduling, communication or allocation of limited logical resources.

The extracted simulation model targets our system level simulator, *RTSSim*. The core of RTSSim is a C library, which allows C programs (the task models) to be executed in an isolated "sandbox", where time is represented in a discrete manner, using a simulation clock (an integer counter) which is only advanced explicitly. RTSSim provides typical RTOS services to the task models, such as task scheduling (e.g. FPPS), inter-process communication and synchronization (semaphores), all with respect to where time-related events (such as timeouts) are driven by simulation clock. Thus, the simulation result is not affected by other programs running on the same PC. RTSSim provides two modeling primitives which are not found in typical operating systems:

- The *execute* function, which advances the simulation clock and thereby models the consumption of CPU time.
- The *P* function, which is used for probabilistic modeling of selections.

The data for these modeling primitives, i.e. how much CPU time to consume and how often to select a certain branch is obtained from measurements on the runtime system. The commercial simulator "Virtual Time", from Rapita Systems [3], uses a very similar approach.

In [9], we proposed a well-defined, system-level modeling process which utilizes two complementary methods for the task modeling; *Model Synthesis*, a fully automated method complemented by a novel method which we refer to as *hybrid model extraction*. Moreover, in order to support hybrid model extraction automatically, a tool called MXTC (Model Extraction Tool for C), is in development. The tool targets large implementations in C, consisting of millions of lines of code (LOC). The resulting model includes probabilistic selection of execution times, interarrival times and behaviors, in order to represent behavioral variations that would otherwise require extremely detailed models, also including hardware behavior, which would slow down simulations by several magnitudes. More information can be found in [9].

III. Approximate Timing Analysis

A. MABERA Framework

MABERA is an implementation of an iterative process, where each iteration consists of a set of s simulations, which produce a *generation* of simulation results. Each simulation has the length l, i.e. model time when to stop each simulation. The first (initial) generation is produced by running s independent probabilistic simulations. Each simulation of a generation is evaluated and assigned a *fitness score* based on three properties of the task in focus: the highest observed response time, execution time and preemption count for any task instance during the simulation. The fitness scores are used to select p number of *parent simulations* for next generation. Each parent simulation is used to produce s/p child simulations for the next generation, which are *mutations* of the parent simulation. The child simulations explore the state-space "close" to the parent simulation and are likely to detect a response time for T higher than the highest response time for T of the parent simulation, unless the parent simulation already have discovered the WCRT. The algorithm iterates in this manner until a termination condition is reached, which depends on the termination threshold, $\mathtt{tt}$. The value of $\mathtt{tt}$ decides how many "unsuccessful" generations that are allowed before termination, i.e. generations that failed to discover a response time higher than the highest response time of the previous generations.

1) Seed Schedules: In MABERA, each chromosome (i.e. simulation input parameters) in the genetic algorithm represents a *seed schedule*. The seed schedule specifies the random seed values that are used for generation of pseudo-random numbers and thereby outcomes of all non-deterministic selections during the simulation [8], and thereby exactly decides the simulation result. More formally, a seed schedule is simply a set of n pairs (s_i, t_i), where s_i is a seed used to initiate the random number generator in RTSSim and t_i is a simulation time instant. Each seed-time pair (s_i, t_i) in the set is used to change the seed from the current one, originating in s_{i-1}, to s_i, at the specific simulation time instant t_i.

A seed corresponds to the specific sequence of random numbers, which in the context of MABERA is used for many different purposes; task arrival jitter, execution time of part of a task, environmental input stimulus etc. In particular, the specific sequence of random numbers corresponds exactly to a particular execution of the modeled application.

In MABERA, mutation is done by inserting a new generated seed at a specific simulation time point, which has the effect of changing the execution trace completely for the rest of the simulation. This makes MABERA efficient at finding seed values early in the chain that maximizes response time, but also makes it impossible to exchange only some specific simulation parameters, for example variable input stimulus values, or specific execution times of a task, that might on their own severely affect the response time.

This is the reason why the genetic crossover operator, combining two different individuals (the parents) into two new individuals (the offspring), was never implemented. Using seed schedules, it is in practice not meaningful to combine two "good" chromosomes into a better offspring, since only the initial part of the seed schedule will keep its properties when recombined, and the offspring will thus not inherit other properties than the first part of the seed schedule from their parents. In essence, crossover would only serve to introduce more randomness, and can therefore be replaced by mutation. The randomness caused by mutation diversifies the search in the large system state space that we are exploring, but we still need to introduce the randomness in a more systematic manner, more similar to other evolutionary frameworks [10], [11].

B. A New Solution

Our previous evaluation showed that MABERA is significantly more efficient than traditional probabilistic simulation in finding approximations of the WCRT. Unfortunately, there still exists cases with even higher response times which MABERA fails to find. Moreover, the obtained results are unstable, i.e. almost 50% replications failed in finding higher RT. We therefore propose a substantially more structured approach using genetic algorithms (GA) also including crossover, and where certain key aspects of the system at hand is encoded directly as parameters in the genetic algorithm. The new approach is outlined below.

1) A New Representation of Chromosome: To remedy the shortcomings of MABERA, we propose a solution centered around *consumers* of random values. A random value consumer is a property of the system that affects the outcome of response time analysis. Examples of such properties are environmental stimulus variables, runtime jitter results, and execution time samples. In our new approach, we propose to represent each random value consumer as either a separate seed (which when applied yields a sequence of random values) or in a more explicit representation, i.e. a sequence of random numbers. This sequence is then used directly to provide input to simulator. The advantage of this approach is that there is a direct relationship between genes in the chromosome and properties in the program to be simulated, and that different consumers are fully separated from each other. These properties make it possible to refine specific aspects of the chromosome by using crossover and local improvement techniques, in order to improve MABERA. If the number of random value consumers are substantial, we plan to partition these into groups of similar random value consumers sharing a single seed. Partitioning should take care of scaling issues with only a marginal loss of detail.

2) Representation of parameters: Formally, let J_i be a jitter value consumer, X_i be an input stimulus value consumer, and C_i be an execution time consumer. We propose to encode each chromosome by a string

$$\langle J_1, \dots J_n, X_1, \dots, X_m, C_1, \dots, C_b \rangle, \tag{1}$$

where n is the number of tasks which are associated with jitter, m is the number of input stimulus variables, and b is the maximum number of execution time samplings in the model to be simulated. In order to make the new representation more applicable to the general model whilst decreasing the calculation complexity, we propose a representation rule, i.e. the consumers which impact the model behavior most will be represented in a more explicit way, i.e. real values representation. The less impacted consumers will be represented in a higher level abstraction, i.e. seeds-level representation. In the simulator, it is now possible to construct a sequence of random numbers in different representation levels, which can be used for different purposes depending on the type of consumer Y_i represents, i.e. Y_i can represent either J, X or C in the simulation model.

The new chromosome representation has the advantage of a tight coupling of representation (as genes) and particular behavior (as value consumer) in the simulator, which makes it possible to reuse and recombine good genes in the sense of high response time. This allows us to use more advanced genetic operators such as single and multi point crossover. Artificial randomness can be introduced as separate components to diversify the search in a more systematic manner and to generate a sufficient coverage of possible input values.

3) Results: The new solution is applied to the same, described model in [8] using our newly developed prototype tool. In [8], about 50% replications failed in finding higher response times above 8,000 time units on the particular simulation model used for this study. The proposed new solution, using GA, gives very promising and reliable performance, i.e. very high response time of 8,324 time units, which was found in about half of earlier MABERA approach runs, is obtained in 100% of the runs of the new prototype tool. Better yet, the population size necessarily in the new tool is decreased by almost 80%. The new approach is thereby more stable and efficient, and has higher scalability of analyzing more complex industrial-size system in terms of consuming less computation time and replications.

IV. Conclusions and Future Work

In this paper, we presented an approach based on simulation optimization analysis, for approximate timing analysis of complex real-time legacy systems. We proposed a new solution based on *consumers*, which form a direct relationship between genes in the chromosome and properties of the simulation model, for example simulation input stimulus, task jitter and execution time variables. This makes it possible to improve the previous algorithm by using advanced evolutionary mechanisms such as crossover, multiple fitness criteria and multiple population abstractions. As part of the future work, we will continue with the evaluation by comparing the results of this new approach with the old one from [8] as well as other analyses, for instance FPA [2], Monte Carlo simulation etc. We also plan to use our approach to derive more accurate execution times of tasks in the model, which can then be used to improve the results of FPA. We aim to evaluate our analysis method by using a complex simulation model extracted from the target legacy system, the ABB IRC5 robot control system. To analyze other system properties by using our method and relevant tools, e.g. best case and worst case buffer usage analysis, is another reasonable application.

References

[1] S. Perathoner, E. Wandeler, L. Thiele, A. Hamann, S. Schliecker, R. Henia, R. Racu, R. Ernst, and M. G. Harbour, “Influence of different system abstractions on the performance analysis of distributed real-time systems,” in *EMSOFT ’07: Proceedings of the 7th ACM & IEEE international conference on Embedded software*, 2007.

[2] N. C. Audsley, A. Burns, R. I. Davis, K. W. Tindell, and A. J. Wellings, “Fixed priority pre-emptive scheduling: an historical perspective,” *Real-Time Syst.*, pp. 173–198, 1995.

[3] “Rapita systems, www.rapitasystems.com, 2008.”

[4] “Sym TA/S, www.symtavision.com/symtas.html, 2008.”

[5] “Real-time calculus, www.mpa.ethz.ch/rtctoolbox/overview, 2008.”

[6] “Uppaal, www.uppaal.com, 2008.”

[7] D. Decotigny and I. Puaut, “Artisst: An extensible and modular simulation tool for real-time systems,” in *In Proceedings of the Fifth IEEE International Symposium on Object-Oriented Real-Time Distributed Computing (ISORC ’02)*, 2002.

[8] J. Kraft, Y. Lu, C. Norström, and A. Wall, “A metaheuristic approach for best effort timing analysis targeting complex legacy real-time systems,” in *Proceedings of the 14th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS 08)*, April 2008.

[9] J. Kraft, J. Huselius, A. Wall, and C. Norström, “Extracting simulation models from complex embedded real-time systems,” in *Real-Time in Sweden 2007*, August 2007. [Online]. Available: http://www.mrtc.mdh.se/index.php?choice=publications&id=1314

[10] T. Back, F. Hoffmeister, and H. Schwefel, “A survey of evolution strategies,” 1991.

[11] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Professional, January 1989. [Online]. Available: http://www.amazon.ca/exec/obidos/redirect?tag=citeulike09-20&amp;path=ASIN/0201157675