Measurement-Based Estimation of the Worst-Case Execution Time of Consecutive Jobs

Rômulo Silva de Oliveira
Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina
Florianópolis, Brasil
romulo.deoliveira@ufsc.br

Luís Almeida
CISTER / Faculdade de Engenharia
Universidade do Porto
Porto, Portugal
lda@fe.up.pt

***Abstract*—The classic task model used for real-time scheduling analysis includes a single parameter C that represents the worst-case execution time of a task, and it can occur at every single job of that task. Many works in the past extended this model, providing a richer description of a task in terms of the execution time of several consecutive jobs. However, the collective execution time of consecutive jobs is directly influenced by the combinations of hardware states, application states and input data patterns between jobs. The resulting complexity can easily go beyond what typical static analysis can handle. In this paper we consider the use of measurement-based probabilistic timing analysis to obtain estimates for the worst-case execution time of a task, considering several consecutive jobs. We define the Gamma Factor function to describe by how much the interference a task generates is reduced when sequences of jobs are explicitly considered. A set of experiments is used to investigate the applicability and potential gain of this method.**

***Index Terms*—extreme value theory, work load, probabilistic worst-case execution time**

I. Introduction

The classic task model used for real-time scheduling analysis includes a single parameter C that represents an upper bound on the Worst-Case Execution Time (WCET) of task τ, and it can occur at every single job of τ.

Previous works extended this model with a richer description of the execution time of consecutive jobs in a task [1] [2] [3] [4]. One example is the workload curve $\gamma(k)$ that gives an upper bound on the processor time needed to execute any k consecutive jobs of task τ and which is tighter than $k \times C$. Thus, replacing $k \times C$ with $\gamma(k)$ in classic Response-Time Analysis (RTA) also reduces the analysis pessimism [1] [2]. The benefits are particularly noticeable when computing the worst-case response time of tasks of lower priority than τ, but with sufficiently long execution time so that τ interferes k times. If RTA uses $\gamma(k)$ to account for the worst-case cumulative execution time of such k jobs, then it will produce shorter response times than using $k \times C$.

Nevertheless, RTA seldom uses $\gamma(k)$ due to the difficulty of obtaining such values. When one considers the collective execution time of successive jobs of the same task, the number of combinations of hardware state, application state and input data patterns across jobs explode. This complexity may preclude using static analysis as typical for WCET determination. In this case, we have to resort to Measurement-Based Probabilistic Timing Analysis (MBPTA), instead.

MBPTA employs statistical tools from the Extreme Value Theory (EVT) [5] to estimate probabilistic worst-case execution times [6]. MBPTA can produce WCET estimates associated to arbitrarily low exceedance probabilities, which are defined in accordance to the specification of the system. For instance, the probabilistic WCET of task τ with an exceedance probability of 10^{-8} is denoted by $pWCET(10^{-8})$. Since pWCET is obtained from measurements only, it is in principle possible to use MBPTA to obtain $\gamma(k)$ for specific values of k. However, MBPTA poses two main requirements. First, it is necessary to apply a measurement protocol that considers real use scenarios, so input states, hardware states and application software states are representative of those that will occur during system operation. Second, it is necessary to provide evidence that the execution times are analyzable through EVT.

In this paper we use MBPTA, based on EVT, to obtain probabilistic estimates for the worst-case execution time of a task, in terms of a single job but also for several consecutive jobs. To the best of the authors' knowledge, this is the first time MBPTA is proposed to obtain $\gamma(k)$ for $k > 1$, potentially contributing to lower pessimism of schedulability analysis.

II. System Model

Similarly to [1] and [2], the system is composed of a set of tasks. Each individual task τ gives rise to a potentially infinite sequence of jobs with execution times $[E_1, E_2, E_3, ...]$. The following function tells us how much time is needed in the worst-case by any sub-sequence of k jobs starting from jth job in the sequence:

$$\gamma(j,k) = \sum_{l=j}^{j+k-1} E_l, \forall j, k > 0 \tag{1}$$

The workload upper bound curve $\gamma(k)$ gives an upper bound on the time needed to process any k consecutive activations of task τ without inter-task interference:

This work has been partially supported by CAPES, The Brazilian Agency for Higher Education, project PrInt CAPES-UFSC "Automation 4.0", and by FCT/MCTES (PIDDAC), The Portuguese Foundation for Science and Technology, with CISTER Research Unit Base Funding (UIDB/04234/2020).

$$\gamma(k) = \max_{\forall j>0} \gamma(j,k) \tag{2}$$

This definition implies that $\gamma(k)$ is strictly increasing and that $\gamma(1)$ is the worst-case execution time of τ (i.e., $\gamma(1) = C$). Moreover, we can state that:

$$\gamma(k) \leq k \times \gamma(1) \tag{3}$$

Note that $\gamma(k) > k \times \gamma(1)$ would imply that one of the execution times used to compute $\gamma(k)$ would be greater than $\gamma(1)$ contradicting the definition. Nevertheless, when a task presents strong auto-correlation, jobs with long execution times will appear close to each other and $\gamma(k)$ will be close to $k \times \gamma(1)$.

A. The Gamma Factor Function

The *gamma factor* function $GF(k)$ describes by how much the interference caused by task τ is reduced when sequences of jobs are explicitly considered.

$$GF(k) = \frac{\gamma(k)}{k \times \gamma(1)} \tag{4}$$

From Equation (3) we have $GF(k) \leq 1$. Also, $\gamma(k)/k$ represents the average execution time of the k longest jobs. As k tends to infinity, this ratio includes the execution time of all jobs, resulting in the average execution time $\mu(1)$ of the jobs of task τ:

$$\lim_{k\to\infty} \frac{\gamma(k)}{k} = \mu(1) \tag{5}$$

We can, then, define the gamma factor for task τ, denoted by GF, which is the ratio between the average-case and the worst-case execution times:

$$GF = \lim_{k\to\infty} GF(k) = \lim_{k\to\infty} \frac{\gamma(k)}{k \times \gamma(1)} = \frac{\mu(1)}{\gamma(1)} \tag{6}$$

Note that $GF(k)$ varies from 1 (for $k = 1$) to GF (for $k = \infty$) in an asymptotic, usually not monotonic, way.

III. Extreme Value Theory

Measurement-Based Probabilistic Timing Analysis (MBPTA) is an approach to generate probabilistic estimates of the Worst-Case Execution Time (pWCET) of a task. The rationale is that the probability of exceeding pWCET should not be larger than a known bound. MBPTA applies Extreme Value Theory (EVT) to measurements of execution time of the task running on its target environment. The distribution of execution times is adjusted to an asymptotic distribution of extreme values (see the surveys in [7] and [8]).

The applicability of EVT relies on certain assumptions, though. Observed execution times must be amenable to representation as independent and identically distributed (iid) random variables. The assumptions for EVT applicability are tested using several techniques [9] [10] [11] [12], e.g., tests Wald-Wolfowitz, Turning Point, Ljung-Box, Kolmogorov-Smirnov and Anderson-Darling. These tests are expected to produce so-called *p-values* uniformly distributed in $[0, 1)$. These values are usually presented in box and whisker plots highlighting the 0%, 5%, 50%, 95% and 100% quantiles.

The application of EVT requires the following steps:

- Measuring the execution time of several jobs.
- Providing evidence that the method can be applied.
- Selecting a sample of values at the tail of the frequency distribution, e.g., using the Block Maxima technique.
- Fitting a Generalised Extreme Value (GEV) distribution to the maxima by estimating the parameters *Location*, *Scale* and *Shape*, e.g., the *Lmoments* method.
- Checking the goodness of fit between the measurements and the fitted GEV, e.g., using quantile-quantile (QQ) plots.
- Obtaining the execution time value with the desired exceedance probability (or vice-versa).

We are interested in modelling the total execution time of a sequence of consecutive τ jobs. The moving sum of k execution times over all executions of τ generates measurements of the execution time of k consecutive jobs, i.e., $\gamma(k)$. These measurements are not independent, but can be analysed with the Block Maxima approach, which retains only the maximum value of each block. In practice only the block maxima need to be independent for the theory to apply [7].

IV. Proof-of-Concept Experiments

Tests were performed on a microcomputer with an Intel(R) Core(TM)2 Q9300, a quad-core CPU. This particular model does not support hyper-threading and the frequency of all cores were fixed to approximately 2GHz. This is a complex processor, with several probabilistic hardware acceleration components, where static analysis of worst-case execution time is unfeasible. We used the Ubuntu 18.04.3 LTS distribution, which is based on the Linux kernel 5.3.

The test task was implemented as a C function of a Linux thread that calls the function each 10 ms. We first generate the input data for the defined number of executions (sample size) and then measure the execution time of each execution. The overall computer load does not vary much during the experiment, as common in embedded systems. Moreover, no particular cache setting was used.

Execution times are measured using the Perf tool, a profiler for Linux 2.6+ based systems. It is based on the *perf_events* interface, exported by recent versions of the Linux kernel, and offers an integrated collection of sub-commands [13]. By observing tracing events one can compute the task execution time with reasonable accuracy, at the resolution of microseconds.

In all experiments we collect the execution time of 10000 jobs. These values are used to estimate $\gamma(1)$. We then use a moving sum of size k to obtain measurements for k consecutive task executions. We use an exceedance probability of 10^{-8} to obtain an estimate for each value of $\gamma(k)$. We fitted the measurements to a GEV distribution using Lmoments through the Block Maxima approach with block size of 100. We applied EVT using the R statistical software with the extRemes [14] package.

A. Experiment 1: Fixed Size Bsort

Here we take the Bsort task from the Mälardalen WCET benchmark [15] to sort an array of fixed size with 900 random integers using the bubble sort method. The p-values yielded by the iid tests from Section III are acceptable, since they produced p-values distributed in the range $[0, 1)$. We fitted the execution times to a GEV distribution using Lmoments and the resulting Q-Q plot was acceptable.

We used an exceedance probability of 10^{-8} to obtain an estimate for each value of $\gamma(k)$ in microseconds. Table I shows, for some values of k, the mean $\mu(k)$, the coefficient of variation (i.e., the ratio of the standard deviation to the mean) $CV(k)$, the high water mark $HWM(k)$, the estimated value for $\gamma(k)$, the commonly used $k * \gamma(1)$ and the gamma factor function $GF(k)$. For this scenario we have $\mu(1) = 4760$ and $\gamma(1) = 4993$ which results in $GF = 0.953$.

TABLE I
RESULTS FOR FIXED SIZE BSORT

k	$\mu(k)$	$CV(k)$	HWM	$\gamma(k)$	$k * \gamma(1)$	$GF(k)$
1	4760	0.010	4942	4993	4993	1.000
2	9519	0.007	9761	9822	9986	0.984
3	14279	0.006	14583	14659	14979	0.979
4	19039	0.005	19441	19572	19972	0.980
5	23798	0.004	24273	24418	24965	0.978
6	28558	0.004	29016	29256	29958	0.977
7	33318	0.004	33802	34227	34951	0.979
8	38077	0.003	38573	39095	39944	0.979
9	42837	0.003	43377	44178	44937	0.983
10	47597	0.003	48227	49131	49930	0.984
20	95193	0.002	95878	96570	99860	0.967
30	142790	0.002	143582	144070	149790	0.962
40	190387	0.002	191317	191598	199720	0.959
50	237983	0.001	238993	239580	249650	0.960

Since all values of $\gamma(k)$ are statistically estimated, they are always approximations. Notwithstanding, the values presented in Table I respect the property $\gamma(k) \leq \gamma(k_1) + \gamma(k_2), k = k_1 + k_2$. For instance, we can use $\gamma(20) + \gamma(30)$ instead of $\gamma(50)$, although the resulting $GF(50)$ would be smaller, i.e., the advantage of using $\gamma(k)$ in RTA would be lesser.

B. Experiment 2: Random Size Bsort

In this experiment, each job of task bsort has to sort a new array of random integers using the bubble sort method. The size of the array is a sample taken from a random variable with uniform distribution between 800 and 1000. The p-values yielded by the statistic tests, considering the execution time of one job, are acceptable. We fitted the measurements to a GEV distribution using method Lmoments. The resulting Q-Q plot is showed in Fig. 1 and deemed acceptable. Good fitting in Q-Q plots is evidenced by the dots being disposed over or randomly distributed around and close to the $1:1$ line, meaning that the sampled values are compatible with those expected when the fitted distribution properly models the observed data.

Clearly the input data, generated with moving sums for $k > 1$, makes the execution times of two consecutive measurements highly correlated as k increases. The method Block Maxima naturally reduces the auto-correlation because only the maximum of each block is used. Fig. 2 shows the box-and-whisker plots of the iid tests applied to the block maxima of $k = 10$. As the value of k increases, the auto-correlation also increases, and larger blocks may be necessary.

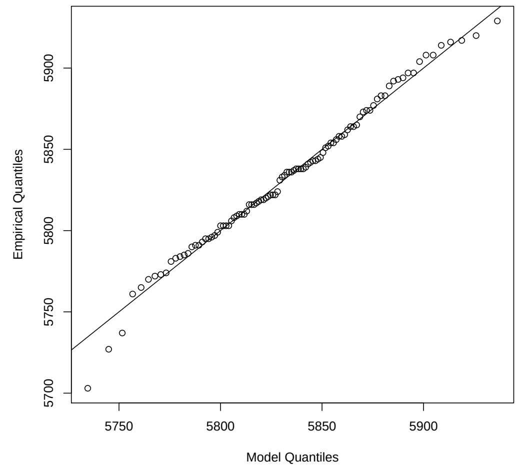

Fig. 1. Quantile plot for $k = 1$, random size array.

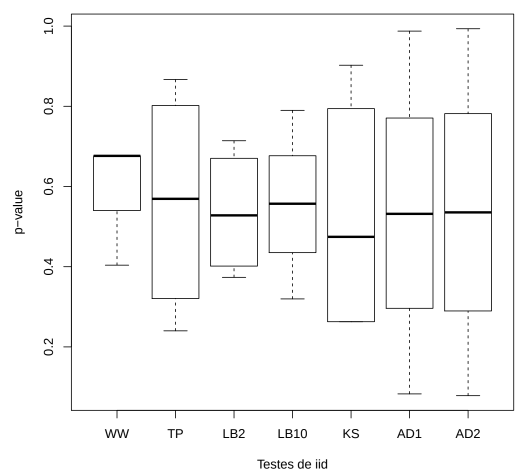

Fig. 2. Iid tests, block maxima for $k = 10$, random size array.

We fit the measurements for $k = 10$ to a GEV distribution using Lmoments. Fig. 3 shows the respective QQ-plot, which shows adherence of the empirical data to the fitted distribution. We consider the GEV model applicable in this case and proceed to compute $\gamma(k)$ for this experiment.

An exceedance probability of 10^{-8} is used again in order to obtain an estimate for each value of function $\gamma(k)$. Table II shows, for some values of k, the average $\mu(k)$, the coefficient of variation $CV(k)$, the high water mark $HWM(k)$, the estimated value for $\gamma(k)$, the commonly used $k * \gamma(1)$ and the

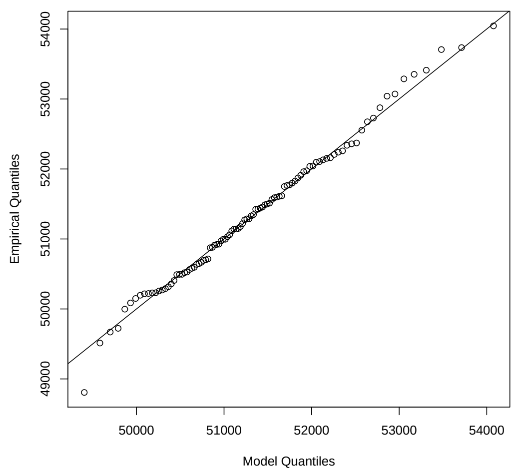

Fig. 3. Quantile plot for $k = 10$, random size array.

gamma factor $GF(k)$. For this scenario we have $\mu(1) = 4732$ and $\gamma(1) = 5989$ which results in $GF = 0.790$.

TABLE II
RESULTS FOR RANDOM SIZE BSORT

k	$\mu(k)$	$CV(k)$	HWM	$\gamma(k)$	$k * \gamma(1)$	$GF(k)$
1	4732	0.129	5929	5989	5989	1.000
2	9463	0.091	11614	11654	11978	0.973
3	14194	0.074	17293	17393	17967	0.968
4	18926	0.064	22905	22991	23956	0.960
5	23658	0.057	28127	28367	29945	0.947
6	28389	0.052	33246	33702	35934	0.938
7	33121	0.048	38739	39534	41923	0.943
8	37853	0.045	43985	45368	47912	0.947
9	42585	0.042	49027	50761	53901	0.942
10	47317	0.040	54045	57284	59890	0.956
20	94632	0.029	105037	107274	119780	0.896
30	141943	0.023	153095	155712	179670	0.867
40	189252	0.020	203178	206862	239560	0.864
50	236562	0.018	253803	260804	299450	0.871

C. Experiment 3: Correlated Size Bsort

In this experiment, again each job has to sort a new array of random integers using the bubble sort method. The first job receives an array of 900 elements. For the next job, the size of the array is changed by a quantity that is a sample taken from a random variable with uniform distribution between -5 and $+5$. The array size is limited to the interval $[800, 1000]$.

The statistic tests showed that the execution times of two consecutive jobs are highly correlated. We tried to fit the measurements to a GEV distribution using Lmoments. The quantile plot showed significant systematic discrepancies indicating the data does not adhere to the target distribution. So, we did not compute $\gamma(k)$ for this experiment.

V. CONCLUSION AND WORK IN PROGRESS

This paper proposed using MBTPA (and EVT) to determine the worst-case cumulative execution time of sequences of consecutive jobs, as opposed to the common use for a single job, only. Preliminary results show the potential of the method to tighten the interference a task may cause in lower priority tasks. For instance, the random size Bsort experiment revealed $GF(20) = 0.896$, implying a reduction of more than 10% in the potential interference this task causes on a lower priority task that has sufficiently long execution time, so that it causes interference for 20 consecutive jobs.

We are currently working on a deeper characterization of the applicability and benefits of this approach.

REFERENCES

[1] A. Maxiaguine, S. Kunzli, and L. Thiele, "Workload characterization model for tasks with variable execution demand," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, Feb 2004, pp. 1040–1045 Vol.2.

[2] M. Jersak, R. Henia, and R. Ernst, "Context-aware performance analysis for efficient embedded system design," in *Proceedings Design, Automation and Test in Europe Conference and Exhibition*, vol. 2, Feb 2004, pp. 1046–1051 Vol.2.

[3] S. Quinton, M. Hanke, and R. Ernst, "Formal analysis of sporadic overload in real-time systems," in *2012 Design, Automation Test in Europe Conference Exhibition (DATE)*, March 2012, pp. 515–520.

[4] S. Tobuschat, R. Ernst, A. Hamann, and D. Ziegenbein, "System-level timing feasibility test for cyber-physical automotive systems," in *2016 11th IEEE Symposium on Industrial Embedded Systems (SIES)*, May 2016, pp. 1–10.

[5] S. Coles, *An introduction to statistical modeling of extreme values*, ser. Springer Series in Statistics. London: Springer-Verlag, 2001.

[6] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quiñones, and F. J. Cazorla, "Measurement-based probabilistic timing analysis for multi-path programs," in *2012 24th Euromicro Conference on Real-Time Systems*, July 2012, pp. 91–101.

[7] R. Davis and L. Cucu-Grosjean, "A survey of probabilistic timing analysis techniques for real-time systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, p. 60, 2019. [Online]. Available: https://hal.inria.fr/hal-02158973

[8] F. J. Cazorla, L. Kosmidis, E. Mezzetti, C. Hernandez, J. Abella, and T. Vardanega, "Probabilistic worst-case timing analysis: Taxonomy and comprehensive survey," *ACM Comput. Surv.*, vol. 52, no. 1, Feb. 2019. [Online]. Available: https://doi.org/10.1145/3301283

[9] L. Santinelli, J. Morio, G. Dufour, and D. Jacquemart, "On the sustainability of the extreme value theory for wcet estimation," in *WCET*, vol. 39, 07 2014.

[10] J. Abella, E. Quiñones, F. Wartel, T. Vardanega, and F. J. Cazorla, "Heart of gold: Making the improbable happen to increase confidence in mbpta," in *2014 26th Euromicro Conference on Real-Time Systems*, July 2014, pp. 255–265.

[11] F. W. Scholz and M. A. Stephens, "K-sample anderson-darling tests," *Journal of the American Statistical Association*, vol. 82, no. 399, pp. 918–924, 1987. [Online]. Available: http://www.jstor.org/stable/2288805

[12] G. M. Ljung and G. E. P. Box, "On a measure of lack of fit in time series models," *Biometrika*, vol. 65, no. 2, pp. 297–303, 1978. [Online]. Available: http://www.jstor.org/stable/2335207

[13] Perf. Linux perf tool. (2020, January 31). [Online]. Available: https://perf.wiki.kernel.org/

[14] E. Gilleland and R. Katz, "extremes 2.0: An extreme value analysis package in r," *Journal of Statistical Software, Articles*, vol. 72, no. 8, pp. 1–39, 2016. [Online]. Available: https://www.jstatsoft.org/v072/i08

[15] J. Gustafsson, A. Betts, A. Ermedahl, and B. Lisper, "The mälardalen wcet benchmarks: Past, present and future." in *WCET*, ser. OASICS, B. Lisper, Ed., vol. 15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2010, pp. 136–146.