UNIVERSITÄT AUGSBURG
Universitätsbibliothek

OPEN ACCESS
OPUS AUGSBURG

Progress on static probabilistic timing analysis for systems with random cache replacement policies

Sebastian Altmeyer, L. Cucu-Grosjean, Robert I. Davis, B. Lesage

Angaben zur Veröffentlichung / Publication details:

Altmeyer, Sebastian, L. Cucu-Grosjean, Robert I. Davis, and B. Lesage. 2014. "Progress on static probabilistic timing analysis for systems with random cache replacement policies." In *Proceedings of the 5th Real-Time Scheduling Open Problems Seminar (RTSOPS 2014), co-located with the 26th Euromicro Conference on Real-Time Systems (ECRTS 2014), Madrid, Spain, July 8-11, 2014*, edited by Marko Bertogna and Sathish Gopalakrishnan, 7–8. RTSOPS.

Nutzungsbedingungen / Terms of use:

licgercopyright

Dieses Dokument wird unter folgenden Bedingungen zur Verfügung gestellt: / This document is made available under these conditions:
Deutsches Urheberrecht
Weitere Informationen finden Sie unter: / For more information see:
https://www.uni-augsburg.de/de/organisation/bibliothek/publizieren-zitieren-archivieren/publiz/

Progress on static probabilistic timing analysis for systems with random cache replacement policies

Sebastian Altmeyer
University of Amsterdam,
altmeyer@uva.nl

Liliana Cucu-Grosjean
INRIA Paris-Rocquencourt
liliana.cucu@inria.fr

Robert I. Davis
University of York,
rob.davis@york.ac.uk

Benjamin Lesage
University of York,
benjamin.lesage@york.ac.uk

I. Original Problem Statement

Real-time systems such as those deployed in space, aerospace, automotive and railway applications require guarantees that the probability of the system failing to meet its timing constraints is below an acceptable threshold (e.g. a failure rate of less than 10^{-9} per hour). Advances in hardware technology and the large gap between processor and memory speeds, bridged by the use of cache, make it difficult to provide such guarantees without significant over-provisioning of hardware resources. The use of deterministic cache replacement policies means that pathological worst-case behaviours need to be accounted for, even when in practice they may have a vanishingly small probability of actually occurring. The use of cache with random replacement policies [3] can negate the effects of pathological worst-case behaviours while still achieving efficient average-case performance, hence providing a way of increasing guaranteed performance in hard real-time systems.

The timing behaviour of programs running on a processor with a random cache replacement policy can be determined using Static Probabilistic Timing Analysis (SPTA). SPTA computes an upper bound on the probabilistic Worst-Case Execution Time (pWCET) in terms of an exceedence function, which gives the probability, as a function of all possible values for an execution time budget x, that the execution time of the program will not exceed that budget on any single run. SPTA [5] requires a probability function that can be used to compute an estimate of the probability of a cache hit for each memory access. This probability function is *valid* if it provides a lower bound on the probability of a cache hit. As shown last year at RTSOPS 2013 [4], the only valid cache-hit probability known by then is given as follows:

$$\hat{P}^D(k) = \begin{cases} \left(\frac{N-1}{N}\right)^k & N > k \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where N denotes the associativity of the cache and k the reuse distance, i.e., the number of intervening memory accesses that could cause an eviction, since the memory block was last accessed. All other estimations of the hit-probability [7, 6] that had been proposed by then have been refuted as they may lead to optimistic results. The complexity of deriving a sound estimate of the hit-probability is caused by the dependency of the current event of a cache hit or miss on the history of prior events; caused by the finite size of the cache. In Equation (1), this dependency is accounted for by setting the probability of a cache hit to zero in cases where the reuse distance exceeds the associativity; which results in a large over-approximation even for simple access sequences. The open problem presented in last year's RTSOPS [4] was thus: *how to improve upon the simple SPTA analysis?*

II. Correctness Conditions and Optimality

Instead of immediately answering the open problem, we tried to learn from the failed approaches to improve upon Equation (1) and identified the correctness conditions [2] that any sound approximation of the cache-hit probability must fulfil. Sound in this context means that for any sequence of cache accesses $[e_1, \ldots, e_n]$, the approximation $\hat{P}$ complies with two constraints: (**C1**) it does not over-estimate the probability of a cache hit, and (**C2**) the value obtained from convolution of the approximated probabilities for any subset of a trace T describing the probability that all elements in the subset are a hit, is at most the precise probability of such an event occurring:

C1 $\forall e \in [e_1, \ldots, e_n]\colon P(e^{hit}) \geq \hat{P}(e^{hit})$,
C2 $\forall E \subseteq [e_1, \ldots, e_n]\colon P\left(\bigwedge_{e\in E} e^{hit}\right) \geq \prod_{e\in E} \hat{P}(e^{hit})$.

Using these soundness conditions, we have been able to clearly identify why former approaches [7, 6] failed and we have been able to show that Equation (1) is not only correct, but also optimal with respect to the limited information it uses: any cache-hit probability that only uses the associativity and the reuse distance is either at most as precise as Equation (1) or optimistic. Due to space limitation, we refer to [1] for the proof of optimality.

III. Using other information

The negative result that we can not improve the existing cache-hit probability by using the same information also gives the key to providing better bounds: we have to include additional information which is not yet taken into account.

A. Stack Distance

$\hat{P}^D(k)$ can be pessimistic in the commonly observed case of sequences with repeated accesses (e.g. loops). For example, the trace $a, b, c, d, c^1, d^1, c^1, d^1, a^7, b^7$ repeats the accesses c, d three times within the reuse distance of the final accesses to a and b. Assuming an associativity of 4, then $\hat{P}^D(k)$ gives zero probability of a cache hit for these accesses, since their reuse distance exceeds the associativity of the cache. However, it is possible for the cache to contain all four distinct memory blocks a, b, c, d accessed in this sequence, and so a zero value for the probability of a cache hit for the final accesses to a and b is pessimistic.

Let Δ be the *stack distance* of element e_l, i.e., the total number of pair-wise *distinct* memory blocks that are accessed within the reuse distance k of element e_l. The maximum number of distinct cache locations loaded during the reuse

distance of e_l is upper bounded by Δ, hence it follows that a lower bound on the probability that e_l will survive all of the loads and remain in the cache is given by:

$$\hat{P}^A(\Delta, k) = \begin{cases} \left(\frac{N-\Delta}{N}\right) & (N > \Delta) \wedge (k \neq \infty) \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

We note that $\hat{P}^A(\Delta, k)$ and $\hat{P}^D(k)$ are incomparable, yet both give valid lower bounds on the probability of a cache hit. We thus may use the maximum of them to compute an improved lower bound that dominates each individually.

B. Cache Contention

Equation (1) and Equation (2) both provide a tight lower bound on the probability of a cache hit, but are imprecise even for simple access sequences. If we consider for instance a random cache with associativity 4 and the following access sequence, $a, b, c, d, f, a^4, b^4, c^4, d^4, f^4$ all accesses are considered cache misses. The reason for this is that for each of the last five accesses, the probability of a cache hit is set to 0 to ensure correctness with respect to condition **C2**, i.e, that the probability of the last five access all being hits is zero. However, this can also be ensured by considering the probability of a cache hit for the preceding accesses. To this end, we define the concept of the *cache contention* of a memory block e_l which denotes the number of memory accesses within the reuse distance of e_l that potentially contend with e_l for space in the cache. We only need to set the probability of a cache hit for an access e_l to zero when the cache contention is greater than or equal to the associativity N.

$$\hat{P}^N(e_l^{hit}) = \begin{cases} 0 & con(e_l, T) \geq N \\ max\left(\hat{P}^A(\Delta, k), \left(\frac{N-1}{N}\right)^k\right) & \text{otherwise} \end{cases} \tag{3}$$

Conceptually, the cache contention assumes that each access within the reuse distance of e_l that has been assigned non-zero probability of being a hit as requiring its own separate location in the cache. Due to space limitation, we refer to [1] for the exact definition of the cache contention.

IV. Collecting Semantics and Combined Approach

An orthogonal approach to compute the pWCET is to enumerate all possible cache states and the associated probabilities. As this solution is computationally intractable, we have developed a combined approach with scalable precision: The idea is to use the precise approach for a small subset of *relevant* memory blocks, while using the imprecise approach for the remaining blocks. So, instead of enumerating all possible cache states, we abstract the set of cache states and focus only on the m most important memory blocks, where m can be chosen to control both the precision and the runtime of the analysis. In this way, we effectively reduce the complexity of the precise component of the analysis for a trace with l distinct elements from 2^l to 2^m (typically with $m \ll l$). We again refer to [2] for the details of this approach.

V. Open Problems and Future Work

This progress report presents the solutions to one of last year's open problems: *how to improve upon the simple SPTA analysis?* A first negative result, namely that the original hit probability can not be improved without additional information, has led us towards (i) the discovery of alternative approaches to bound cache-hit probability that rely on additional information such as the stack distance and the cache contention and (ii) the development of an orthogonal approach that relies on complete, or partial enumeration of the cache contents. As according to George Bernard Shaw *science never solves a problem without creating ten more*, the recent advancements lead to new, open problems. Foremost, how to extend the analysis to control-flow graphs and how to select the relevant memory blocks for the combined approach.

Acknowledgements

This work was funded by COST Action IC1202 (TACLe), and the EU FP7 Integrated Project PROXIMA (611085).

References

[1] Sebastian Altmeyer and Robert I. Davis. On the correctness, optimality and precision of static probabilistic timing analysis. Technical Report YCS-2013-487, University of York, 2013. Available from http://www.cs.york.ac.uk/ftpdir/reports/2013/YCS/487/YCS-2013-487.pdf.

[2] Sebastian Altmeyer and Robert I. Davis. On the correctness, optimality and precision of static probabilistic timing analysis. In *DATE 2014*, page tbp, 2014. Available from http://www.cs.york.ac.uk/ftpdir/reports/2013/YCS/487/YCS-2013-487.pdf.

[3] Francisco J. Cazorla, Eduardo Quiñones, Tullio Vardanega, Liliana Cucu, Benoit Triquet, Guillem Bernat, Emery D. Berger, Jaume Abella, Franck Wartel, Michael Houston, Luca Santinelli, Leonidas Kosmidis, Code Lo, and Dorin Maxim. Proartis: Probabilistically analyzable real-time systems. *ACM Trans. Embedded Comput. Syst.*, 12(2s):94, 2013.

[4] Robert I. Davis. Improvements to static probabilistic timing analysis for systems with random cache replacement policies. In *RTSOPS 2013*, pages 22–24, 2013.

[5] Robert I. Davis, Luca Santinelli, Sebastian Altmeyer, Claire Maiza, and Liliana Cucu-Grosjean. Analysis of probabilistic cache related pre-emption delays. In *ECRTS 2013*, pages 129–138, 2013.

[6] Leonidas Kosmidis, Jaume Abella, Eduardo Quiñones, and Francisco J. Cazorla. A cache design for probabilistically analysable real-time systems. In *DATE 2013*, pages 513–518, 2013.

[7] Shuchang Zhou. An efficient simulation algorithm for cache of random replacement policy. In *NPC 2010*, pages 144–154, 2010.