CHAIR OF CYBER-PHYSICAL SYSTEMS IN PRODUCTION ENGINEERING

Static timing analysis for CPU caches with new replacement policies

Point of Contact: Dr. Tomasz Kloda
tomasz.kloda@tum.de

Type: Master

Description:

While the CPU caches are likely to decrease the average execution times of the programs, the worst-case can remain unaffected or even increase. Over the years, practitioners have been disabling the caches in real-time systems due to the lack of appropriate cache-aware worst-case execution time analysis. Such analysis attempts to determine for each point in the task execution the cache content and predict the potential cache misses that can take tens or hundreds of cycles to resolve. For straight-line programs and known initial cache content, this is easy and can be done by a standard simulator. However, for multi-path programs whose control flow depends on input data, it is, in general, an undecidable and complex problem.

The cache replacement policy is a key design parameter that controls the cache runtime behavior. Caching the most recently accessed memory entries is the most common strategy (*LRU* replacement policy) implemented in the modern general-purpose processors as the programs tend to reuse the data from the same memory region (*i.e.*, spatial locality) over a short period of time (*i.e.*, temporal locality). Such replacement policy, achieving a good hit ratio for most workloads, is, however, susceptible to *thrashing* (*i.e.*, the useful data is evicted by the incoming data that is not going to be reused) for memory-intensive workloads that have a working set exceeding the cache size. Artificial intelligence and computer vision are examples of today's data-intensive applications. Fortunately, the cache thrashing in these applications can be significantly reduced by a minor modification in the common *LRU*

replacement policy: placing the incoming line in the lowest-rank position (instead of the highest-rank) and promoting it to the highest-rank only upon the next reuse [1]. Whereas the simulation results for the selected applications indicate improved performance, the hard real-time systems require strong deterministic guarantees of timing correctness that can be provided only by a formal analysis.

The main objective of the thesis is to derive a static timing analysis for deterministic caches under the *Last-position Insertion Policy* (*LIP*) [1]. The new analysis can be built on a large body of literature [2] and tools available in the context of the standard replacement policies (*e.g.*, *LRU*, *FIFO*, *RANDOM*). Its implementation can take the form of an extension to the existing frameworks or a new stand-alone tool.

[1] M. Qureshi, A. Jaleel, Y. Patt, S. Steely, and J. Emer. 2007. Adaptive insertion policies for high performance caching. SIGARCH Comput. Archit. News 35, 2 (May 2007), 381–391.
https://people.csail.mit.edu/emer/papers/2007.06.isca.dip.pdf

[2] M. Lv, N. Guan, J. Reineke, R. Wilhelm, and W. Yi. 2016. A survey on static cache analysis for real-time systems. Leibniz Transactions on Embedded Systems, vol. 3, no. 1. p. 05:1-48.
http://user.it.uu.se/~yi/pdf-files/2016/lgyrw-acm15.pdf

Requirements:

Ability to think logically about computer architectures.
Programming skills.

Chair of Cyber-Physical Systems in Production Engineering,
Technical University of Munich (TUM),
Boltzmannstr. 15, 85748 Garching b. München

https://rtsl.cps.mw.tum.de/theses