Poster

ResilienceP Analysis: Bounding Cache Persistence Reload Overhead for Set-Associative Caches

Syed Aftab Rashid

CISTER-TR-190607

ResilienceP Analysis: Bounding Cache Persistence Reload Overhead for Set-Associative Caches

Syed Aftab Rashid

CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

https://www.cister-labs.pt

Abstract

This work presents different approaches to calculate CPRO for set-associative caches. The PCB-ECB approach uses PCBs of the task under analysis and ECBs of all other tasks in the system to provide sound estimates of CPRO for set-associative caches. The resilienceP analysis then removes some of the pessimism in the PCB-ECB approach by considering the resilience of PCBs during CPRO calculations. We show that using the state-of-the-art (SoA) resilience analysis to calculate resilience of PCBs may result in underestimating the CPRO tasks may suffer. Finally, we have also presented a multi-set alike resilienceP analysis that highlights the pessimism in the resilienceP analysis and provides some insights on how it can be removed.

ResilienceP Analysis: Bounding Cache Persistence Reload Overhead for Set-Associative Caches

Syed Aftab Rashid[1]

[1]CISTER, ISEP, Polytechnic Institute of Porto, Portugal

Abstract

This work presents different approaches to calculate CPRO for set-associative caches. The PCB-ECB approach uses PCBs of the task under analysis and ECBs of all other tasks in the system to provide sound estimates of CPRO for set-associative caches. The resilienceP analysis then removes some of the pessimism in the PCB-ECB approach by considering the resilience of PCBs during CPRO calculations. We show that using the state-of-the-art (SoA) resilience analysis to calculate resilience of PCBs may result in underestimating the CPRO tasks may suffer. Finally, we have also presented a multi-set alike resilienceP analysis that highlights the pessimism in the resilienceP analysis and provides some insights on how it can be removed.

1. Motivation and Introduction

In modern systems, the latency of an access to the main memory is much higher than the latency of an individual computation on the processor. Cache memory bridge this performance gap between the main memory and processor by holding frequently required data and instructions. Intuitively, caches are used to decrease average-case memory access latency; however, due to their limited capacity in comparison to main memory the use of caches can also cause large variations in the execution times of the tasks. This is mainly because of the limited cache space, not all data and instructions of all tasks can simultaneously reside in the cache. Hence, tasks may compete for cache space, with the execution of one task potentially evicting memory blocks previously loaded into the cache by other tasks. This may result in increasing the worse-case execution/response time (WCET/WCRT) of tasks depending on whether the instructions and data needed by the tasks are already present in the cache (i.e. cache hit) or not (i.e. cache miss).

The impact of caches on the WCET/WCRT of tasks is more evident under preemptive scheduling. In preemptive scheduling, tasks may suffer two types of additional execution delays depending on the state of the cache, i.e., Cache Related Preemption Delays (CRPDs) and Cache Persistence Reload Overheads (CPROs). CRPDs refer to the delay in execution time of the preempted tasks due to reloading of Useful Cache Blocks (UCBs) (i.e., blocks that may be cached and that may be reused later) that are evicted from the cache during the execution of the preempting tasks. Whereas, CPROs result from the eviction of Persistent Cache Blocks (PCBs) (i.e., memory blocks that are once loaded into cache by the task will never be invalidated or evicted by the task itself) of tasks due to interleaved or preemptive execution of all other tasks in the system. Considering that CRPDs and CPROs can induce significant delay in task's WCET/WCRT, many different approaches have been presented in the state-of-the-art (SoA) to bound CRPDs (Altmeyer et al. 2011, Altmeyer et al. 2012, Busquets et al. 1996, Lee et al. 1998, Staschulat et al. 2005, Tan et al. 2007, Tomiyama et al. 2000) and CPROs (Rashid et al. 2016, Rashid et al. 2017). In these approaches, CRPDs are usually calculated using the UCBs of the preempted tasks (Lee et al. 1998), Evicting Cache Blocks (ECBs) (i.e.,. all cache block used by the task during its execution) of the preempting tasks (Busquets et al. 1996, Tomiyama et al. 2000) or a combination of both (Altmeyer et al. 2011, Altmeyer et al. 2012, Tan et al. 2007). Whereas, CPROs are calculated using the PCBs of the task under analysis and ECBs of all other tasks in the system (Rashid et al. 2016, Rashid et al. 2017).

However, most of the SoA approaches that focus on CRPD/CPRO calculation consider a direct-mapped cache. In a direct-mapped cache, each cache set can only hold at most one memory block. Therefore, in case of a cache conflict between two tasks τ_i and τ_j, one ECB of τ_j may only evict one UCB/PCB of τ_i and vice versa. This makes the CRPD/CPRO calculation for direct-mapped caches relatively simpler and more precise in comparison to a set-associative cache. In a set-associative cache any cache set may hold more than one memory block(s) depending on the number of cache ways or the cache associativity (i.e., the number of memory blocks that can be mapped to a single cache set). Therefore, under a set-associative cache a single cache conflict between two tasks τ_i and τ_j, e.g., one ECB of τ_j being mapped to the same cache set S used by τ_i, may lead to multiple cache misses depending on the cache replacement policy.

Few approaches that have been presented in literature considering set-associative caches only focus on CRPD computations. However, it has been shown in recent works (Rashid et al. 2016, Rashid et al. 2017) that only considering CRPDs for tasks scheduled under fixed-priority preemptive scheduling may result in pessimistic WCRT bounds. Moreover, it is also shown that the WCRT analysis that considers both CRPD and CPRO (Rashid et al. 2016, Rashid et al. 2017) dominate the WCRT analysis that only consider CRPD (Altmeyer et al. 2011, Altmeyer et al. 2012). Considering that the existing approaches for CPRO calculation only consider direct-mapped caches, in this paper we present different approaches to bound the CPRO for set-associative caches. First, we present the PCB-ECB approach that only considers PCBs of the task under analysis and ECBs of all other tasks in the system to calculate CPRO for set-associative caches. We then introduce the resilienceP analysis that determines how much disturbance a PCB of task τ_i may endure before being evicted from the cache due to other task executions. ResilienceP analysis removes some of the pessimism in the PCB-ECB approach. Finally, we present a multi-set alike extension to the resilienceP analysis that considers variation in the resilience of PCBs over different job executions of a task in order to have a tighter CPRO bound in a time interval of length t.

Author Keywords. Preemptive Scheduling, Cache Related Preemption Delays, Cache Persistence Reload Overhead,

Acknowledgments. This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UID/CEC/04234).

References

Altmeyer, Sebastian, Robert I. Davis, and Claire Maiza. "Improved cache related pre-emption delay aware response time analysis for fixed priority pre-emptive systems." Real-Time Systems 48, no. 5 (2012): 499-526.

Altmeyer, Sebastian, Robert I. Davis, and Claire Maiza. "Cache related pre-emption delay aware response time analysis for fixed priority pre-emptive systems." In 2011 IEEE 32nd Real-Time Systems Symposium, pp. 261-271. IEEE, 2011.

Altmeyer, Sebastian, Claire Maiza, and Jan Reineke. "Resilience analysis: tightening the CRPD bound for set-associative caches." In ACM Sigplan Notices, vol. 45, no. 4, pp. 153-162. ACM, 2010.

Busquets-Mataix, José V., Juan José Serrano, Rafael Ors, Pedro Gil, and Andy Wellings. Adding instruction cache effect to schedulability analysis of preemptive real-time systems. IEEE, 1996.

Lee, Chang-Gun, Hoosun Hahn, Yang-Min Seo, Sang Lyul Min, Rhan Ha, Seongsoo Hong, Chang Yun Park, Minsuk Lee, and Chong Sang Kim. "Analysis of cache-related preemption delay in fixed-priority preemptive scheduling." IEEE transactions on computers 47, no. 6 (1998): 700-713.

Rashid, Syed Aftab, Geoffrey Nelissen, Damien Hardy, Benny Akesson, Isabelle Puaut, and Eduardo Tovar. "Cache-persistence-aware response-time analysis for fixed-priority preemptive systems." In 2016 28th Euromicro Conference on Real-Time Systems (ECRTS), pp. 262-272. IEEE, 2016.

Rashid, Syed Aftab, Geoffrey Nelissen, Sebastian Altmeyer, Robert I. Davis, and Eduardo Tovar. "Integrated analysis of cache related preemption delays and cache persistence reload overheads." In 2017 IEEE Real-Time Systems Symposium (RTSS), pp. 188-198. IEEE, 2017.

Staschulat, Jan, Simon Schliecker, and Rolf Ernst. "Scheduling analysis of real-time systems with precise modeling of cache related preemption delay." In 17th Euromicro Conference on Real-Time Systems (ECRTS'05), pp. 41-48. IEEE, 2005.

Tan, Yudong, and Vincent Mooney. "Timing analysis for preemptive multitasking real-time systems with caches." ACM Transactions on Embedded Computing Systems (TECS) 6, no. 1 (2007): 7.

Tomiyama, Hiroyuki, and Nikil D. Dutt. "Program path analysis to bound cache-related preemption delay in preemptive real-time systems." In Proceedings of the eighth international workshop on Hardware/software codesign, pp. 67-71. ACM, 2000.