2014 IEEE 22nd Annual International Symposium on Field-Programmable Custom Computing Machines

Abstract: Shared L2 Cache Management in Multicore Real-time System

Gang Chen, Biao Hu, Kai Huang, Alois Knoll
Technical University of Munich
{cheng,hub,huangk,knoll}@in.tum.de

Kai Huang
Zhejiang university
huangk@vlsi.zju.edu.cn

Di Liu
Leiden University
dliu@liacs.nl

***Abstract*—In multicore system, shared cache interference has been recognized as one of the major factors that degrade the average performance as well as predictability of system. How to manage the shared cache in order to optimize the system performance while guaranteeing the system predictability is still an open issue. State-of-the-art techniques on this topic use page coloring to partition the shared cache at OS level. In this paper, we present a shared cache management scheme for multicore system. This shared cache management scheme supports way-based cache partitioning at hardware level, building task-level time-triggered reconfigurable-cache multicore system. We evaluated the proposed scheme w.r.t. different numbers of cores and cache modules and prototyped the constructed MPSoCs on FPGA.**

I. Introduction

With increasing requirement for high-performance, multi-processor system-on-chip (MPSoC) has emerged as the dominated architecture choice for modern computing platforms. To alleviate the high latency of the off-chip memory, MPSoC architectures are typically equipped with hierarchical cache subsystems. Due to this inherent complex cache hierarchy, the analysis of shared cache subsystem has received much attention [3], [5], in recent years.

The main problem of cache hierarchy is that the behavior of shared cache is hard to predict and analyze statically [2] in MPSoCs. How to manage the shared cache in order to optimize the system performance while guaranteeing the system predictability is still an open issue [6]. To address this problem, most of the state-of-the-art techniques [3], [5] on the multicore cache management for real-time systems use page coloring, i.e., a software approach in the OS level, to partition the cache by sets. The problem of page-coloring based techniques is the significantly large timing overhead when computing the color of a page. In [4], the observed overhead of page coloring based dynamic cache partitioning is reported to reaches 7% of the total execution time even after the optimization. To solve this problem, we implement multicore cache management in hardware level and propose an integrated cache management scheme that improves the execution predictability for MPSoCs. This paper summarizes the results built in [1].

II. Cache Management Approach

One ILP formulation is developed to model the time-triggered scheduling as well as the cache configuration for a given mapping of a task set on an MPSoC [2]. With this ILP formulation, optimal task schedule and cache configuration can be computed to minimize the cache misses while preventing deadline misses of all tasks in the task set. The generated schedule and the cache configurations together minimize the cache miss of the cache subsystem while preventing deadline misses of all tasks. According to the generated configurations, tasks can be scheduled with inserting cache configuration instructions in each task invocation. High performance code can be generated by this scheme.

III. Hardware Prototype

In this paper, we developed a parameterized reconfigurable way-based cache memory and prototyped it on FPGA. The cache size, line size, and associativity of the cache memory can be parameterized during compile time while the partition of the cache can be reconfigured during runtime. We also design a complete set of APIs with atomic operation, such that the application tasks can reconfigure their cache sizes during runtime. In addition, We developed a share-clock multi-port timer component that enables the precisely time-triggered schedule for the MPSoCs generated from the above scheme. With the customized multicore system with a reconfigurable way-based cache memory and share-clock mutli-port timer component, we prototyped and evaluated the above cache management scheme on the generated MPSoCs on Altera Statrix III FPGA using benchmarks.

References

[1] G. Chen, B. Hu, K. Huang, and A. Knoll. Cache management and time-triggered scheduling for hard real-time mpsocs. Technical Report TUM, 2014.

[2] G. Chen, K. Huang, J. Huang, and A. Knoll. Cache partitioning and scheduling for energy optimization of real-time mpsocs. In *24th IEEE International Conference on Application-specific Systems, Architectures and Processors (ASAP)*, June 2013.

[3] H. Kim, A. Kandhalu, and R. Rajkumar. A coordinated approach for practical os-level cache management in multi-core real-time systems. In *2013 25th Euromicro Conference on Real-Time Systems (ECRTS)*, 2013.

[4] J. Lin, Q. Lu, X. Ding, Z. Zhang, X. Zhang, and P. Sadayappan. Gaining insights into multicore cache partitioning: Bridging the gap between simulation and real systems. In *14th IEEE International Symposium on High Performance Computer Architecture (HPCA)*, 2008.

[5] R. Mancuso, R. Dudko, E. Betti, M. Cesati, M. Caccamo, and R. Pellizzoni. Real-time cache management framework for multi-core architectures. In *2013 IEEE 19th Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2013.

[6] S. Zhuravlev, J. C. Saez, S. Blagodurov, A. Fedorova, and M. Prieto. Survey of scheduling techniques for addressing shared resources in multicore processors. *ACM Computing Surveys*, 2012.

978-1-4799-5111-6/14 $31.00 © 2014 IEEE
DOI 10.1109/FCCM.2014.52

IEEE computer society

Authorized licensed use limited to: Technische Universitaet Muenchen. Downloaded on August 01,2022 at 19:34:26 UTC from IEEE Xplore. Restrictions apply.