A Study of Low Power and High Performance Cache Hierarchy for Multi-Core Processor

by

Geng Tian

B.Engineering. (Electrical and Electronic),
Xi'an Jiaotong University, 2006
M.Engineering. (Electrical and Electronic),
University of Adelaide, 2009

Thesis submitted for the degree of

Doctor of Philosophy

in

Electrical and Electronic Engineering
University of Adelaide

2015

ADELAIDE
UNIVERSITY
AUSTRALIA

Contents

Abstract

The increasing levels of transistor density have enabled integration of an increasing number of cores and cache resources on a single chip. However, power, as a first order design constraint may bring this trend to a dead end. Recently, the primary design objective has been shifted from pursuing faster speed to higher power-performance efficiency. This is also reflected by the fact that design preference has transitioned from fast super-scalar architecture to slower multi-core architecture.

Tiled chip multiprocessors (CMPs) have shown unmatched advantages in recent years, and they are very likely to be the mainstream in the future. Meanwhile, increasing number of cores will exert higher pressure on the cache system. Expanding cache storage can ease the pressure but will incur higher static power consumption. More importantly, very large caches in future multi-core systems may not be fully utilised. Under-utilised caches consume static power for no productivity. Off-line profiling of applications to determine optimal cache size and configuration is not practical.

This thesis describes dynamic cache reallocation techniques for tiled multi core architectures. We proposed the idea of Break Even number of Misses (BEM). BEM defines, for a given cache configuration and time interval, the maximum number of misses that can be tolerated without increasing the energy delay product. We use BEM as the upper bound to determine a set of thresholds that are used to periodically evaluate the utility of cache. Based on this scheme, we then propose a conservative increase-only resizing method to tune the cache size at tile-level granularity. The increasing only method can be further extended to a dynamic downsizing scheme.

In simulations, our tuning scheme can reduce the static power significantly with a very minor degradation of IPC (Instruction Per Cycle). One thing that can be improved from our resizing scheme is the replacement policy. The estimation of cache utility is based on stack-distance which relies on the recency position of a real *LRU* (least recently used) replacement policy. As is commonly known, LRU is not easy to implement, especially in high associativity caches, which are becoming more commonly used. LRU also suffers from “cache thrashing ”, “scan ”and “inter-thread interference ”. To solve

these three problems, we further propose a novel replacement policy, the Effectiveness-Based Replacement policy (*EBR*) and a refinement, Dynamic *EBR* (*D-EBR*), which combines measures of recency and frequency to form a rank sequence inside each set and evict blocks with lowest rank. To evaluate our design, we simulated all 30 applications from SPEC CPU2006 for uni-core system and a set of combinations for 4-core systems, for different cache sizes. The results show that both of them can achieve higher performance with only half the hardware overhead of real *LRU*. With the help of *EBR*, we further extend our last level cache resize scheme. We discuss how to estimate equivalent utility of cache using *EBR* replacement policy rather than *LRU*, and introduce an *EBR* based resizing scheme. Since *EBR* replacement policy is hardware economical and cache thrashing protected, it is more suitable for the utility estimation.

Finally, to shorten average cache access latency, we propose the idea of using a private replica region to store useful data replicas. Keeping them close to the requester can significantly reduce average access latency, however, it also reduces effective storage size causing higher cache miss rates. We leverage our *EBR* based cache utility estimation method to dynamically change the partition based on cache access patterns to achieve a near optimal result.

Statement of Originality

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

Signed

Date

Acknowledgments

First and foremost, I would like to thank my supervisors A.Prof Michael Liebelt and Dr Braden Phillips sincerely. Without their guidance and encouragement, I would not be able to finish my research. It is my honour to pursue Ph.D under their supervision. They have taught me not only how to do research in this particular area but also taught me how to divide and conquer the puzzle.

I also want to acknowledge my colleagues and academic members in the computer system research group for their help and advice. I would like to thank the staff of School of Electrical and Electronics Engineering at the University of Adelaide for their assistance and technical support.

I would like to acknowledge Microsoft for giving me a job before I finish my thesis which makes my thesis writing much easier.

Finally, I would like to express my gratitude to my wife Jessica for her encouragement, support and patience throughout my entire candidature.

Publications

Journal

Geng Tian, Michael Liebelt, (2014). An effectiveness-based adaptive cache replacement policy, *Microprocess. Microsyst. 38, 1 (February 2014), 98-111. DOI=10.1016/j.micpro.2013.11.011 http://dx.doi.org/10.1016/j.micpro.2013.11.011* . Published.

Conference

Geng Tian, Michael Liebelt, (2011). Utility-based dynamic cache resizing, *Computer Science and Network Technology (ICCSNT), 2011 International Conference on , vol.1, no., pp.610,616, 24-26 Dec. 2011* . Published.

List of Figures

List of Tables