The Journal of Systems and Software 111 (2016) 200–212

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Multi-objective optimization of energy consumption and execution time in a single level cache memory for embedded systems

Josefa Díaz Álvarez [a], José L. Risco-Martín [b,*], J. Manuel Colmenar [c]

[a] Centro Universitario de Mérida, Universidad de Extremadura, Mérida 06800, Spain
[b] Department of Computer Architecture and Automation, Complutense University of Madrid, C/Prof. José García Santesmases 9, Madrid 28040, Spain
[c] Department of Computer Science, Rey Juan Carlos University, Móstoles 28933, Spain

ARTICLE INFO

Article history:
Received 13 January 2015
Revised 9 July 2015
Accepted 3 October 2015
Available online 20 October 2015

Keywords:
Cache memory
Energy
Performance

ABSTRACT

Current embedded systems are specifically designed to run multimedia applications. These applications have a big impact on both performance and energy consumption. Both metrics can be optimized selecting the best cache configuration for a target set of applications. Multi-objective optimization may help to minimize both conflicting metrics in an independent manner. In this work, we propose an optimization method that based on Multi-Objective Evolutionary Algorithms, is able to find the best cache configuration for a given set of applications. To evaluate the goodness of candidate solutions, the execution of the optimization algorithm is combined with a static profiling methodology using several well-known simulation tools. Results show that our optimization framework is able to obtain an optimized cache for Mediabench applications. Compared to a baseline cache memory, our design method reaches an average improvement of 64.43 and 91.69% in execution time and energy consumption, respectively.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Multimedia embedded systems like digital cameras, audio and video players, smartphones, etc., are one of the major driving forces in technology. Currently, they have less powerful resources than desktop systems, but these systems must run multimedia software (video, audio, gaming, etc.). These applications require high performance and consume much energy, which reduces the battery lifetime. The battery in embedded systems is limited in capacity and size because of design constraints. Hence, embedded systems designers must be very concerned on both increasing performance and reducing energy consumption, which in turn will also affect to the lifetime of the device.

In recent years, a number of scientific papers have been published indicating that the memory subsystem acts as an energy bottleneck of the system (Kandemir, 2006). In fact, cache memory behavior affects both performance and energy consumption. The best cache configuration gives us the minimum execution time and the lowest energy consumption. Total cache and block sizes, associativity, and algorithms for search, prefetch and replacement, or write policies are some of the parameters that form a cache configuration. Finding optimal values for these parameters will guide us to reach the best performance and energy consumption. Finding an optimal cache configuration for one single application is a bad choice for other applications with different memory access patterns (Hennessy and Patterson, 2011). Thus, we tackle the problem of finding the optimal cache configuration for all the applications executed in an embedded device, which will improve performance and energy consumption.

Energy optimization directly affects aging of transistors, which is a limiting factor for long term reliability of devices. In a common context where the lifetime of a device is determined by the earliest failing component, the aging impact is more serious on memory arrays, where failure of a single SRAM cell would cause the failure of the whole system. Previous works have shown that saving energy in the memory subsystem can effectively control aging effects and can extend lifetime of the cache significantly (Cai et al., 2006; Mahmood et al., 2014). Our approach, which optimizes performance and energy, is also indirectly improving the long term reliability of the target device.

A first brute-force approach to obtain the best cache configuration would require the execution and evaluation of time and energy for all available cache configurations and target applications, which is an unmanageable task given current time-to-market reduced windows. In addition, execution time and energy consumption are conflicting objectives in practice. For example, if associativity is increased, the number of misses is reduced, as well as the execution time. However, a high associativity increases the hardware complexity and thus the energy consumed by the cache memory (Hennessy and Patterson, 2011). Therefore, we present in this paper a new

* Corresponding author. Tel.: +343947603; fax: +34913947527.
E-mail addresses: mjdiaz@unex.es (J. Díaz Álvarez), jlrisco@dacya.ucm.es (J.L. Risco-Martín), josemanuel.colmenar@urjc.es (J.M. Colmenar).

http://dx.doi.org/10.1016/j.jss.2015.10.012
0164-1212/© 2015 Elsevier Inc. All rights reserved.

methodology to evaluate cache configurations in order to customize cache designs with the aim of reducing both the execution time and the energy consumption by means of a multi-objective optimization (Deb, 2009). In particular, our optimization framework is built around the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb et al., 2002). In order to evaluate our approach, we have automatically designed caches optimized for a set of multimedia applications taken from Mediabench benchmarks (Lee et al., 1997), since they are representative for image, audio and video processing. Our hardware architecture is based on the ARM920T processor (Ltd., 2013), broadly used in multimedia embedded devices.

The rest of the paper is organized as follows. Next section summarizes the related work on the topic. Section 3 describes the design of the search space for our multi-objective optimization. Section 4 shows details of our multi-objective function, describing both performance and energy models. Our optimization framework is integrated and explained in Section 5. Then, Section 6 analyzes our experimental results. In Section 7, we present our conclusions based on the results obtained, and explain the main lines of our future work.

2. Related work

The optimization of performance and energy consumption in the memory subsystem have received a lot of attention in the last decade. Regarding performance, multiple research works have been developed with the aim of improving performance through changing architectural parameters. With respect to energy, previous studies have demonstrated that half of the energy consumption in embedded systems is due to the cache memory (Kandemir, 2006). The optimization of these parameters has been conducted mainly using two different techniques: dynamic reconfiguration and static profiling.

Regarding dynamic reconfiguration, Givargis (2006) improved cache performance by choosing a variable set of bits used as index into the cache. Zhang minimized the energy consumption introducing a new cache design method called way concatenation to reconfigure the cache by software (Zang and Gordon-Ross, 2013). However, this approach provided a limited number of cache configurations, allowing the system engineer to optimize associativity (one-way, two-way or four-way), cache size and line size. Chen et al. (2007) proposed an efficient reconfiguration management algorithm to optimize three parameters: cache size, line size and associativity. Similarly, Gordon-Ross and Vahid (2007) presented a dynamic tuning methodology to optimize cache sizes (2, 4, or 8 KB), line sizes (16, 32, or 64 bytes), and associativity (1-way, 2-way or 4-way). Lopez et al. (2007) proposed an on-line algorithm on *Simultaneous Multithreading (SMT)* to decide the cache configuration after a fixed set of instructions, a technique based on a cache working-set adaptation (Gracia et al., 2014). Dynamic reconfiguration in soft real-time embedded systems on single-level cache hierarchy was proposed by Wang and Mishra (2009), and on a multi-level cache hierarchy in Wang et al. (2011). More recently, Wang et al. (2012) minimize energy consumption in real-time embedded systems performing dynamic analysis during runtime. All these approaches optimize cache size (1, 2 or 4 KB), line-size (16, 32 or 64 bytes) and associativity (1-way, 2-way or 4-way). The main inconvenient of dynamic reconfiguration is the addition of extra complexity in the design of the memory subsystem. We also see that these approaches only optimize a few number of cache parameters, minimizing either execution time or energy consumption. In addition, it is proved in this work that an offline multi-objective optimization may find optimal cache parameter values, without the need of adding hardware complexity to the standard memory subsystem design.

With respect to the use of static profiling, Rackesh Reddy in Reddy and Petrov (2010) studied the effect of multiprogramming workloads on the data cache in a preemptive multitasking environment, and proposed a technique that mapping tasks to different cache partitions, significantly reduced both dynamic and leakage power. Our approach is different, since we try to obtain the behavior of a target set of applications, obtaining their full static profile and the best memory cache configuration (i.e., size, associativity, and replacement and prefetching algorithms for both data and instruction caches) for the whole set. Andrade et al. presented in Andrade et al. (2007) an extension of a systematic analytical modeling technique based on probabilistic miss equations, allowing the automated analysis of the cache behavior for codes with irregular access patterns resulting from indirections. Nevertheless, these models can only optimize cache size and associativity. Feng et al. (2011) applied a new cache replacement policy to perform the replacement decision based on the reuse information of the cache lines and the requested data developing two reuse information predictors: a profile-based static predictor and a runtime predictor. Similarly, Xingyan and Hongyan (2010), based on a profiling scheme of the OPT cache replacement, presented a method to generate best static cache hints. However, these approaches only improve the replacement algorithm. Gordon-Ross et al. (2013) studied the interaction of code reordering and cache configuration, obtaining excellent results. However, this technique is applied to the instruction cache, and our systematic optimization method is applied to the full configuration of both the instruction and data caches.

Additionally, all the aforementioned approaches minimized either execution time or energy consumption. We propose the use of multi-objective optimization to simultaneously minimize both objectives. To this end, we use the concept of multi-objective optimization, which can be easily applied in evolutionary computation. Evolutionary computation and multi-objective optimization are being widely used in *Computer Aided Design (CAD)* problems. Close to cache optimization, Risco-Martín et al. (2008) applied a novel parallel multi-objective evolutionary algorithm to optimize desktop applications for their use in multimedia embedded systems, improving performance, memory usage and energy consumption of the memory subsystem. In Díaz et al. (2009) a simple online *Genetic Algorithm (GA)* was used to obtain the best cache associativity to improve the performance of SMT processors. In this line, Bui et al. (2008) proposed a solution for the cache interference problem applying cache partitioning techniques using a simple GA whose solution sets the size of each cache partition and assigns tasks to partitions such that system worst-case utilization is minimized thus increasing real-time schedulability. An approach based on NSGA-II algorithm was used in Silva-Filho et al. (2008) to evaluate cache configurations on a second cache-level in order to improve energy consumption and performance, optimizing cache size, line size and associativity. However, none of these approaches is able to simultaneously optimize cache performance and energy consumption for a target set of applications as our methodology performs.

To the best of our knowledge none of the previous works tackle the optimization of all the parameters that we propose in this research work. Most of the cited papers focus their space exploration on cache size, line size and associativity, even though the possible values for each configurable parameter is quite small. In this work, we optimize the following cache parameters: cache size, line size, associativity, replacement policy, prefetch policy and write policy. We also consider first-level (L1) instruction/data caches, although the methodology proposed can be applied to other cache types. All the aforementioned configurable parameters complete the chromosome in the *Multi-Objective Evolutionary Algorithm (MOEA)* proposed. The aim is to find the best cache configurations that minimizes memory access time (performance) and energy consumption. As we try to minimize two conflicting objectives, multi-objective optimization is suitable to address this problem. Our approach is valid on embedded systems, where the small number of applications allows the engineer to select one cache design among all the optimizations performed, as we show in this work.

Download English Version:

https://daneshyari.com/en/article/6885542

Download Persian Version:

https://daneshyari.com/article/6885542

Daneshyari.com