



# Multi-objective optimization applied to unified second level cache memory hierarchy tuning aiming at energy and performance optimization



Filipe Rolim Cordeiro\*, Abel Guilhermino da Silva-Filho

Informatic Center, Universidade Federal de Pernambuco, CEP: 50740-540 Recife, PE, Brazil

## ARTICLE INFO

### Article history:

Received 20 January 2014  
Received in revised form 12 January 2016  
Accepted 2 September 2016  
Available online 7 September 2016

### Keywords:

Embedded systems  
Multi-objective optimization  
Energy consumption  
Evolutionary algorithms

## ABSTRACT

Cache memory optimization has an important impact on the energy consumption of the embedded system. However, optimization is a hard task due to the large exploration space and conflicting objectives. In this work five multiobjective optimization techniques are applied to cache memory optimization. The PESA-II, NSGAII, SPEA2, PAES and NPGA approaches were applied to 18 different applications from MiBench and PowerStone benchmark suites. Results compared the quality of results in terms of the metrics of general distance, diversity, hypervolume and precision. All techniques had good performance to cache optimization, but PESA-II showed a better performance for all metrics analyzed, having better results in 83% and 88% of cases, compared with the metrics of generational distance and hypervolume, respectively. Additionally, PESA-II needs to explore only 1.47% of exploration space, finding solutions near to Pareto Optimal.

© 2016 Elsevier B.V. All rights reserved.

## 1. Introduction

One of the main aspects of optimization in recent projects is related to energy consumption of an embedded system [1,2]. It is known that in a microprocessor system, one of the main factors responsible for energy consumption is the cache memory hierarchy, which can consume up to 50% of the energy required by the complete system [3,4]. Therefore, tuning a memory hierarchy has an important impact on a processors total energy, and, consequently, influences directly the energy consumed by the embedded system. The main purpose of a cache subsystem is to provide high performance access to memory, and the cache optimization should not only save energy, but also prevent the depreciation of the applications performance.

Studies reveal that tuning performed on cache memory parameters for a specific application can save on average 60% of the energy consumption [5]. However, each application usually requires a specific kind of architecture and the search for optimal configurations can lead to an elevated cost due to the size of the exploration space. In most commercially used memory hierarchies, with a unified

second level cache memory, the size of the exploration space can involve thousands of configurations, due to the interdependency between the instruction and the data cache [6]. In those cases, finding optimal architecture configurations in the least time possible becomes a big challenge, considering that it is practically unfeasible for the use of exhaustive mechanisms to perform searches in the space project.

Some optimization techniques have been proposed in the last years with the objective of finding a set of optimal solutions, analyzing only a small subregion of the total exploration space. Among those techniques, one that has been successfully applied is the multi-objective approach, whose proposal is to find a set of solutions which are in the optimal region of exploration space. The use of multi-objective optimization algorithms in the area of embedded systems is well applied because it usually presents typical characteristics of problems that can be solved by those algorithms. Among the main characteristics is the search for configurations in a large exploration space, involving project restrictions and aiming to optimize attributes to be defined by the designer. Despite the large number of existing optimization algorithms, few works have been proposed aiming at the optimization of cache memory hierarchy with a unified second level.

In this work, five multi-objective optimization algorithms were applied to the optimization of cache memory with a unified second level. The optimization of the algorithms was done aiming to optimize the energy consumption and the number of cycles necessary

\* Corresponding author.

E-mail addresses: [frc@cin.ufpe.br](mailto:frc@cin.ufpe.br) (F.R. Cordeiro), [agsf@cin.ufpe.br](mailto:agsf@cin.ufpe.br) (A.G. da Silva-Filho).

to execute an application. A set of 18 applications was analyzed to validate the analysis of this work.

## 2. Background

Different strategies have been developed to solve multi-objective problems, such as memory hierarchy optimization. One approach which succeeded for parameters optimization is genetic algorithms (GAs), which was introduced by Holland [7]. This strategy is based on the evolution of a group of individuals, representing the solutions of a problem, which are submitted to a selection procedure to find the best individuals adapted to the problem. The set of the best solutions equally optimal in the exploration space is located in the region called Pareto Optimal, whereas the best solution set found by the algorithm is called Pareto Front. In GAs, each candidate solution is mapped as an individual of a population. The population can create new individuals through genetic recombination and selection, which may generate a new population.

Based on the GA approach, several techniques were proposed in the literature, such as NPGA [8], PAES [9], NSGAI [10], PESA-II [11] and SPEA2 [12], which vary in terms of selection operators and parameters to obtain the Pareto Optimal. In NPGA, the selection mechanism is based on Pareto dominance, where a niche counter indicates the number of neighbor solutions, such that the region density can be used as a selection criterion. In PAES, just one individual is produced at a time and compared with its mutated solution. A grid is used to control the density of best solutions. NSGAI uses a concept of non-dominance to select individuals, separating the solutions in Fronts. The crowding distance metric is used to select the best individuals of a non-dominated solution set. PESA-II uses the concept of hyper grids to separate solutions by density, but unlike PAES, all the population is submitted to the operators of crossover and mutation. SPEA2 also uses a neighbor density strength measure to select the individuals of the best solution set, following the basic sequence of GAs.

In Silva-Filho et al. [13] applies NSGAI algorithm to the problem of memory hierarchy tuning. The technique was applied to a set of 12 applications, trying to optimize the objectives of energy consumption and number of cycles necessary to execute an application. The results obtained were compared with the techniques TECH-CYCLES and TEMGA and it was possible to show good results of the technique to 67% of analyzed cases.

Palermo et al. [14] uses the Discrete Particle Swarm Optimization (DPSO) algorithm for hierarchies of memory with two levels, with unified second level. It is based on the traditional particle swarm optimization (PSO) algorithm, being adapted to the problem of cache memory. In the results, the DPSO is compared only with the exhaustive approach, accelerating the search mechanism by 5 times with a precision of 70%.

Gordon-Ross et al. [6] presents the ACE-AWT heuristic, also for cache with unified second level, aiming to reduce energy consumption. This approach is based on the concatenation method, which allows the banks of memory to be logically concatenated, allowing

different configurations of memory hierarchies. Results were compared with the SERP heuristic, achieving a 61% energy reduction.

In this work the techniques NPGA, PAES, NSGAI, PESA-II and SPEA2 were applied, which were selected based on a previous study of the multi-objective approaches most used in the literature. The techniques were adapted to optimize cache parameters of cache hierarchy with unified second level.

## 3. Experimental environment

### 3.1. Architecture specification

The architecture used in this work is composed of a MIPS processor, level 1 instruction cache (IC), level 1 data memory (DM), level 2 unified cache (L2) and a main memory (MEM), such as shown in Fig. 1. It also uses an input tension of 1.7 V, with write-through write policy and transistor technology of 70 nm.

Commercial cache configurations that are used in embedded systems applications were adapted to the exploration space. The parameters used to adjust the architecture configuration were cache size, line size and associativity, for each cache component. The ranges of the parameters used are shown in Table 1.

As shown in Table 1, each parameter of a cache component varies in a range of three values. The full set of combinations of Table 1, which represents the exploration space, is composed of 9084 valid configurations of cache hierarchy. During the exploration process of the optimization techniques, the cache parameters are tuned, trying to find the cache configuration which has the best tradeoff between consumed energy and cycles necessary to run each application.

In this work, we used 18 applications from MiBench [15] and Power Stone [16] benchmarks. The applications involve different areas, allowing the validation of the optimization techniques in different exploration contexts. The architecture behavior changes in relation to energy consumed and cycles, depending on which application is running. The values for total energy consumption and number of cycles necessary to run each application to a specific configuration were obtained through the use of the tools SimpleScalar [17] and eCACTI [18]. The cache memory energy model used supports both energy components: static and dynamic.

### 3.2. Metrics

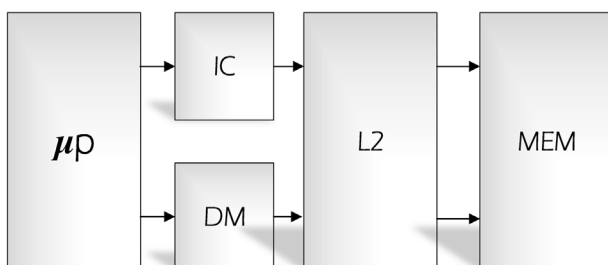
Metrics are used to measure characteristics of each multi-objective algorithm, helping to understand its behavior and allowing more concrete evaluations of the performance of the algorithm. The metrics are also an important parameter of comparison between algorithms, since many times it is hard to notice which algorithm presents a better solution set for a problem. The metrics used in this work are described below.

The metric of generational distance was proposed by Van Veldhuizen and Lamont in [19] and it is used to measure the Euclidean distance between the solutions. The generational distance is calculated by the following equation:

$$GD = \frac{\sqrt{\sum_{i=1}^s d_i^2}}{s}, \quad (1)$$

**Table 1**  
Range of cache parameters.

Parameter	Level one cache	Level two cache
Cache size	2KB, 4KB, 8KB	16KB, 32KB, 64KB
Line size	8B, 16B, 32B	8B, 16B, 32B
Associativity	1, 2, 4	1, 2, 4



**Fig. 1.** Cache memory hierarchy.

Download English Version:

<https://daneshyari.com/en/article/4963586>

Download Persian Version:

<https://daneshyari.com/article/4963586>

[Daneshyari.com](https://daneshyari.com)