Microprocessors and Microsystems 000 (2016) 1–13

Contents lists available at ScienceDirect

Microprocessors and Microsystems

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

A pareto-optimal runtime power budgeting scheme for many-core systems☆

Xiaohang Wang[a,*], Baoxin Zhao[b], Ling Wang[c], Terrence Mak[d], Mei Yang[e], Yingtao Jiang[e], Masoud Daneshtalab[f]

[a] South China University of Technology, China
[b] Shenzhen Institute of Advanced Technology, CAS, China
[c] Harbin Institute of Technology, China
[d] Southampton University, UK and Guangzhou Institute of Advanced Technology, CAS, China
[e] University of Nevada, Las Vegas, USA
[f] KTH Royal Institute of Technology, Sweden

ARTICLE INFO

Article history:
Received 14 October 2015
Revised 22 January 2016
Accepted 3 March 2016
Available online xxx

Keywords:
Power budgeting
Many-core systems
Dynamic programming

ABSTRACT

Due to the ever-escalating power consumption, a significant proportion of the future many-core chips is mandatory to be switched off to meet the power budgets. This trend has brought up a paradigm shift from conventional low-power to power budgeting designs, where performance optimization needs to be performed under a tight power budget constraint. There are two key issues to be considered when moving this new design paradigm forward. Firstly, with per-core frequency scaling, the number of frequency combinations of the cores grows exponentially. As more cores are integrated onto a chip, it becomes more challenging to achieve the optimal performance over a given power budget. Secondly, the power budgets of many-core system might undergo a rapid fluctuation. Consequently, the power budgeting scheme needs to be prompt to make appropriate changes to track such power budget variation. This paper is aiming at resolving the problem of optimizing overall performance over a power budget using frequency scaling technique. To solve the problem efficiently at runtime, we propose a parallel dynamic programming network, in which the Pareto-optimal solutions can be obtained using linear time complexity. Experimental results have confirmed that the proposed approach can reduce the execution time by 45% when compared to other existing methods. The runtime overhead and hardware cost of the proposed approach are reasonably small, such as the average area and power consumption are less than 1% of the whole network-on-chip. This paper demonstrates an effective formulation for delivering Pareto-optimal solutions for power budgeting in future many-core systems.

1. Introduction

Driven by the continuous advances of the CMOS technology and requests of emerging applications, multi-core and many-core chips are widely used in cloud computing, mobile computing, high-performance computing, as well as many other important areas [1]. However, this explosive growth of computation power comes with a rapid increase in power consumption. According to the international technology roadmap for semiconductors (ITRS), by the year 2020, the chip power consumption will increase by a factor of 10 over that in the year 2012 [1]. This level of power consumption will be much higher than the affordable power budget (maximum power supply or thermal-compliant power budget), as clearly shown in Fig. 1(a). Consequently, the real performance of many-core chips is far below their peak (theoretical) performance (see Fig. 1(b)). For instance, although transistor count will increase by a factor of 32 in 2020, the performance speedup can only be 7 ~ 8 folds, according to a study indicated in [2]. This wide performance gap is caused by the "dark silicon" phenomenon [2] that a large proportion of a chip then has to be powered off so that it can still confine to its given power budget. This "dark silicon" problem

☆ This research program is supported by the Natural Science Foundation of China No. 61376024, 61306024 and 61176025, Natural Science Foundation of Guangdong Province No. S2013040014366 and 2015A030313743, and Basic Research Programme of Shenzhen No. JCYJ20140417113430642 and JCYJ20140901003939020, and Special Program for Applied Research on Super Computation of the NSFC-Guangdong Joint Fund (the second phase).

* Corresponding author. Tel.: +86 150 1418 2716
E-mail addresses: baikeina@163.com, xiaohangwang@scut.edu.cn (X. Wang), bx.zhao@giat.ac.cn (B. Zhao), ling.wang@giat.ac.cn (L. Wang), tmak@ecs.soton.ac.uk (T. Mak), mei.yang@unlv.edu (M. Yang), yingtao@egr.unlv.edu (Y. Jiang), masdan@kth.se (M. Daneshtalab).

http://dx.doi.org/10.1016/j.micpro.2016.03.006
0141-9331/

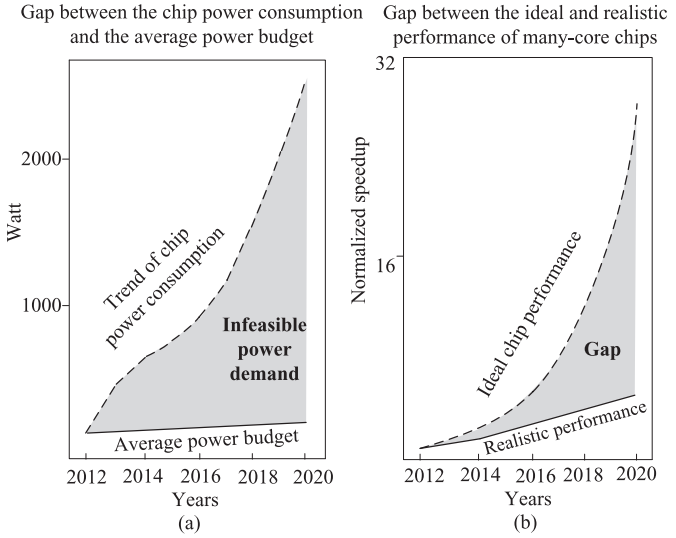

Fig. 1. (a) Widening gap between the power consumption of chips and the power budget that could be delivered [1]. (b) Gap between ideal and realistic performance of many-core chips [2].

becomes more prevalent at 8 nm, where half of the chip may have to be powered off at any given time [2].

These power and performance gaps, as illustrated in Fig. 1, bring up a paradigm shift from conventional low-power to power budgeting designs [3–7] with an objective to optimize performance under a limited power budget. Power budgeting problem aims at optimizing the overall performance under a power limit, a vast departure from the conventional online power management approaches [8–15] whose goals are to minimize the power consumption or temperature under a performance constraint (*e.g.*, deadline satisfaction). In the literature, online power budgeting can be achieved by applying various circuit and/or architectural techniques, including frequency and/or voltage scaling [3,16–19], power gating to ensure that the number of working resources at a time is kept on check [20,21], dynamic cache resizing [16], and dynamic pipeline reconfiguration [22].

Frequency and/or voltage scaling can be performed either at the chip or at the core level. Chip-wide frequency scaling [23] treats the frequencies of the on-chip cores as one variable, and these frequencies are all scaled up/down together with the same scaling factor. This rigid scheme, although simple, tends to result in quite poor chip performance. To improve the power control granularity, per-core frequency scaling and power gating are preferred. For example, [21] proposed a method where the number of cores that can be powered on is coarsely determined, after which fine-grained frequency adjustment is followed. Instead of dealing with cores as [21] does, the schemes proposed in [3,16] even look down to the components like the last level cache and network-on-chip. Cache resizing and frequency scaling are used for cache power budgeting [16], and both frequency scaling and sub-network power gating are used for network-on-chip power budgeting [3,24].

Further performance improvement is possible by exploring *a priori* knowledge of the application behaviors. In this case, performance-power models are estimated either online or offline for each application, followed by online power adjustment [20,25]. Applications are categorized and controlled according to their respective degree of parallelism, *i.e.*, applications of high parallelism will be re-mapped to more cores running at lower frequency, while applications of a low degree of parallelism to smaller number of cores but at higher frequency, given a power budget [26].

The problem of performance optimization under power budget is solved either using some sub-optimal control [3,8,26],

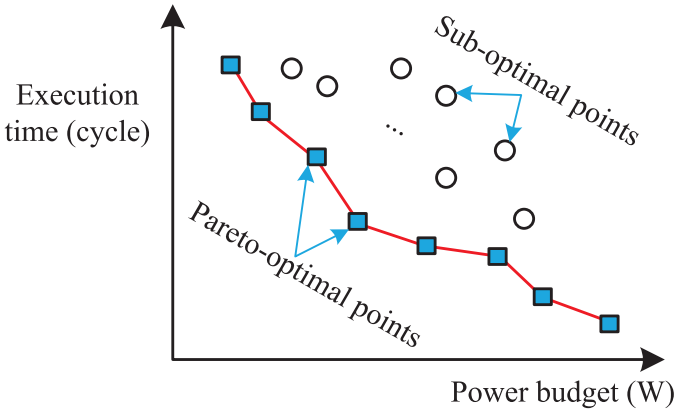

Fig. 2. The objective of the power budgeting scheme is to find the Pareto frontier, where the Pareto-optimal points can generate best performance given the power budgets.

formulating it as a linear programming [16], or following heuristic approaches [17,26,27]. Sub-optimal and heuristic approaches do not generate optimal solutions and their performance can degrade severely with the increase of the number of resources to be controlled; chip performance can be even worsened when the behavior of the applications becomes more complex. The linear programming-based approach might take long time to find a solution, and thus it is not able to track the rapid change of power budget. Another big problem of all these aforementioned methods is that they dedicate a master processor explicitly or implicitly to find the solutions in a centralized way, which scales poorly and consumes excess power for the purpose of power control. This scaling problem has inspired us to seek a solution using dynamic programming that will be detailed in the following sections.

Besides, the power budgeting problem is complicated due to the following challenges observed in many-core systems.

First and foremost, there is a need of achieving the optimal performance over a given power budget. An optimal solution will be a minimum execution time over a given power budget, referred as a *Pareto-optimal* solution, as in Fig. 2. Those Pareto-optimal points are optimal solutions for multiple objectives in terms of both performance and power. Given a power budget, Pareto-optimal points correspond to the best performance that can be possibly achieved in a many-core system. In contrast, performance of the sub-optimal solutions represented as circle points in Fig. 2 are worse than that of their corresponding Pareto-optimal points. In simple words, our goal here is to find the Pareto-optimal points for the given power budgets.

However, in a state-of-the-art many-core system, many of its cores' (or IPs') operating frequencies are tunable. For example, consider a 64-core system, where each core can operate at one of the four available frequencies. The total number of frequency combinations is as large as 4^{64}, which is around 3×10^{38}. Heuristic-based methods [20,21,25,26] converge to sub-optimal solutions given such a vast solution space, which will cause performance degradation in the many-core systems.

Second, there is a need of prompt control (low runtime overhead) to track and then respond to the rapid fluctuation of power budget. This is because power budget might change quickly and abruptly, due to a number of reasons, including IR drop, power supply noise [28], temperature emergency [29], or change in external power supply due to the scheduling or power regulation policy [30]. Unfortunately, most existing methods [20,21,25,26], as they involve a long run time, are unable to track the rapid variation in power budget.

To address the above mentioned challenges, a power budgeting scheme, optimal power budgeting using dynamic programming

Please cite this article as: X. Wang et al., A pareto-optimal runtime power budgeting scheme for many-core systems, Microprocessors and Microsystems (2016), http://dx.doi.org/10.1016/j.micpro.2016.03.006