J. Parallel Distrib. Comput. 87 (2016) 43–54

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

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

Application configuration selection for energy-efficient execution on multicore systems

Shinan Wang [a], Bing Luo [a,*], Weisong Shi [a], Devesh Tiwari [b]

[a] Department of Computer Science, Wayne State University, United States
[b] Oak Ridge National Lab, United States

HIGHLIGHTS

- We present a hybrid method to achieve an energy efficiency configuration.
- Our method utilizes concurrency levels, thread allocation, and DVFS settings.
- We propose a model to capture the relationship between C, P, and T in detail.
- We apply an analytical speedup model to predict an optimal/nearoptimal configuration.

ARTICLE INFO

Article history:
Received 3 March 2015
Received in revised form
9 September 2015
Accepted 11 September 2015
Available online 21 September 2015

Keywords:
Energy consumption
High performance computing
Speedup model
Power model
Parallel

ABSTRACT

Modern computer systems are designed to balance performance and energy consumption. Several run-time factors, such as concurrency levels, thread mapping strategies, and dynamic voltage and frequency scaling (DVFS) should be considered in order to achieve optimal energy efficiency for a workload. Selecting appropriate run-time factors, however, is one of the most challenging tasks because the run-time factors are architecture-specific and workload-specific.

While most existing works concentrate on either static analysis of the workload or run-time prediction results, in this paper, we present a hybrid two-step method that utilizes concurrency levels and DVFS settings to achieve the energy efficiency configuration for a workload. The experimental results based on a Xeon E5620 server with NPB and PARSEC benchmark suites show that the model is able to predict the energy efficient configuration accurately. On average, an additional 10% EDP (Energy Delay Product) saving is obtained by using run-time DVFS for the entire system. An off-line optimal solution is used to compare with the proposed scheme. The experimental results show that the average extra EDP saved by the optimal solution is within 5% on selective parallel benchmarks.

1. Introduction

The focus of computing has shifted from performance-centered to energy efficiency. As a result, energy-efficient techniques have been adopted across different layers in almost every system, from single chips to large data centers [32,36]. Power dissipation and energy consumption are priority concerns when designing computer systems, especially in the High Performance Computing (HPC) field. A recent article suggests that the benefits of the multi-core architecture diminishes as the power constraint on a chip rises [14].

Generally, there are two major factors that affect energy consumption for a specific workload: execution time and average power dissipation. Speedup models are used to describe the benefits introduced by parallel implementations in terms of execution time, while power models are used to estimate power dissipation of a workload. Energy efficiency can be defined as the workload over the required energy, which in turn is equal to $\frac{W}{(P_{AI}+\sum_i^C P_t)T}$ [40], where C is concurrency level, P_{AI} and P_t denote active idle power of a system and average power dissipation of each thread, respectively, and T is execution time. A concurrency level with a thread mapping strategy is referred to a configuration in the rest of the paper.

In-depth analysis of these three factors, C, P, and T, is necessary to achieve better energy efficiency. For example, a speedup model is usually used to quantify the benefits introduced by

* Corresponding author.
E-mail addresses: ez3716@wayne.edu (S. Wang), ez6913@wayne.edu (B. Luo), weisong@wayne.edu (W. Shi), tiwari@ornl.gov (D. Tiwari).

http://dx.doi.org/10.1016/j.jpdc.2015.09.003
0743-7315/

parallel computing in terms of execution time [27]. Higher concurrency levels, however, affect power dissipation (P) because not only additional computing units are activated but also the power dissipation of common components on a chip will be shared by more cores. An analytical model is needed to understand the energy efficiency of a workload in a multi-core computing scenario. While allocating a workload to multiple CPUs is an effective way to reduce computation energy consumption, DVFS is usually used to explore slacks during execution to save extra power dissipation [20].

Workload characteristics and micro-architectures have major influence on the three factors. The speedup factor of a workload is closely related to the serial portion of different programs [16,6]. In addition, memory boundedness affects the scalability of a workload in the sense that an individual thread or a process competes for off-chip resources with other threads so that concurrency hazards, such as false sharing, might occur [13]. That information cannot be exposed without run-time profiling. On the other side of the spectrum, modern computer systems deploy different mechanisms to improve memory performance. For example, Intel processors use Quickpath technology [43] as an implementation of NUMA architecture.

An empirical model is used to predict the configuration as the second step, and voltage/frequency levels of a workload [38,33] by using Performance Monitoring Counters (PMCs). However, one of the major drawbacks of using an empirical model is architecture dependency, which requires different sets of PMCs to be used for different architectures. Ge et al. propose an energy-performance estimation using an analytical model [17]. However, the model mainly analyzes the behavior of a multi-core based power aware system by case studies. No prediction is used to select the appropriate configuration for each workload.

In this paper, we propose an approach to predict the appropriate configuration of a workload for energy efficiency purposes. First, we propose an analytical speedup model that utilizes PMCs to predict potential speedup of various configurations from two threads execution information. The collected information is used to build the power estimation of various concurrency levels. Once the optimal concurrency level is selected, we apply a run-time DVFS to select an appropriate frequency for each phase. The contributions of this paper include:

- The approach proposed in this paper, rather than training hundreds of PMCs to find the best fit PMCs suite, utilizes only four PMCs based on workload analytical model. The four PMCs are available in most of modern platforms.
- Our model carefully captures the relationship between C (Concurrency Level), P(Power Dissipation), and T(Execution Time), so that we are able to use execution information that using two threads to predict the energy consumption of different configurations on a specific architecture.
- To predict the optimal/near-optimal configuration of a parallel workload, we apply a speedup model. Unlike an empirical model, we reserve the applicability of the model on different architectures by using an analytical model.
- We observe that some parallel applications may have up to 87% of serial part making the power consumption grow more than benefits obtained by increasing the number of threads.

The rest of the paper is organized as follows: we start the paper by introducing our observation between workload concurrency level and execution time/power in Section 2; then, we present a two-step prediction model in Section 3, followed by evaluation of the prediction model on a Intel Xeon E5620 platform in Section 4. Related work is discussed in Section 5. Finally, we summarize the paper in Section 6.

2. Observation

In a multi-core or many-core system, the scalability and power dissipation of a workload are closely related to the system architecture and workload characteristics. The execution time could be considered as infinity if no computation unit is involved. As the core number and thread number increase, execution time drops since more computation power is involved. The lower bound of the execution time, however, is limited by two factors, namely the serial portion of the workload and the off-chip resources. As a result, the execution time of a workload approaches its lower bound and even slowly rises as a system allocating more cores to the workload. On the other hand, a system power dissipation increases as the workload occupies more cores. The system consumes idle power if no computation is invoked. Although the system power dissipation increases, it is bounded by the Thermal Design Power (TDP). As a result, the speedup bound and power bound can be modeled using mathematical equations.

Specifically, we model the speedup and power dissipation as a function of the number of threads utilized as Eqs. (1) and (2), respectively. In particular, we derive Eq. (2) from the logistic function, which has a maximum value β_1 and exponential growth. This scenario corresponds to the fact that the system power is bounded by the design. $\frac{\beta_1}{1+\exp^{\beta_2}}$ represents the system idle power when there are no cores engaged for the workload.

$$T = f(C) = \alpha_1 + \alpha_2 \times C^{\alpha_3} + \alpha_4 \times C^{-\alpha_3} \quad (\alpha_3 > 0) \tag{1}$$

$$P = g(C) = \frac{\beta_1}{1 + \exp^{\beta_2 - \beta_3 \times C}}. \tag{2}$$

Fig. 1 shows the model fitting results for NPB and PARSEC benchmarks. The solid line represents the proposed model while the dots are actual measurements. It is easy to observe that all the benchmarks present a similar pattern, which can be captured by the mathematical models. As the number of threads increases, the workload speedup is limited by the off-chip resources such as cache and memory. On the other hand, the power dissipation, starting from idle power when no thread is running, grows to meet the maximum bound. The mathematical models carefully capture the features of concurrency level and its relationship to the power and speedup. Particularly, the execution time and power dissipation of raytrace benchmark fall into a small range because of the large serial portion of the benchmark. Our model is able to capture this unusual case as well.

We derive models to estimate speedup and power dissipation under various configurations for a workload in order to achieve the best energy efficiency and energy delay product.

3. Model derivation

In the first step, we focus on concurrency levels (C), power dissipation of the selected concurrency level (P), and execution time (T). By increasing the concurrency level of an application, the average power dissipation increases while the execution time required to finish the same work shrinks. Usually the speedup benefited from a higher concurrency is substantial so that the total energy consumption drops. However, a large number of threads in a system usually results in a competing situation, especially regarding front bus usage. Although NUMA [7] is proposed to solve this problem, off-chip activities affect speedup dramatically. Thread mapping is another technique that is proposed to reduce contention. However, the effects of thread mapping on energy efficiency are uncertain [19,8]. Although a compact scheme (allocating threads to as less physical processors as possible) generates less power dissipation, this scheme tends to reduce speedup. Scatter scheme (allocating threads evenly to different physical processors), on the other hand, generates higher power dissipation but alleviates contention.