Sustainable Computing: Informatics and Systems xxx (2017) xxx–xxx

Contents lists available at ScienceDirect

Sustainable Computing: Informatics and Systems

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

Performance and energy metrics for multi-threaded applications on DVFS processors

Thomas Rauber[a,*], Gudula Rünger[b], Matthias Stachowski[a]

[a] University Bayreuth, Germany
[b] Chemnitz University of Technology, Germany

ARTICLE INFO

Article history:
Received 22 July 2016
Received in revised form 14 June 2017
Accepted 31 October 2017
Available online xxx

Keywords:
Energy
DVFS
Metrics
Multithreading
Performance
PARSEC
SPLASH-2
Intel Core i7
ARM

ABSTRACT

Due to their internal execution characteristics, application programs exploit the hardware very differently, which leads to a quite diverse behavior concerning their performance or the energy consumed for their execution. A change of the operational frequency of DVFS processors leads to further variations in performance and energy consumption, as does the exploitation of thread parallelism on multicores. This article combines frequency scaling and thread-parallelism and considers several new metrics for the evaluation of an application's performance and energy consumption. As application programs, the PARSEC benchmark suite and the SPLASH-2 benchmark suite are investigated. The PARSEC benchmark suite provides an up-to-date collection of applications with different workloads on chip-multiprocessors. The SPLASH-2 is a common suite for scientific studies on parallel shared memory machines. Intel Core i7 processors are used as hardware platforms for the evaluation.

1. Introduction

A low energy consumption as well as a good performance are important properties of today's application codes. These properties are influenced by the software structure, the execution characteristics, and the way in which the code exploits the hardware details of the execution platform. The internal organization of the processor architecture may have a large influence on the resulting runtime and energy consumption, depending on how well the operations defined by the application code can be mapped onto the functional units of the processor. Also, recent processors provide features, such as chip-multiprocessing or dynamic voltage frequency scaling (DVFS), which can be exploited to achieve a better performance or energy balance. However, it is not a priori predictable whether the use of more threads leads to a smaller execution time of an application and whether the use of a smaller operational frequency leads to a reduction in the energy consumption. Instead, for a given application there is typically an optimal number of threads beyond which the execution time cannot be reduced further. Similarly, there is often an optimal operational frequency beyond which a further frequency reduction does not lead to a further energy reduction.

In this article, we study the combination and interaction of energy efficiency and runtime performance for multithreaded applications on multicore DVFS processors. We consider the dependence of energy consumption as well as runtime performance from both, the number of threads used and the operational frequency chosen. The runtime performance usually improves with an increasing number of threads but decreases with a down-scaling of the frequency. Also the energy used for an application varies when varying the number of threads or when scaling the frequencies. And although the energy consumed depends on the execution time, it can be observed that the lowest execution time achieved for a specific setting of number of threads and operational frequency does not necessarily lead to the lowest energy consumption. This is caused by an application-specific power consumption.

The diverse interactions of the effects of a varying number of threads and varying frequencies on the runtime performance and the energy consumption of applications can be quite intricate. Thus, the question arises whether appropriate metrics can be defined which can capture the dependence of performance and energy consumption on the two varying parameters, i.e., the number of threads and the operational frequency, and how an improvement effect can be expressed. Since both runtime performance and energy

[*] Corresponding author.
E-mail addresses: rauber@uni-bayreuth.de (T. Rauber), matthias.stachowski@uni-bayreuth.de (M. Stachowski).

https://doi.org/10.1016/j.suscom.2017.10.015
2210-5379/© 2017 Elsevier Inc. All rights reserved.

consumption are considered together, a metrics combining them is needed. In this article, we propose several metrics, including speedup and reduction factors for the execution time and the energy in isolation as well as combined metrics. All these metrics depend on two variables, which are the number of threads and the frequency. Also, metrics for the power consumption are introduced with which application-specific power values can be assessed.

The Princeton Application Repository for Shared-Memory Computers (PARSEC) [6] is a collection of multithreaded benchmarks with different parallel workloads and execution characteristics. PARSEC provides programs from a wide range of applications based on the latest techniques in the specific domains and with an emphasis on large workloads as well as multicore parallelism. The contribution concerning the PARSEC benchmarks has two aspects. First, we demonstrate and evaluate our metrics with respect to their expressiveness and validity. Second, we provide an intensive study of the performance and energy characteristics of the PARSEC benchmarks on recent Intel multicore processors that adds another study to the investigation of the PARSEC benchmarks. The results concerning energy might be valuable for users of the benchmark suite.

The SPLASH-2 benchmark suite [32] is a popular benchmark suite comprising eleven applications mainly from scientific computing and graphics, which were the main application areas for parallel programming when the SPLASH-2 benchmark suite has been assembled. The benchmarks have been chosen and optimized for parallel shared memory machines of the 90th. Thus, the optimizations naturally address different hardware features then are now available in recent multicore architectures. Nevertheless, the programming model of the SPASH-2 benchmarks is suitable for multicore programming and it is worthwhile to study these application codes with respect to multicore parallelism and frequency scaling, which will gives insight into the behavior of scientific applications, which are usually used for a longer period, on recent architectures.

The contribution of this article comprises an intensive empirical investigation of the performance and the energy behavior of multithreaded applications with respect to varying numbers of cores and varying frequencies on recent multicore DVFS processors. As hardware platform, we use two Intel Core i7 processors from the 4th (Haswell) and the 6th (Skylake) generation, which both provide frequency scaling, and an ARM big.LITTLE architecture with an Samsung Exynos5422 Cortex A15 2.0 GHz quadcore as well as a Cortex 1.4 GHz A7 quadcore CPU, provided by an Odroid XU4 Board. The study gives insight into the performance and energy characteristics of the PARSEC and the SPLASH-2 benchmarks, thus providing results for application optimized for recent as well as for older parallel shared memory machines. The investigations and evaluations have been done with the help of well-known and also newer metrics capturing both criteria, energy and performance.

The rest of the article is structured as follows: Section 2 describes the energy model and the evaluation strategy used. Section 3 presents the new metrics. Section 4 evaluates of the PARSEC and SPLASH-2 benchmarks with these metrics and discusses the results. Section 5 discusses related work and Section 6 concludes the article.

2. Multivariable energy model

The execution time T [s] of an application code varies with the number of threads p used for the execution and the operational frequency f chosen, as does the power drawing P [Watt]. To express the dependency of the energy consumption $E = T \cdot P$ [Joule] on the number of threads and the operational frequency, we develop an energy model which explicitly uses the two variables p and f.

2.1. Energy model for frequency scaling

When considering the energy consumption of application programs for DVFS processors, it is useful to introduce frequency scaling factors as follows: the frequency scaling for a DVFS processor is expressed by a dimensionless scaling factor $s \geq 1$ which describes a smaller frequency $\tilde{f} < f_{\max}$ as $\tilde{f} = f_{\max}/s$ where f_{max} is the maximum frequency possible for the processor.

Power models for DVFS processors distinguish the dynamic power consumption and the static power consumption. The dynamic power consumption $P_{dyn}(f)$ is related to the supply voltage and the switching activity during the computing activity of the processor and can be expressed by $P_{dyn}(f) = \alpha \cdot C_L \cdot V^2 \cdot f$ where α is the switching probability, C_L is the load capacitance, and V is the supply voltage. The frequency f depends linearly on the supply voltage V, i.e., $V = \beta \cdot f$ with some appropriate constant β. Thus, the dependence of the dynamic power consumption on the frequency f can be expressed as $P_{dyn}(f) = \gamma \cdot f^3$ with $\gamma = \alpha \cdot C_L \cdot \beta^2$ or when using the corresponding scaling factor s as $P_{dyn}(s) = s^{-3} \cdot P_{dyn}(1)$ where $P_{dyn}(1)$ is the dynamic power consumption in the un-scaled case.

The static power consumption $P_{stat}(f)$ is intended to capture the leakage power consumption as well as the power consumption of peripheral devices and can be expressed by $P_{stat}(f) = V \cdot N \cdot k_{\text{design}} \cdot I_{\text{leak}}$, where N is the number of transistors, k_{design} is a design dependent parameter, and I_{leak} is a technology-dependent parameter [7]. Again using $V = \beta \cdot f$ leads to a linear dependence of the static power on f, i.e., $P_{stat}(f) = \delta \cdot f$ with $\delta = N \cdot k_{\text{design}} \cdot I_{\text{leak}} \cdot \beta$ or $P_{stat}(s) = s \cdot P_{stat}(1)$ where $P_{stat}(1)$ is the static power consumption in the un-scaled case. Other authors have also proposed to make the simplified assumption that P_{static} is independent of the voltage or frequency scaling [33], i.e., $P_{stat}(s) = P_{stat}(1) = P_{static}$. In the following, we use this assumption. The total power consumption includes both power components $P_{dyn}(s)$ and P_{static}.

Reducing the operational frequency of a processor by a scaling factor of s usually decreases the power consumption, however, it increases the execution time $T(1)$ of an application program by the same factor compared to an un-scaled execution, i.e., $T(s) = s \cdot T(1)$. Using the power and the execution time depending on the scaling used for the execution of the application program leads to an energy model depending on the scaling factor s:

$$\begin{aligned} E(s) &= (P_{dyn}(s) + P_{static}) \cdot s \cdot T(1) \\ &= (s^{-3} \cdot P_{dyn}(1) + P_{static}) \cdot s \cdot T(1) \qquad (1) \\ &= (s^{-2} \cdot P_{dyn}(1) + s \cdot P_{static}) \cdot T(1) \end{aligned}$$

2.2. Energy model for parallel execution

The execution of multithreaded programs introduces a further variable into the model, which is the number of threads used for a specific code execution. The execution time usually decreases with an increasing number of threads (typically in a non-linear way) until a saturation point is reached, which strongly depends on the application.

On the other hand, also the power drawing varies with the number of threads, usually in a way that the power drawing increases with an increasing number of threads. Table 1 shows the power consumption and Fig. 4 illustrates the energy consumption impacts of multithreading and frequency scaling for the example applications swaption and blackscholtes from the PARSEC suite. The table shows that the power drawing increases with the frequency for a fixed number of threads. For a fixed frequency, the power drawing increases up to four threads, corresponding to the number of physical cores, and then it may increase further or may slightly decrease. The table shows the enormous differences in the power

Please cite this article in press as: T. Rauber, et al., Performance and energy metrics for multi-threaded applications on DVFS processors, Sustain. Comput.: Inform. Syst. (2017), https://doi.org/10.1016/j.suscom.2017.10.015