Energy Consumption Modeling for Hybrid Computing

Ami Marowka

Department of Computer Science
Bar-Ilan University, Israel
amimar2@yahoo.com

Abstract. Energy efficiency is increasingly critical for embedded systems and mobile devices, where their continuous operation is based on battery life. In order to increase energy efficiency, chip manufacturers are developing heterogeneous CMP chips.

We present analytical models based on an energy consumption metric to analyze the different performance gains and energy consumption of various architectural design choices for hybrid CPU-GPU chips. We also analyzed the power consumption implications of different processing modes and various chip configurations. The analysis shows clearly that greater parallelism is the most important factor affecting energy saving.

Keywords: Analytical model, CPU-GPU architecture, Performance, Power estimation, Energy.

1 Introduction

Energy efficiency is one of the most challenging problems confronting multi-core architecture designers. Future multi-core processors will have to manage their computing resources while maintaining their power consumption within a power budget. This constraint is forcing the microprocessor designers to develop new computer architectures that deliver better performance per watt rather than simply yielding higher sustainable performance.

Recent research shows that integrated CPU-GPU processors have the potential to deliver more energy efficient computations, which is encouraging chip manufacturers to reconsider the benefits of heterogeneous parallel computing. The integration of CPU and DSP cores on a single chip has provided an attractive solution for the mobile and embedded market segments, and a similar direction for CPU-GPU computing appears to be an obvious move. It is known that the integration of thin cores and fat cores on a single processor achieves a better performance gain per watt. For example, a study of analytical models of various heterogeneous multi-core processor configurations found that the integration of many simplified cores in a single complex core achieved greater speedup and energy efficiency when compared with homogeneous simplified cores [1]. Thus, it is generally agreed that a heterogeneous chip integrating different core architectures, such as CPU and GPU, on a single die is the most promising

C. Kaklamanis et al. (Eds.): Euro-Par 2012, LNCS 7484, pp. 54–64, 2012.

technology [2–5]. Chip manufacturers such as Intel, NIVIDIA, and AMD have already announced such architectures, i.e., Intel Sandy Bridge, AMD's Fusion APUs, and NVIDIA's Project Denver.

Despite some criticisms [6, 7] Amdahl's Law [8] is still relevant as we enter a heterogeneous multi-core computing era. Amdahl's Law is a simple analytical model that helps developers to evaluate the actual speedup that can be achieved using a parallel program. However, the future relevance of the law requires its extension by the inclusion of constraints and architectural trends demanded by modern multiprocessor chips. Here, we extend a study conducted by Woo and Lee [1] and apply it to the case of hybrid CPU-GPU multi-core processors.

We investigate how energy efficiency and scalability are affected by the power constraints imposed on modern CPU-GPU based heterogeneous processors. We present analytical models that extend Amdahl's Law by accounting for energy limitations and we analyze the three processing modes available for heterogeneous computing, i.e., symmetric, asymmetric, and simultaneous asymmetric.

The rest of this paper is organized as follows. Section 2 presents an analytical model of a symmetric multi-core processor that reformulates Amdahl's Law to capture power constraints. In Section 3 we continue by applying energy constraints to an analytical model of an asymmetric processor. In Section 4 we study how performance and power consumption are affected by simultaneous asymmetric processing. In Section 5 we compare the three analytical models. Section 6 presents related works and Section 7 concludes the paper.

2 Symmetric Processors

In this section we reformulate Amdahl's Law to capture the necessary changes imposed by power constraints. We start with the traditional definition of a symmetric multi-core processor and continue by applying energy constraints to the equations following the method of Woo and Lee [4].

2.1 Symmetric Speedup

Amdahl's law posts an upper limit on the *symmetric speedup* ($speedup_s$) that can be achieved by parallelization of a symmetric multi-core processor, as follows:

$$Speedup_s = \frac{1}{(1-f) + \frac{f}{c}} \tag{1}$$

where c is the number of cores, and f is the fraction of a program's execution time that is parallelizable ($0 \leq f \leq 1$).

2.2 Symmetric Performance per Watt

To model power consumption in realistic scenarios, we introduce the variable k_c to represent the fraction of power a single CPU core consumes in its idle state

$(0 \leq k_c \leq 1)$. In the case of a symmetric processor, one core is active during the sequential computation and consumes a power of 1, while the remaining $(c-1)$ CPU-cores consume $(c-1)k_c$. During the sequential computation period, the processor consumes a power of $1 + (n-1)k_c$. Thus, during the parallel computation time period, c CPU-cores consume c power. It requires $(1-f)$ and f/c to execute the sequential and parallel codes respectively, so the formula for the average power consumption W_s of a symmetric processor is as follows.

$$W_s = \frac{(1-f)\cdot\{1+(c-1)k_c\}+\frac{f}{c}\cdot c}{(1-f)+\frac{f}{c}} = \frac{1+(c-1)k_c(1-f)}{(1-f)+\frac{f}{c}} \tag{2}$$

Next, we define the *performance per watt (Perf/W)* metric to represent the amount of performance that can be obtained from 1 W of power. The *Perf* of a single CPU-core execution is 1, so the $Perf/W_s$ achievable for a symmetric processor is formulated as follows.

$$\frac{Perf}{W_s} = \frac{Speedup_s}{W_s} = \frac{1}{1+(c-1)k_c(1-f)} \tag{3}$$

2.3 Symmetric Performance Per Joule

The definition of $Perf/W$ metric allows us to evaluate the performance achievable by a derived unit of power (watt). Power is the rate at which energy is converted, so we can define a *Performance per Joule (Perf/J)* metric where the joule is the derived unit of energy, representing the amount of performance stored in an electrical battery. The $Perf/J$ of a single CPU-core execution is 1, so the $Perf/J_s$ achievable by a symmetric processor is formulated as follows.

$$\frac{Perf}{J_s} = Speedup_s \cdot \frac{Perf}{W_s} = \frac{1}{(1-f)+\frac{f}{c}} \cdot \frac{1}{1+(c-1)k_c(1-f)} \tag{4}$$

Figure 1 plots the $Perf/J_s$ as a function of the number of CPU-cores in a symmetric multi-core processor. It is immediately obvious that there is a huge gap between the $Perf/J_s$ obtainable when a high degree of parallelism is available $(f = 0.99)$ and that when the available parallelism is only 10% less $(f = 0.9)$. Thus, the major factor affecting the energy saving of mobile devices is the development of extremely parallel applications. When an abundance of parallelism is available $(f = 0.99)$, the $Perf/J_s$ increases linearly with the increase in the number of cores whereas with $f < 0.9$ the $Perf/J_s$ reaches it maximum at a small number of cores before decreasing slowly.

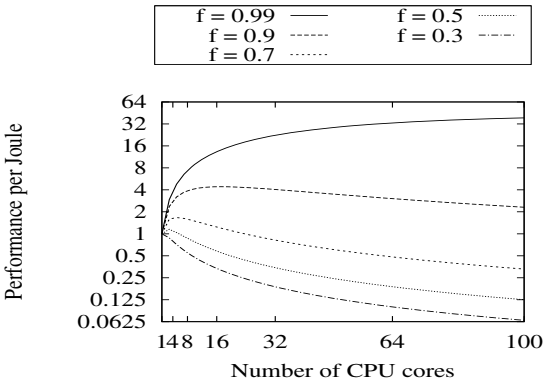

Fig. 1. Performance per joule as a function of the number of CPU-cores in a symmetric multi-core processor when $k_c = 0.3$ and various values of f

3 Asymmetric CPU-GPU Processors

We assume that a program's execution time can be composed of a time period where the program runs in parallel (f), a time period where the program runs in parallel on the CPU cores (α), and a time period where the program runs in parallel on the GPU cores ($1 - \alpha$).

To model the power consumption of an asymmetric processor we introduce another variable, k_g, to represent the fraction of power a single GPU-core consumes in its idle state ($0 \leq k_g \leq 1$). We introduce two further variables, α and β, to model the performance difference between a CPU-core and a GPU-core. The first variable represents the fraction of a program's execution time that is parallelized on the CPU-cores ($0 \leq \alpha \leq 1$), while the second variable represents a GPU core's performance normalized to that of a CPU-core ($0 \leq \beta$). For example, comparing the performance of a single core of Intel Core-i7-960 multi-core processor against the performance of a single core of a NVIDIA GTX 280 GPU processor yields values of β between 0.4 and 1.2. Moreover, recent studies such as [9] show that the GPU processor (NVIDIA GTX 280) achieves only 2.5x speedup in average compared to multi-core processor (Intel Core-i7-960).

We assume that one CPU-core in an active state consumes a power of 1 and the *power budget (PB)* of a processor is 100. Thus, $g = (PB - c)/w_g$ is the number of the GPU-cores embedded in the processor, where variable w_g represents the active GPU core's power consumption relative to that of an active CPU-core ($0 \leq w_g \leq 1$).

3.1 Asymmetric Speedup

Now, if the sequential code of the program is executed on a single CPU-core the following equation represents the theoretical achievable *asymmetric speedup* ($speedup_a$).

$$Speedup_a = \frac{1}{(1-f) + \frac{\alpha f}{c} + \frac{(1-\alpha)f}{g \cdot \beta}} \tag{5}$$

3.2 Asymmetric Performance per Watt

To model the power consumption of an asymmetric processor we assume that one core is active during the sequential computation and consumes a power of 1, while the remaining $c-1$ idle CPU-cores consume $(c-1)k_c$ power and g idle GPU-cores consume $g \cdot w_g \cdot k_g$ power. Thus, during the parallel computation period of the CPU-cores, c active CPU-cores consume c power and g idle GPU-cores consume $g \cdot w_g \cdot k_g$ power. During the parallel computation period of the GPU-cores, g active GPU-cores consume $g \cdot w_g$ power and c idle CPU-cores consume $c \cdot k_c$ power. Let P_s, P_c *and* P_g denote the power consumption during the sequential, CPU, and GPU processing phases, respectively.

$$\begin{aligned} P_s &= (1-f)\{1 + (c-1)k_c + g \cdot w_g \cdot k_g\} \\ P_c &= \frac{\alpha f}{c}\{c + g \cdot w_g \cdot k_g\} \\ P_g &= \frac{(1-\alpha)f}{g \cdot \beta}\{g \cdot w_g + c \cdot k_c\} \end{aligned}$$

It requires $(1-f)$ to perform the sequential computation, and $\frac{\alpha f}{c}$ and $\frac{(1-\alpha)f}{g \cdot \beta}$ to perform the parallel computations on the CPU and GPU, respectively, so the average power consumption W_a of an asymmetric processor is as follows.

$$W_a = \frac{P_s + P_c + P_g}{(1-f) + \frac{\alpha f}{c} + \frac{(1-\alpha)f}{g \cdot \beta}} \tag{6}$$

Consequently, $Perf/W_a$ of a an asymmetric processor is expressed as

$$\frac{Perf}{W_a} = \frac{Speedup_a}{W_a} = \frac{1}{P_s + P_c + P_g} \tag{7}$$

3.3 Asymmetric Performance per Joule

Based on our definition of performance per joule, the $Perf/J_a$ of a an asymmetric processor is expressed as follows.

$$\frac{Perf}{J_a} = Speedup_a \cdot \frac{Perf}{W_a} = \frac{1}{(1-f) + \frac{\alpha f}{c} + \frac{(1-\alpha)f}{g \cdot \beta}} \cdot \frac{1}{P_s + P_c + P_g} \tag{8}$$

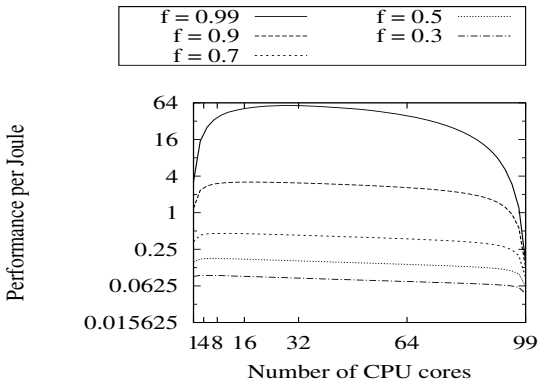

Fig. 2. Performance per joule as a function of the number of CPU-cores in an asymmetric processor when $\alpha = 0.1, k_c = 0.3, k_g = 0.2, w_g = 0.25, \beta = 0.5$ and various values of f

Figure 2 plots the $Perf/J_a$ as a function of the number of CPU-cores in an asymmetric processor. It can be observed again that high energy efficiency in a heterogeneous system is obtainable only if hybrid parallel programming models will be available for building extremely parallel programs. Such programs will need the support of runtime systems to find the optimal chip configuration for maximum battery continues operation. For example, in Figure 2 the optimal configuration (for $f = 0.99$) is achieved for 28 CPU-cores and 312 GPU-cores.

4 CPU-GPU Simultaneous Processing

In the previous analysis we assumed that a program's execution time is divided into three phases as follows: a sequential phase where one core is active, a CPU phase where the parallelized code is executed by the CPU-cores and a GPU phase where the parallelized code is executed by the GPU-cores. However, the aim of hybrid CPU-GPU computing is to divide the program while allowing the CPU and the GPU to execute their codes simultaneously.

4.1 Simultaneous Asymmetric Speedup

We conduct our analysis assuming that the CPU's execution time overlaps with the GPU's execution time. Such an overlap occurs when the CPU's execution time $\frac{\alpha f}{c}$ equals the GPU's execution time $\frac{(1-\alpha)f}{g \cdot \beta}$. Let α' denote the value of α that applies to this equality:

$$\alpha' = \frac{c}{g \cdot \beta + c}$$

Now, if the sequential code of the program is executed on a single CPU-core the following equation represents the theoretical achievable *simultaneous asymmetric speedup (speedup$_{sa}$)*.

$$Speedup_{sa} = \frac{1}{(1-f) + \frac{\alpha' f}{c}} = \frac{1}{(1-f) + \frac{f}{g \cdot \beta + c}} \tag{9}$$

4.2 Simultaneous Asymmetric Perf/W

To model the power consumption of an asymmetric processor for the case of simultaneous processing, we assume that one core is active during the sequential computation and consumes a power of 1, while the remaining $c-1$ idle CPU-cores consume $(c-1)k_c$ power and g idle GPU-cores consume $g \cdot w_g \cdot k_g$ power. During the parallel computation period, c active CPU-cores consume c power and g GPU-cores consume $g \cdot w_g$ power. It requires $(1-f)$ to execute the sequential code, and $\frac{\alpha' f}{c}$ to execute the parallel code on the CPU and GPU simultaneously, so the formula for the average power consumption W_{sa} of an asymmetric processor during simultaneous processing is as follows.

$$W_{sa} = \frac{P_s + \frac{\alpha' f}{c}\{c + g \cdot w_g\}}{(1-f) + \frac{\alpha' f}{c}} \tag{10}$$

Consequently, $Perf/W_{sa}$ of an asymmetric processor during simultaneous processing is expressed as

$$\frac{Perf}{W_{sa}} = \frac{Speedup_{sa}}{W_{sa}} = \frac{1}{P_s + \frac{\alpha' f}{c}\{c + g \cdot w_g\}} \tag{11}$$

4.3 Simultaneous Asymmetric Perf/J

Based on our definition of performance per joule, the
$Perf/J_{sa}$ of an asymmetric processor in the simultaneous processing mode is expressed as follows.

$$\frac{Perf}{J_{sa}} = Speedup_{sa} \cdot \frac{Perf}{W_{sa}} = \frac{1}{(1-f) + \frac{\alpha' f}{c}} \cdot \frac{1}{P_s + \frac{\alpha' f}{c}\{c + g \cdot w_g\}} \tag{12}$$

Figure 3 shows the $Perf/J_{sa}$ as a function of number of CPU-cores with an asymmetric processor where the CPU and the GPU are in simultaneous processing mode. As expected, a low degree of parallelism decreases significantly

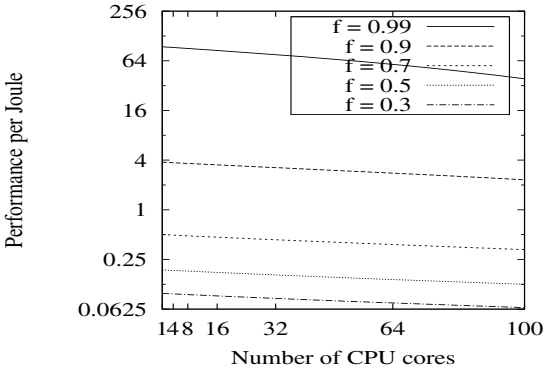

Fig. 3. Performance per joule as a function of the number of CPU-cores for an asymmetric processor in simultaneous processing mode when $\alpha = 0.1, k_c = 0.3, k_g = 0.2, w_g = 0.25, \beta = 0.5$ and various values of f

the energy efficiency. On the other hand, when an abundance of parallelism is available the energy efficiency is very high. In simultaneous processing mode, the obtainable $Perf/J_{sa}$ decreases slowly with the increase in the number of CPU-cores. This phenomenon means that it is not always necessary to support a dynamic reconfigurable processor and an associated runtime optimizer when finding the best chip configuration, because all possible chip configurations yield optimal or near-optimal configuration.

5 Synthesis

Figure 4 shows the three $Perf/J$ graphs for the analytical models investigated, i.e., symmetric (s), asymmetric (a) and simultaneous asymmetric (sa). This comparison shows that greater parallelism yields better energy efficiency and offers more chip configurations choices, while encouraging the search or better scalable software with energy saving. Simultaneous processing yields an excellent $Perf/J$ with peak performance using a chip configuration of a single CPU-core. It then decreases as the number of CPU-cores increases untill the point where all cores in the chip are CPU-cores, which is also the intersection point with the symmetric $Perf/J$. In contrast, the asymmetric processor delivers poor $Perf/J$ at extreme points where the number of CPU-cores is small or large, which requires that the dynamic configuration is identified and set for optimal chip organization.

6 Related Work

Hill and Marty [11] studied the implications of Amdahl's law on multi-core hardware resources and proposed the design of future chips based on the overall chip performance rather than core efficiencies. The major assumption in that model

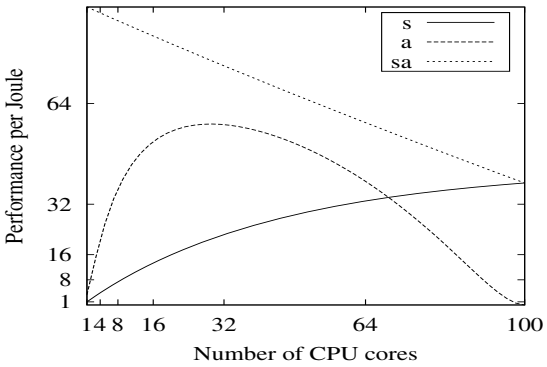

Fig. 4. Comparison between symmetric Perf/J (s), asymmetric Perf/J (a), and simultaneous asymmetric Perf/J (sa) when $\alpha = 0.1, k_c = 0.3, k_g = 0.2, w_g = 0.25, \beta = 0.5$ and $f = 0.99$

was that a chip is composed of many basic cores and their resources can be combined dynamically to create a more powerful core with higher sequential performance. Using Amdahl's law, they showed that asymmetric multi-core chips designed with one fat core and many thin cores exhibited better performance than symmetric multi-core chip designs. For example, with $f = 0.975$ (the fraction of computation that can parallelize) and $n = 256$ (Base Core Equivalents), the best asymmetric speedup was 125.0, whereas the best symmetric speedup was 51.2. Individual core resources could be dynamically combined to increase performance of the sequential component, so the performance was always improved. In our example, the speedup was increased to 186.0.

Woo and Lee [1] developed a many-core performance per energy analytical model that revisited Amdahl's Law. Using their model the authors investigated the energy efficiency of three architecture configurations. The first architecture studied contained multi-superscalar cores, the second architecture contained many simplified and energy efficient cores, and the third architecture was an asymmetric configuration of one superscalar core and many simplified energy efficient cores. The evaluation results showed that under restricted power budget conditions the asymmetric configuration usually exhibited better performance per watt. The energy consumption was reduced linearly as the performance was improved with parallelization scales. Furthermore, improving the parallelization efficiency by load balancing among processors increased the efficiency of power consumption and increased the battery life.

Sun and Chen [11] studied the scalability of multi-core processors and reached more optimistic conclusions compared with the analysis conducted by Hill and Marty [11]. The authors suggested that the fixed-size assumption of Amdahl's law was unrealistic and that the fixed-time and memory-bounded models might better reflect real world applications. They presented extensions of these models for multi-core architectures and showed that there was no upper bound on the

scalability of multi-core architectures. However, the authors suggested that the major problem limiting multi-core scalability is the memory data access delay and they called for more research to resolve this memory-wall problem.

Esmaeilzadeh et al. [12] performed a systematic and comprehensive study to estimate the performance gains from the next five multi-core generations. Accurate predictions require the integration of as many factors as possible. Thus, the study included: power, frequency and area limits; device, core and multi-core scaling; chip organization; chip topologies (symmetric, asymmetric, dynamic, and fused); and benchmark profiles. They constructed models based on pessimistic and optimistic forecasts, and observations of previous works with data from 150 processors. The conclusions were not encouraging. Over five technology generations only a 7.9x average speedup was predicted with multi-core processors, while over 50% of the chip resources will be turned off due to power limitations. Neither multi-core CPUs nor many-core GPUs architectures were considered to have the potential for delivering the required performance speedup levels.

Cho and Melhem [13] studied the mutual affects of parallelization, program performance, and energy consumption. Their analytic model was applied to a machine that could turn off individual cores, while others do not make this assumption. The main prediction was that greater parallelism (a greater ratio of the parallel portion in the program) and more cores helped reduce energy use. Moreover, it was shown that is possible to reduce the processor speeds and gain further dynamic energy reductions before static energy becomes the dominant factor determining the total amount of energy used.

Hong and Kim [14] developed an integrated power and performance modeling system (IPP) for the GPU architecture. IPP is an empirical power model that aims to predict performance-per-watt and the optimal number of active cores for bandwidth-limited applications. IPP uses predicted execution times to predict power consumption. In order to predict the execution time the authors used a special-purpose GPU analytical timing model. Moreover, to obtain the power model parameters, they designed a set of synthetic micro-benchmarks that stress different architectural components in the GPU.

The evaluation of the proposed model was done by using NVIDIA GTX280 GPU. The authors show that by predicting the optimal number of active cores, they can save up to 22.09% of runtime GPU energy consumption and on average 10.99% of that for five memory bandwidth-limited benchmarks. They also calculated the power savings if a per-core power gating mechanism is employed, and the result shows an average of 25.85% in energy reduction. IPP predicts the power consumption and the execution time with an average of 8.94% error for the evaluated benchmarks GPGPU kernels. It can be used by a thread scheduler in order to manage the power system more efficiently or by the programmers to optimize program configurations.

7 Conclusions

We investigated three analytical models of symmetric, asymmetric, and simultaneous asymmetric processing. These models extended Amdahl's Law for symmetric multi-core and heterogeneous many-core processors by taking in account power constraints. The analysis of speedup and the performance per watt of various chip configurations suggests that future CMPs should be a priori designed to include one or a few fat cores alongside many efficient thin cores to support energy efficient hardware platforms. On the software side, this study shows without a doubt that increased parallelism should not be the exception, because the standard parallel programming paradigm can create energy saving applications that can be used to efficiently underpin future multi-core processor architectures.

References

1. Woo, D.H., Lee, H.S.: Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era. IEEE Computer 38(11), 32–38 (2005)
2. Mantor, M.: Entering the Golden Age of Heterogeneous Computing. In: Performance Enhancement on Emerging Parallel Processing Platforms (2008)
3. Kogge, P., et al.: ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems. DARPA, Washington, D.C (2008)
4. Fuller, S.H., Millett, L.I.: Computing Performance: Game Over or Next Level? IEEE Computer 44(1), 31–38 (2011)
5. Borkar, S.: Thousand core chips: a technology perspective. In: Proc. 44th Design Automation Conference, pp. 746–749. ACM Press (2007)
6. Gustafson, J.L.: Reevaluating Amdahl's Law. Communication of ACM 31(5), 532–533 (1988)
7. Hillis, D.: The pattern on the stone: The simple ideas that make computers work. Basic Books (1998)
8. Amdahl, G.M.: Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capabilities. In: Proc. Am. Federation of Information Processing Societies Conf., pp. 483–485. AFIPS Press (1967)
9. Lee, V.W., et al.: Debunking the 100X GPU vs. CPU myth: an evaluation of throughput computing on CPU and GPU. In: Proceedings of the 37th Annual International Symposium on Computer Architecture (2010)
10. Hill, M.D., Marty, M.R.: Amdahl's Law in the Multicore Era. IEEE Computer 41(7), 33–38 (2008)
11. Sun, X.-H., Chen, Y.: Reevaluating Amdahl's law in the multicore era. Journal of Parallel and Distributed Computing 70(2), 183–188 (2010)
12. Esmaeilzadeh, H., Blem, E., Amant, R.S., Sankaralingam, K., Burger, D.C.: Dark Silicon and the End of Multicore Scaling. In: Proceeding of 38th International Symposium on Computer Architecture (ISCA), pp. 365–376 (June 2011)
13. Cho, S., Melhem, R.G.: On the Interplay of Parallelization, Program Performance, and Energy Consumption. IEEE Trans. Parallel Distrib. Syst. 21(3), 342–353 (2010)
14. Hong, S., Kim, H.: An Integrated GPU Power and Performance Model. In: Proceeding of ISCA 2010, pp. 19–23. ACM (June 2010)