Extending Gustafson-Barsis's Law for Dual-Architecture Computing

Ami Marowka(✉)

Parallel Research Lab, Jerusalem, Israel
amimar2@yahoo.com

Abstract. This study has investigated how *scaled* performance is affected by the energy constraints imposed on dual-architecture processors. Theoretical models were developed to extend the Gustafson-Barsis Law by accounting for energy limitations before examining the three processing modes available to hybrid processors: symmetric, asymmetric, and simultaneous asymmetric. Analysis shows that by choosing the optimal chip configuration, energy efficiency and energy savings can be increased considerably.

Keywords: Energy efficiency · Gustafson-Barsis's Law · Hybrid architecture · Performance per watt · Modeling techniques

1 Introduction

The major challenge that microprocessor designers will face in the coming decade is not just power, but also energy efficiency. Although Moore's Law [1] continues to offer solutions with more transistors, power budgets limit our ability to use them. However, there are promising solutions such as heterogeneous many-core architectures that will provide higher performance at lower energy requirements and reduced leakage. 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 [3–8]. 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 [9,10], Amdahl's law [11] and Gustafson-Barsis's Law [12] are still relevant at the dawn of a heterogeneous many-core computing era. Both laws are simple analytical models that help developers to evaluate the actual speedup that can be achieved using a parallel program. They represent two points of view that are not contradictory, but rather complement each other. However, neither of these laws is perfect. Amdahl's Law and Gustafson-Barsis's Law do not account for overheads associated with the creation/destruction of processes/threads and with maintaining cache coherence. Neither do they account for other types of serial tasks such as identification of critical sections, synchronization, lock management, and load balancing.

R. Wyrzykowski et al. (Eds.): PPAM 2015, Part II, LNCS 9574, pp. 123–132, 2016.
DOI: 10.1007/978-3-319-32152-3_12

Furthermore, the future relevance of the laws requires their extension by the inclusion of constraints and architectural trends demanded by modern multiprocessor chips. In [13] we extended Amdahl's law according to the work of Woo and Lee [2] and applied it to the case of a hybrid CPU-GPU multi-core processor. In this work we repeat on our previous study, but this time we extend the Gustafson-Barsis's Law. The main contributions of this paper are as follows:

- To define and formulate two metrics: speedup and performance per watt.
- Using the above metrics, to evaluate the energy efficiency and scalability of three processing schemes available for heterogeneous computing: symmetric, asymmetric and simultaneous asymmetric.
- For each processing scheme, to examine how performance and power are affected by different chip configurations.
- Finally, to analyze and compare the outcomes of the three analytical models and to show how considerable energy savings can be achieved by choosing the optimal chip configuration.

2 Symmetric Processors

In this section we reformulate Gustafson-Barsis'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 [2].

2.1 Symmetric Speedup

Gustafson-Barsis's Law begins with a parallel computation and estimates how much faster the parallel computation is than the same computation executing on a single core. Gustafson argues that, as processor power increases, the size of the problem set also tends to increase. This is why the speedup determined by Gustafson-Barsis's Law, also called *scaled speedup*, is the time required by a parallel computation divided into the time hypothetically required to solve the same problem on a single core.

According to the Gustafson-Barsis's Law, a typical program has a serial portion that cannot be parallelized (and therefore can be executed only by a single core) and a parallel portion that can be parallelized (and therefore can be executed by any number of cores in the processor). Let the parallel execution time of the program be normalized to 1, and let the serial and parallel portions be denoted by s and p respectively. Then the following equation concisely describes the law:

$$Speedup_s = s + (1 - s) \cdot c = c + (1 - c) \cdot s \tag{1}$$

where c is the number of cores and s is the fraction of a program's execution time that is spend in serial code ($0 \leq s \leq 1$).

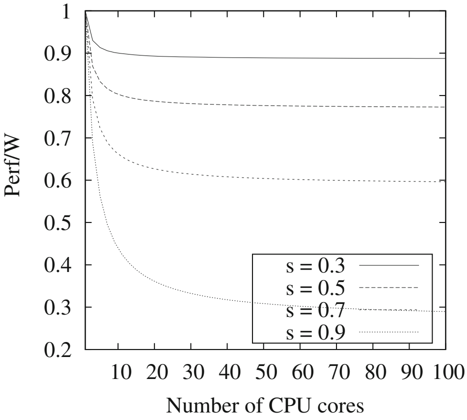

Fig. 1. Performance per watt as a function of the number of CPU cores of a symmetric multi-core processor when $k_c = 0.3$

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 $1+(c-1)k_c$. Thus, during the parallel computation time period, c CPU cores consume c power. It requires s and $(1 - s)$ 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.

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

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

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

Figure 1 plots the performance per watt for a symmetric multi-core processor as modeled by Eq. (3), showing that the performance per watt decreases rapidly for a small number of cores. However, as the number of cores increases, so does the problem size, and the inherently serial portion becomes much smaller as a proportion of the overall problem. Therefore, the performance per watt remains almost constant as the number of cores increases and reflects the assumption that the execution time remains fixed.

3 Asymmetric CPU-GPU Processors

In this section, an asymmetric CPU-GPU processor where CPU and GPU cores are **integrated on the same die and share the same memory space and power budget** will be referred to as a ***hybrid processor***.

We assume that a program's execution time can be composed of a time period where the program runs sequentially (s), 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 ($1 - \alpha$). **Note that in this case it is assumed that the program runs in parallel on the CPU cores *or* on the GPU cores, but not on both at the same time. Simultaneous asymmetric processing will be the topic of the next section.**

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 CPU core (Intel Core-i7-960 multicore processor) against the performance of a single GPU core (NVIDIA GTX 280 GPU processor) yields values of β between 0.4 and 1.2.

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 ($0 \leq w_g$).

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*$_a$).

$$Speedup_a = s + N \cdot (1 - s) \cdot \{\alpha \cdot c + \frac{(1 - \alpha) \cdot g}{\beta}\} \tag{4}$$

where N is the number of hybrid processors. Each hybrid processor contains c CPU cores and g GPU cores.

3.2 Asymmetric Performance per Watt

To model the power consumption of an asymmetric processor we assume that during the sequential computation phase, one CPU core is in active state and the amount of power it consumes is 1, the $c - 1$ idle CPU cores consume $(c - 1)k_c$ and the g idle GPU cores consume $g \cdot w_g \cdot k_g$. During the parallel computation

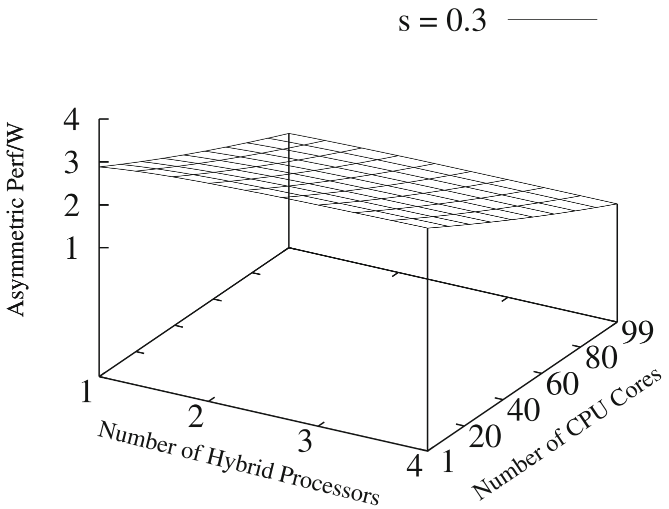

Fig. 2. Asymmetric perf/W as a function of the number of hybrid processors and various CPU-GPU chip configurations for $s = 0.3, w_g = 0.25, \alpha = 0.5, k_c = 0.3, k_g = 0.2$ and $\beta = 1.0$.

on the CPU cores, the CPU cores consume c and the g idle GPU cores consume $g \cdot w_g \cdot k_g$. During the parallel computation on the GPU cores, the GPU cores consume $g \cdot w_g$ and the idle CPU cores consume $c \cdot k_c$.

Let P_s, P_c, and P_g denote the power consumption during the sequential, CPU, and GPU processing phases, respectively.

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

It requires time $(1-p)$ to perform the sequential computation, and times $\alpha \cdot p$ and $(1-\alpha) \cdot p$ 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.

$$W_a = P_s + P_c + P_g \tag{5}$$

Consequently, $Perf / W_a$ of N asymmetric processors is expressed as

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

Figure 2 shows the performance per watt of an asymmetric processor for s = 0.3 as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be seen that the *Perf*/*W*$_a$ decreases slowly with the increase in the number of hybrid processors, as expected, and decreases faster as the number of the CPU cores increases. Furthermore, the optimal *Perf*/*W*$_a$ is obtained for a chip configuration of 1 CPU core and 396 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 will 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 $\alpha \cdot p \cdot c$ equals the GPU's execution time $\frac{(1-\alpha)\cdot p \cdot g}{\beta}$. Let α' denote the value of α that applies to this equality:

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

We assume that the sequential code of the program is executed on a single CPU core. Thus, the following equation represents the theoretical achievable *simultaneous asymmetric speedup (speedup$_{sa}$)*:

$$\begin{aligned} Speedup_{sa} &= s + N \cdot (1-s) \cdot \{\alpha' \cdot c\} \\ &= s + N \cdot (1-s) \cdot \{\frac{(1-\alpha') \cdot g}{\beta}\} \end{aligned} \tag{7}$$

where N is the number of hybrid processors. Each hybrid processor contains c CPU cores and g GPU cores.

4.2 Simultaneous Asymmetric Perf/W

To model the power consumption of an asymmetric processor in a simultaneous processing mode, 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 $(c-1)k_c$ and g idle GPU cores consume $g \cdot w_g \cdot k_g$. Thus, during the parallel computation time period, c active CPU cores consume c and g active GPU cores consume $g \cdot w_g$. It requires $(1-p)$ to execute sequential code and $\alpha' \cdot p$ to execute the parallel codes on the CPU and the GPU simultaneously,

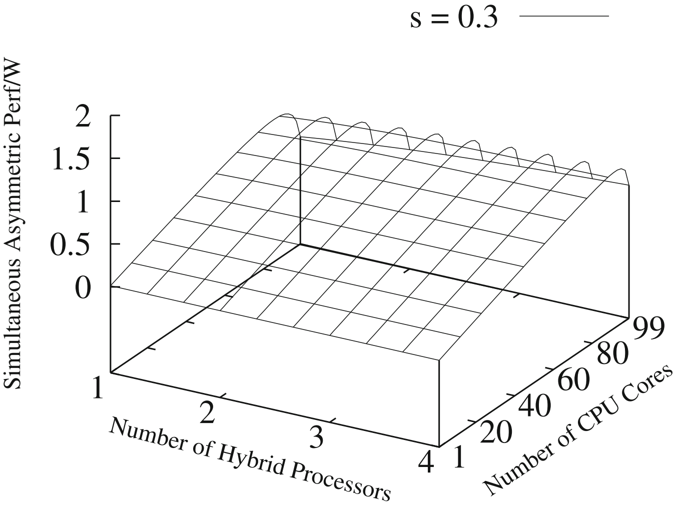

Fig. 3. Simultaneous Asymmetric Perf/W as a function of the number of hybrid processors and various CPU-GPU chip configurations for $s = 0.3, w_g = 0.25, k_c = 0.3, k_g = 0.2$ and $\beta = 1.0$.

so the average power consumption of an asymmetric processor in a simultaneous processing mode is

$$W_{sa} = P_s + P_c + P_g \tag{8}$$

where

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

Consequently, $Perf/W_{sa}$ of N asymmetric processors in a simultaneous processing mode is expressed as follows.

$$\frac{Perf}{W_{sa}} = \frac{s + N \cdot (1-s) \cdot \{\alpha' \cdot c\}}{P_s + N \cdot (P_c + P_g)} \tag{9}$$

Figure 3 shows the performance per watt of an asymmetric processor, as modeled by Eq. (9), for $s = 0.3$ as a function of the number of hybrid processors and as a function of CPU cores within each hybrid processor. It can be observed that the $Perf/W_{sa}$ slightly decreases with the increase in the number of hybrid processors. When the performance of the CPU cores dominates, the graph increases rapidly as the number of CPU cores increases (and the number of GPU cores is decreases). Then, it reaches the point beyond which the performance per watt decreases very rapidly because the dominance of the GPU cores is negligible.

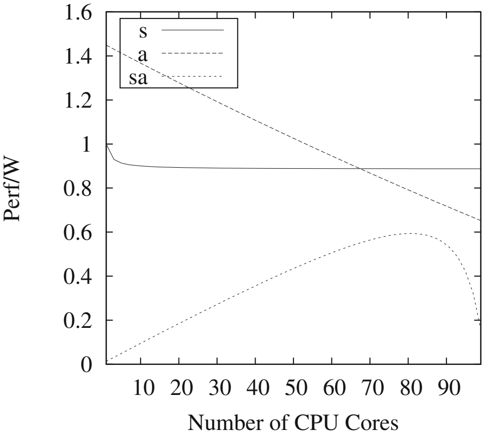

Fig. 4. Symmetric (s) Asymmetric (a) and Simultaneous Asymmetric (sa) Perf/W as a function of the number of CPU cores for one hybrid processor and for $s = 0.3$, $w_g = 0.25, \alpha = 0.5$ and $\beta = 2.0$.

5 Synthesis

Figure 4 shows the performance per watt of the three processing schemes that were studied in this research (symmetric (s), asymmetric (a), and simultaneous asymmetric (sa)) and how they are affected by chip configuration. First, it can be observed that the chip configuration has no effect on *Perf*/*W* while processing in symmetric mode, as can be expected. In simultaneous asymmetric processing mode, *Perf*/*W* improves with increasing number of CPU cores until it reaches peak performance for a chip configuration of approximately 85 CPU cores and 60 GPU cores. Beyond this point, *Perf*/*W* decreases rapidly to a point where the contribution of the GPU cores is negligible. On the other hand, in asymmetric processing mode, a chip configuration consisting of a single CPU core yields an optimal performance per watt, and any attempt to increase the number of CPU cores in the chip organization leads to a significant decrease in performance per watt.

6 Related Work

Hill and Marty [14] 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 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.

Woo and Lee [2] 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 [15] studied the scalability of multi-core processors and reached more optimistic conclusions compared with the analysis conducted by Hill and Marty [14]. 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. [16] 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.

7 Conclusions

The analysis of three analytical models of symmetric, asymmetric, and simultaneous asymmetric processing using two performance metrics with regard to various chip configurations suggest that future many-core processors should be a priori designed to include one or a few fat cores alongside many efficient thin cores to support energy efficient hardware platforms. Moreover, to achieve optimal scalability and energy savings, a dynamic configuration mechanism is required for identifying and implementing the optimal chip organization.

References

1. Moore, G.: Cramming more components onto integrated circuits. Electronics **38**(8), 114–117 (1965)
2. Woo, D.H., Lee, H.S.: Extending Amdahl's law for energy-efficient computing in the many-core era. IEEE Comput. **38**(11), 32–38 (2005)
3. Kumar, R., et al.: Heterogeneous chip multiprocessors. IEEE Comput. **38**(11), 32–38 (2005)
4. Mantor, M.: Entering the Golden Age of Heterogeneous Computing. C-DAC PEEP 2008. http://ati.amd.com/technology/streamcomputing/IUCAA_Pune_PEEP_2008.pdf
5. Kogge, P., et al.: ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems. DARPA, Washington, D.C (2008)
6. Fuller, S.H., Millett, L.I.: Computing performance: game over or next level? IEEE Comput. **44**(1), 31–38 (2011)
7. Borkar, S.: Thousand core chips: a technology perspective. In: Proceedings of 44th Design Automation Conference (DAC 2007), pp. 746–749. ACM Press (2007)
8. Marowka, A.: Back to thin-core massively parallel processors. IEEE Comput. **44**(12), 49–54 (2011)
9. Hillis, D.: The Pattern on the Stone: The Simple Ideas that Make Computers Work. Basic Books, New York (1998)
10. Shi, Y.: Reevaluating Amdahl's Law and Gustafson's Law (1996). http://www.cis.temple.edu/shi/docs/amdahl/amdahl.html
11. Amdahl, G.M.: Validity of the single-processor approach to achieving large-scale computing capabilities. In: Proceeidngs of American Federation of Information Processing Societies Conference, pp. 483–485. AFIPS Press (1967)
12. Gustafson, J.L.: Reevaluating Amdahl's law. Comm. ACM **31**, 532–533 (1988)
13. Marowka, A.: Analytical modeling of energy efficiency in heterogeneous processors. Comput. Electr. Eng. J. **39**(8), 2566–2578 (2013). Elsevier press
14. Hill, M.D., Marty, M.R.: Amdahl's law in the multicore era. IEEE Comput. **41**, 33–38 (2008)
15. Sun, X.-H., Chen, Y.: Reevaluating Amdahl's law in the multicore era. J. Parallel Distrib. Comput. **70**, 183–188 (2010)
16. Esmaeilzadeh, H., Blem, E., St. Amant, R., 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