International Journal of New Computer Architectures and their Applications (IJNCAA) 3(3): 30-46
The Society of Digital Information and Wireless Communications, 2013 (ISSN: 2220-9085)

A Survey on Amdahl's Law Extension in Multicore Architectures

Bashayer M. Al-Babtain, Fajer J. Al-Kanderi, Maha F. Al-Fahad, and Imtiaz Ahmad
Computer Engineering Department, College of Computing Sciences and Engineering
Kuwait University, P.O Box 5469, Safat 13060, Kuwait
Eng.bashayer@gmail.com, eng.fjk@gmail.com, maha@al-fuhaid.net,
imtiaz.ahmad@ku.edu.kw

ABSTRACT

Multicore architectures represent the future of computing since they provide cost effective solutions to improve throughput and performance of parallel programs while keeping power consumption manageable. Amdahl's law continues to serve as a guideline for parallel programmers to assess the upper bounds of attainable performance in multicore architectures. In this article, we review the key papers related to the extension of Amdahl's law for multicore architectures by characterizing them into five categories. For each category we briefly survey the main analytic modeling techniques and discuss their inherent advantages and disadvantages. All the analytic models discussed in this article are compared against a number of attributes hindering the performance enhancement of multicore computing to pinpoint their strengths and weaknesses. Finally, we recommend directions for future work to inspire the research community to invent new ideas to model and evaluate the emerging multicore computing paradigms.

KEYWORDS

Amdahl's Law, Multicore Architecture, Performance, Analytic Models, Power.

1 INTRODUCTION

The hunger for faster performance in the computer domain is never satisfied; every new performance enhancement in processors leads to another level of greater performance demands from businesses and consumers. Today these performance demands are not just for speed, but also for smaller, more powerful mobile devices, longer battery life, better price/performance per watt and lower cooling costs [1]. In the past, processor performance trends were dominated by increasingly complex feature sets, higher clock speeds, and increasing power dissipation. Recently, clock speeds have tapered and power dissipation envelopes have remained flat [2]. However, the demand for increasing performance still continues and as single core processors reaches their physical limits of possible complexity and speed, the movement towards multicore processors begins.

A multicore processor is an integrated circuit (IC) to which two or more processors have been attached for enhanced performance, reduced power consumption, and more efficient simultaneous processing of multiple tasks [3]. Multicore architectures have been a major design trend over the past decade, starting with high-end server processors and moving to low-end hand-held mobile devices. These chips provide an effective solution to improve throughput performance of parallel programs while keeping power consumption manageable. They allow for faster execution of applications by taking advantage of parallelism, or the ability to work on multiple problems simultaneously. Computing vendors

have been announcing chips with multiple processor cores. The general trend in processor development has moved from dual-, tri-, quad-, hexa-, octo-core chips to ones with tens or even hundreds of cores [4]. In addition, multi-core chips mixed with simultaneous multithreading, memory-on-chip, and special-purpose "heterogeneous" cores promise further performance and efficiency gains, especially in processing multimedia, recognition and networking applications. Similarly, the highly parallel graphic processing unit (GPU) is rapidly gaining maturity as a powerful engine for computationally demanding applications.

In 1967, Gene Amdahl proposed an often overlooked law of scaling: A program's sequential computation largely limits the maximum achievable speedup [5]. A simple, yet insightful, observation, Amdahl's law continues to serve as a guideline for parallel programmers to assess the upper bounds of attainable performance. As the wealth and complexity of the data around us grows, the importance of multicore processors will increase significantly and new processors with hundreds or even more cores will be developed. Since the multicore processors represent the future of computing, extending Amdahl's law to model and evaluate their performance would be extremely beneficial to the current and future generation's multicore architectures. Several previous studies [8-32] have extended Amdahl's law to model and evaluate multicore architectures. In this article, we review the key papers related to extension of Amdahl's law for multicore architectures to pinpoint their main contributions and limitations with the goal of inspiring and directing future research work in this area. First we describe the fundamental concepts of Amdahl's law. Second, we classify the modeling techniques into the following categories:

• Performance Modeling Techniques
• Power/Energy Modeling Techniques
• Latency Modeling Techniques
• Synchronization Modeling Techniques
• Communication Modeling Techniques

For each category, we review the contribution of key papers and highlight the state of the research. In each paper, we focus on identifying the design challenges and limitations that are critical. Note that most of the categories are nonexclusive; that is, a paper may focus on both performance and energy, however we assign each paper to one category based on the main objective of the paper. Thus, our classification is subjective. Finally, we recommend directions for future research to inspire the research community to invent new ideas to model and evaluate the future multicore computing paradigms.

The rest of this paper is organized as follows. Section 2 presents an overview on Amdahl's law. In Section 3, we briefly describe the different types of multicore architectures. In Section 4, we discuss a number of recent approaches that extend Amdahl's law for the multicore architectures. Section 5 presents a comparison of the different analytical models discussed in Section 4. Finally, Section 6 concludes the paper.

2 AMDAHL'S LAW

Amdahl's Law [5] is one of the few fundamental laws of computing that contribute to systems' performance enhancement. It is used to calculate the performance gain that can be achieved by improving some portion of a computer system. The Law states that the performance improvement to be

gained from using some faster mode of execution is limited by the fraction of time the faster mode can be used [6]. Amdahl's Law guides the computer architects on how much a particular enhancement will improve the overall performance and how to distribute resources to improve the cost to performance ratio. Spending resources should be proportional to where time is spent.

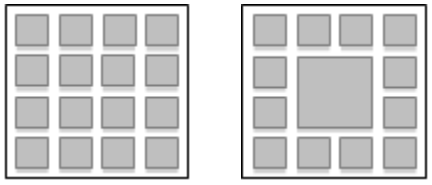
Amdahl defined his law for the special case of using *n* processors in parallel when he argued for the single-processor approach's validity for achieving large scale computing capabilities. Amdahl used a limit argument to assume that a fraction *f* of a program's execution time was infinitely parallelizable with no scheduling overhead, while the remaining fraction, *1−f*, was totally sequential. Amdahl noted that the speedup on *n* processors is governed by:

$$\text{Speedup}(f, n) = 1/((1-f) + f/n) \quad (1)$$

Despite its simplicity, Amdahl's law is applied broadly and gives important insights such as it when *f* is small, optimization will have little effect. Moreover, the sequential part *(1-f)* limits the speedup. Even if *n* approaches infinity, speedup is bounded by *1/(1−f)*. Amdahl's law continues to serve as a guideline for parallel computing. Before discussing recent models that extend Amdahl's law for the multicore architecture, we first present a brief overview on the different types of multicore architectures in Section 3.

3 MULTICORE ARCHITECTURES

As the number of transistors on a chip increases, the flexibility to determine a processor's configuration also increases. The current trend is to use them to integrate multiple cores on a chip [7]. Many different types of multicore architectures have been proposed in literature and can be classified into: symmetric, asymmetric, dynamic, distributed and CPU-GPU. A symmetric multicore processor (or homogeneous) requires the core to be fixed and multiple copies of the core are integrated on the chip and have the same cost. For example, a symmetric multicore chip which can support 16 processors with a single core is shown in Fig. 1(a). Asymmetric multicore chip (or heterogeneous) has only one large processor and the other cores remain small. For example, an asymmetric multi-core chip with a cost of 16 cores can have one big processor with four cores and 12 small cores as shown in Fig. 1(b). The dynamic multicore system provides the ability for cores to dynamically adjust their computational resources during execution to provide near optimal power/performance hardware configurations. It solves the problem of achieving the right balance of power and performance for an application that is challenging with today's multicore processors. They simply dynamically configure the number of cores and the size of each core, as shown in Fig. 1(c). When the currently executing portion of a program is easy to parallelize, the dynamically configurable multi-core processor increases the number of cores. On other hand, it combines some cores into a large core. The adaptability will improve the performance [7].



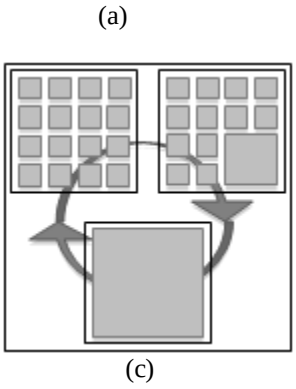


Figure 1. (a) Symmetric, (b) asymmetric and (c) dynamic multicore chips.

Distributed systems are collections of independent computing systems which are connected by some network and work together to solve an overall task using the notion of divide and conquer [8]. Fig. 2 shows an example of a multicore distributed system where a network combines a number of multicore machines to function as one.

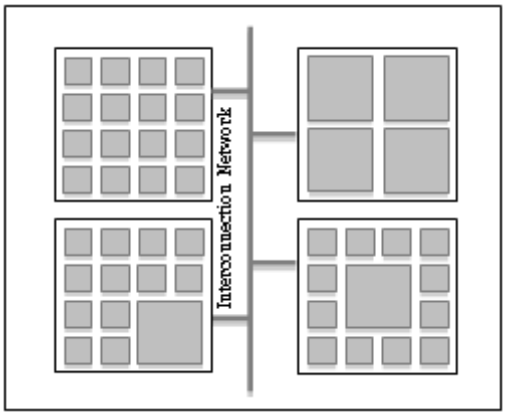


Figure 2. Distributed multicore chips.

A graphical processing unit, GPU is a good example of specialized massively parallel processors with over a hundred of cores in the latest products. Internally, the GPU contains a number of small core processors that are used to perform calculations. GPUs are similar to multicore CPUs but with two main differences. CPUs are made for speedup and GPUs for throughput. CPUs improve the execution of a single instruction stream while GPUs take the opposite route obtaining benefits from massively threaded streams of instructions and/or data (SIMD). The second difference is how threads are scheduled. The operating system schedule threads over different cores of a CPU in a preemptive fashion. GPUs have dedicated hardware for the cooperative scheduling of threads.

GPUs started out as independent units for program execution but there are clear trends towards tight-knit CPU-GPU integration (see Fig. 3) [9]. Not only does CPU-GPU chip integration offer performance benefits but it also enables new directions in system development. Reduced communication costs and increased bandwidth have the potential to enable new optimizations that were previously not possible.

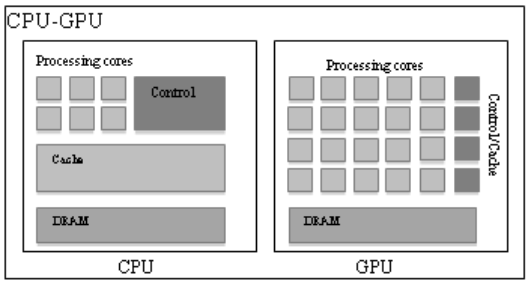


Figure 3. Architecture of a CPU-GPU chip.

Symmetric multi-core processors have several advantages such as flexibility and can run on different processes simultaneously. The symmetric multi-core processor that simply replicates a superscalar processor on a die provides the best single-thread performance. Additionally, it can run independent threads spawned from one process to improve a single application's performance. Asymmetric processor has difficulty in the flexibility to run different processes simultaneously. However, since it guarantees state-of-

the-art sequential performance for certain applications, the single-thread performance on the host processor should be high. The advantages of this architecture that it provides highly parallel performance when the efficient cores are in use. The dynamic processor solves the problem of achieving the right balance of power and performance for an application that is challenging with today's multicore processors. These chips can offer speedup that can be greater than asymmetric chips. Moreover, distributed computation differs from parallel computation in the way in which memory is used. In parallel systems, all processing elements use the same shared memory for communication, whereas distributed systems are autonomous systems with private memory connected by a network which is used for communication between the processing nodes. 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.

4 AMDAHL'S LAW EXTENSION IN MULTICORE ARCHITECTURE

In this section, we describe the analytic modeling techniques based on Amdahl's law for multicore architectures. For each model, we present its assumptions, the key contributions and limitations.

4.1 Performance Modeling Techniques

4.1.1 Amdahl's law in the multicore era [10]

Hill and Marty [10] did one of the pioneer works by extending Amdahl's law to multicore architectures by constructing a cost model for the number and performance of cores that the chip can support. According to the cost model, they classify the architecture of multicore chips into three types: symmetric, asymmetric and dynamic multicore chips. Based on the type of the multicore chip, the authors developed a formula for the overall speedup of the chip relative to using one single base core as an extension to Amdahl's law [5]. By comparing the different architectures, it was shown that asymmetric multicore chips offer higher speedups than that of symmetric multicore chips. Moreover, methods to speedup the sequential execution and extracting more parallelism are going to play a critical role in multicore era. Their proposed models have received much attention in the parallel computing and multicore area and the results obtained encouraged multicore designers to view the entire chip's performance rather than focusing on core efficiencies. However, the model has a number of limitations and required further research. As indicated by Hill and Marty [10] themselves, the presented model ignores the important effects of dynamic and static power, as well as on- and off-chip memory system and interconnects design, etc. Moreover, the communication cost between cores and overhead that may result from developing parallel software are not considered in their model. Despite its limitation and weaknesses, Hill and Marty [10] have successfully encouraged researchers to develop better models and their work has been used over the years as a basis for new performance models to measure the performance of multicore architectures.

4.1.2 Extending Amdahl's law in the multicore era [11], [12]

Yao et al. [11], [12] investigated the theoretical analysis of Hill and Marty's work [10] and extended their result to a more general framework to provide computer architects with a better and quantitative understanding of multicore scalability. The potentials of the maximum of speedups using architecture of a symmetric, asymmetric or dynamic multicore is obtained. Moreover, the precise quantitative conditions are given to determine how to obtain optimal multicore performance. However, their precise quantitative results are unreliable because many performance factors were removed from the model, including cache contention, cache coherence, synchronization, communication cost etc.

4.1.3 Scalable computing in the multicore era [13], [14]

Sun et al. [13], [14] followed the Hill and Marty [10] cost model to study the scalability of symmetric multicore processors from the scalable computing point of view. Gustafson [15] introduced the concept of scalable computing and the fixed-time speed-up model, which is a linear function of the number of processors if the workload is scaled up to maintain a fixed time execution time. The authors proposed three speedup models of multicore scalability which are: fixed-size (Amdahl's law), fixed-time and memory-bounded speedup. Amdahl's law assumes that the problem size is fixed and provides a fixed-size speed-up model. The theoretical analysis of their models reveals that, from the scalable computing viewpoint, the multicore architecture is linearly scalable and suitable for large-scale manufacturing as long as the data communication time is fixed. Their conclusions were that multicore architectures are scalable and not limited by Amdahl's law rather limited by the data access delay (memory-wall). In contrast to Hill and Marty work [10], the authors were optimistic about the scalability of multicore architectures. However, in their model the context switching overhead was not considered.

4.1.4 Single-chip heterogeneous computing: Does the future include custom logic, FPGAs, and GPGPUs [16]?

Chang et al. [16] compared the power efficiency of general purpose cores with three forms of "unconventional cores"; custom logic, FPGA and GPGPU. They extended Hill and Marty's model [10] to include area, power and bandwidth implications of unconventional computing cores. The objective of their model is identifying important trends worthy of future investigation. Their study shows that the complex interplay between bandwidth, power, and parallelism has tremendous implications for various heterogeneous and non-heterogeneous computing approaches. The study reveals the U-cores achieve performance gain over asymmetric and symmetric multicores despite the scarcity of memory bandwidth. Moreover, it was shown that sufficient parallelism must exist before U-core offered significant performance gains. However, dynamic multicore machines were not considered in their study.

4.1.5 Amdahl's law for predicting the future of multicores considered harmful [17]

One of the bottlenecks of Amdahl's law is that it assumes a fixed-size problem; that is; the parallelizable fraction remains constant, no matter how many cores are employed. Gustafson [15] addressed this limitation and indicated that such a situation with a fixed problem size is very rare. He argued that in almost all application domains, more cores are used to solve larger and more complex problems and assumed that the parallel fraction grows linearly with the number of cores. Hence, it is more realistic to assume that run time, not problem size, is constant. Juurlink and Meenderinck [17] generalized both Amdahl's and Gustafson's laws by assuming that the parallel fraction does not stay constant as in Amdahl's law, nor that it grows linearly with the number of cores as in Gustafson's law, but something in between (scaling function based on number of cores). Based on that assumption, a generalized scaled speedup equation (GSSE) was proposed. The authors applied Gustafson's law and the GSSE model to symmetric, asymmetric, and dynamic multicores and showed that they produce results that are fundamentally different from the results obtained by Hill and Marty model [10], which is based on Amdahl's law. The authors argue that while Amdahl's law makes a strong case for asymmetric and dynamic multicores, Gustafson's law and the GSSE show that asymmetric and dynamic multicores can still provide a performance advantage over symmetric multicores, but much less so than under Amdahl's assumptions. The authors emphasized that one has to consider the scaling properties of the targeted applications before applying Amdahl's law and making decisions based on the results. However, the proposed model can be extended to incorporate power constraints, memory bandwidths and workload behavior, etc.

4.1.6 Multicore model from abstract single core inputs [18]

Blem et al. [18] proposed and implemented a fine grained analytic model to improve multi-core performance accuracy by getting a tighter upper bound on performance as compared with the previous Amdahl's law based approaches. The authors model a heterogeneous chip with a mix of CPU and GPU like cores with varying performance. The chip's topology may be symmetric, asymmetric, dynamic, or even dynamically compose cores together (fused). The model consisting of five components: core performance, memory bandwidth performance, chip constraints, multicore performance, and overall speedup. The model takes into consideration the cache handling by allowing either a hit rate number or an analytical model input of its own. The authors included study for the CPU validation, the number of cores and the impact of memory bandwidth for the GPU validation, as well as a comparison against Amdahl's Law projection. The result showed that the value calculated in their model is more close to the measured value as compared with the value obtained by Amdahl's law. The model's accuracy is limited by optimistic assumptions which include homogeneous workload, no synchronization and interconnect overhead or memory stalls, thus making speedup projections over predictions. However, their extended model exposed more bottlenecks than the Amdahl's law while remaining simple and flexible

enough to be adapted for many applications.

4.1.7 MultiAmdahl: How should I divide my heterogeneous chip? [19], [20]

The authors introduced an analytic optimization technique called MultiAmdahl to design heterogeneous multicore systems by taking into consideration the workload, the performance of each computational unit, and the total available resources. The algorithm relies on modeling the performance of each computational unit as a function of the resources it uses such as the unit area, power or energy. The optimization problem is solved using Lagrange multipliers. The authors generalized the Amdahl's law from two types (serial and parallel) to n types and by directly modeling various design constraints and accounting for their impacts. The model is applicable to different resource constraints, efficiency models and objective functions. However, in their model, only one of the computational units (accelerators) is active at a time, which is not the case in real life applications.

4.1.8 Speedup for multi-level parallel computing [21]

State-of-the-art high performance computing systems support parallelism at multiple levels which takes processes for course-grained parallelism across the nodes and threads for fine-grained parallelism within the node at the same time. The original Amdahl's law [5] and Gustafson's law [15] modeled speedup based on the single-level parallelism and do not consider multi-level parallelism. Both of the laws do not differentiate and capture the varying granularity of multi-level parallelism. Therefore, to evaluate performance for homogeneous multi-level parallel computing systems, Tang et al. [21] extended Amdahl's law (called E-Amdahl's Law) for fixed-size speedup and Gustafson's law (called E-Gustafson's Law) for fixed-time speedup. The formulations consider the performance degradation due to uneven allocation and communication latency. However, for simulation purposes, the communication cost was not considered. The author's study revealed that the E-Amdahl's Law indicates that if the degree of parallelism at the first level is not large, increasing the degree of parallelism at the second level will not significantly improve the overall performance. The author's findings show that the estimated speedup based on E-Amdahl's Law is much more accurate than that with the original Amdahl's Law. Moreover, the estimated speedup of Amdahl's Law becomes more inaccurate when there are more processors used for fine-grained parallelism. However, the authors did not model the heterogeneous multi-level parallelism by taking into account the different computing capacities of heterogeneous processing elements such as combination of CPUs and GPUs. The author's main contribution lies in extending the model to capture the multi-level parallelism for better understanding in the performance and scalability of distributed computing systems.

4.2 Power/Energy Modeling Techniques

4.2.1 Extending Amdahl's law for energy-efficient computing in the many-core era [22]

Woo and Lee [22] extended the model of Hill and Marty [10] to include energy efficiency of superscalar, symmetric and asymmetric types of multicore architectures. The authors formulated and evaluated an analytical power model based on two metrics performance per watt and performance per joule for each design. The results of evaluating this model showed that, unfortunately, parallel execution on a superscalar core consumes much more energy than sequential execution to complete the task. This is because performance doesn't scale linearly but the amount of idle power does scale linearly with the number of cores. The results also imply that maximizing and balancing parallelization among processors is important, not only for higher performance but also for power-supply efficiency and extended battery life. The authors also performed a cross design comparison on the three previous models. It was concluded that a symmetric many-core processor can easily lose its energy efficiency as the number of cores increases as compared to asymmetric multicore system with many small energy-efficient cores integrated with a full-blown processor.

4.2.2 On the interplay of parallelization, program performance, and energy consumption [23], [24]

Cho and Melhem [23], [24] proposed analytic models to study the interaction between parallelization, program performance and energy consumption for variable speed processors. Their analytic modes were applied to machines that could turn off individual cores, while others do not make this assumption. The main prediction was that greater parallelism and more cores helped reduce energy. The authors derive optimal operating frequency of DVFS based processors to minimize the dynamic energy consumption without compromising on parallelism. Moreover, it was shown that it is possible to reduce the processor speeds and gain further energy reduction before static energy becomes the dominating factor in determining the total amount of energy used. However, the authors assumed that the parallel section of an application is fully parallelizable and ignore the overheads because of changing speeds.

4.2.3 Extending Amdahl's law for energy efficiency [25]

The authors investigated the impact of parallelization in the total dynamic energy consumed by considering dynamic voltage and frequency scaling in a multicore system. Under the given ratio between the serial and parallel potion of an application, the authors [25] extended Amdahl's law to derive analytic expressions to obtain the optimum frequency, supply voltage and energy improvement while the execution time remains the fixed to design energy efficient multicore architecture.

4.2.4 Extending Amdahl's law for heterogeneous computing [26]

Heterogeneous chips integrating different core architectures such as CPU and GPU on a single die is predicted to be the most promising solution for next generation of energy efficient multi-core architectures. Marowka [26] extended Woo and Lee [22] work for hybrid CPU-GPU multicore processors and examined the three processing modes available to heterogeneous processors such as single CPU process where one core is active,

CPU-GPU process where the code executed either by the CPU-cores or by the GPU-cores, and simultaneous CPU-GPU that allowing CPU and GPU execute codes simultaneously. The author focused on how energy efficiency and scalability are affected by the power constraints imposed on CPU-GPU processors. A simulation has been done that showed the performance per watt curves in the three cases. The analysis results show that simultaneous processing outperforms the other cases. Therefore, their finding suggests that future multicore system should be designed to include one or a few fat cores alongside many efficient thin cores to support energy efficient hardware platforms. However, they assume interconnect has zero latency (no overhead) and ignore the impact of delay in the proposed model.

4.2.5 Generalizing Amdahl's law for power and energy [27], [28]

The authors extended Amdahl's law to identify optimal power-performance configurations by considering the interactive effects of power, performance, and parallel overhead in power-scalable multicore systems. The authors focused on power aware speedup that describes the situation in which the workload is fixed and both the power modes (different power states) and the amount of parallelism change. Algorithm developers or programmers introduce parallel overhead when they redesign a program for a parallel system. For example, data might need to be exchanged between parallel processes where no such communications were required in the sequential version. The authors emphasized that the communication phase were the excellent times to slow down the processor for saving energy without affecting performance.

4.2.6 Analytical latency-throughput model of future power constrained multicore processors [29]

Single thread performance is still an important and an essential metric in today's multicore architectures for workloads that are difficult to parallelize due to intrinsic serial dependencies and even for parallel workloads due to critical sections that bottleneck throughput performance. Therefore, depending upon the mix of parallel and serial workloads balancing the tradeoff between single thread and throughput performance metrics is a critical design decision. Tseng and Brooks [29] proposed and evaluated an analytical model (an extension of Amdahl's Law) to compute single thread latency and throughput performance under a given power budget for both symmetric and asymmetric multicore architectures. The results showed that asymmetric designs achieve better tradeoffs between these metrics and that asymmetric design effort is more effective for higher power budgets. Asymmetric benefits are greater for highly parallel rather than more sequential single task workloads, when considering throughput under a fixed single thread performance constraint. The analytical model reduces the complexity of searching through a large design space of core designs, core counts, and asymmetric configurations. This model is very useful for early design phase studies. Simulations based on this model can improve the architectural configuration. However, as mentioned by the authors, their analytic model did not consider many other

important factors such as critical section modeling and context switching overhead, cache capacity and memory bandwidth sharing etc.

4.2.7 Beyond Amdahl's law: An objective function that links multiprocessor performance gains to delay and energy [30]

Cassidy and Andreou [30] presented cost function formulation of Amdahl's law for multicore systems. The generalized cost function incorporates delay and energy costs which is linked to characteristics of processors, memories and communication networks. As reported by the authors, Hill and Marty [10] performance model and Woo and Lee [22] energy cost model are special cases of their proposed model. To develop the analytic model, the authors generalized Amdahl's law to incorporate multiple degree of speedup, which is critical to capture varying nature of parallelism during different execution phases of applications. In applications, some portion of a program is serial, another portion can be parallelized across more processors and a third fraction can only be parallelized across less number of processors. Moreover, the generalization is important for modeling heterogeneous systems. The energy cost is considered only for active processors based on the degree of parallelism. The model can be applied to study the design tradeoffs for a wide range of architectures, including symmetric, asymmetric chips and shared access structures such as buses, memories, and networks and allows the inclusion of more detailed models for more accurate system modeling for optimal performance and efficiency.

4.3 Network Latency Modeling Technique

4.3.1 Performance prediction model for distributed applications on multicore clusters [8]

The authors propose a theoretical model to measure the performance of distributed multicore clusters, which consists of a number of computers interconnected by a network. Due to that fact, it is necessary to take into account the network influence on the performance of the overall architecture. The most important factors that affect the performance of these clusters are latency and bandwidth. This work focuses on the impact of those two factors, which affect the cost of inter-processor communication. The authors follow the work of Hill and Marty [10] and Sun et al. [13] to obtain a fixed-time speedup for distributed clusters. An experiment has been conducted to evaluate the accuracy of the presented model and to compare it with the results from the measured results, Hill and Marty [10] and Sun et al. [13]. It was noticed that Sun et al.'s model over estimates the speedup, whereas Hill and Marty's model [10] under estimates it. Both models do not consider the effects of communication associated with distributed systems. The presented model considers both the computation capability of multicore systems as well as the limitations of the network.

4.4 Critical Section Modeling Technique

4.4.1 Modeling critical sections in Amdahl's law and its implications for multicore design [32]

Locking and synchronization overhead has detrimental effects on the

performance and scalability of multicore systems [31]. Therefore, in order to quantify the critical sections effect on scalability and performance, analysis models are needed. The original Amdahl law assumes that the program's execution time is either parallelizable or it is totally sequential and does not take into account the impact of critical sections. Eyerman and Eeckhout [32] extended Amdahl's law for parallel performance to address the synchronization problems through critical sections. The authors derived a simple analytical (probabilistic) model for how long it takes to execute a critical section. The simple model reveals that the time spent in critical sections can be modeled as completely sequential part plus a parallel part. Augmenting Amdahl's law with this probabilistic model for critical sections shows that the parallel performance is not only limited by the sequential part (as suggested by Amdahl's law) but is also limited by synchronization of the parallel part.

This fundamental law has important implications for multicore design. In other words, the larger time spent in critical sections, the lower the maximum speedup. It shows that parallel performance is fundamentally limited by synchronization and more specifically, critical section size and their contention probability. The result has important implications for asymmetric processor design. The performance benefit of asymmetric multicore processor may not be as high as suggested by Hill and Marty model [10]. The core should not be tiny as suggested by Hill and Marty model [10], but should instead be larger to execute critical sections faster. The discussion of this paper is limited to critical sections only and does not tackle other types of synchronization. Another limitation is that it assumes that the parallel workload is homogeneous (all threads execute the same code and have the same probability for a critical section).

5 SUMMARY AND FUTURE DIRECTIONS

We provide a summary of various analytic models discussed in Section 4 by comparing against a number of attributes to pin point their strengths and weaknesses as shown in Table 1. The attributes selected for evaluation are: the type architecture, power/energy, memory latency, communication overhead, synchronization and degree of parallelism. In Table 1, the first five columns indicate the different design styles (symmetric, asymmetric, dynamic, distributed, CPU-GPU) to which model is applicable. The most critical attribute is the power and energy since the future of computing will be driven by the need for energy efficiency. Next we are considering memory latency in the evaluation criteria which is an important factor in multicore architectures. The communication overhead of parallelism is considered as another attribute for the evaluation. When a sequential program is redesigned for a parallel system, parallel overhead is introduced. For example, data might need to be exchanged between parallel processes where no such communications were required in the sequential version [28]. This communication overhead includes the intercommunication cost that is needed between cores. The next attribute shows whether the process of synchronization is taken into consideration while designing the model. The last attribute shows the degree of parallelism (multi-level parallelism)

which is very critical for distributed computing.

Table 1. Summary of results.

Technique	Symmetric	Asymmetric	Dynamic	Distributed	CPU-GPU	Pow./Energy	Mem. Latency	Comm. Over.	Synch.	Parallelism
[10]	✓	✓	✓	×	×	×	×	×	×	×
[11] [12]	✓	✓	✓	×	×	×	×	×	×	×
[13] [14]	✓	×	×	×	×	×	×	×	×	×
[16]	✓	✓	×	×	✓	✓	✓	×	×	×
[17]	✓	✓	✓	×	×	×	×	×	×	×
[18]	✓	✓	✓	×	✓	×	✓	×	×	×
[19] [20]	×	✓	×	×	✓	✓	×	×	×	✓
[21]	✓	×	×	✓	×	×	×	✓	×	✓
[22]	✓	✓	×	×	×	✓	×	×	×	×
[23] [24]	✓	×	×	×	×	✓	×	×	×	×
[25]	✓	×	×	×	×	✓	×	×	×	×
[26]	✓	✓	×	×	✓	✓	×	×	×	×
[27] [28]	×	×	×	×	×	✓	×	✓	×	×
[29]	✓	✓	×	×	×	✓	✓	×	×	×
[30]	✓	✓	✓	×	×	✓	✓	×	×	✓
[8]	×	×	×	✓	×	×	✓	✓	×	✓
[32]	✓	✓	×	×	×	×	×	×	✓	×

After studying, analyzing, and evaluating the recent proposed work related to the performance of multicore architectures, we have identified the main bottlenecks that hinder the performance enhancement of multicore computing to be power, latency, communication overhead, and synchronization. It can be observed from Table 1 that there is no recent model that takes into consideration all of the attributes that can affect the performance of multicore architectures. Most of the proposed works focuses on power while neglecting all other attributes [22-30]. Power is certainly one of the most important attributes however it does not diminish the importance of other attributes. The work done by [8], [18] concentrated on the latency metric, while [29], [30] combined latency with power to develop their models. The communication overhead and synchronization attributes did not receive as much attention as power and latency in the multicore research area. Only few built performance models that included communication overhead such as [8], [28]. In [28], a general formula that relates the performance of multicore architectures to the overhead attribute was presented without specifying which architecture is being used. On the other hand, the work presented by [8] focused on overhead in the distributed multicore architectures only. As far as we know, only [32] discussed the problem of synchronization between parallel cores. It was shown that the synchronization process specially related to critical sections has a huge impact on the overall performance.

It is of paramount importance to exploit all levels of parallelism such as fine-grained parallelism at the thread level and course grained parallelism at the task level to make the most effective use of multicore systems. Few techniques incorporated the degree of parallelism in their analytic model [8], [19-21], [30]. We also noticed that there is no recent work that studies the performance relative to all the different types of multicore architectures explained in Section 3. Most of the works consider

mainly the symmetric, asymmetric, and dynamic multicore architectures specially the models that extend and improve the work of Hill and Marty [10] such as [11], [17], [18]. The performance of distributed multicore architectures has been addressed by [8]. Following the clear trends towards tight-knit CPU-GPU integration, [16], [18-20], [26] built performance model to evaluate this type of architecture.

The different attributes may not be equally likely effective on the performance of the multicore architectures and a specific attribute may have a different degree of influence on performance relative to the type of architecture. For example, the communication overhead impacts the overall performance of any multicore architecture however, it has an extremely high level of influence in distributed architectures where an interconnection network is present. In the reported techniques, sometimes the authors have drawn contradictory conclusions based on the importance of attributes. For example, Hill and Marty model [6] suggests that the core in asymmetric architecture be tiny and more in number, whereas Eyerman and Eeckhout [22] extended model suggests that the cores should be more powerful to reduce the synchronization overhead in critical sections.

Ideally speaking it would be extremely beneficial to develop analytic models and frameworks that are highly flexible and extendable to incorporate novel multicore architectures and constraints impacting the performance. The analytic models based on Amdahl's law can be extended in the following directions:

• The main memory has become a significant energy dissipater in recent years as investigations [33] show that it accounts for up to 40% of total energy consumption in modern server systems. Since the future of computing will be driven by the need for energy efficiency [34-36], optimizing energy consumption of main memory is crucial to the overall energy budget of the system. One step in the direction of reducing energy consumption, researchers are predicting that future memory systems will contain hybrid memories such as Phase Change Memory and DRAM [37]. Therefore, analytic modeling should consider the hybrid nature of the future memory systems. Moreover, it is important to incorporate the memory load time into Amdahl's law for architectural model with deeper cache hierarchies and to consider more factors (e.g., cache misses in critical sections and non-critical sections).

• In order to perform energy-proportional computing at the lowest-possible levels of energy, efficient data orchestration to maximize locality will increasingly be critical. Algorithms will need to be developed to minimize the need to move data and maximize the reuse of data that is already locally available. Doing tens of extra operations instead of moving one operand from DRAM saves energy and the savings can be greater if the data is even further away on another processor die [34], [35]. Recently, Yuan and Zhang [38] introduced a locality function to model the reuse ability of an algorithm and propose a corresponding performance model to analyze the optimal cache utilization scheme for a cache partition design (leveraging both shared and private caches). It is worth modeling and exploring locality of an algorithm to save energy in multicore architectures.

• Recent research shows that integrated CPU-GPU processors with the aggressive use of application-specific hardware accelerators have the most potential to deliver more energy efficient computations on many applications [18-20], [26], [34], [35]. Given the budgets of power, area and other physical and design constraints, one has to consider task partitioning and allocation, granularities of tasks, parallel task fractions, load balancing, and selection of the accelerators to harvest the maximum energy-proportional computing of heterogeneous systems for a selected set of workload. Therefore, Amdahl's law needs to be extended to the leverage of HW/SW co-design of the heterogeneous multicore architecture.

6 CONCLUSIONS

Multicore architectures have a tremendous potential for the future of computing. However, there exists some bottlenecks that can affect the performance of such architectures and limit the speed up gained. Identifying and eliminating those bottlenecks in the process of designing multicore architectures are of paramount importance to get the maximum benefit out of the resources available. Amdahl's law continues to serve as a general analytic model to identify the bottlenecks and assess the upper bounds of attainable performance in the multicore architectures. In this article, we surveyed, analyzed and compared a number of key Amdahl's law based analytic modeling techniques for multicore architectures to identify their strengths and limitations. The goal of this survey is to emphasize the importance of full system optimization in the area of multicore processors and encourage researches to continue what Amdahl started back in 1967 by extending it to fit in the future multicore computing paradigms.

7 REFERENCES

1. Ramanathan, R.: Intel® Multi-Core Processors: Making the Move to Quad-Core and Beyond. Technology@Intel Magazine, Intel® Quad-Core Processors (2006).
2. Parkhurst, J., Darringer, J., Grundmann, B.: From Single Core to Multi-Core: Preparing for a New Exponential. Proc. IEEE ICCAD'06, pp. 67-72 (2006).
3. Wikipedia:http://en.wikipedia.org/wiki/Multi-core_processor
4. Borkar, S.: Thousand Core Chips - A Technology Perspective. Proc. ACM/IEEE 44th annual Design Automation Conference, pp. 746-749 (2007).
5. Amdahl, G.M.: Validity of the Single-Processor Approach to Achieving Large-Scale Computing Capabilities. Proc. AFIPS Conference, pp. 483-485 (1967).
6. Hennessy, J., Patterson, D.: Computer Architecture: A Quantitative Approach. Morgan Kaufman (2012).
7. Sato, T., Mori, H., Yano, R., Hayashida, T.: Importance of Single-Core Performance in the Multicore Era. Proc. Thirty-Fifth Australasian Computer Science Conference (ACSC 2012), pp. 107-114 (2012).
8. Khanyile, N.P., Tapamo, J.R., Dube, E.:An Analytic Model for Predicting the Performance of Distributed Applications on Multicore Clusters. IAENG International Journal of Computer Science 39(3), 312-320 (2012).
9. Arenas, M., Romero, G., Mora, A., Castillo, P., Merelo, J.: GPU Parallel Computation in Bioinspired Algorithms: A Review. Advances in Intelligent Modeling and Simulation: Studies in Computational Intelligence 422, 113-134 (2012).
10. Hill, M., Marty, M.: Amdahl's Law in the Multicore Era. IEEE Computer 41(7), 33-38 (2008).
11. Yao, E., Bao, Y., Tan, G., Chen, M.: Extending Amdahl's Law in the Multicore Era. SIGMETRICS Performance Evaluation Review 37 (2), 24-26 (2009).

12. Yao, E., Bao, Y., Chen, M.: What Hill-Marty Model Learn from and Break Through Amdahl's Law?. Information Processing Letters 111 (23-24), 1092-1095 (2011).
13. Sun, X.H., Chen, Y., Byna, S.: Scalable Computing in Multicore Era. Proc. Int. Symp. on Parallel Algorithms, Architectures and Programming (PAAP'08), (2008).
14. Sun, X.H., Chen, Y.: Reevaluating Amdahl's Law in the Multicore Era. J. Parallel and Distributed Computing 70(2), 183-188 (2010).
15. Gustafson, J.: Reevaluating Amdahl's Law. Communications of the ACM 31(5), 532-533 (1988).
16. Chung, E.S., Milder, P.A., Hoe, J.C., Mai, K.: Single-Chip Heterogeneous Computing: Does the Future Include Custom Logic, FPGAs, and GPGPUs?. Proc. 43rd IEEE/ACM International Symposium on Microarchitecture (MICRO-43), pp. 225-236 (2010).
17. Juurlink B.H.H., Meenderinck, C.H.: Amdahl's Law for Predicting the Future of Multicores Considered Harmful. ACM SIGARCH Computer Architecture News 40(2), 1-9 (2012).
18. Blem, E., Esmaeilzadeh,H., Amant, R.St., Sankaralingam, K., Burger, D.: Multicore Model from Abstract Single Core Inputs. IEEE Computer Architecture Letters 99 (2012).
19. Zidenberg, T., Keslassy, I., Weiser, U.: MultiAmdahl: How Should I Divide my Heterogeneous Chip?. IEEE Computer Architecture Letters 11(2), 65-68 (2012).
20. Morad, A., Morad, T.Y., Yavits, L., Ginosar, R., Weiser, U.: Generalized MultiAmdahl: Optimization of Heterogeneous Multi-Accelerator SoC. IEEE Computer Architecture Letters 99, 99-102 (2012).
21. Tang, S., Lee, B.S., He, B.: Speedup for Multi-Level Parallel Computing. Proc. IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum, pp. 537-546 (2012).
22. Woo D.H., Lee, H.S.: Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era. IEEE Computer 41(12), 24-31 (2008).
23. Cho, S., Melhem, R.G.: Corollaries to Amdahl's Law for Energy. IEEE Computer Architecture Letters 7(1), 25-28 (2008).
24. Cho, S., Melhem, R.G.: On the Interplay of Parallelization, Program Performance, and Energy Consumption. IEEE Transactions of Parallel and Distributed Systems 21(3), 342-353 (2010).
25. Londono, S.M., Gyvez, J.P.: Extending Amdahl's Law for Energy Efficiency. Proc. International Conference on Energy Aware Computing, pp. 1-4 (2010).
26. Marowka, A.: Extending Amdahl's Law for Heterogeneous Computing. Proc. IEEE International Symposium on Parallel and Distributed Processing with Applications, Leganes, pp. 309-316 (2012).
27. Ge, R., Cameron, K.: Power-Aware Speedup. Proc. IEEE International Parallel and Distributed Processing Symposium, pp. 1-10 (2007).
28. Ge, R., Cameron, K.: Generalizing Amdahl's Law for Power and Energy. IEEE Computer 45(3), 75-77 (2012).
29. Tseng, A.C.N., Brooks, D.: Analytical Latency-Throughput Model of Future Power Constrained Multicore Processors. ISCA Workshop on Energy-Efficient Design (WEED) (2012).
30. Cassidy, A.S., Andreou, A.G.: Beyond Amdahl's Law: An Objective Function That Links Multiprocessor Performance Gains To Delay and Energy. IEEE Transactions on Computers 61(8), 1110-1126 (2012).
31. Cui, Y., Wu, W., Guo, X., Chen, Y., Shi, Y.: Reinventing Lock Modeling for Multi-Core Systems. Proc. 18th IEEE/ACM Inter. Symp. On Modeling, Analysis and Simulation of Computer and Telecommunication Systems, pp. 455-457 (2010).
32. Eyerman, S., Eeckhout, L.: Modeling Critical Sections in Amdahl's Law and its Implications for Multicore Design. Proc. of the 37th Intern. Symp. on Computer Architecture, pp. 362-370 (2010).
33. Barroso L.A., Holzle. U.: The Case for Energy-Proportional Computing. IEEE Computer 40(12), 33-37 (2007).
34. Borkar, S., Chien, A.: The Future of Microprocessors. Communications of the ACM 54(5), 66-77 (2011).
35. Taylor, G.: The Next Decade of Computing. Proc. AIP Conference 1456, pp. 55-61 (2012).
36. Esmaeilzadeh, H., Blem, E., Amant, R.St., Sankaralingam, K., Burger, D.: Power Challenges May End the Multicore Era.

Communications of the ACM 56(2), 93-102 (2013).

37. Tian, W., Li, J., Zhao, Y., Xue, C.J., Li, M., Chen, E.: Optimal Task Allocation on Non-Volatile Memory Based Hyrbid Main Memory. Proc. IEEE 2011 ACM Symposium on Research in Applied Computation (RACS'11), pp. 1-6 (2011).
38. Yuan, L., Zhang, Y.: A Locality-based Performance Model for Load-and-compute Style Computation. Proc. IEEE International Conference on Cluster Computing, pp. 566-571 (2012).