Analyzing Performance Asymmetric Multicore Processors for Latency Sensitive Datacenter Applications

Vishal Gupta
Georgia Institute of Technology
vishal@cc.gatech.edu

Ripal Nathuji
Microsoft Research
ripaln@microsoft.com

Abstract

The semiconductor industry is continuing to harness performance gains through Moore's Law by developing multicore chips. While thus far these architectures have incorporated symmetric computational components, asymmetric multicore processors (AMPs) have been proposed as a possible alternative to improve power efficiency. To quantify the tradeoffs and benefits of these designs, in this paper we perform an opportunity analysis of performance asymmetric multicore processors in the context of datacenter environments where applications have associated latency SLAs. Specifically, we define two use cases for asymmetric multicore chips, and adopt an analytical approach to quantify gains in power consumption over area equivalent symmetric multicore designs. Based upon our findings, we discuss the practical merits of performance asymmetric chips in datacenters, including the issues that must be addressed in order to realize the theoretical benefits.

1 Introduction

Using Moore's Law to improve single core performance has been hindered by power and design complexity issues. To continue scaling performance, the industry is increasingly moving towards multicore architectures for both mobile and enterprise platforms. Multicore designs offer improved performance per Watt for workloads that can make use of multiple cores. This is often the case in datacenter environments where workloads perform parallel computations or utilize task-level parallel processing. Recognizing that power and cooling are key challenges in datacenters, there is a continued emphasis to improve power efficiency using multicore chips, including the pursuit of alternative architectural designs.

Asymmetric multicore processors (AMPs) have been proposed as an energy-efficient alternative to symmetric multicore processors (SMPs). AMPs are comprised of cores that differ in power characteristics due to performance or functional asymmetry. Performance asymmetry signifies that cores within a chip support identical instruction set architectures (ISAs), but exhibit different performance characteristics because of differences in issue width, number of functional units, etc. Conversely, functional asymmetry occurs when a subset of cores have different computational capabilities, exposed, for example, through ISA extensions.

In this paper, we begin to quantify the possible benefits of AMP architectures in datacenter environments, where workloads often have service level agreements (SLAs) defined in terms of request latency. Our goal is to better allow system designers to assess the tradeoffs and merits of moving from SMP systems, which are already well supported, to AMP architectures that require changes across both hardware and software. In this work, we limit our scope to performance asymmetry, and perform a theoretical analysis to estimate the power benefits of AMPs when compared to area equivalent SMP configurations. We begin by defining two possible ways of improving power efficiency using performance asymmetric multicores. We then evaluate each of these cases to better understand their relative merits. Our results provide a perspective on the practical benefits of these architectures in datacenters and also help to understand the qualitative issues and complexities that must be addressed in order to realize the gains in practice.

2 AMP Use Cases and Related Work

Prior work on AMPs motivates multiple usage models for exploiting performance asymmetric cores to improve power efficiency within a given resource budget (e.g. chip area). Figure 1 illustrates two scenarios considered in this paper, based upon whether requests for a latency sensitive application are processed using serial or parallel computations. *Energy scaling* (ES) is a technique that has been proposed to improve efficiency for serial com-

putations, while the coupling of large and small cores for parallel computations has been considered to realize *parallel speedup* (PS).

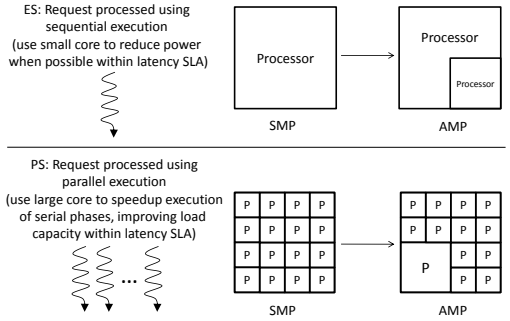


Figure 1: Illustrating AMP usage scenarios: Energy scaling (ES) and parallel speedup (PS).

Energy scaling (ES): Kumar et al. showed that by using a mix of cores with different power and performance characteristics, different phases within an application can be mapped to the core which can run it most efficiently [7]. For example, by running CPU intensive phases on faster large cores and memory intensive phases on slower small cores, overall power consumption can be reduced with minimal performance degradation. When latency SLAs are present, it is possible to use a smaller core even when execution time is impacted as long as the SLA is met. For example, at lower request loads when queuing delay is minimal, work can be offloaded to a smaller processor, allowing the larger core to go offline. In addition to considering energy scaling with latency SLAs, our analysis builds on prior results by accounting for the power cost of system components beyond the CPU, thereby introducing a tradeoff between 'race-to-halt' and increased execution time at a reduced power level [9].

Parallel speedup (PS): Another technique to exploit AMPs is to use them for speeding up serial portions of a parallel computation [1, 5]. During parallel phases, a request can be concurrently processed across many small cores. If available, a larger big core can be used to speedup serial phases. A theoretical analysis of such an approach has been presented by Hill and Marty [5], where they conclude that AMPs can provide significant speedup compared to area equivalent SMPs. As we show in Section 3, improved request processing time maps to the ability to support a higher overall throughput within the same latency. We extend the model developed by Hill and Marty with a queuing model to determine the overall power benefit of this effect.

Falling into one or both of the above two categories, many asymmetry-aware schedulers have been proposed in literature [6, 8, 10]. These systems are evaluated by measuring the benefits of including AMP-awareness when making scheduling decisions. They do not, however, provide insight into the relative benefits of a properly scheduled AMP architecture when compared to an area equivalent SMP alternative. Our goal is to perform such a comparison by quantifying the advantages of asymmetric architectures over symmetric multicore chips in datacenters.

3 Analytical Models

In this section, we provide a brief overview of the analytical modeling methodology used to evaluate the use cases outlined in Section 2.

M/M/1 queuing model: Metrics such as throughput per Watt have been used in the past to evaluate tradeoffs between symmetric and asymmetric architectures. We take a key departure from prior work in this regard by considering latency as a performance metric. In order to accomplish this, we utilize a simple queuing model that allows us to calculate the response time exhibited when processing requests as a function of computational capacity and request arrival rate. Specifically, we adopt the standard M/M/1 queuing model which assumes an exponentially distributed request inter-arrival time with mean $\frac{1}{\lambda}$, and a server which processes requests with an exponentially distributed service time with mean $\frac{1}{\mu}$. Based upon these assumptions, the expected average time spent in the system, including both service and queuing time, is provided by Equation 1.

$$E[T] = \frac{1}{\mu - \lambda} \tag{1}$$

We experimentally validate this equation using a simple benchmark that performs a parallelized computation (matrix multiplication) when processing a request. Requests arrive at a specified rate based upon an exponential distribution. We vary the service rate μ by changing the number of cores used to execute requests, as well as by varying computation size. We pick multiple combinations of arrival rates and service rates to achieve utilizations (defined as $\frac{\lambda}{\mu}$) of 25% and 75%.

Figure 2 compares the measured total response times of our experimental workload, where requests are queued and dispatched using Windows thread pooling, against theoretical curves based on Equation 1. We see that the expected queuing and response time effects of the theoretical model are indeed valid for real systems. Moreover, we observe that the determining factor for latency is not just processing capacity, but also the utilization that the system is run at as this directly impacts the queuing time experienced by requests. Thus, when a latency SLA must be met, it dictates a maximum arrival rate (or

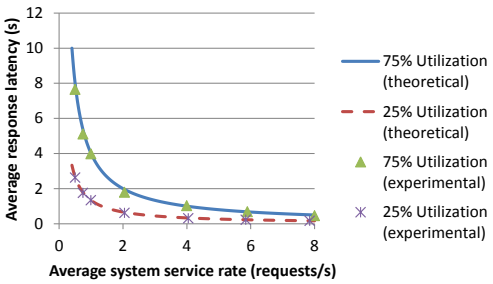


Figure 2: Experimental validation of the M/M/1 queuing model for a parallelized computation.

throughput) that can be supported given some service rate μ. We denote this maximum load as λ_{max}. For example, if we define the latency SLA to be T_{SLA}, λ_{max} can be calculated as shown in Equation 2.

$$\lambda_{max} = \mu - \frac{1}{T_{SLA}} \quad (2)$$

Equations 1 and 2 help illustrate the opportunities for our AMP use cases when observing latency constraints. First, when the system is experiencing less than peak load, there is a difference between T_{SLA} and $E[T]$ from Equation 1. The goal of ES is to utilize this "slack" to offload computation onto a smaller, lower power, processor for some fraction of the processing time spent on a request. How much of the processing that can be offloaded is of course a function of the performance impact of switching to the small core. On the other hand, the PS scenario allows for an increased μ and hence λ_{max}, thereby increasing the achievable utilization of the system while still meeting SLAs, and improving throughput capacity per Watt.

AMP performance modeling: In order to evaluate the performance tradeoffs across asymmetric options, we leverage the models used in prior work including that by Hill and Marty [5]. First, we presume that a core of area r has a normalized performance, $perf(r)$, of $\sqrt{r}$. For example, a core that takes up four times the area should provide twice the performance. This relationship allows us to model the worst case performance impact between a big and small core for the ES scenario. The actual impact, however, is workload specific as there are computation characteristics that may create reduced degradation. For example, memory bound computations tend to exhibit significantly less degradation than CPU bound computations. Prior work has described this tradeoff as computations having small core or large core bias [6].

In our analysis, we use a probability distribution function to capture the relative impact of computing a request on a smaller sized core. Figure 3 shows three cases that we consider in our analysis, where the small core bias distribution is centered around 25% of the worst case impact. For example, if the small core is one fourth the area of the large core, the execution time increase if computed completely on the small core would be 25% as opposed to the worst case of 100%. In our analysis, we assume that portions of the request are scheduled to the smaller core in an optimal way whenever possible under SLA constraints, and with zero overhead.

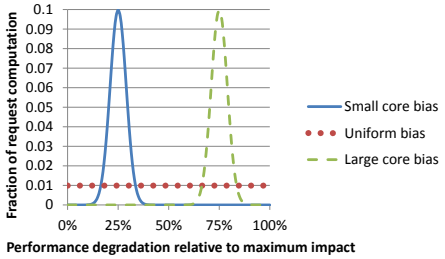


Figure 3: Bias profiles: Fraction of request computation that exhibits a particular performance impact.

The PS case, however, assumes a parallel computation. Hence, we need to understand the performance difference between an SMP and AMP based parallel computation, holding total area constant. Hill and Marty extended the well known formulation of Amdahl's law to capture tradeoffs across multicore designs. Specifically, if we define a total area budget of n, core sizes of area r, and f to be the fraction of a computation that can be parallelized, Equation 3 provides the speedup extended by an SMP multicore configuration. Similarly, if we assume an AMP chip composed of multiple small cores of equivalent area one, and one large core of area r, Equation 4 provides the achievable speedup.

$$Speedup_{SMP}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f*r}{perf(r)*n}} \quad (3)$$

$$Speedup_{AMP}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r)+n-r}} \quad (4)$$

Based upon the above equations, Figure 4 compares the performance of the best AMP configuration versus the best SMP configuration where the total area budget n is 64. The graph exhibits different curves where we limit the value of r, the area that can be consumed by a large core compared to the baseline small core of normalized area one. We observe a best case performance improvement of 85% with an appropriate f, and the ability to design a large core that takes up 32 times the area of a small core thereby providing a 5.7x performance improvement for serial phases.

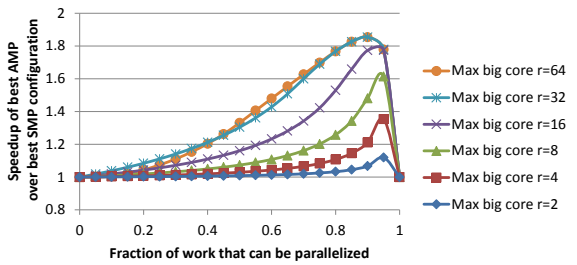


Figure 4: Comparing performance of AMP versus SMP multicore configuration with a total area budget of n=64.

Comparing AMP and SMP power consumption: Recognizing that datacenter systems have varying load across time [3], when comparing AMP and SMP designs for our two use cases, we estimate the power differences across load capacities up to a total provisioned capacity $\lambda_{provisioned}$. As shown by Equation 2, application latency SLAs along with computational capacity dictate the request throughput that can be supported by a given processor. Hence we assume that the load is scaled out equally across $\frac{\lambda_{provisioned}}{\lambda_{max}}$ compute elements of either design. At any given load, each of these elements consumes power according to Equation 5, which exemplifies a simple and *power-proportional* model.

$$P(\lambda) = \frac{\lambda}{\mu} * (P_{CPU} + P_{other}) + (1 - \frac{\lambda}{\mu}) * P_{Idle} \quad (5)$$

In our calculations, we assume P_{CPU} is proportional to area, and normalized to one for the SMP case for both ES and PS. We further assume that the power consumption of other system components during active periods, P_{other}, is one as well, and it scales linearly with CPU utilization. Finally, we use a value of 0.1 for P_{Idle}. In order to compute overall power savings, we weight the power difference calculated between AMP and SMP configurations across load from zero to $\lambda_{provisioned}$ using a load distribution based upon real data [3].

4 Evaluation

In this section, we present the benefits of AMPs over SMPs in terms of power savings while meeting a specified latency requirement.

Energy scaling: Figure 5 shows power savings as a function of the amount of area sacrificed for a small core, and the normalized latency SLA (higher values indicate larger latencies can be tolerated) for the three bias distributions in Figure 3. When the request computation has significant small core bias, we observe power savings of nearly 18% when 20% of the SMP core area is used for a smaller core. Interestingly, if we ignore the system power component P_{other}, the CPU power savings can be as high as 59%. When considering system overheads, though, the power savings is reduced since there is a cost of running at a higher utilization on the smaller core even though it consumes less power. This result is in line with prior work that highlights the tradeoff between CPU and system-level power reduction in the context of frequency scaling [9].

We observe that as we consider less ideal biases, savings drop to a maximum of 5%, and when requests have a strong large core bias, there is a power penalty of using the AMP configuration. Overall, we can conclude that energy scaling may realistically only provide limited benefits, and likely only for computations with strong small core bias.

Parallel speedup: Our final results consider the benefits of using AMP configurations over SMP configurations when requests are processed using parallel computations. Figure 6 provides the data from our analysis, where we calculate power savings as a function of f and the maximum possible size r of a large core in relation to the small core. For space we only provide results with a normalized SLA of two as larger SLAs exhibit similar, though slightly lower, savings.

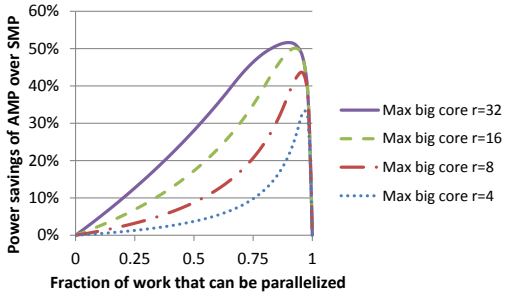
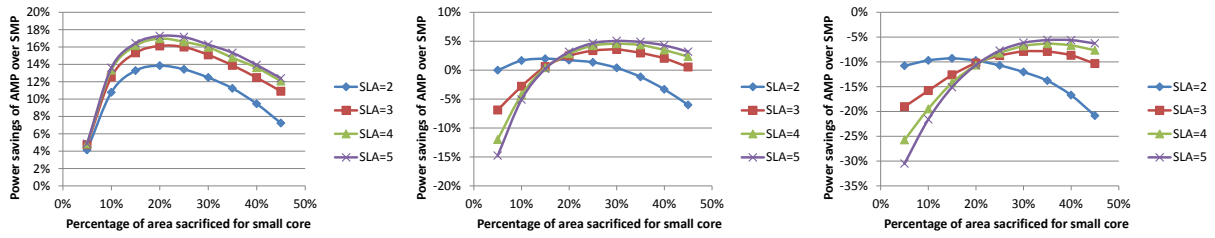


Figure 6: Power impact of using AMP configurations for parallel speedup (SLA=2).

We observe many interesting trends from the data in Figure 6. First, under certain conditions, power savings can be as high as 52% using an area equivalent AMP configuration. However, this requires the ability to design a large core that consumes 32 times the area to improve performance by a factor of 5.7 which may not be practical today. For example, Koufaty et al. from Intel use 3 as a reasonable value for r in their work [6], which reduces the maximum theoretically predicted savings by about 20%. Moreover, these savings are only significant for applications exhibiting a particular range of f. Overall, though, the parallel speedup scenario is significantly more promising than the energy scaling use case for AMPs.



(a) Small core bias. (b) Uniform bias. (c) Large core bias.

Figure 5: Power impact of using AMP configurations for energy scaling.

5 Discussion and Future Work

Power savings results shown in Section 4 are derived from our theoretical analysis. Therefore, we must temper these numbers by considering aspects of our modeling that may reduce savings in practice, as well as dependencies that must be met in order to apply these methods.

First, Amdahl's law assumes unbounded scalability of parallel applications. However, this is generally not true for real workloads. For example, a recent study from Eyerman et al. shows that when critical sections are incorporated into Amdahl's law, the relative performance benefits of AMPs over SMPs is reduced, and in some cases AMPs even provide worse performance [4]. Second, our theoretical model assumes perfect and overhead free migration between cores. Again, in real systems this may not occur. Finally, achieving higher power savings for PS requires large values of r (ratio of the big core to the small core area) which are not practical today, and these savings are significant only for applications having a particular range of f (fraction of computation that can be parallelized). Hence, all of these assumptions will bring down actual power savings in practice.

Moreover, Balakrishnan et al. showed that certain applications behave unpredictably when run on AMPs even after adding asymmetry-awareness to the operating system scheduler [2]. This brings additional challenges for adopting AMPs in datacenters as predictable application behavior is critical for meeting application SLAs.

We also observe that our PS results are based upon workloads whose requests are executed as parallel computations. This is different than the task level parallelism that is typical of, for example, web servers, where additional cores improve throughput but not latency. Hence, in order to exploit AMPs, software developers of enterprise applications must place emphasis on parallelizing the computations performed to process a request.

In summary, this paper presented an opportunity analysis of AMPs for datacenter applications with SLAs. We considered two use cases of AMPs, i.e., energy scaling and parallel speedup. Our results strongly indicate that of the two use cases, PS is the more promising avenue for AMPs as ES becomes less rewarding due to CPU power becoming a smaller component of the overall system power. Our opportunity analysis indicates benefits of up to 52% in power consumption for PS, however, there are practical considerations which must be addressed in order to reap full benefit out of them.

As future work, we plan to extend our analysis to consider the tradeoffs in using chip area for functional asymmetry. This includes the use of accelerators and heterogeneous multicore configurations including programmable processors that can provide a significant benefit in terms of speedup per area and power, but may only be used for a fraction of the request execution time.

References

[1] ANNAVARAM, M., GROCHOWSKI, E., AND SHEN, J. Mitigating amdahl's law through epi throttling. *SIGARCH Comput. Archit. News 33*, 2 (2005), 298–309.

[2] BALAKRISHNAN, S., RAJWAR, R., UPTON, M., AND LAI, K. The impact of performance asymmetry in emerging multicore architectures. *SIGARCH Comp. Arch. News 33* (2005), 506–517.

[3] BARROSO, L. A., AND HOLZLE, U. The case for energy-proportional computing. *Computer 40* (2007), 33–37.

[4] EYERMAN, S., AND EECKHOUT, L. Modeling critical sections in amdahl's law and its implications for multicore design. In *ISCA* (Saint-Malo, France, June 2010).

[5] HILL, M. D., AND MARTY, M. R. Amdahl's law in the multicore era. *Computer 41* (2008), 33–38.

[6] KOUFATY, D., REDDY, D., AND HAHN, S. Bias scheduling in heterogeneous multi-core architectures. In *EuroSys '10* (2010).

[7] KUMAR, R., FARKAS, K. I., JOUPPI, N. P., RANGANATHAN, P., AND TULLSEN, D. M. Single-isa heterogeneous multi-core architectures: The potential for processor power reduction. In *MICRO* (2003).

[8] LI, T., BAUMBERGER, D., KOUFATY, D. A., AND HAHN, S. Efficient operating system scheduling for performance-asymmetric multi-core architectures. In *SC '07* (2007).

[9] MIYOSHI, A., LEFURGY, C., VAN HENSBERGEN, E., RAJAMONY, R., AND RAJKUMAR, R. Critical power slope: Understanding the runtime effects of frequency scaling. *SC '02* (2002).

[10] SAEZ, J. C., PRIETO, M., FEDOROVA, A., AND BLAGODUROV, S. A comprehensive scheduler for asymmetric multicore systems. In *EuroSys '10* (New York, NY, USA, 2010), ACM, pp. 139–152.