Amdahl's Law for Predicting the Future of Multicores Considered Harmful

B.H.H. Juurlink
Berlin University of Technology
Berlin, Germany
b.juurlink@tu-berlin.de

C.H. Meenderinck
IntelliMagic
Leiden, The Netherlands
cor.meenderinck@intellimagic.net

ABSTRACT

Several recent works predict the future of multicore systems or identify scalability bottlenecks based on Amdahl's law. Amdahl's law implicitly assumes, however, that the problem size stays constant, but in most cases more cores are used to solve larger and more complex problems. There is a related law known as Gustafson's law which assumes that runtime, not the problem size, is constant. In other words, it is assumed that the runtime on p cores is the same as the runtime on 1 core and that the parallel part of an application scales linearly with the number of cores. We apply Gustafson's law to symmetric, asymmetric, and dynamic multicores and show that this leads to fundamentally different results than when Amdahl's law is applied. We also generalize Amdahl's and Gustafson's law and study how this quantitatively effects the dimensioning of future multicore systems.

1. INTRODUCTION

Several recent works predict the future of multicore systems or identify scalability bottlenecks based on Amdahl's well-known law [1]. For example, Hill and Marty [11] extended Amdahl's law with an area-performance model and applied it to symmetric, asymmetric, and dynamic multicore chips. Their results show that obtaining optimal multicore performance requires extracting more parallelism as well as making sequential cores faster. Basically, sequential cores need to be made relatively larger and hence faster to execute the serial part of an application faster, since the serial part will eventually dominate when the number of cores increases. Based on Amdahl's law they also showed that dynamic multicores that can dynamically combine all resources to form one large, sequentially cores provide the optimimal solution. Based on Hill and Marty's findings, two dynamic multicore designs were presented at ISCA 2010: WiDGET [18] and Forwardflow [7]. Other examples of the use of Amdahl's law include the work of Eyerman and Eeckhout [6], who show that parallel speedup is not only limited by the serial fraction but also by synchronization through critical sections, and the work of Cho and Melhem [4], who use Amdahl's law to determine the optimal processor frequencies in the serial and parallel regions with the goal of minimizing the total energy consumption.

The implicit assumption in Amdahl's law as well as the extensions mentioned above is, however, that the problem size remains constant. As observed by Gustafson [10], this is virtually never the case. More cores are used to solve larger and more complex problems. For example, more cores are used to perform weather forecasting on a larger area (e.g., [12]) or for video decoding with higher resolutions [3]. Furthermore, for many applications, when the problem size scales the parallel part scales faster than the serial part. Gustafson therefore proposed an alternative to Amdahl's law which is now known as Gustafson's law but which he himself attributed to E. Barsis. Gustafson's law is much more optimistic than Amdahl's law. While Hill and Marty briefly mention Gustafson's law, they state that in their view multicore designs should also operate well under Amdahl's more pessimistic assumptions and do not consider it further.

The main contribution of this work is two-fold. First we generalize 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 (e.g., it is proportional to $\sqrt{n}$, where n is the number of cores). We refer to this equation as the *generalized scaled speedup equation* (GSSE), since it encompasses both Amdahl's and Gustafson's law by substituting the appropriate application scaling function. Second, we apply Gustafson's law and the GSSE to symmetric, asymmetric, and dynamic multicores and show that they produce results that are fundamentally different from the results obtained by Hill and Marty based on Amdahl's law. Our results have several important implications of which we mention three. First, while Amdahl's law indicates that symmetric multicore processors should consist of fewer but larger and more powerful cores, Gustafson's law suggests that many tiny cores yield the highest performance. The GSSE indicates that fewer, more powerful cores can deliver a performance improvement, but the improvement is much smaller and only occurs for smaller parallel fractions f than when Amdahl's law is assumed. Second, when the serial fraction is only 1%, Amdahl's law implies that one-eighth of the area of a large asymmetric multicore chip should be devoted to a large, high-performance core, while the remaining area is devoted to many small cores. Gustafson's law, on the other hand, indicates that in this case an asymmetric design barely performs better than a symmetric design, while the GSSE indicates that only 3.1% of the area should be devoted to the large core and, even then, the asymmetric design is only 7.8% faster (in theory) than a symmetric design. Third, under Amdahl's law the speedup of a dynamic multicore that can contain up to 256 simple cores is limited to 30.1. Gustafson's law, on the other hand, shows that a speedup of 242 can be achieved, while the GSSE which assumes

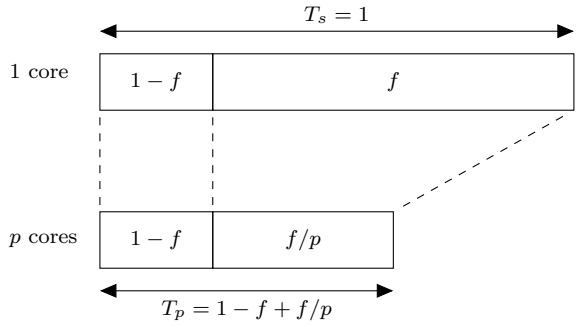


Figure 1: Illustration of Amdahl's law. Amdahl's assumes that the normalized runtime on one core is 1.

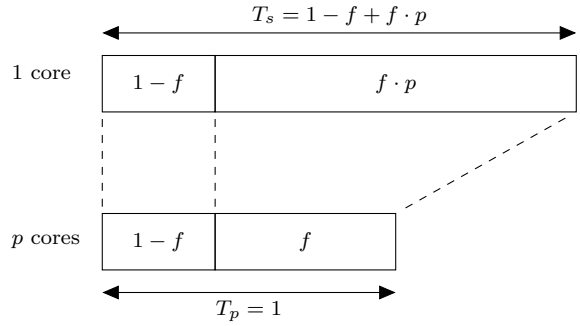


Figure 2: Illustration of Gustafson's law. Gustafson assumes that the normalized runtime on p cores is 1.

less than perfect application scaling, indicates that still a speedup of 136 can be achieved.

This paper is organized as follows. Section 2 reviews Amdahl's and Gustafson's law and explains their underlying assumptions. In addition, it presents the generalized scaled speedup equation that subsumes both Amdahl's and Gustafson's law. Section 3 applies Amdahl's and Gustafson's law and the generalized scaled speedup equation to symmetric, asymmetric, and dynamic multicore chips using the area-performance model proposed by Hill and Marty, and presents and discusses the analytical speedup results. Conclusions are drawn in Section 4.

2. AMDAHL'S AND RELATED LAWS

Amdahl's law assumes that a fraction f of a serial program's execution time is perfectly parallelizable with no communication and synchronization overhead, while the remaining fraction $s = 1 - f$ is totally sequential. Consequently, if T_s is the execution time of the serial program, the parallel execution time on p cores, $T(p)$, is given by

$$T(p) = (1 - f) \cdot T_s + f \cdot T_s/p,$$

since only a fraction f of the serial program's execution time is parallelizable. Speedup is the ratio of sequential execution time to parallel execution time, giving

$$S^{Amdahl} = \frac{T_s}{T(p)} = \frac{T_s}{(1 - f) \cdot T_s + \frac{f \cdot T_s}{p}} = \frac{1}{(1 - f) + \frac{f}{p}}. \quad (1)$$

The last expression in this equation goes to $1/(1 - f)$ when p goes to infinity. So, for example, when the serial fraction $s = 1 - f$ is 20%, the speedup is limited to 5, no matter how many cores are employed. Amdahl used this equation to argue for the validity of the single-processor approach. Amdahl's law is illustrated in Figure 1.

Amdahl's equation assumes, however, that the problem size does not change when using more cores to execute the application. In other words, the parallelizable fraction remains constant, no matter how many cores are employed. As observed by Gustafson, this is very rare. One does not take a fixed-sized problem and run it on as many cores as possible. In virtually all application domains, more cores are used to solve larger and more complex problems. Gustafson therefore argued that it is more realistic to assume that runtime, not problem size, is constant.

Gustafson's law is illustrated in Figure 2. Gustafson assumes that the normalized runtime **on p cores** is $(1 - f) + f = 1$. If $(1-f)+f$ is the runtime on p cores, the runtime on one core will be $(1-f)+p \cdot f$. Consequently, the speedup according to Gustafson (which he referred to as *scaled speedup*) is given by

$$S^{Gustafson} = \frac{(1 - f) + f \cdot p}{1 - f + f} = (1 - f) + f \cdot p. \quad (2)$$

In this equation, if the serial fraction $(1 - f)$ is 20%, the speedup will be 80.2 on 100 cores, which is much more optimistic than the speedup of 4.8 predicted by Amdahl's law.

Figure 3, based on [17], illustrates the differences between Amdahl's and Gustafson's law. Amdahl assumes that the amount of work that can be parallelized, W_p, is constant and independent of the number of cores p. This can be considered overly pessimistic. Gustafson assumes that the amount of work that can be parallelized grows linearly with the number of cores p. This, on the other hand, can be considered overly optimistic. For example, although video coding is a domain that can take benefit from multi- and many-cores [3], the resolution and computational requirements will not grow indefinitely.

To address the limitations of Amdahl's and Gustafson's law, we propose an equation that is somewhere in between. In this equation the amount of work that can be parallelized is not constant as in Amdahl's law, nor does it grow linearly with the number of cores as in Gustafson's law. Instead, the amount of parallel work is proportional to a function $scale(p)$ that is sub-linear in p (e.g., $scale(p) = \sqrt{p}$). Consequently, the normalized execution time on p cores is given by

$$(1 - f) + \frac{f \cdot scale(p)}{p}.$$

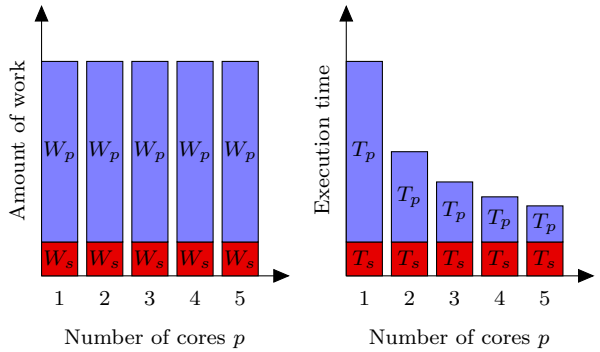
If this equation gives the normalized runtime on p cores, the runtime on one core will be

$$(1 - f) + f \cdot scale(p).$$

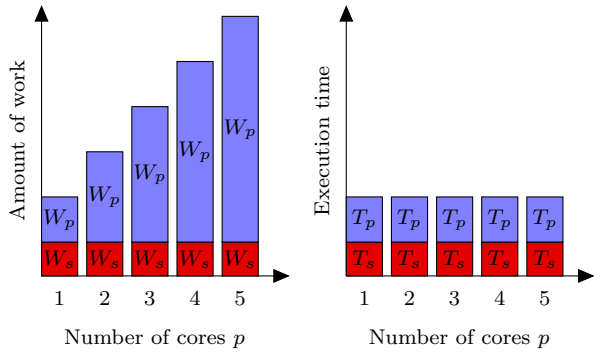
The speedup is therefore given by

$$S^{General} = \frac{(1 - f) + f \cdot scale(p)}{(1 - f) + \frac{f \cdot scale(p)}{p}}. \quad (3)$$

Note that when $scale(p) = 1$, this equation is identical to Amdahl's law, and when $scale(p) = p$, it is identical to Gustafson's law. The precise scaling function is, of course,



(a) Amdahl's assumption.



(b) Gustafson's assumption.

Figure 3: Amdahl's and Gustafson's assumption. Amdahl assumes the input size (or amount of work) to be constant, while Gustafson assumes it to be dependent on N.

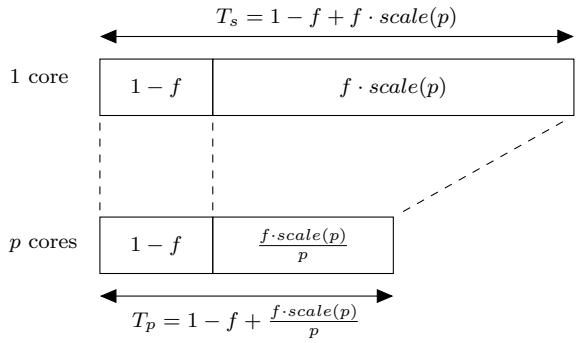


Figure 4: Illustration of the generalized scaled speedup equation.

application dependent. We refer to this equation as the generalized scaled speedup equation (GSSE). The GSSE is illustrated in Figure 4.

Figure 5 plots the speedups given by Amdahl's law, Gustafson's law, and the GSSE for $f = 0.5$ and $f = 0.9$ (where we assume that $scale(p) = \sqrt{p}$, which will be done throughout this article). While Amdahl's law indicates that the maximum speedups that can be achieved are 2 and 10 for $f = 0.5$ and $f = 0.9$, respectively, the speedups on 100 cores obtained using Gustafson's law are 50.5 and 90.1, and the speedups on 100 cores calculated using the GSSE are 10 and 47.9. We note, however, that it is misleading to plot these functions in a single figure, since the underlying assumptions differ. For example, a value of f of 0.5 on 100 cores in Gustafson's law implies that the parallel fraction *on a single core* is 100 times as large as the serial fraction. This corresponds to a parallel fraction f of $100/101 = 99\%$ in Amdahl's law.

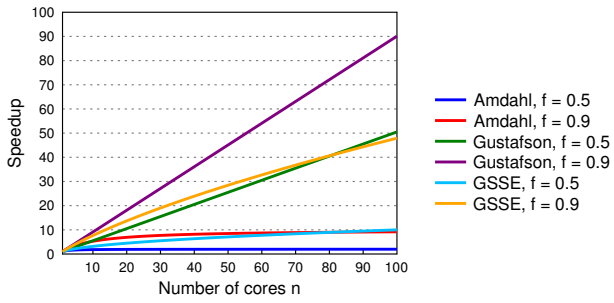


Figure 5: Speedup as a function of the number of cores for $f = 0.5$ and $f = 0.9$ assuming Amdahl's law, Gustafson's law, and the GSSE.

3. IMPLICATIONS FOR MULTICORE DESIGN

Hill and Marty [11] used Amdahl's law to make assertions about the organization of multicore chips. In particular, should a multicore chip consist of many small and simple cores, a few large, high-performance cores, or some mixture of both? To do so, an area-performance model is needed to estimate the number of cores a chip can contain, and the performance of a core as a function of its size (in number of transistors).

Like Hill and Marty, we assume that a multicore chip of a given size implemented in a given technology node can contain n *Base Core Equivalents* (BCEs) and each BCE has a performance of 1. Furthermore, a core with an area of r BCEs has a performance of $perf(r)$, where $perf(r)$ is between 1 and r. $perf(r)$ can be an arbitrary function, but like Hill and Marty, in all speedup graphs we assume $perf(r) = \sqrt{r}$, which corresponds to Pollack's / Borkar's rule [15, 2].

3.1 Symmetric Multicores

In a *symmetric* multicore all cores have the same size and performance. So a symmetric multicore chip of n BCEs can contain n/r cores of r BCEs each, and the performance of each core is $perf(r)$. Under Hill and Marty's collorary of Amdahl's law, the speedup of a symmetric multicore over a single-BCE core is a function of the fraction that is parallelizable (f), the chip area in BCEs (n), and the size of each core in BCEs (r). The serial part $1 - f$ is executed sequentially by one core at performance $perf(r)$, and the parallel part f is executed in parallel by all n/r cores, each with a performance of $perf(r)$. Using this reasoning Hill and Marty obtained the following equation for symmetric multicores:

$$S^{Amdahl}_{symmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r)\cdot n/r}} = \frac{1}{\frac{1-f}{perf(r)} + \frac{f\cdot r}{perf(r)\cdot n}}. \quad (4)$$

We refer to this equation as Amdahl's law for symmetric multicores.

The same reasoning can be applied to Gustafson's law. We use one core to execute the serial part $1-f$ and all n/r cores to execute the parallel part. The parallel part, however, is now larger. The unnormalized parallel fraction on a single core is not f but $f \cdot n$, since the speedup is relative to a 1-BCE core. Consequently, we obtain:

$$S^{Gustafson}_{symmetric}(f,n,r) = \frac{1-f+f\cdot n}{\frac{1-f}{perf(r)}+\frac{f\cdot r}{perf(r)}} = \frac{(1-f+f\cdot n)\cdot perf(r)}{1-f+f\cdot r}. \quad (5)$$

This equation will be referred to as Gustafson's law for symmetric multicores.

Similarly, for the GSSE we obtain

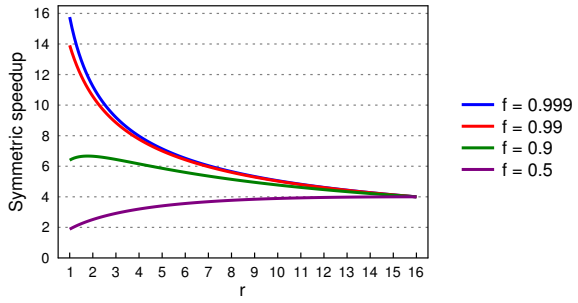
$$S^{General}_{symmetric}(f,n,r) = \frac{1-f+f\cdot scale(n)}{\frac{1-f}{perf(r)}+\frac{f\cdot scale(n)}{n/r\cdot perf(r)}} = \frac{(1-f+f\cdot scale(n))\cdot perf(r)}{1-f+\frac{f\cdot r\cdot scale(n)}{n}}. \quad (6)$$

Again we observe that if $scale(n)=1$, this equation is identical to Amdahl's law for symmetric multicores (Eq. (4)), and when $scale(n)=n$, it is identical to Gustafson's law for symmetric multicores (Eq. (5)).

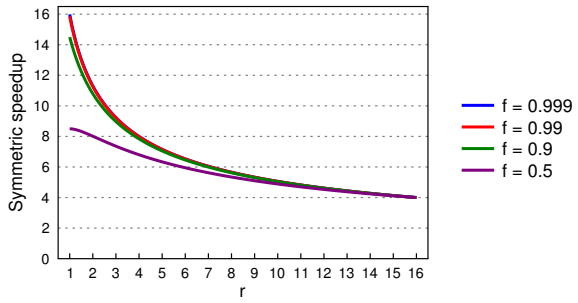
Figure 6(a) depicts the results obtained using Amdahl's law for symmetric multicores for $n=16$ and as a function of the size of each core r. The results obtained using Gustafson's law and the GSSE for symmetric multicores are shown in Figure 6(b) and Figure 6(c), respectively. In all figures we assume that $perf(r)=\sqrt{r}$ and that $scale(n)=\sqrt{n}$.

The results indicate that when application scaling is considered leads to fundamentally different results than when Amdahl's law is applied. For example, Amdahl's law for symmetric multicores indicates that using 16 cores of 1 BCE each is not necessarily the optimal solution, depending on the parallel fraction f. When $f=0.5$, the optimal solution under Amdahl's law is to use a single large core that occupies all chip resources. In fact, it can be shown analytically that when $f \leq 0.5$, a single large core will always achieve the highest performance under Amdahl's law for symmetric multicores, independent of n, since $r \leq n$. Even when $f=0.9$, the optimal solution is not a single large core, but 8 cores of 2 BCEs each. On the other hand, Gustafson's law for symmetric multicores indicates that 16 1-BCE cores achieve the highest performance for each value of $f \geq 0.5$. Thus, under the assumption that the parallel fraction scales linearly with the chip area n, many simple cores is the best approach. The impression changes slightly when non-perfect application scalability is assumed (Figure 6(c)). For $f \geq 0.9$, many simple cores is still the best approach, but for $f=0.5$, 4 cores of 4 BCEs each perform slighlty better. However, the performance difference between the optimal organization and 16 cores of 1 BCE each is much smaller when the GSSE is assumed (with $scale(n)=\sqrt{n}$) than suggested by Amdahl's law for symmetric multicores (1.25x versus 2.125x).

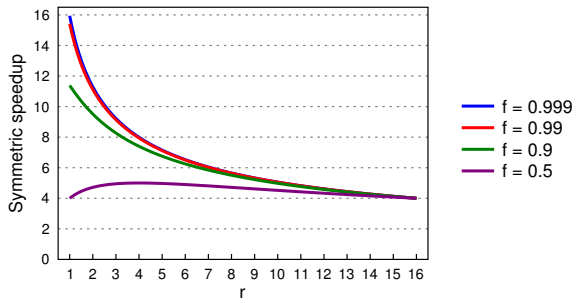
Based on Amdahl's law for symmetric multicores, Hill and Marty concluded that researchers should seek methods of increasing core performance even at high cost. While we agree with this conclusion, this conclusion cannot be drawn solely on the basis of Amdahl's law. Rather, it is likely that



(a) Amdahl, $n=16$.



(b) Gustafson, $n=16$.

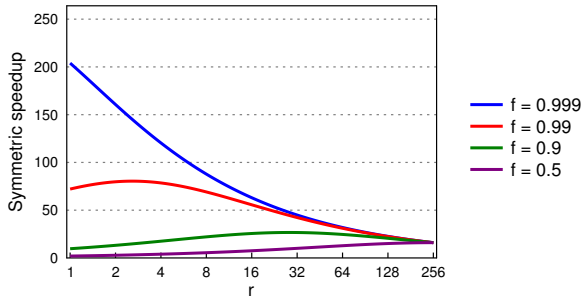


(c) GSSE, $n=16$.

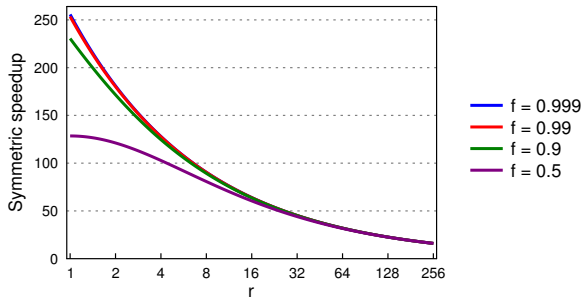
Figure 6: Speedup of symmetric multicores of $n=16$ BCEs over a single-BCE core assuming Amdahl's law, Gustafson's law, and the GSSE.

there will be abundant single-threaded legacy codes for a few decades to come.

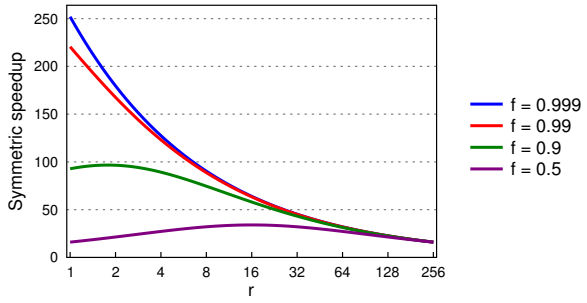
An important question is how these results change when Moore's law allows many more BCEs per chip. To answer this question, Figure 7 depicts the results for $n=256$ BCEs. While the precise results are slightly different, in general the same conclusions can be drawn. While Amdahl's law for symmetric multicores indicates that 256 single-BCE cores never yield the highest performance (unless when $f=0.999$), Gustafson's law suggests the opposite. Furthermore, assuming less that perfect scaling of the parallel part, the GSSE shows that now non-single BCE cores can provide a performance advantage, not only for $f=0.5$ but also for $f=0.9$, but for $f=0.5$ the performance advantage is much smaller than predicted by Amdahl's law (2.13x versus 8.03x),



(a) Amdahl, $n = 256$.



(b) Gustafson, $n = 256$.



(c) GSSE, $n = 256$.

Figure 7: Speedup of symmetric multicores of $n = 256$ BCEs over a single-BCE core assuming Amdahl's law, Gustafson's law, and the GSSE.

and for $f = 0.9$ the performance advantage is really minor (less than 1.04x). It is questionable, however, if large symmetric multicores should be organized in such a way as to obtain optimal performance for applications with a parallel fraction f of 0.5.

3.2 Asymmetric Multicores

In an *asymmetric* multicore, one or more cores are larger and more powerful than the others. Asymmetric multicores are also called performance heterogeneous multicores, since all cores implement the same instruction set architecture (ISA) but at different performance levels. Besides performance heterogeneous, there are functionally heterogeneous multicores, where different cores support different ISAs. An example of an asymmetric multicore is the single-ISA heterogeneous multicore proposed by Kumar et al. [13]. Examples of functionally heterogeneous multicores in industry and academia are the Cell processor [9] and the SARC architecture [16].

Amdahl's law makes a case for asymmetric multicores with one large core to accelerate the serial part of the execution time of an application, while many small cores are used to execute the parallel part. In this section we investigate if the same holds under the assumptions of Gustafson's law and the GSSE.

Under Amdahl's law for asymmetric multicores, the serial part is executed by one large core of size r BCEs and performance $perf(r)$. The parallel part, on the other hand, is executed by both the $n - r$ single-BCE cores and the large core. Overall, this yields

$$S^{Amdahl}_{asymmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{perf(r)+n-r}}. \quad (7)$$

Note that this equation assumes that perfect dynamic scheduling is performed, i.e, the large core performs a larger part of the parallel fraction than the single-BCE cores.

Similarly, we obtain

$$S^{Gustafson}_{asymmetric}(f,n,r) = \frac{1-f+f\cdot n}{\frac{1-f}{perf(r)} + \frac{f\cdot n}{perf(r)+n-r}} \quad (8)$$

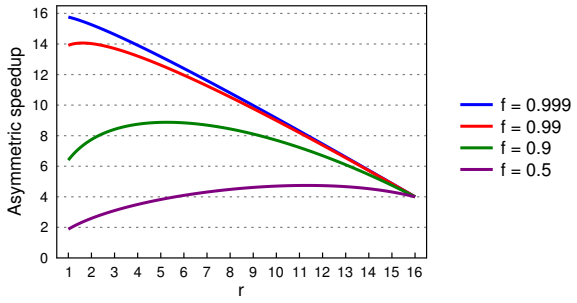
and

$$S^{General}_{asymmetric}(f,n,r) = \frac{1-f+f\cdot scale(n)}{\frac{1-f}{perf(r)} + \frac{f\cdot scale(n)}{perf(r)+n-r}} \quad (9)$$

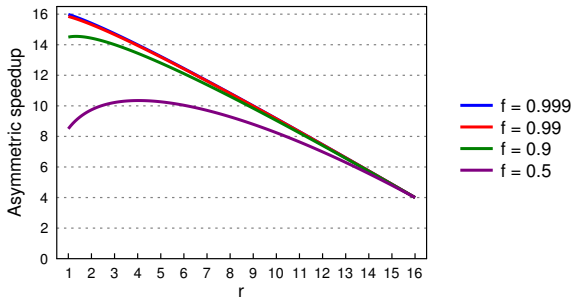
These equations will be referred to as Gustafson's law and the GSSE for asymmetric multicores, respectively.

Figure 8 depicts the speedup attained by asymmetric multicores over a single-BCE core for $n = 16$ BCEs, while Figure 9 depicts the speedup curves for $n = 256$. Overall, the results indicate that asymmetric multicores indeed provide a performance advantage compared to symmetric multicores, but when application scaling is considered, the performance advantage is smaller, occurs for smaller values of f, and the optimal size of the large core is smaller than under Amdahl's law for assymetric multicores. For example, when $n = 16$ and assuming perfect application scaling (Gustafson's law), Figure 8(b) shows that asymmetric multicores only provide a performance advantage for $f = 0.5$ and in that case the performance benefit (compared to when $r = 1$) is limited to 1.22x. When non-perfect application scaling is assumed, Figure 8(c) shows that asymmetric multicores also improve performance for $f = 0.9$, but the performance benefit is small (1.06x). When $f = 0.5$, the performance benefit predicted by the GSSE for asymmetric multicores is larger, but smaller than predicted by Amdahl's law for asymmetric multicores (1.73x versus 2.38x).

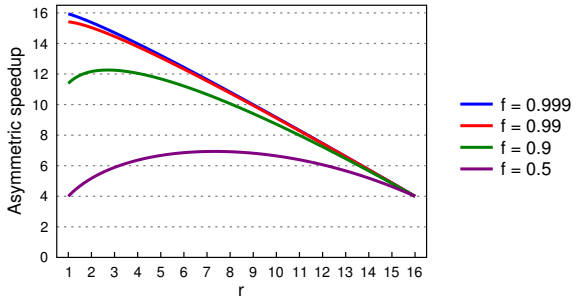
When $n = 256$, the speedup benefits of asymmetric multicores increase, but still the conclusions are different when application scaling is considered. As depicted in Figure 9(b), Gustafson's law indicates that asymmetric designs still only provide a substantial performance improvement for $f = 0.5$, while Amdahl's law suggests that they provide substantial improvements for any value of f. The GSSE, on the other hand, indicates that asymmetric multicores of $n = 256$ BCEs also yield substantially higher performance for $f = 0.9$, but unlike Amdahl's law for asymmetric multicores, not for $f = 0.99$. The equations that assume application scaling also suggest that the optimal size of the large core is smaller



(a) Amdahl, $n = 16$.

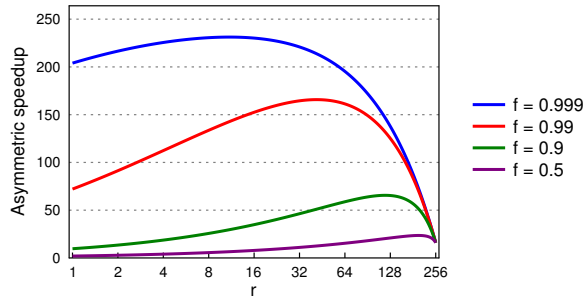


(b) Gustafson, $n = 16$.

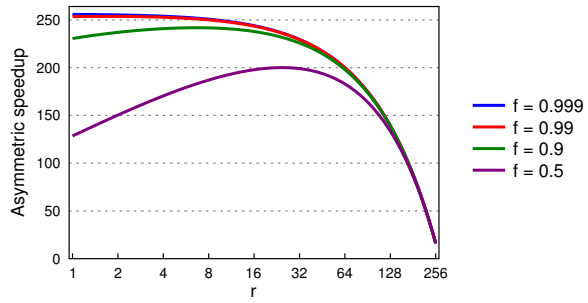


(c) GSSE, $n = 16$.

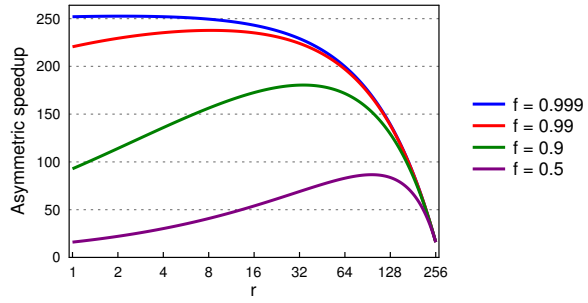
Figure 8: Speedup of asymmetric multicores assuming Amdahl's law, Gustafson's law, and the GSSE for $n = 16$ BCEs.



(a) Amdahl, $n = 256$.



(b) Gustafson, $n = 256$.



(c) GSSE, $n = 256$.

Figure 9: Speedup of asymmetric multicores assuming Amdahl's law, Gustafson's law, and the GSSE for $n = 256$ BCEs.

than the size predicted by Amdahl's law. For example, when $f = 0.5$, Amdahl's law indicates that half the chip resources (128 BCEs) should be devoted to the large core and still the speedup is limited to 20.9. Gustafson's law (Figure 9(b)), on the other hand, indicates that when the large core is 16 BCEs (6.3% of the chip resources), near optimal performance and a speedup of 198 are achieved. This is a much more optimistic result, since it is questionable if a core that is 128 times larger than a base core can and should be built with a performance that is 11.3 times as high. It is even more questionable if the chip area should be statically divided such that optimal performance is achieved for poorly scalable applications with a serial fraction as large as 0.5, especially considering that such a static division will hurt the performance of applications with larger parallel fractions. Under the GSSE when $f = 0.9$, an asymmetric multicore with a large core of 16 BCEs performs slightly worse (1.05x) than the optimal design (large core of 32 BCEs).

3.3 Dynamic Multicores

In a dynamic multicore, it is assumed that up to r cores can be temporarily aggregated to accelerate the sequential components of an application. During the parallel phases, the resources are divided into n 1-BCE cores again to attain maximum speedup during the parallel phases. As indicated by Hill and Marty, helper threads conceptually boost the performance of a single core, since the helper threads may e.g. prefetch data needed by the sequential main thread. Furthermore, two dynamic multicore designs were presented at ISCA 2010: WiDGET [18] and Forwardflow [7].

Amdahl's law suggests that we should use a large dynamic core of r BCEs during the serial phases and n single-BCE cores during the parallel phases. The speedup achieved by such a dynamic multicore processor over a single-BCE core is given by:

$$S^{Amdahl}_{dynamic}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{n}}. \quad (10)$$

Gustafson's law and the GSSE for dynamic multicores are obtained similarly:

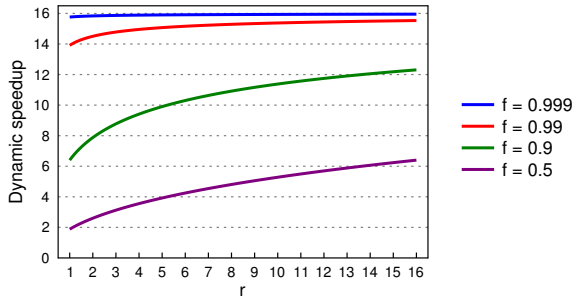
$$S^{Gustafson}_{dynamic}(f,n,r) = \frac{1-f+f\cdot n}{\frac{1-f}{perf(r)} + \frac{f\cdot n}{n}} = \frac{1-f+f\cdot n}{\frac{1-f}{perf(r)} + f} \quad (11)$$

and

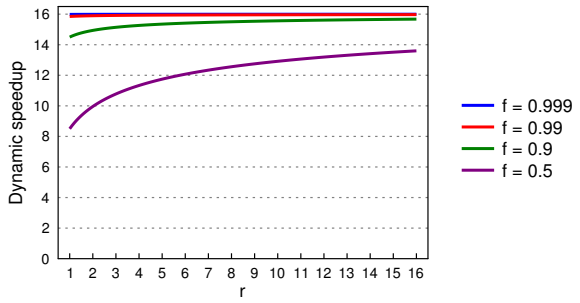
$$S^{General}_{dynamic}(f,n,r) = \frac{1-f+f\cdot scale(n)}{\frac{1-f}{perf(r)} + \frac{f\cdot scale(n)}{n}}. \quad (12)$$

Figure 10 (for $n = 16$ BCEs) and Figure 11 (for $n = 256$ BCEs) depict the speedups calculated using these equations as a function of the size (in BCEs) of the large dynamic core. While these figures display the results for up to a large dynamic core of n BCEs, practical considerations might keep r much smaller than n. Obviously, in all cases the speedup increases with the size of the large dynamic core. But similar to asymmetric designs, the advantage of a dynamic multicore is more pronounced under Amdahl's law than under Gustafson's law and the GSSE. For example, when $n = 16$, Gustafson's law and the GSSE indicate that dynamic multicores only provide a significant (more than 20%) performance improvement when $f = 0.5$. A large analytical performance improvement is important, since a dynamic multicore naturally incurs a higher overhead than asymmetric designs, as additional data paths are needed to be able to aggregate several cores. Even when $n = 256$, if perfect application scaling is assumed (Figure 11(b)), dynamic multicores only provide a significant performance improvement for $f = 0.5$. When less than perfect application scaling is assumed (Figure 11(c)), dynamic multicores also provide a significant advantage for $f = 0.9$, but if the dynamic large core consists of 16 BCEs with a performance that is 4 times higher than that of a single BCE, a speedup of 170.5 is attained (versus 232 for $r = 256$), which is much better than the speedup of 35.1 for $r = 16$ (versus 102 for $r = 256$) achieved under Amdahl's assumptions. This also has a positive implication, since it is questionable if a dynamic large core of 256 BCEs with a performance that is 16 times higher than that of a single BCE can be constructed. If we optimistically assume that it takes one cycle to route a signal through one BCE and that the BCEs have to be layed out in 2D space, it takes at least 16 cycles to route an operation from the middle of the chip to a functional unit of a core at the corner. It is doubtful if such large operation latencies can be completely hidden.

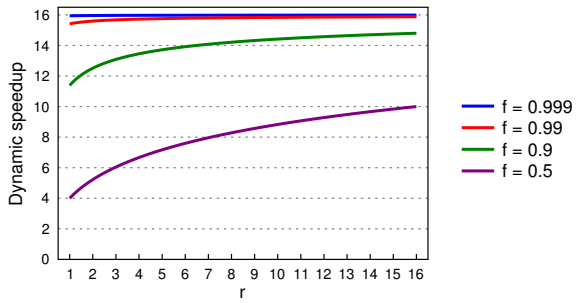
Figure 12 depicts the speedup of the optimal dynamic design (with a dynamic large core of $r = n$ BCEs) over the optimal asymmetric design. It is easy to see that this can be at most 2, since we can devote half of the asymmetric multicore chip to the large core and the other half to $n/2$ small (single-BCE) cores. Not suprisingly, under all three scaling equations, dynamic multicores provide a performance advantage over asymmetric designs (by at most 1.63x). It remains to be seen, however, if this is achievable in practice



(a) Amdahl, $n = 16$.



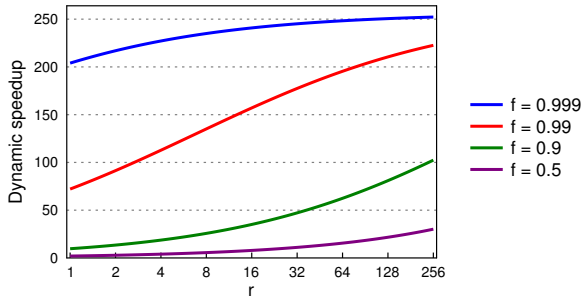
(b) Gustafson, $n = 16$.



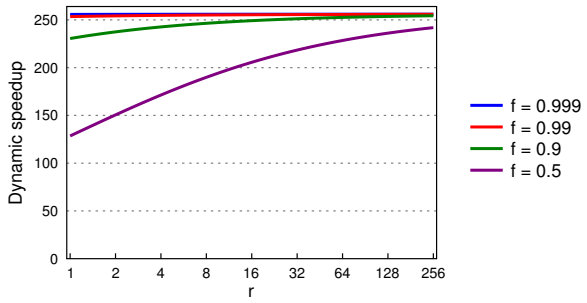
(c) GSSE, $n = 16$.

Figure 10: Speedup of dynamic multicores assuming Amdahl's law, Gustafson's law, and the GSSE for $n = 16$ BCEs.

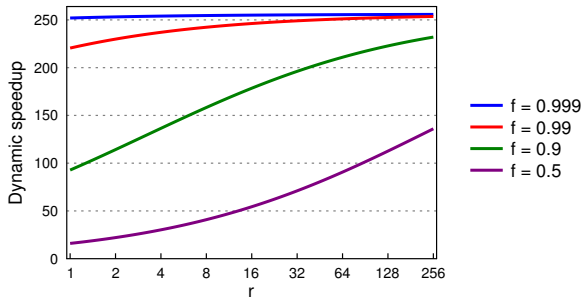
since dynamic multicores naturally incur a higher overhead than asymmetric designs. Under Gustafson's law, however, dynamic multicores only provide a significant performance advantage if $f = 0.5$. Somewhat surprisingly, the largest improvement is obtained under the GSSE (for $f = 0.5$ and $n = 256$) However, it needs to be kept in mind that this is for a dynamic large core of $r = 256$ BCEs, while practical considerations might keep r much smaller than its maximum of n. If we limit the size of the large core to 16 BCEs, the speedup of the optimal dynamic design over the optimal asymmetric design is less than 1.01 when the GSSE is assumed. Furthermore, the performance advantage of dynamic multicores over asymmetric designs diminishes when f increases under Gustafson's law and the GSSE, while under Amdahl's law this is not necessarily the case.



(a) Amdahl, $n = 256$.

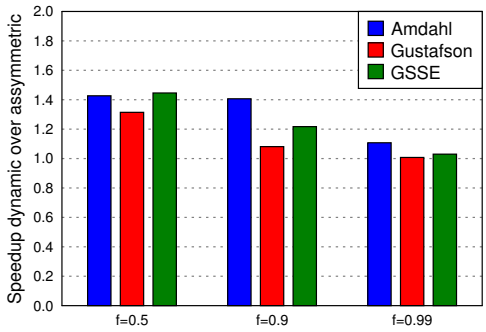


(b) Gustafson, $n = 256$.

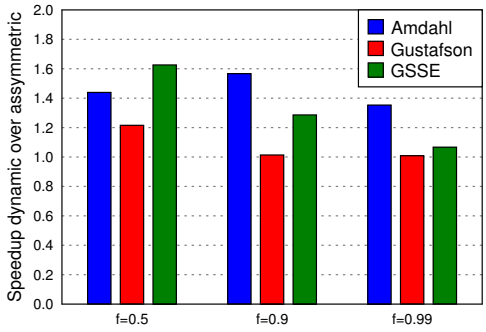


(c) GSSE, $n = 256$.

Figure 11: Speedup of dynamic multicores assuming Amdahl's law, Gustafson's law, and the GSSE for $n = 256$ BCEs.



(a) $n = 16$ BCEs.



(b) $n = 256$ BCEs.

Figure 12: Speedup of optimal dynamic design over optimal asymmetric design assuming Amdahl's law, Gustafson's law, and the GSSE for $n = 16$ and $n = 256$ BCEs.

4. CONCLUSIONS

The main contribution of this paper is two-fold. First, we have presented a generalized scaled speedup equation (GSSE) that encompasses both Almdahl's and Gustafson's law by substituting the appropriate application scaling function. Second, we have applied Amdahl's and Gustafson's law and the generalize scaled speedup equation to the area-performance model developed by Hill and Marty, and showed that substantially different results are obtained. 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 point of this paper is not to question the contribution of Hill and Marty. On the contrary, we thank them for starting a stimulating discussion (see Acknowledgment). Furthermore, under the assumption that future multicore will be used to accelerate fixed-size applications, their conclusions still hold. The main point of this paper is, however, that one has to consider the scaling properties of the targeted applications. One cannot simply take Amdahl's law and use it to determine the organization of next-generation multicores. Moreover, it seems unlikely that multicores should be organized such that optimal performance is achieved for parallel applications with a serial fraction as large as 0.5. It also seems unlikely that general-purpose multicore processors will be time-shared and thus at any time execute a single application. It is more likely that they will be space-shared and time-shared between several applications. Such a scenario indicates that a design with a few (but not one) large (but not huge) cores and several (but not too many) small cores will provide optimal throughput. The few large cores will be used to execute single-threaded applications and applications with large serial fractions as well as to accelerate the serial phases of applications with moderate serial fractions. The several small cores will be used to execute the parallel phases of several applications in a time-shared manner.

Many possibilities for future work exist. Like Hill and Marty, we have assumed that the performance is limited by area. One could also consider, for example, power constraints [14], pin count constraints, as well as Thermal Design Power, area/performance, power/performance, ITRS scaling factors, memory bandwidth, workload behavior, etc. [5]. One could consider a multi-application scenario as described above and analyze how this affects the results. Another possibility would be to determine how the parallel fraction scales with the problem size for typical applications. We note that this would be somewhat reminiscent to iso-efficiency analysis [8]. Iso-efficiency analysis, however, can only be applied to relatively simple and well-understood parallel algorithms and architectures. The simplicity of Amdahl's law, Gustafson's law, and the GSSE is both their strength as well as their weakness.

5. ACKNOWLEDGMENTS

Thanks to Mark Hill and Michael Marty for challenging us to develop a better model. Thanks to Peter Hofstee for suggesting the title. Thanks to Chi Ching Chi for making some of the figures.

6. REFERENCES

[1] G. Amdahl. Validity of the Single Processor Approach to Achieving Large Scale Computing Capabilities. *AFIPS Conference Proceedings*, 30(8):483–485, 1967.

[2] S. Borkar. Getting Gigascale Chips: Challenges and Opportunities in Continuing Moore's Law. *ACM Queue*, 1:26–33, October 2003.

[3] C. C. Chi and B. Juurlink. A QHD-Capable Parallel H.264 Decoder. In *Proc. Int. Conf. on Supercomputing*, ICS'11, 2011.

[4] S. Cho and R. Melhem. Corollaries to Amdahl's Law for Energy. *IEEE Computer Architecture Letters*, 12, 2007.

[5] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger. Dark silicon and the end of multicore scaling. *SIGARCH Comput. Archit. News*, 39(3):365–376, June 2011.

[6] S. Eyerman and L. Eeckhout. Modeling Critical Sections in Amdahl's Law and its Implications for Multicore Design. *SIGARCH Comput. Archit. News*, 38:362–370, June 2010.

[7] D. Gibson and D. A. Wood. Forwardflow: a scalable core for power-constrained CMPs. In *Proc. 37th annual Int. Symp. on Computer Architecture*, ISCA '10, 2010.

[8] A. Y. Grama, A. Gupta, and V. Kumar. Isoefficiency: Measuring the Scalability of Parallel Algorithms and Architectures. *IEEE Concurrency*, pages 12–21, August 1993.

[9] M. Gschwind, H. P. Hofstee, B. K. Flachs, M. Hopkins, Y. Watanabe, and T. Yamazaki. Synergistic Processing in Cell's Multicore Architecture. *IEEE Micro*, 26(2):10–24, 2006.

[10] J. Gustafson. Reevaluating Amdahl's Law. *Communications of the ACM*, 31(5):532–533, 1988.

[11] M. D. Hill and M. R. Marty. Amdahl's Law in the Multicore Era. *IEEE Computer*, 41(7):33–38, 2008.

[12] T. Kauranne. *Introducing Parallel Computers into Operational Weather Forecasting*. PhD thesis, Lappeenranta University of Technology, 2002.

[13] R. Kumar, D. M. Tullsen, P. Ranganathan, N. P. Jouppi, and K. I. Farkas. Single-ISA Heterogeneous Multi-Core Architectures for Multithreaded Workload Performance. In *Proc. 31st Int. Symp. on Computer Architecture*, ISCA '04, Washington, DC, USA, 2004. IEEE Computer Society.

[14] C. Meenderinck and B. Juurlink. (When) Will CMPs hit the Power Wall? In *Proceedings of the Euro-Par 2008 Workshops (HPPC)*, August 2008.

[15] F. J. Pollack. New microarchitecture challenges in the coming generations of CMOS process technologies (keynote address)(abstract only). In *Proc. 32nd annual ACM/IEEE Int. Symp. on Microarchitecture*, MICRO 32, 1999.

[16] A. Ramirez, F. Cabarcas, B. Juurlink, M. A. Mesa, F. Sanchez, A. Azevedo, C. Meenderinck, C. Ciobanu, S. Isaza, and G. Gaydadjiev. The SARC Architecture. *IEEE Micro*, pages 16–29, Sept./Oct. 2010.

[17] X. Sun and L. Ni. Another View on Parallel Speedup. *Proc. of Supercomputing'90*, pages 324–333, 1990.

[18] Y. Watanabe, J. D. Davis, and D. A. Wood. WiDGET: Wisconsin Decoupled Grid Execution Tiles. In *Proc. 37th annual Int. Symp. on Computer Architecture*, ISCA '10, pages 2–13, 2010.