Extending Amdahl's Law in the Multicore Era

Erlin Yao, Yungang Bao, Guangming Tan and Mingyu Chen

Institute of Computing Technology, Chinese Academy of Sciences
yaoerlin@gmail.com, {baoyg,tgm,cmy}@ncic.ac.cn

1. INTRODUCTION

The scalability problem is in the first place of the dozen long-term information-technology research goals indicated by Jim Gray [2]. Chip multiprocessors (CMPs) or multicores are emerging as the dominant computing platform. In the multicore era, the scalability problem is still an interesting long-term goal, and it will become more urgent in the next decade. Hill and Marty [4] augment Amdahl's law to multicore hardware by constructing a cost model for the number and performance of cores that the chip can support. They conclude that obtaining optimal multicore performance will require further research in both extracting more parallelism and making sequential cores faster. Woo and Lee [6] develop Hill's work by taking power and energy into account. The revised models provide computer architects with a better understanding of multicore scalability, enabling them to make more informed tradeoffs. However, as far as we know, no work has investigated theoretical analysis of these types of works, existing works are all carried out using programs and experiments.

This paper investigates the theoretical analysis of multicore scalability. For asymmetric multicore chips, although the architecture of using one large core and many base cores is assumed originally for simplicity, it is proved to be the optimal architecture in the sense of speedup. The potentials of the maximum of speedups using architecture of symmetric, asymmetric or dynamic multicore are obtained. Given the parallel fraction, performance index and the number of base core resources, precise quantitative conditions are given to determine how to obtain optimal multicore performance. Our quantitative analysis not only explains Hill's work [4] theoretically, but also extends their result to a more general framework. The analytical tools in this paper can also be used to the theoretical analysis of Woo and Lee's works [6].

2. MODEL OF MULTICORE SCALABILITY

Four decades ago, Gene 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 [1]. He 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. Without presenting an equation, he noted that the speedup on n processors is governed by:

$$Speedup_{parallel}(f,n) = \frac{1}{(1-f)+\frac{f}{n}}.$$

Despite its simplicity, Amdahl's law applies broadly and gives important insights such as: (i) Attack the common case: When f is small, optimization will have little effect. (ii) The aspects you ignore also limit speedup: Even if n approaches infinity, speedup is bounded by $\frac{1}{(1-f)}$.

Hill and Marty augment Amdahl's law to multicore hardware by constructing a cost model for the number and performance of cores that the chip can support [4]. We adopt the same cost model constructed by them. They first assume that a multicore chip of given size and technology generation can contain at most n base core equivalents (BCE) (where a single BCE implements the baseline core). Second, they assume that architects can use the resources of multiple BCEs to create a core with greater sequential performance. Let the performance of a single-BCE core be 1, we assume that architects can expend the resources of r BCEs to create a powerful core with sequential performance $perf(r)$ $(1 < perf(r) < r)$. According to the cost model, they classify the architecture of multicore chips into three types: symmetric, asymmetric and dynamic multicore chips.

A symmetric multicore chip requires that all its cores have the same cost. A symmetric multicore chip with a resource budget of n BCEs will have n/r cores of r BCEs each. Under Amdahl's law, the speedup of a symmetric multicore chip (relative to using one single-BCE core) is:

$$Speedup_{symmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)}+\frac{fr}{perf(r)n}}. \quad (1)$$

An alternative to a symmetric multicore chip is an asymmetric (or heterogeneous) multicore chip, in which one or more cores are more powerful than the others. With the simplistic assumptions of Amdahl's law, it makes most sense to devote extra resources to increase only one core's capability. With a resource budget of n BCEs, an asymmetric multicore chip can have $1+n-r$ cores with one larger core (with r BCEs) and $n-r$ base cores (with 1 BCE each). This chip uses the one core with r resources to execute sequentially at performance $perf(r)$. In the parallel fraction, it gets performance $perf(r)$ from the large core and performance 1 from each of the $n-r$ base cores. Under Amdahl's law, the speedup of an asymmetric multicore chip is:

$$Speedup_{asymmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)}+\frac{f}{perf(r)+n-r}}. \quad (2)$$

A dynamic multicore chip can dynamically combining up to r cores to boost performance of only the sequential component. In sequential mode, this dynamic multicore chip can execute with performance $perf(r)$ when the dynamic techniques can use r BCEs. In parallel mode, a dynamic multicore gets performance n using all base cores in parallel. Overall, the speedup of a dynamic multicore chip is:

$$Speedup_{dynamic}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{f}{n}}. \tag{3}$$

3. A THEORETICAL ANALYSIS

3.1 Symmetric Multicore Chips

It is clear that for fixed n and r, $Speedup_{symmetric}(f,n,r)$ is an increasing function of f. And for fixed f and r, speedup is also an increasing function of n, which indicate that we should increase both the parallel fraction f and the number of base core equivalents n to enhance the speedup of symmetric multicore chip. For fixed f and n, we have the following theorem:

THEOREM 1. *For symmetric multicore chip with speedup*

$$Speedup_{symmetric}(f,n,r) = \frac{1}{\frac{1-f}{perf(r)} + \frac{fr}{perf(r)n}},$$

suppose $perf(r) = r^c, 0 < c < 1$, then it holds that: (i) if $n \le \frac{\frac{1}{c}-1}{\frac{1}{f}-1}$, then the maximum of speedup occurs at $r = 1$ and the speedup is a decreasing function of r; (ii) if $c \ge f$, then the maximum of speedup occurs at $r = n$ and the speedup is an increasing function of r; (iii) if $c < f$ and $n > \frac{\frac{1}{c}-1}{\frac{1}{f}-1}$, then the maximum of speedup occurs at $r = \frac{n(\frac{1}{f}-1)}{\frac{1}{c}-1}$.

PROOF. Let $Speedup_{symmetric}(f,n,r)$ be $S(x)$ and $S'(x)$ be the first derivative of it, then the maximum of speedup can be determined according to the positive or negative of $S'(x)$. □

Note that for any fixed $0 < f < 1$ and $0 < c < 1$, if the number of base cores n is big enough, then $n \le \frac{\frac{1}{c}-1}{\frac{1}{f}-1}$ can not hold, so the maximum of speedup will not occur at $r = 1$, which means that moving to denser chips increases the likelihood that cores will be nonminimal. And no matter how many base cores there are, if the parallel fraction f is less than the performance index c, then the maximum of speedup will surely occur at $r = n$, which indicates that we should build a chip with only one big core including all the BCEs to obtain optimal multicore performance. If f is bigger than c and n is big enough ($n > \frac{\frac{1}{c}-1}{\frac{1}{f}-1}$), to obtain optimal multicore performance we should devote $\frac{n(\frac{1}{f}-1)}{\frac{1}{c}-1}$ BCE resources to increase each core's performance.

3.2 Asymmetric Multicore Chips

Similar to the symmetric multicore chip, we should also increase both the parallel fraction f and the number of BCEs n to enhance the speedup of asymmetric multicore chip. And for fixed f and n, the following theorem can be given:

THEOREM 2. *For asymmetric multicore chip with speedup*

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

suppose $perf(r) = r^c, 0 < c < 1$, then it holds that: (i) if $\frac{f}{1-f}\frac{1-c}{c} \ge n^2$, then the maximum of speedup occurs at $r = 1$ and the speedup is a decreasing function of r; (ii) if $\frac{c}{f} \ge n^{1-c}$, then the maximum of speedup occurs at $r = n$ and the speedup is an increasing function of r; (iii) if $\frac{f}{1-f}\frac{1-c}{c} < n^2$ and $\frac{c}{f} < n^{1-c}$, then the maximum of speedup occurs at some unique $r_0 \in (1,n)$.

PROOF. Let $Speedup_{asymmetric}(f,n,r)$ be $S(x)$, $S'(x)$ and $S''(x)$ be the first derivative and second derivative of $S(x)$. When $perf(x) = x^c, 0 < c < 1$, it can be proved that $S''(x) < 0$ for any $x \in [1,n]$, so $S(x)$ is a concave function on $[1,n]$. Then the maximum of speedup can be determined according to the positive or negative of the two ends of $S'(x)$, i.e., $S'(1)$ and $S'(n)$. □

It is clear that if the parallel fraction f is greater than the performance index c, then $\frac{c}{f} \ge n^{1-c}$ can not hold, so the maximum of speedup will not occur at $r = n$, which means that we should not build a chip with only one big core including all the BCEs. If f is less than c, then $\frac{f}{1-f}\frac{1-c}{c} \ge n^2$ can not hold, so the maximum of speedup will not occur at $r = 1$, which indicates that we should devote more resources to make a faster sequential core. Note that for any fixed $0 < f < 1$ and $0 < c < 1$, if the number of base cores n is big enough ($\frac{f}{1-f}\frac{1-c}{c} < n^2$ and $\frac{c}{f} < n^{1-c}$), then the maximum of speedup will occur at $1 < r_0 < n$, which indicates that we should devote r_0 resources to build a faster sequential core to obtain optimal multicore performance. Note that the optimal value r_0 in Theorem 2 can not be solved analytically like in Theorem 1. But the following corollary can be given:

COROLLARY 1. *The optimal value r_0 in Theorem 2 can not be solved analytically, but it can be determined using at most log_2^n times of computation, and it has an estimation of*

$$\frac{2c(1-f)}{2c(1-f)+f}n < r_0 < n. \tag{4}$$

And for any $\epsilon > 0$, if n is big enough, e.g.,

$$n \ge \left(\frac{f}{c(1-f)}\frac{(1-\epsilon)^{1+c}}{\epsilon^2}\right)^{\frac{1}{1-c}}, \tag{5}$$

then it holds that $r_0 > (1-\epsilon)n$.

PROOF. According to Theorem 2, r_0 is the root of the equation: $S'(x) = 0$, which is a transcendental function for any $0 < c < 1$, so r_0 can not be solved analytically. However, according to the monotonicity of $S'(x)$ and the uniqueness of r_0 in interval $(1,n)$, it can be determined using at most log_2^n times of computation with the dichotomy method. The estimation in Eq. (4) and Eq. (5) can be obtained under the equivalent transformation and zoom in and out of the equation: $S'(x) > 0$. □

It can be seen that the optimal r_0 is linear with the number of BCE resources n, and if n is big enough, r_0 will approach n to any extent. Theorem 2 indicates that the architecture of using one large core with r_0 BCEs and $n - r_0$

base cores is better than other architectures with one large core and many base cores. Then what is the case of other possible architectures of asymmetric multicore chip? The following theorem will answer this question:

THEOREM 3. *For asymmetric multicore chip, the architecture of using one large core and many base cores is optimal, i.e., the speedup of architecture using one large core with* r_0 *BCEs and* $n-r_0$ *base cores is bigger than the speedup of any other possible architecture.*

PROOF. Any of the possible architectures of an asymmetric multicore chip with a resource budget of n BCEs is as the following: there are in total m cores of r_i BCEs each, satisfying $\sum_{i=1}^{m} r_i = n$. Suppose the chip use the core with r_k BCEs to handle the sequential phase, then according to the assumption that $perf(r) < r$, it is clear that the speedup of this architecture is less than $Speedup_{asymmetric}(f, n, r_k)$. And according to Theorem 2, $Speedup_{asymmetric}(f, n, r_k) \le Speedup_{asymmetric}(f, n, r_0)$ holds. □

Theorem 3 indicates that although the architecture of asymmetric multicore chip using one large core and many base cores is assumed originally for simplicity, it is indeed the optimal architecture in the sense of speedup.

3.3 Dynamic Multicore Chips

Similarly, we should increase both the parallel fraction f and the number of base core equivalents n to enhance the speedup of dynamic multicore chip continuously. For fixed f and n, it is clear that if $perf(r)$ is an increasing function of r, then $Speedup_{dynamic}(f, n, r)$ is also an increasing function of r, which indicates that the maximum of $Speedup_{dynamic}(f, n, r)$ always occurs at $r = n$.

Since $perf(r) < r$, it holds that $perf(r)\frac{n-r}{r} < n - r$, so it is clear that $Speedup_{symmetric}(r) < Speedup_{asymmetric}(r)$. Likewise, $Speedup_{asymmetric}(r) < Speedup_{dynamic}(r)$ holds according to $perf(r)+n-r < n$. This indicate that dynamic multicore chips can offer potential speedups that are greater and never worse than symmetric or asymmetric multicore chips with identical $perf(r)$ functions. So researchers should continue to investigate methods that approximate a dynamic multicore chip.

3.4 The Potentials of Maximum Speedups

Recall that in the Amdahl's law, even if the number of processors n approaches infinity, the speedup is bound by $\frac{1}{1-f}$. Then what is about the speedup considered here? The following theorem can be given:

THEOREM 4. *Suppose there are* n *base core resources, the parallel fraction is* f *and the performance function is* $perf(r) = r^c, 0 < c < 1$. *Then it holds that: (i) if* $n^2 \le \frac{f}{1-f}\frac{1-c}{c}$, *then no matter we adopt the symmetric or asymmetric multicore architecture, the maximum of speedup can be obtained is* $\frac{1}{1-f+\frac{f}{n}}$; *(ii) if* $n^{1-c} \le \frac{c}{f}$, *then no matter we adopt the symmetric or asymmetric multicore architecture, the maximum of speedup can be obtained is* n^c; *(iii) if* n *is big enough, then no matter we adopt the symmetric, asymmetric or dynamic multicore architecture, the maximum of speedup can be obtained is between* n^c *and* $\frac{n^c}{1-f}$.

PROOF. According to Theorem 1 and Theorem 2, (i) and (ii) are clear. (iii) If n is big enough, it is clear that the maximum of $Speedup_{symmetric}(r)$ (or $Speedup_{asymmetric}(r)$) is bigger than n^c. The maximum of $Speedup_{dynamic}(r)$ always occurs at $r = n$. According to Eq.(3),

$$Speedup_{dynamic}(f, n, n) = \frac{1}{\frac{1-f}{n^c} + \frac{f}{n}}. \qquad (6)$$

And it is clear that $n^c < Speedup_{dynamic}(f, n, n) < \frac{n^c}{1-f}$. □

Theorem 4 tells that the increasing of n can enhance the speedup continuously. Under the assumption $perf(r) = r^c$, when the number of base core resources approaches infinity, the speedup can also approach infinity even if the performance index c is small.

4. CONCLUSION

In this paper, we investigate a theoretical analysis of multicore scalability, and quantitative conditions are given to determine how to obtain optimal multicore performance. The theorems and corollary we offer provide computer architects with a better understanding of multicore design types, enabling them to make more informed tradeoffs. However, our precise quantitative results are suspect because the real world is much more complex. Many performance factors were removed from the model, including pipeline efficiency, branch prediction, cache contention, cache coherence, synchronization, etc. In practice, the optimal configuration must be decided through experiments and based on designer experience, with the help of architectural simulators or other performance tools. This theoretical analysis seek to provide insights to stimulate discussions and future works. Since more cores might advantageously allow greater parallelism from larger problem size, future works should also consider extending the Gustafson's law [3] and Sun-Ni's law [5] in the multicore era.

5. ACKNOWLEDGMENTS

This research is funded by the National Natural Science Foundation of China under grant 60633040 and 60803030.

6. REFERENCES

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

[2] Jim Gray. What next?: A dozen information-technology research goals. *Journal of the ACM* 50(1): 41-57 (2003).

[3] J. L. Gustafson. Reevaluating Amdahl's Law. *Communications of the ACM*, May 1988, 532-533.

[4] Mark D. Hill and Michael R. Marty. Amdahl's Law in the Multicore Era. *IEEE Computer*, vol. 41, no. 7, pp. 33-38, July 2008.

[5] X.-H. Sun and L. M. Ni. Another View on Parallel Speedup. In Proc. *Supercomputing'90* (NY, 1990), 324-333.

[6] D. H. Woo and H. S. Lee. Extending Amdahl's Law for Energy-Efficient Computing in the Many-Core Era. *IEEE Computer*, December 2008, 24-31.