

(When) Will CMPs hit the Power Wall?

Technical Report

Abstract

The power wall is currently one of the major obstacles computer architecture is facing. In this paper we analyze the impact of the power wall on CMP design. As a case study we model a CMP consisting of Alpha 21264 cores, scaled to future technology nodes according to the ITRS roadmap. When running at the maximum clock frequency, such a CMP would greatly exceed the power budget. Although power limits performance significantly, technology improvements will still provide performance growth. Amdahl's Law highly threatens this performance growth, but might not be valid for all application domains. In those cases Gustafson's Law could be valid which is much more optimistic. From our results we derive two principles that prevent CMPs from hitting the power wall.

August 13, 2008
CE-TR-2008-04

Cor Meenderinck and Ben Juurlink
Computer Engineering Department
Delft University of Technology
{Cor, Benj}@ce.et.tudelft.nl



1 Introduction

It is commonly believed that we have reached the *power wall*, meaning that uniprocessor performance improvements have come to an end due to power constraints. The main drivers of the increased power consumption are higher clock frequencies and power inefficient techniques to exploit more Instruction Level Parallelism (ILP), such as wide-issue superscalar execution. Hitting the power wall is also one of the reasons why industry has shifted towards multi-cores or chip multiprocessors (CMPs). Because CMPs exploit explicit Thread Level Parallelism (TLP), their cores can be simpler and do not need additional hardware to extract ILP. In other words, CMPs allow exploiting parallelism in a power efficient way.

Figure 1, taken from [1], illustrates the power wall. Uniprocessors have basically reached the power wall. As argued above, multicores can postpone hitting the power wall but they are also expected to hit the power wall. Several questions arise like when will CMPs hit the power wall, what limitation will cause this to happen, what can computer architects do to avoid the problem, etc. It is generally believed that power efficiency of CMPs can be improved by designing asymmetric or heterogeneous multicores [2]. For example, several domain specific accelerators could be employed which are turned on and shut down according to the actual workload. But, is the power saving it provides worth the area cost? In this paper we try to answer those questions.

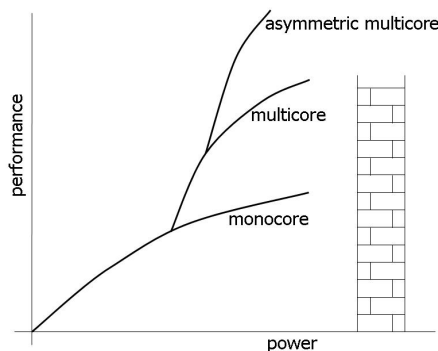


Fig. 1: The power wall problem.

Specifically, in this paper we focus on technology improvements as they have been one of the main drivers of performance growth in the past. According to the ITRS roadmap [3], technology improvements are expected to remain. It predicts a clock frequency of 14 GHz for the year 2022 and an astonishing amount of available transistors. Because of power constraints, however, it might not be possible to exploit those technology improvements. In the next section we analyze the limits of performance growth due to technology improvements with respect to power constraints. First we assume perfectly parallelizable applications but also performance growth for non-perfect parallelization is analyzed using both Amdahl's and Gustafson's Law. In Section 3 based on the results of our experiment we conclude that CMPs can offer significant performance

Tab. 1: Parameters of the Alpha 21264 chip.

year	1998
technology node	350nm
supply voltage	2.2V
die area	314mm ²
dynamic power (400MHz)	48W
dynamic power (600MHz)	70W

improvements provided a number of principles are followed.

Of course our model is necessarily rudimentary. For example, it does not consider bandwidth constraints nor static power dissipation due to leakage. Nevertheless, the power wall has been predicted, multicores are expected to be the remedy, asymmetric multicores have been envisioned, but to the best of our knowledge this has never been quantified. From our results we hope to derive some principles which can be the basis for future work.

2 Performance and Power of Future Multicores

To analyze the effect of technology improvements on the performance of future CMPs, and to investigate the power consumption trend, the following experiment was performed. We take an Alpha 21264 chip (that is core and caches), scale it to future technology nodes according to the ITRS roadmap, create a hypothetical CMP consisting of the scaled cores, and derive the power numbers. Specifically, we calculate the power consumption of a CMP for full blown operation, i.e., all cores are active and run at the maximum possible frequency. Furthermore, we analyze the performance growth over time if the power consumption is restricted to the power budget allowed by packaging.

The Alpha 21264 [4] core was chosen as subject of this experiment for two reasons. First, the Alpha has been well documented in literature, providing the required data for the experiment. Second, the 21264 is a moderately sized core lacking the aggressive ILP techniques of current high performance cores. Thus, it is a good representative of what is generally expected to be the processing element in future many-core CMPs. Table 1 provides an overview of the key parameters of the 21264 relevant for this analysis.

2.1 Scaling of the Alpha 21264

The 21264 is scaled according to data in the 2007 edition of the International Technology Roadmap for Semiconductors (ITRS) [3]. The relevant parameters are given in Table 2. The time frame considered is 2007-2022, which is the exact time span of the roadmap. The values of the technology node and the on-chip frequency were taken from Page 79 of the Executive Summary Chapter. The on-chip frequency is based on the fundamental transistor delay, and an assumed maximum number of 12 inverter delays. The die area values were taken from the table on Page 81 of the Executive Summary. Finally, the values of the supply voltage and the gate capacitance (per micron device) were taken from the table starting at Page 11 of the Process Integration, Devices, and Structures Chapter of the roadmap.

Tab. 2: Technology parameters of the ITRS roadmap.

	2007	2008	2009	2010	2011	2012	2013	2014
technology (nm)	68	57	50	45	40	36	32	28
frequency (MHz)	4700	5063	5454	5875	6329	6817	7344	7911
die area (mm ²)	310	310	310	310	310	310	310	310
supply voltage (V)	1.1	1	1	1	1	0.9	0.9	0.9
$C_{g,total}$ (F/μm)	7.10E-16	8.40E-16	8.43E-16	8.08E-16	6.5E-16	6.29E-16	6.28E-16	5.59E-16
	2015	2016	2017	2018	2019	2020	2021	2022
technology (nm)	25	22	20	18	16	14	13	11
frequency (MHz)	8522	9180	9889	10652	11475	12361	13351	14343
die area (mm ²)	310	310	310	310	310	310	310	310
supply voltage (V)	0.8	0.8	0.7	0.7	0.7	0.65	0.65	0.65
$C_{g,total}$ (F/μm)	5.25E-16	5.07E-16	4.81E-16	4.58E-16	4.1E-16	3.91E-16	3.62E-16	3.42E-16

To model the experimental CMP for future technology nodes, we scale all required parameters of the 21264 core. The values that are available in the ITRS, we use as such. The others we scale using the available parameters by taking the ratio between the original 21264 parameter values and the predictions of the roadmap. Below we describe in detail for each scaled parameter how this was done. The gate capacitance of the 21264 was not found in literature, thus we extrapolated the values reported in the roadmap and calculated a value of $1.1 \times 10^{-15} F/\mu m$ for 1998.

First, the area of one core was scaled. Let $L(t)$ be the process technology size for year t and let L_{orig} be the process technology size of the original core. The area of one 21264 core in year t will be $A_1(t) = A_{orig} \times \left(\frac{L(t)}{L_{orig}}\right)^2$. Figure 2 depicts the results for the time frame considered. The area of one core decreases more or less quadratically over time, and will be about one third of a square millimeter in 2022.

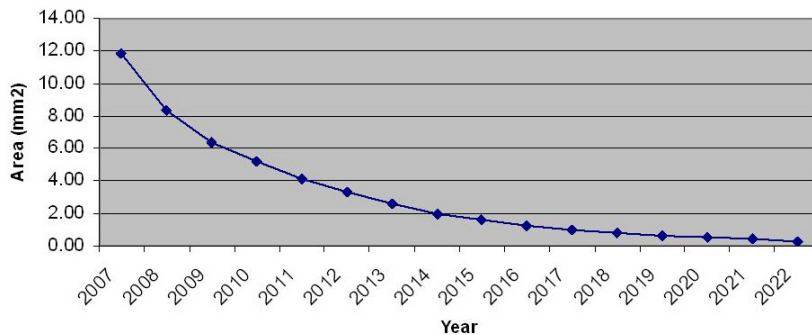


Fig. 2: Area of one core.

Second, using the scaled area of one core, the number of cores that could fit on a single die was calculated. The ITRS roadmap assumes a die area of $310mm^2$ for the entire time frame. Thus, the total number of cores per die in year t is $N(t) = \frac{310}{A_1(t)}$, which is depicted in Figure 3. For 2008 it was calculated that a 37-core CMP would be possible. This seems reasonable considering two examples of current state-of-the-art CMPs, the Tiler Tile64 [5] and the Clear-speed CSX600 [6]. The first has 64 cores, each having their own L1 and L2 cache and can run a full operating system autonomously. These cores are similar to the Alpha 21264 from an architectural point of view. The latter has 96 cores

which are simpler and targeted at exploiting data level parallelism. From 2007 on the graph shows a doubling of cores roughly every three years which is in line with the expected increase [7]. In 2022 our calculations predict a CMP with 999 cores.

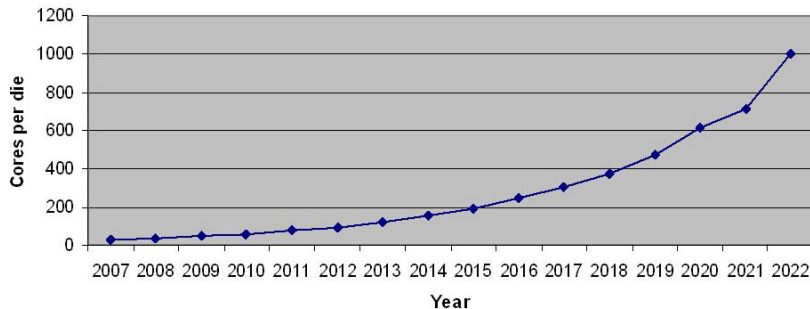


Fig. 3: Number of cores per die.

Finally, the power of one core was scaled. Power consumption consists of a dynamic and a static part, of which the latter is dominated by leakage. The data required to scale the static power is not available to us and thus we restrict this power analysis to dynamic power. It is expected that leakage remains a problem and thus our estimations are conservative.

The dynamic power is given by $P_{dyn} = \alpha C f V^2$, where α is the transistor activity factor, C is the gate capacitance, f is the clock frequency, and V is the power supply voltage. The activity factor α of the 21264 processor is unknown and also depends on the application, but since this does not change with scaling, it can be assumed to be constant. The capacitance C (F) in the equation is different from capacitance $C_{g,total}$ (F/ μm) in Table 2, but they relate to each other according to $C \propto C_{g,total} \times L$. Thus, the dynamic power at year t is calculated as:

$$P(t) = P_{orig} \times \frac{C_{g,total}(t) \times L(t)}{C_{g,total,orig} \times L_{orig}} \times \frac{f(t)}{f_{orig}} \times \left(\frac{V(t)}{V_{orig}} \right)^2. \quad (1)$$

It is noted that this analysis assumes that the cores run at the maximum possible frequency. Figure 4 depicts the power of one core over time. As the curve shows it roughly decreases linearly, resulting in less than 2 W in 2022.

2.2 Power and Performance Assuming Perfect Parallelization

Now that all parameters have been scaled, it is possible to calculate the power consumption of the total CMP. It is assumed that all cores are active and thus $P_{total}(t) = N(t) \times P(t)$. Figure 5 depicts the total power over time and shows that for the assumptions of this analysis the total power consumption in 2008 is 600 W, gradually increases, and reaches 1.5 kW in 2022. The roadmap predicts that the power budget allowed due to packaging constraints is 198 W. It is clear that for the entire calculated time span the power consumption of our hypothetical CMP exceeds the power budget. This is why power has become one of the main design constraints nowadays.

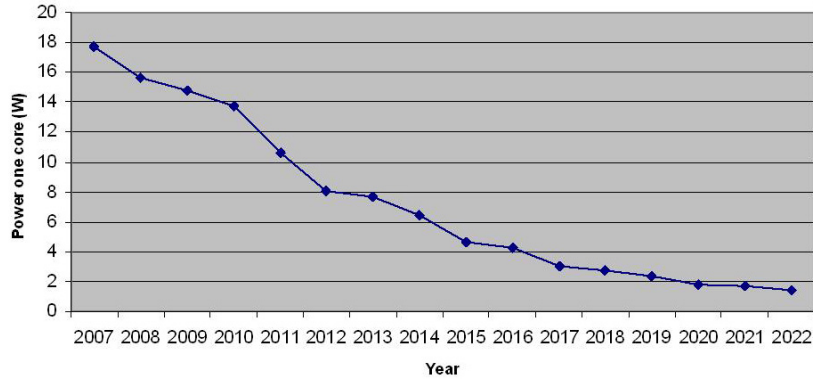


Fig. 4: Power of one core.

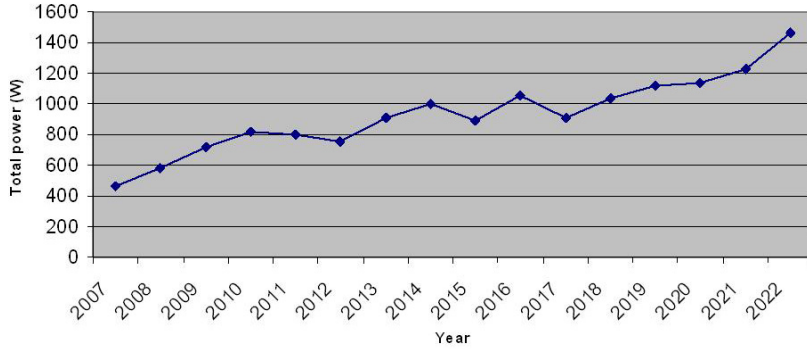


Fig. 5: Total power of the case study CMP.

The figure also shows that the difference between the power budget and the power consumption of the full blown hypothetical CMP is increasing over time. That means that a large part of the technology improvement cannot be put into effect for performance growth. For example, between 2011 and 2020 technology allows doubling the on-chip frequency. However, the power consumption would increase with a factor 1.5. Thus, for equal power only a small frequency improvement would be possible.

Next we analyze the performance improvement that can be achieved by this CMP. Assuming that the application is perfectly parallelizable, the parameters that influence performance are frequency and the number of cores. In this case the speedup in year t $S(t)$, relative to the original 21264 core, is given by $S(t) = \frac{f(t)}{f_{orig}} \times \frac{N(t)}{1}$. This speedup is depicted in Figure 6 as the non-constrained speedup.

We are interested in the speedup achieved by CMPs that meet the power budget of 198 W. As the results show the power budget is exceeded when all cores are used concurrently at the maximum frequency. Thus, the non-constrained speedup is not achievable in practice. To meet the power budget, either the frequency could be scaled down or a number of cores could be shut down. Since both measures are linear to the speedup, the power-constrained

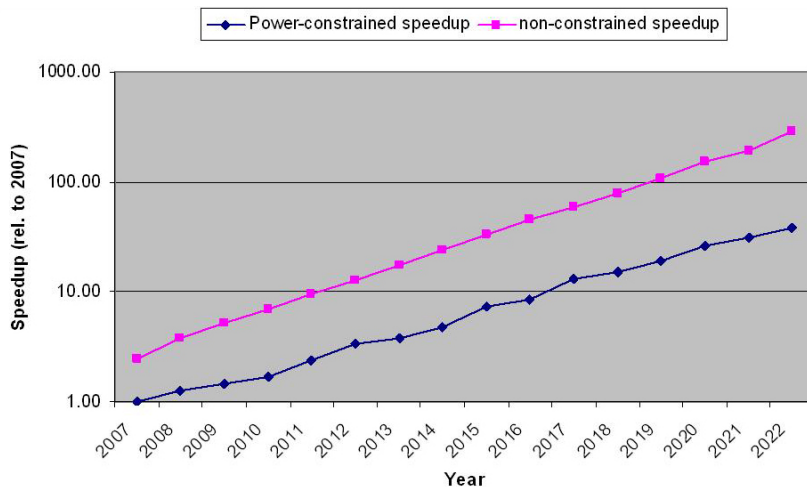


Fig. 6: Power-constrained performance growth.

speedup can be defined as:

$$S_{power_constr.}(t) = \frac{f(t)}{f_{orig}} \times \frac{N(t)}{1} \times \frac{P_{budget}}{P_{total}(t)}, \quad (2)$$

where P_{budget} is the power budget allowed by packaging.

Figure 6 depicts the power-constrained performance of the case study CMP over time. To increase readability we normalized the result to 2007. The curve shows a performance growth of 27% per year. Also the non-constrained performance growth is depicted. Note that the latter is growing with 37% per year and that the gap between the two is increasing. To put these results in historical perspective we compare to Figure 2.1 of Hennessey and Patterson [8]. From the mid-1980s to 2002 the graph shows an annual performance growth of 52%. Then from 2002 the annual performance growth dropped to 20%. Considering this historical perspective, the predicted annual performance growth of 27% is a bit on the high side, but not far of. The main conclusion from these results is that although power severely limits performance substantial performance growth is still possible using the CMP paradigm.

2.3 Intermezzo: Amdahl and Gustafson

So far we assumed a perfectly parallelizable application. In practice this is not always the case as there might be purely serial code. If this is the case we can apply either Amdahl's Law [9] or Gustafson's Law [10]. As these laws are often misunderstood we review both.

Amdahl's Law is the most well known of the two. In general terms it states that the performance improvement achieved from some feature is limited by the fraction of time the feature can be used. Originally Amdahl formulated his law to argue against parallel computers. He observed that 40% of the instructions were overhead which consisted of serial code that cannot be parallelized. Therefore, the speedup that can be achieved by a parallel computer would be very

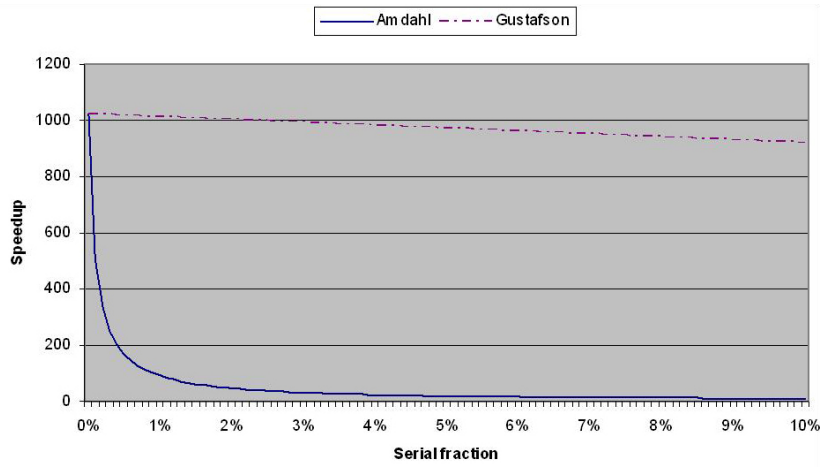


Fig. 7: Wrong comparison of speedup under Amdahl’s and Gustafson’s Law for $N = 1024$.

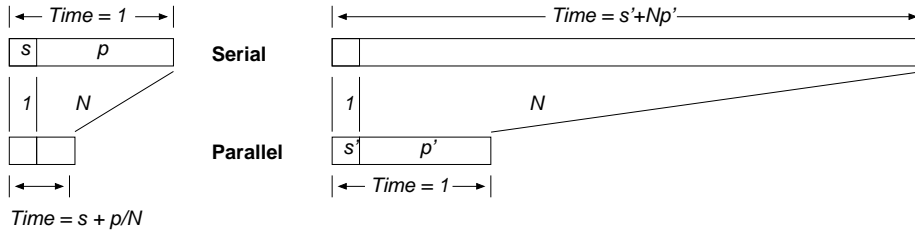


Fig. 8: Gustafson’s explanation of the difference between his speedup (right) and Amdahl’s speedup (left). Yuan Shi proved the two to be equivalent; they just used a different definition of s (denoted by author as s and s').

limited. In a formal way, Amdahl’s Law is given by:

$$\begin{aligned}
 \text{Speedup} &= (s + p)/(s + p/N) \\
 &= \frac{1}{(s + p/N)},
 \end{aligned}
 \tag{3}$$

where N is the number of processors, s is the amount of time spent (by a serial processor) on serial parts of a program, and p is the amount of time spent (by a serial processor) on parts of the program that can be done in parallel.

Something that few people realize is that Amdahl’s Law is only valid under certain circumstances. The law assumes firstly that the serial and the parallel part both take a constant number of calculation steps independent of N , and secondly that the program takes the same input size independent of N .

Figure 7 depicts Amdahl’s Law for $N = 1024$ and shows that the function is very steep for s close to zero. On the other hand, Gustafson noticed that on a 1024 processor machine they achieved speedup between 1016 and 1021 for $s = 0.4 - 0.8$ percent. For $s = 0.4$ the latter predicts a maximum speedup of 201 and thus Gustafson’s findings seemingly conflicted with Amdahl’s Law.

From these results Gustafson concluded that Amdahl’s Law was inappropriate to massive parallel systems and he proposed a new law to calculate the

speedup of massive parallel systems. He started reasoning from the parallel machine and defined s' and p'^1 as the serial and parallel fractions of time spend on a parallel system. A serial processor would require time $s' + p' \times N$ to perform the same task. Gustafson explained the difference between his reasoning and that of Amdahl using Figure 8. The reasoning of Gustafson leads to the following equation, now known as Gustafson's Law:

$$\begin{aligned} \text{Scaled speedup} &= (s' + p' \times N)/(s' + p') \\ &= (s' + p' \times N) \\ &= N + (1 - N) \times s' \end{aligned} \tag{4}$$

Gustafson called the results of his equation the 'scaled speedup' because he thought he calculated a different speedup. Figure 7 shows, beside the speedup under Amdahl's Law, the speedup using Gustafson's equation. The latter corresponds to the experimental findings of Gustafson.

However, something is wrong here. Gustafson used a different definition of the serial fraction than Amdahl did. The first measured s' in the parallel program while the latter measured s in the serial program (see Figure 8). Yuan Chi proved that actually the two equations are identical [11], which is not difficult to understand. Take the experiment of Gustafson with $s' = 0.4$ (measured on a parallel machine) and $N = 1024$, which results in speedup of 1020 using Gustafson's Law. Assume on a parallel machine the total execution time is 1. Then the serial part takes 0.004 time. The same task on a serial machine would take $s' + Np' = 1019.908$ time. The fraction of the serial part on this serial run is $0.004/1019.908 = 3.92 \times 10^{-6}$. Filling this value of s into Amdahl's Law results in a speedup of 1020, exactly as Gustafson's Law predicted and the experiments showed. This also explains why Figure 7 provides a wrong comparison of the two laws.

However, that is not all there is to say about the two laws. Gustafson showed that Amdahl's pessimistic prediction of the future was wrong and he also pointed out why. Amdahl assumed that the large s measured in his time would remain large in the future, i.e., he considered the fraction s to be a constant, because he assumed a fixed program and input size. Gustafson correctly observed that over time the problem size had grown and also that the serial fraction is dependent on the problem size.

Therefore, in predicting the future Gustafson assumed that the problem size scales with the number of processors. Furthermore, he assumed that the execution time of the serial part is independent on the problem size. That is, he assumed the absolute value of the serial fraction to be constant over the number of processors in contrast to Amdahl who assumed the relative value of the serial fraction to be constant over the number of processors.

The different assumptions are hidden in the two equations in the way the serial fraction is defined. Usually, when predicting the future, people simply take a fixed value s and use either of the two equations to calculate the speedup for a number of processors (see Figure 9). However, taking a fixed value for s means something different in the two Law's and thus the user (unfortunately often unconsciously) agrees with the assumptions corresponding to the chosen

¹ We distinguish between s and p from Amdahl and s' and p' from Gustafson. As we show later, those are not the same although Gustafson did not realize that.

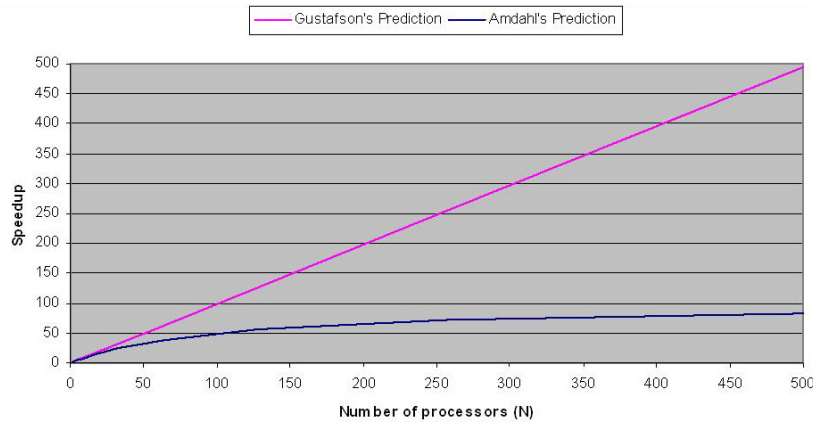


Fig. 9: Prediction of speedup for $s = s' = 1\%$ with Amdahl's and Gustafson's assumptions.

equation. Because the assumptions are different, the equations produce different results although they are identical. It's not the different equation that produces the different results, it is the different set of assumptions that produce the different result. For example, it is possible to use Amdahl's equation with Gustafson's assumptions and find Gustafson's prediction. To do that, for every N you have to recalculate the value of s' to match Amdahl's definition of s .

Therefore, when predicting the future one has to consider carefully the assumptions. Otherwise, one is amenable to making the same mistake as Amdahl who in 1967 pessimistically predicted that there was no future for massive parallel systems. On the other hand, using Gustafson's assumptions one might obtain a more optimistic prediction. The best way is to thoroughly analyze the application domain under consideration and classify it as 'Amdahl' or 'Gustafson'. But this has to be done carefully as the following example shows. H.264 video decoding contains an entropy decoding stage which is bit-serial. It is parallelizable on the slice and frame-level but any strategy exploiting this was generally assumed to be impractical or not scalable. Thus, often Amdahl's assumptions were used to predict a pessimistic future for parallelizing H.264. However, in [12] a new parallelization strategy was proposed where exploiting frame-level parallelism can effectively be deployed. With this parallelization strategy the 'serial' entropy decoding can be parallelized to a certain extent resulting in a much more optimistic prediction. In this case the difference is not in the assumption of a fixed problem size but in the assumption of the fixed program.

Not all application domains can be strictly classified as Amdahl or Gustafson, but something in between would be more appropriate. The following equation can be used to model that:

$$S = \frac{N}{\frac{s(N-1)}{s + \sqrt[n]{N(1-s)}} + 1} \quad (5)$$

This equation is equivalent to Amdahl's and Gustafson's equations but models the assumption on the serial fraction s through variable n . A value $n = 1$ equals Gustafson's assumption while $n = \infty$ equals Amdahl's assumption. Figure 10 depicts the speedups predicted for different values of n .

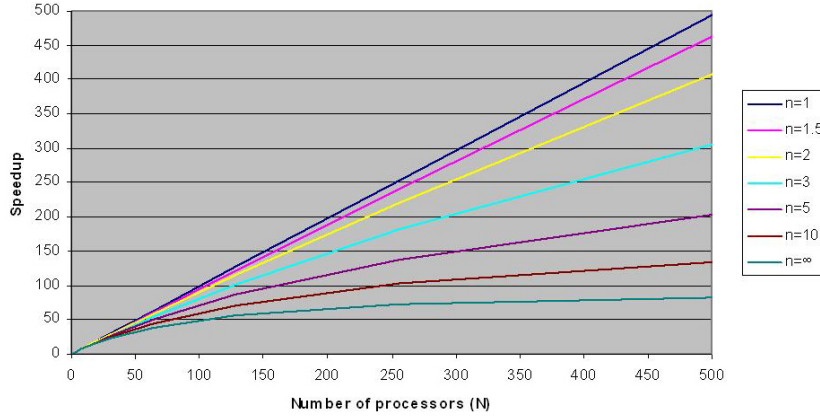


Fig. 10: Prediction of speedup for $s = 1\%$ using Equation 5; $n = 1$ equals Gustafson's assumptions while $n = \infty$ equals Amdahl's assumptions.

An interesting research question is what value n to assign to what application (domain). If this would be possible it would resolve the issue of which assumptions to use in predicting the future and hopefully the confusion about the two laws would diminish.

2.4 Performance Assuming Non-Perfect Parallelization

In the previous section we showed how Amdahl's and Gustafson's assumptions and equations can be applied in predicting the future for non-perfect parallelization. The choice between the two showed to be dependent on the (assumed) characteristics of the application domain. In this section we analyze the impact of non-perfect parallelization on the power-constrained performance growth using both Amdahl's and Gustafson's assumptions. We will leave it up to the reader to connect application domains to either of the two assumptions.

First, we take Amdahl's assumptions to predict the power-constrained performance growth. We assume a symmetric CMP where all cores are being used during the parallel part. The clock frequency of all cores is equal and has been scaled down to meet the power budget. The power-constrained speedup that this symmetric CMP can achieve is given by

$$S_{Amdahl_power-constr._sym.}(t) = \frac{1}{s + \frac{1-s}{N(t)}} \times \frac{f(t)}{f_{orig}} \times \frac{P_{budget}}{P_{total}(t)} \quad (6)$$

and is depicted in Figure 11. We used a serial fraction s ranging from 0.1% to 10%. The figure shows that for $s = 0.1\%$ there is a slight performance drop, compared to ideal, going up to a factor 2 for 2022. However, for $s = 1\%$ there is a performance drop of 10x for 2022 and for $s = 10\%$ there is no performance growth at all.

Indeed we see that Amdahl's prediction is pessimistic, which is an argument for asymmetric or heterogeneous CMPs. If the serial part can be speed up by deploying one faster core, more speedup through parallelism could be achieved. This one fast core could be an aggressive superscalar core, a domain specific

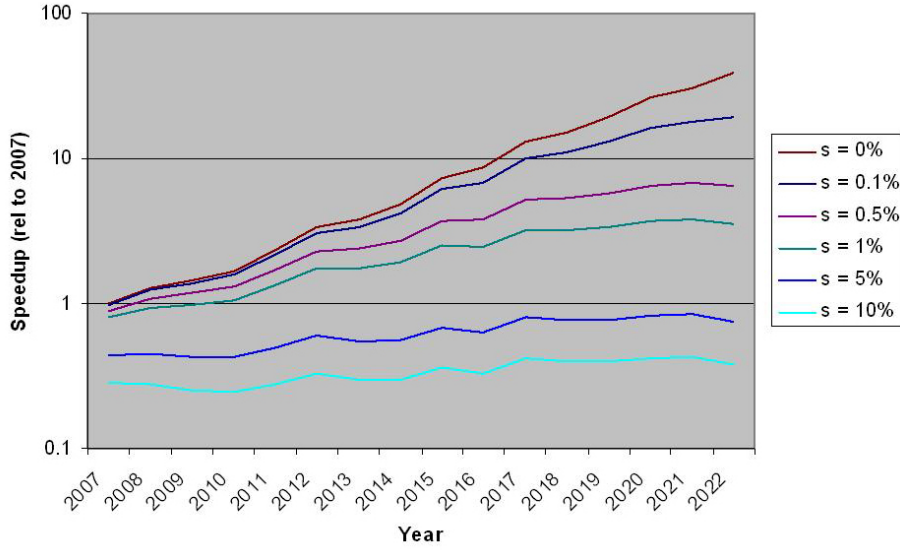


Fig. 11: Prediction of power-constrained performance growth for several fractions of serial code s assuming a symmetric CMP and with Amdahl's assumptions.

accelerator, or a core that runs at a higher clock frequency than the others. For this experiment we assume identical cores but increase the clock frequency of one core during the serial part. Note that during this time the other cores are inactive and thus the power budget is not exceeded. The speedup this asymmetric CMP can achieve is given by

$$S_{Amdahl_power-constr._asym.}(t) = \frac{1}{s \times \frac{f_{orig}}{f(t)} + \frac{1-s}{N(t)} \times \frac{f_{orig}}{f(t)} \times \frac{P_{total}(t)}{P_{budget}}} \quad (7)$$

and is depicted in Figure 12. Note that both this equation and Equation 6 become identical to Equation 2 if $s = 0\%$. The results show that for $s = 0.1\%$ there is only a very small performance drop compared to ideal parallelization. For $s = 1\%$ the performance drop is 2.3x while for $s = 10\%$ the performance drops 14 times compared to ideal parallelization but considerable performance growth is predicted over time. These results show that asymmetric CMPs are a good choice to improve performance, if Amdahl's assumptions are correct and if the serial fraction is larger than approximately 0.5%.

Second, we predict the power-constrained performance growth using Gustafson's assumptions. Again, we assume a symmetric CMP where all cores are being used during the parallel part. The clock frequency of all cores is equal and has been scaled down to meet the power budget. The power-constrained speedup that this symmetric CMP can achieve is given by

$$S_{Gustafson_power-constr._sym.}(t) = (N + (1 - N) \times s') \times \frac{f(t)}{f_{orig}} \times \frac{P_{budget}}{P_{total}(t)} \quad (8)$$

and is depicted in Figure 13. The figure shows that for any value s' between 0% and 10% there will be no significant performance loss compared to ideal

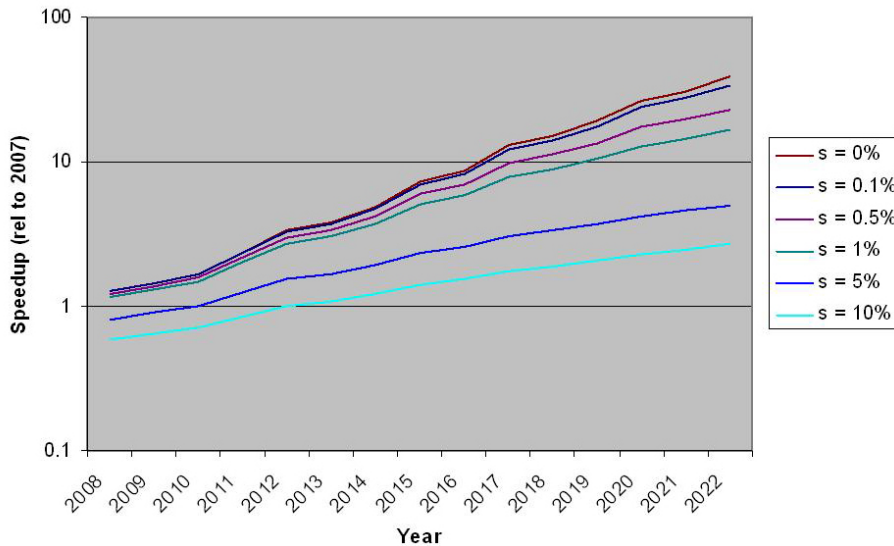


Fig. 12: Prediction of power-constrained performance growth for several fractions of serial code s assuming an asymmetric CMP and with Amdahl's assumptions.

parallelization. For $s' = 10\%$ in 2022 the performance is 11% less than the ideal parallelization $s' = 0\%$ case. Thus we can conclude that homogeneous CMPs are the best choice if Gustafson's assumptions are correct.

3 Conclusions

In this paper we analyzed the impact of the power wall on CMP design. Specifically, we investigated the limits to performance growth of CMPs due to technology improvements with respect to power constraints. As a case study we modelled a CMP consisting of Alpha 21264 cores, scaled to future technology nodes according to the ITRS roadmap. It was found that in 2022 such a CMP would contain 999 cores, each consuming 1.5 W when running at the maximum possible frequency of 14 GHz . The total CMP, at full blown operation, would consume 1.5 kW while the power budget predicted by the ITRS is 198 W .

From these figures it is clear that power has become a major design constraint, and will remain a major bottleneck for performance growth. However, it does not mean that the power wall has been hit for CMPs. We calculated the power-constrained performance growth and showed that technology improvements enables a doubling of performance every three years for CMPs, which is in line with the number of cores per die.

However, there is another threat for CMPs which is Amdahl's Law. The speedup achieved by parallelism is limited by the size of the serial fraction s . Amdahl assumes a fixed program and input size and thus uses a constant fraction s over time. Using Amdahl's equation and assumptions we predicted the power-constrained performance growth for several fractions s . For a symmetric CMP, 1% of serial code decreases the speedup by a factor of up to 10. For an

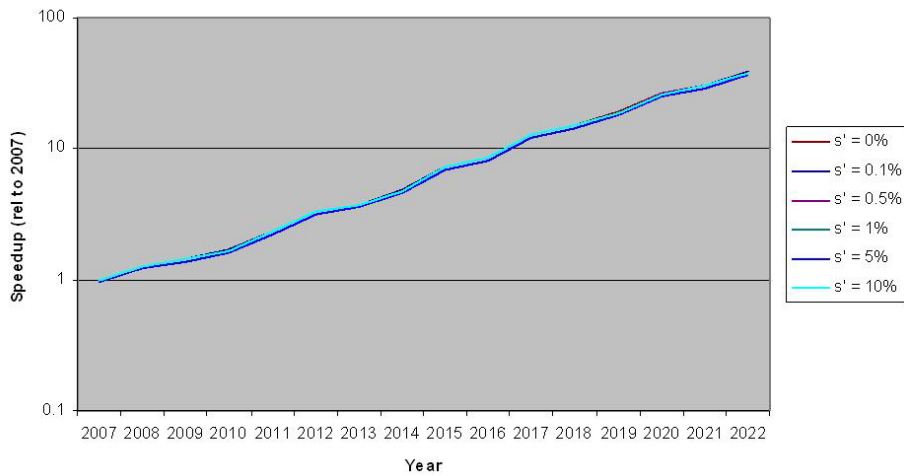


Fig. 13: Prediction of power-constrained performance growth for several fractions of serial code s' with Gustafson's assumptions.

asymmetric CMP, where the serial code is executed on a core that runs at a higher clock frequency, 1% of serial code reduces the achievable speedup by a factor of up to 2.

On the other hand, there is Gustafson's Law which assumes that the input size grows equally with increasing number of processors and that the serial part has a constant absolute execution time independent of the input size. We also predicted the power-constrained performance increase using Gustafson's assumptions. The results are much more optimistic as even for $s' = 10\%$ in 2022 the performance loss is only 11% compared to ideal parallelization.

The question whether Amdahl's or Gustafson's assumptions are believed to be valid for a certain application domain we leave up to the reader. Most likely some can be characterized as 'Amdahl', some as 'Gustafson' and other as something in between.

From the results of this paper we conclude that in order to avoid hitting the power wall the following two principles should be followed. First, at the architectural level power efficiency has to become the main design constraint and evaluation criterion. The transistor budget is no longer the limit but the power those transistors consume. Thus performance alone should no longer be the metric but performance per watt (or a similar power efficiency metric like performance per transistor [13], $BIPS^3/W$ [14], and others [15]). Second, for application domains that follow Amdahl's assumptions asymmetric or heterogeneous designs are necessary. For those the need to speedup serial code remains. A challenge for computer architects is to combine speedup of serial code with power efficiency.

A CMP that follows these principles could for example look like this: a few general purpose high speed cores (e.g. aggressive superscalar), many general purpose power efficient cores (no superscalar, no out-of-order, no deep pipelines, etc.), and domain specific accelerators. The latter provides the most power efficient solution and also allows fast execution of serial code. For example, entropy

coding in video codecs is a perfect candidate to be implemented by an accelerator. It is highly serial and benefits from high clock frequencies, but has mainly bit level operations and thus a simple 8- or 16-bit core with a small instruction set would be best [12]. Furthermore, dynamic voltage/frequency scaling can be applied to optimize the performance-power balance, while hardware support for thread and task management reduces the energy of the overhead introduced by parallelism. A lot more techniques and architectural directions are possible.

Summarizing, from this study we conclude that for the next decade CMPs can provide significant performance improvements without hitting the power wall, even though power severely limits performance growth. Technology improvements will provide the means, however, to achieve the possible performance improvements power efficiency should be the main design criterion at the architectural level.

Acknowledgment

The authors would like to thank Stefanos Kaxiras for his input on the methodology used in this paper.

References

- [1] Mendelson, A.: How many cores are too many cores? Presentation at 3rd HiPEAC Industrial Workshop.
- [2] Hill, M., Marty, M.: Amdahls law in the multicore era. to appear in *IEEE Computer* (2008)
- [3] ITRS: International technology roadmap for semiconductors, 2007 edition (2007) <http://www.itrs.net>.
- [4] Kessler, R.: The Alpha 21264 microprocessor. *Micro, IEEE* **19**(2) (1999) 24–36
- [5] Tiler: TILE64(TM) Processor Family <http://www.tiler.com>.
- [6] ClearSpeed: The CSX600 Processor <http://www.clearspeed.com>.
- [7] Stenström, P.: Chip-multiprocessing and Beyond. In: *Proc. 12th Int. Symp. on High-Performance Computer Architecture*. (2006) 109–109
- [8] Hennessy, J., Patterson, D.: *Modern Computer Architecture - A Quantitative Approach*. 4th edn. Morgan Kaufman Publishers (2007) page 3.
- [9] Amdahl, G.: Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Conference Proceedings* **30**(8) (1967) 483–485
- [10] Gustafson, J.: Reevaluating Amdahl's law. *Communications of the ACM* **31**(5) (1988) 532–533
- [11] Shi, Y.: Reevaluating amdahl's law and gustafson's law (1996) <http://www.cis.temple.edu/shi/docs/amdahl/amdahl.html>.

-
- [12] Meenderinck, C., Azevedo, A., Juurlink, B., Alvarez, M., Ramirez, A.: Parallel scalability of video decoders. Technical report, Delft University of Technology (April 2008) <http://ce.et.tudelft.nl/publications.php>.
 - [13] Hofstee, H.: Power efficient processor architecture and the cell processor. In: 11th Int. Symp. on High-Performance Computer Architecture. (2005) 258–262
 - [14] Lee, B., Brooks, D.: Effects of Pipeline Complexity on SMT/CMP Power-Performance Efficiency. In: Proc. of Workshop on Complexity Effective Design. (June 2005)
 - [15] Hofstee, H., Microelectron, I., Austin, T.: Power-constrained microprocessor design. In: Proc. of IEEE Int. Conf. on Computer Design: VLSI in Computers and Processors. (2002) 14–16