Analyzing specifics of scalability laws for proper modeling of a system’s throughput

Volodymyr Kuharsky[1,†], Dmytro Mykhalyk[1,*,†], Yuri Humen[1,†]

[1] *Ternopil Ivan Puluj National Technical University, Ruska str. 56, Ternopil 46001, Ukraine*

Abstract

The article revisits scalability laws to refine system throughput modeling practices. Amdahl’s Law, Gustafson’s Law, and the Universal Scalability Law are critically analyzed: considering their individual merits and limitations in capturing the essence of system scalability. With theoretical insights and empirical data, this research provides practical guidelines for selecting the most suitable scalability model to ensure accurate and effective modeling of system throughput in diverse computing environments.

Keywords

Scalability, throughput, speedup factor, Amdahl’s Law, Universal Scalability Law, coherence, crosstalk, serial

1. Introduction

In today's world, there are a vast number of artificial systems designed to make our lives easier, more efficient, more safe, more entertaining, etc. Most of them are designed with performance in mind to meet the constantly growing demands of their users. The ability of a particular system to handle a growing amount of work or its potential to accommodate growth defines scalability.

Scalability is crucial in system design for a few reasons:

- Growth handling. A properly scalable product should handle the increasing number of users.
- Improve performance. A properly scalable product should degrade performance more gradually.
- Optimize costs. A properly scalable product should save business investment in hardware to operate according to the load.

Highly scalable systems are crucial components of cloud computing, search engines, container orchestration platforms, and more. These systems play a key role in addressing complex challenges across various industries. For instance, researchers use scalable

CITI’2024: 2nd International Workshop on Computer Information Technologies in Industry 4.0, June 12–14, 2024, Ternopil, Ukraine

[*] Corresponding author.

[†] These authors contributed equally.

volodymyr_kukharskyy@tntu.edu.ua.(V. Kuharsky); d.mykhalyk@gmail.com (D. Mykhalyk); h_yuri@ukr.net (Y. Humen)

0009-0009-6937-9526 (V. Kuharsky); 0000-0001-9032-695X (D. Mykhalyk); 0000-0002-5975-4355 (Y. Humen)

computer simulations to tackle issues such as natural gas dehydration for use as a motor fuel [1] and to study problems related to gas transport in solids and the diffusion of benzene using scalable computer modeling [2]. Let's dive behind the meaning of the “scalability laws” to understand better what they are and what they offer in the system’s throughput modeling.

2. Main Part

So, what are the scalability laws? Unfortunately, in terms of combining computational power, 1 + 1 does not equal 2. It could be some sort of cost-effective for batch work, but this combination was not effective for OLTP (Online transaction processing) and other real-time computations.[3]

Gene Amdahl made an early attempt to quantify the speedup factor – a number that shows how much additional performance a system will get by adding more computational units to that system to battle the increasing load.[1] Later became known as Amdahl’s Law. He simplistically assumed that a fraction of p of the program’s execution time was infinitely parallelizable with no overhead, while the remaining fraction, $1 - p$, is totally sequential.[4]

$$S(N) = \frac{1}{(1-p)+\frac{p}{N}} \tag{1}$$

where S – speedup factor, N – number of computational units, and p – a fraction of the work that can be parallelized.

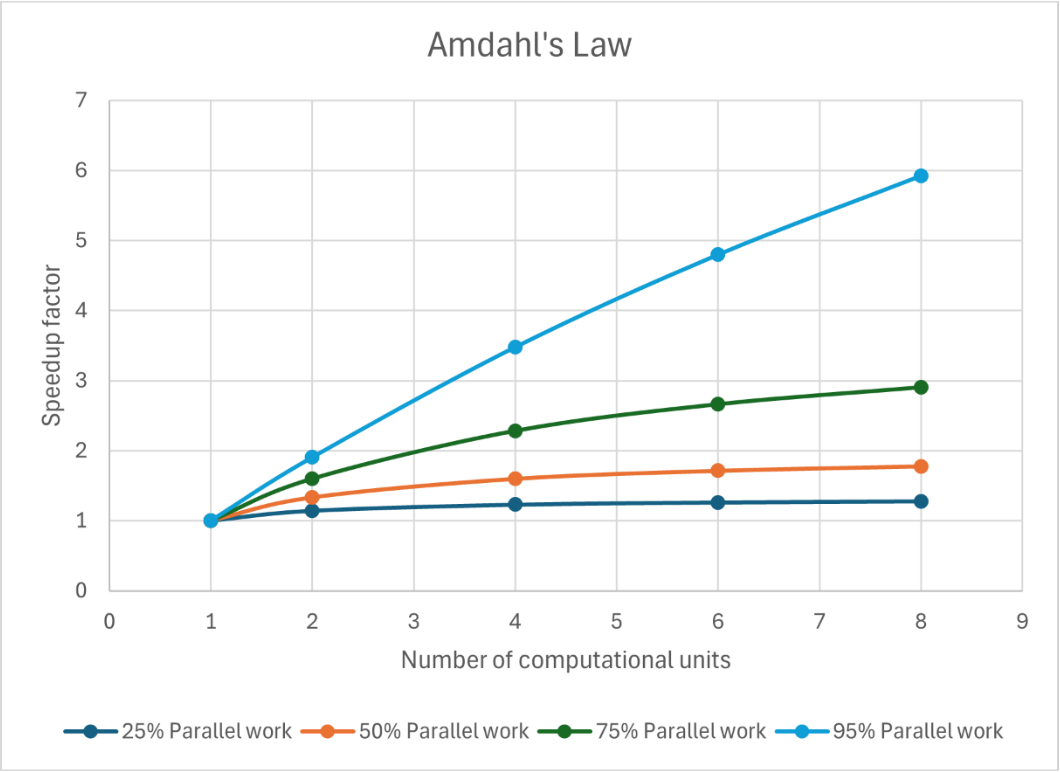

Figure 1: Visualization of the speedup increase for parallelizable work using Amdahl's Law

Figure 1 shows how drastically the speedup factor can degrade if there is even an insignificant amount of serial work in the system's load. The X-axis contains numbers of several computational units dedicated to the workload. The Y-axis shows how much of the potential increase of computational speed for the number of computational units dedicated to the workload. In a linear(ideal) scaling scenario a speedup factor should increase proportionally to the number of computational units, but Figure 1 has already shown how far we are from that situation to happen.

The situation can get much worse if we continue to add more and more computational units to our system. For 128 of the total computational units, our speedup starts to flatten up in Figure 2.

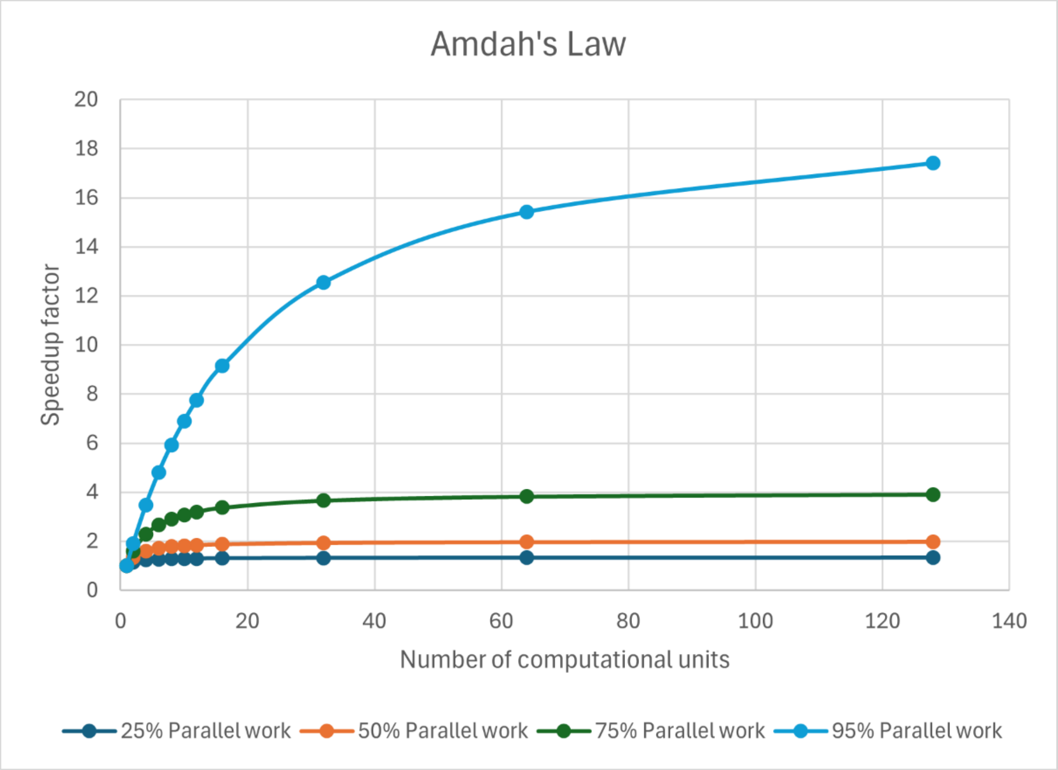

Figure 2: Visualization of the speedup increase at 128 computational units.

As you can see here, the function of S is monotonically increasing and bound on the upper side to the fraction of workload that cannot be parallelized, so the following is true:

$$\lim_{N\to\infty} S(N) = \frac{1}{1-p} \qquad (2)$$

While Amdahl's law is good for theoretical reasoning about peak performance it has its shortcomings. It assumes that the problem size remains the same when utilizing more cores to execute the application. It has no maximum that shows its lack of ability to model the real system behavior with retrograde speedup.[6]

Neil Gunther proposed a slightly different approach to scalability quantification. His model not only uses the serial penalty but also introduces a coherence or crosstalk penalty which results in delays for data to become consistent during the execution for multiprocessor environments. Gunther's Law later became known as the Universal Scalability Law. It can be described as follows:

$$S(N) = \frac{N}{1+\alpha(N-1)+\beta N(N-1)}, \qquad (3)$$

where N – number of computational units, α - fraction between 0 and 1 represents a serial penalty in the workload, β - a fraction between 0 and 1, represents a coherence penalty in the workload.

In application throughput modeling N can represent the number of active users for maximum throughput increase.

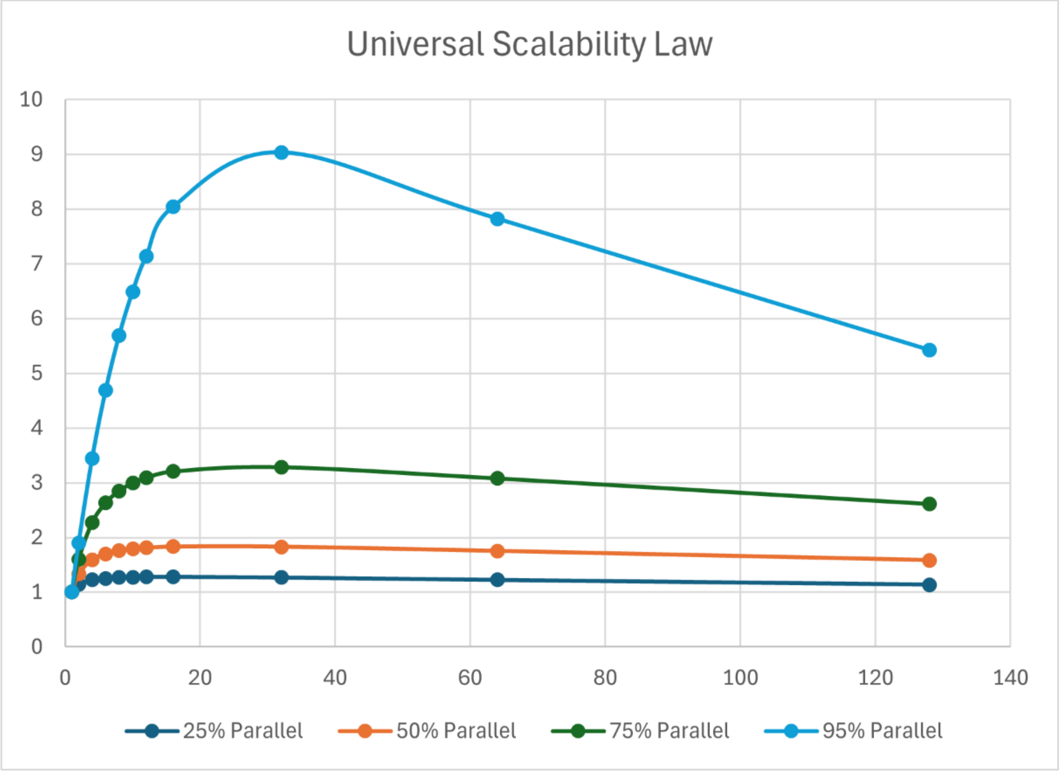

Figure 3: USL visualization with 0.1 percent of coherence penalty.

If *β=0*, then equation (3) simplifies to the well-known Amdahl's Law[5]:

$$S(N) = \frac{N}{1 + \alpha(N-1)} \tag{4}$$

If both coefficients are 0, then we should observe linear scalability for N processors, but it is pure theoretical observation.

As you can see in Figure 3 Universal scalability law has its performance peak followed by a decrease in actual system throughput. This is possible because of the coherence penalty, which stacks fast for workloads with high numbers of parallelism.

In terms of capacity planning, [7] USL has a sufficient number of parameters for predicting the effects of concurrency, contention, and coherency delay, but it can't be super precise because the scalability of any particular system should be considered in light of many factors simultaneously.

Another approach was proposed by Dr. John Gustafson to overcome the pessimistic results of Amdahl's Law speedup dependence from a serial delay as described in (2). Instead of fixing the parallelizable part of the workload, it fixes the running time.[8] In other words, it answers the following question: How much of the execution time would this problem have taken if it ran on a serial processor instead of a multicore or multiprocessor system?[9] Gustafson's observations show more generally that the serial fraction of the workload does not theoretically limit parallel speed enhancement. Algebraically it can be described as follows:

$$S(N) = N - \alpha(N-1), \tag{5}$$

where S – scaled speedup over N number of computational units, α - a fraction between 0 and 1 representing the serial part of the workload.

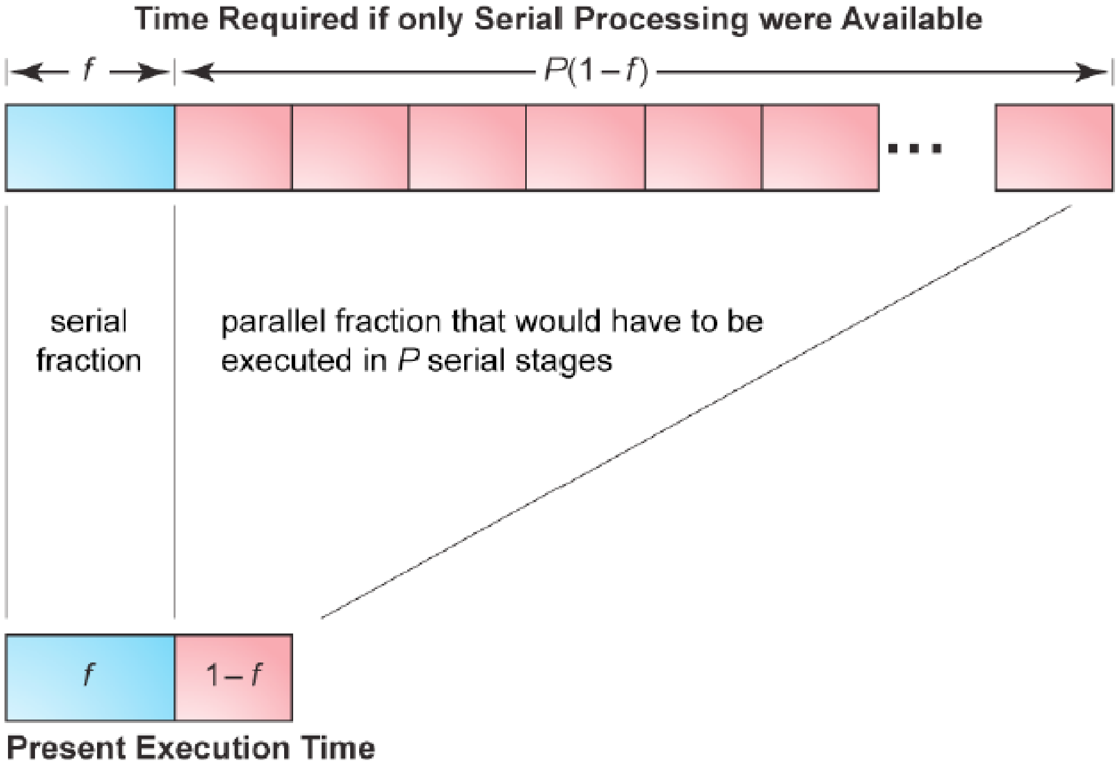
Time Required if only Serial Processing were Available
f
P(1 – f)
serial fraction
parallel fraction that would have to be executed in P serial stages
f
1 – f
Present Execution Time

Figure 4: Visualization of Gustafson-Barsis Law [6]

Later, [10] J. Gustafson summarized his work: "The model is not a contradiction of Amdahl's law as some have stated, but an observation that Amdahl's assumptions don't match the way people use parallel processors. People scale their problems to match the power available, in contrast to Amdahl's assumption that the problem is always the same no matter how capable the computer."

Figure 4 shows how parallel processing can reduce the execution time.

Even after Gustafson's Law provided a more optimistic perspective on parallel computing compared to Amdahl's Law it omits the complexities of modern software design and can poorly predict scaled speedup for algorithms with nonlinear runtimes.[11]

3. Results discussion

After careful analysis of the three proposed laws in this article, the authors advise using the Universal Scalability Law for scalability predictions of real-world systems. Because neither Amdahl's Law nor Gustafson-Barsis Law takes into account the coherence penalty that is almost inevitable in today's multicore and multiprocessor world and, at the same time, the most significant part of the USL equation that will kick diminishing returns early on for throughput scaling that can lead to overall performance degradation.

To achieve better results in throughput modeling using USL, try to use measurement data for throughput and concurrency obtained from stable workloads. USL works best for constant workloads with no variability. Still, the authors admit that obtaining such measurements can sometimes be unrealistic because of the significant complexities in the design and function of real-world applications.

4. Conclusions

All three models can model the capacity of the system throughput. Amdahl’s Law provides a straightforward way to estimate the maximum achievable throughput by considering the proportion of the workload that can be parallelized but completely ignores parallel crosstalk which is inevitable in modern applications.

Gustafson’s Law recognizes the potential for adjusting the problem size to fully utilize additional resources, which can lead to higher system throughput in parallel environments, but also ignores the parallel crosstalk, which can result in more theoretical throughput values.

Universal Scalability Law accounts for saturation effects that may occur as system resources are increased, providing insights into the maximum achievable throughput and the diminishing returns of adding more resources, more practical than the first two but has a cost of additional complexity and more experienced researchers when determining coefficients for an actual system.

In summary, each law has its strengths and weaknesses, and the choice of model depends on the specific characteristics of the system being analyzed and the level of detail required in the analysis.

References

[1] Petryk M., Khimitch A., Petryk M.M., Fraissard J. Experimental and computer simulation studies of dehydration on microporous adsorbent of natural gas used as motor fuel. Fuel. Vol. 239, 1324–1330 (2019) https://doi.org/10.1016/j.fuel.2018.10.134

[2] Petryk M., Leclerc S., Canet D., Fraissard J. Modeling of gas transport in a microporous solid using a sclice selection procedure: Application to the diffusion of benzene in ZSM5. Catalysis Today. Vol. 139, Issue 3, 234-240 (2008) https://doi.org/10.1016/j.cattod.2008.05.034

[3] T. Critchley High Performance IT Services 1st edition. Auerbach Publications 2016

[4] Mark D. Hill, Michael R. Marty Amdahl’s Law in the Multicore Era URL:https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/34400.pdf

[5] Neil J. Gunther A New Interpretation of Amdahl’s Law and Geometric Scaling https://doi.org/10.48550/arXiv.cs/0210017

[6] C. Poola, R. Saxena On Extending Amdahl’s Law to Learn Computer Performance URL: https://arxiv.org/abs/2110.07822

[7] B. Shwartz, E. Fortune Forecasting MySQL Scalability with Universal Scalability Law URL:https://www.percona.com/sites/default/files/white-paper-forecasting-mysql-scalability.pdf

[8] J. Gustafson Gustafson's Law, (2011) ResearchGate.net. doi: 10.1007/978-0-387-09766-4_78

[9] J. Gustafson Reevaluating Amdahl's Law (1988). URL:http://www.johngustafson.net/pubs/pub13/amdahl.pdf

[10] J. Gustafson Gustafson's Law. URL: http://www.johngustafson.net/glaw.html

[11] L. Snyder Type Architectures, shared memory, and the corollary of modest potential (1986) doi: 10.1146/annurev.cs.01.060186.001445