On the Convergence of Amdahl's Law to the Ideal Parallel Speedup Under Specific Conditions

Hendrik Böck

22nd October 2024

1 Motivation

This paper investigates the conditions under which Amdahl's Law achieves equivalence with the ideal parallel speedup predicted by the work-span model. It is to be demonstrated, that this equality is realized exclusively when all parallelizable tasks are independent and have uniform execution times. Additionally, counterexamples are provided, accompanied by example calculations using specific task structures, to illustrate that in cases where tasks exhibit dependencies or non-uniform execution times, this equivalence does not hold. In such scenarios, no direct correspondence exists between Amdahl's Law and the Work/Span model.

2 Definitions & Notation

2.1 Definitions

Definition 2.1 (Amdahl's Law) *Amdahl's Law describes the theoretical speedup S of a computational task when using multiple processors in parallel, constrained by the portion of the task that cannot be parallelized. The speedup is expressed as:*

$$S_p = \frac{1}{s + \frac{f}{p}} = \frac{1}{(1-f) + \frac{f}{p}} \tag{1}$$

*where f represents the fraction of the task that is **sequential** (cannot be parallelized), and p denotes the number of processors.*

Definition 2.2 (Ideal Parallel Speedup) [1] *The Work/Span model defines the **ideal parallel speedup** S_{ideal} achievable by parallelization as the ratio of the total work to the critical path length (span). It is expressed as:*

$$S_{ideal} = \frac{T_w}{T_{span}} \tag{2}$$

where T_w represents the total amount of work (the sum of all operations), and T_{span} denotes the length of the critical path, i.e., the time taken along the longest sequence of dependent tasks.

1: Work/Span Model

2.2 Terminology

- s: Serial fraction of the total workload ($0 \leq s \leq 1$).
- f: Parallelizable fraction of the total workload ($f = 1 - s$).
- p: Number of processors available for parallel execution.
- T_1: Execution time on a single processor.
- T_p: Execution time on p processors.
- S_p: Speedup with p processors ($S_p = \frac{T_1}{T_p}$).
- T_w: Total computational work ($T_w = T_1$).
- T_{span}: Length of the critical path (span).
- T_x: Time for serial tasks before the parallel section. [2]
- T_y: Time for serial tasks after the parallel section.[2]

2: There the serial fraction can be computed as follows:

$$s = \frac{T_x + T_y}{T_1}$$

3 Theorems & Proofs

3.1 Convergence Case

Theorem 3.1 *Under the conditions where all parallelizable tasks are independent and take the same amount of time, the limit of the speedup predicted by Amdahl's Law as $p \to \infty$ equals the ratio of the total work to the critical path length, as defined by the Work/Span model:*

$$\lim_{p \to \infty} S_p = \lim_{p \to \infty} \left(\frac{1}{s + \frac{f}{p}} \right) = \frac{1}{s} = \frac{T_w}{T_{span}} \tag{3}$$

Assumtions / Preconditions:

- **Independent Parallelizable Tasks:** All tasks in the parallel fraction are independent.
- **Uniform Task Time:** Each parallelizable task takes the same amount of time.
- **Serial Tasks:** There are serial tasks before and after the parallel section (T_x and T_y).

Execution time on a single processor (T_1):

The total execution time on a single processor is the sum of the time for the serial tasks and the parallel fraction:

$$T_1 = T_x + T_{\text{parallel}} + T_y \tag{4}$$

where T_{parallel} is the total time for parallelizable tasks executed serially.

Execution time on p processors (T_p):

The total execution time on p processors is the sum of the time for the serial tasks and the parallel fraction, divided by the number of processors:

$$T_p = T_x + \frac{T_{\text{parallel}}}{p} + T_y \tag{5}$$

where no overhead is assumed for parallelization.

Execution time for critical path (T_{span}):

Traditionally, the model determines the span by identifying the longest path through a finite number of sequential and parallel tasks. However, in this generalized version, the number of parallel-executable tasks is unknown, leading to the assumption of a homogeneous task that can be subdivided into infinitely small subtasks based on the number of processors in the system.

In this model, T_x represents the time required to complete an arbitrary number of serial tasks before the parallel section, while T_y is the time needed to complete the serial tasks that follow the parallel section. The parallel portion of the task, T_{parallel}, is divided among p processors. Thus, the critical path length is given by:

$$T_{\text{span}} = T_x + \frac{T_{\text{parallel}}}{p} + T_y = T_p \tag{6}$$

This equation captures the total execution time, where the serial tasks before and after the parallel section contribute to T_x and T_y, and the parallel section is scaled by the number of processors p. The overhead is assumed to be negligible in this context

Execution time for total work (T_w):

The total work T_w is defined as the sum of the time for all tasks, including the serial tasks before and after the parallel section, and the parallel fraction:

$$T_w = T_x + T_{\text{parallel}} + T_y = T_1 \tag{7}$$

Limit as $p \to \infty$:

As the number of processors tends toward infinity, several key observations can be made regarding the behavior of parallel execution. First, the execution time of the parallel portion of the computation approaches zero:

$$\lim_{p \to \infty} \frac{T_{\text{parallel}}}{p} = 0 \tag{8}$$

Consequently, the total execution time on p processors, denoted as T_p, converges to the sum of the execution times of the non-parallelizable portions of the computation, specifically:

$$\lim_{p \to \infty} T_p = T_x + 0 + T_y = T_{\text{span}} \tag{9}$$

Finally, the speedup predicted by Amdahl's Law S_p, reaches a limiting value as the number of processors increases. This limiting speedup is given by:

$$\lim_{p \to \infty} S_p = \lim_{p \to \infty} \left(\frac{1}{s + \frac{f}{p}} \right) = \frac{1}{s} \tag{10}$$

As previously established, the total work T_w is defined as the time required to complete the task using a single processor, represented by $T_w = T_1$. Additionally, the critical path length, denoted as T_{span}, corresponds to the sum of the execution times for the sequential portions of the task, specifically $T_{\text{span}} = T_x + T_y$. Based on this, the ideal speedup according to the Work/Span model is determined by the ratio of the total work to the critical path length, which can be expressed as:

$$S_{\text{ideal}} = \frac{T_w}{T_{\text{span}}} = \frac{T_1}{T_x + T_y} = \frac{1}{s} \tag{11}$$

Result:

As therefore can be established, under the specific conditions of independent parallelizable tasks with uniform execution times, the speedup predicted by Amdahl's Law converges to the ideal parallel speedup as the number of processors increases. This convergence is a direct consequence of the equivalence between the total work and the critical path length, as defined by the Work/Span model:

$$\lim_{p \to \infty} S_p = \frac{1}{s} = \frac{T_w}{T_{\text{span}}} = S_{\text{ideal}} \tag{12}$$

3.2 Failure Case

Theorem 3.2 *If the parallelizable tasks are not independent or not uniform in execution time, then Amdahl's Law does not converge to the ideal parallel speedup defined by the Work/Span model, and there is no direct relation between the two. Therefore should apply, that if there exist dependencies among parallelizable tasks or tasks have non-uniform execution times, then:*

$$\lim_{p \to \infty} S_p = \frac{1}{s} > S_{ideal} = \frac{T_w}{T_{span}} \tag{13}$$

where S_p is the speedup predicted by Amdahl's Law, and $S_{|textideal}$ is the ideal speedup considering task dependencies and non-uniformity after the Work/Span model.

Assumtions / Preconditions:

- **Dependent Parallelizable Tasks:** Some tasks in the parallel fraction have dependencies.
- **Non-uniform Task Time:** Parallelizable tasks take varying amounts of time.
- **Specific Task Structure:** The task sequence is $T_x, t_A, t_B, t_C, t_D, T_y$ where t_C depends on t_A and t_B. (See Figure 1)

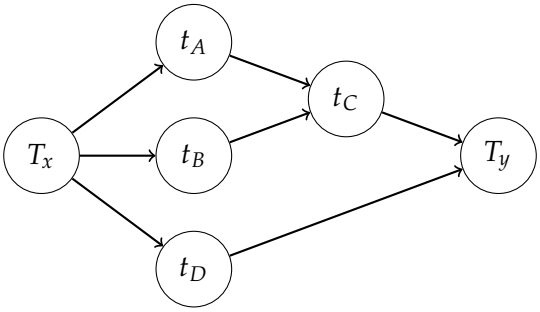

Figure 1: Task Structure with Dependencies

Excution time on a single processor (T_1):

The total execution time on a single processor is the sum of the time for the serial tasks and the parallel fraction:

$$T_1 = T_x + t_A + t_B + t_C + t_D + T_y \tag{14}$$

Execution time on p processors (T_p) with Work/Span model:

The total execution time on p processors is the sum of the time for the serial tasks and the maximum of the longest path through the parallel fraction, taking into account the number of processors:

$$T_{\text{maxpar}} = \max_p((\max_p(t_A, t_B) + t_C), t_D) \tag{15}$$

where T_{maxpar} is the maximum execution time of the parallel fraction on p processors.

Therefore, the total execution time on p processors is:

$$T_p = T_x + T_{\text{maxpar}} + T_y \tag{16}$$

Where the total work T_w is defined as the sum of the time for all tasks, including the serial tasks before and after the parallel section, and the parallel fraction:

$$T_w = T_x + t_A + t_B + t_C + t_D + T_y = T_1 \tag{17}$$

Execution time on p processors (T_p) with Amdahl's Law:

The serial fraction s is calculated as the sum of the time for the serial tasks before and after the parallel section, divided by the total execution time on a single processor:

$$s = \frac{T_x + T_y}{T_1} \tag{18}$$

The parallelizable fraction f is then determined as the difference between the total work and the serial fraction:

$$f = 1 - s = \frac{t_A + t_B + t_C + t_D}{T_1} \tag{19}$$

Therefore Amdahl's Law predicts the speedup on p processors as:

$$S_p = \frac{1}{s + \frac{f}{p}} \tag{20}$$

Limit as $p \to \infty$:

Amdahl's Law predicts the speedup on p processors as:

$$\lim_{p\to\infty} S_p = \frac{1}{s} \tag{21}$$

As a direct consequence of the presence of dependencies among parallelizable tasks or variations in their execution times, Amdahl's Law overestimates the potential speedup. This is a result from the increased critical path length due to the dependencies, which inherently limits parallel efficiency and reduces the achievable speedup. The ideal parallel speedup S_{ideal}, determined by the ratio of the total work to the critical path length, as defined by the Work/Span model, is therefore given by:

$$\lim_{p\to\infty} S_{\text{ideal}} = \frac{T_w}{T_{\text{span}}} = \frac{T_1}{T_x + \max((\max(t_A, t_B), t_C), t_D) + T_y} \tag{22}$$

Result:

As $\max((\max(t_A, t_B), t_C), t_D) > 0$ applies in this case, the ideal parallel speedup S_{ideal} is significantly lower than the limiting speedup predicted by Amdahl's Law. Therefore, in scenarios where tasks exhibit dependencies or non-uniform execution times, the equivalence between Amdahl's Law and the Work/Span model does not hold, and no direct relation exists between the two:

$$\lim_{p\to\infty} S_p = \frac{1}{s} > \lim_{p\to\infty} S_{\text{ideal}} = \frac{T_w}{T_{\text{span}}} \tag{23}$$

4 Conclusion

Amdahl's Law converges to the ideal parallel speedup predicted by the Work/Span model exclusively under stringent conditions, where all parallelizable tasks are both independent and possess uniform execution times. As demonstrated in Theorem 3.2, when tasks exhibit dependencies or have non-uniform execution durations, Amdahl's Law deviates from the Work/Span model's predictions, thereby preventing convergence to the ideal speedup. This outcome underscores the critical importance of task independence and uniformity in ensuring the accuracy of Amdahl's Law for speedup prediction.

Task independence and uniform execution times are essential for Amdahl's Law to accurately predict speedup. When dependencies between tasks are present, the critical path length increases, which inherently limits parallel efficiency and reduces the achievable speedup. In non-ideal conditions characterized by task dependencies and execution time variability, Amdahl's Law tends to overestimate potential speedup, because it fails to account for these dependencies and the resulting load imbalances. In contrast, the Work/Span model provides more accurate speedup estimates by explicitly incorporating the critical path and task dependencies. This enhanced modeling capability allows the Work/Span framework to better reflect the complexities of parallelizable tasks, offering more reliable predictions of parallel performance in realistic scenarios where task dependencies and execution time variability are prevalent.