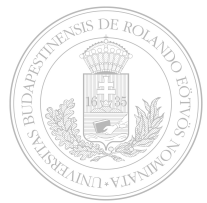
Amdahl's Law

The necessity to parallelize

Balázs Pál

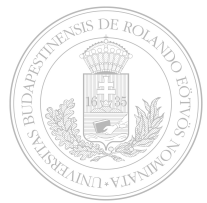
Eötvös Loránd University

Data Models and Databases in Science,
December 3, 2020



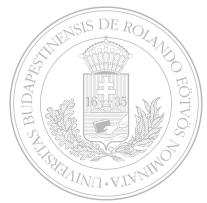
The necessity to parallelize

- Algorithms can easily exceed any reasonable amount of runtime.
  - Example from the slides: *"Algorithms more complex than $\mathcal{O}(n \log(n))$ are hopeless to trace."*



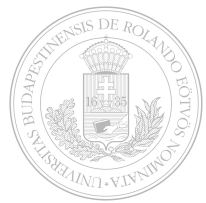
The necessity to parallelize

- Algorithms can easily exceed any reasonable amount of runtime.
  - Example from the slides: *"Algorithms more complex than* $\mathcal{O}(n \log(n))$ *are hopeless to trace."*
  - Solution: parallelization of subtasks!



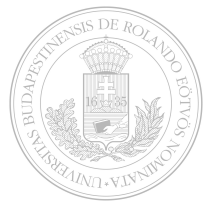
The necessity to parallelize

- Algorithms can easily exceed any reasonable amount of runtime.
  - Example from the slides: *"Algorithms more complex than $\mathcal{O}(n \log(n))$ are hopeless to trace."*
  - Solution: parallelization of subtasks!
- Problems
  - There are task that can be solved only sequentially (eg. reading from disk).
  - There are task which requires a lot of effort and work to parallelize.



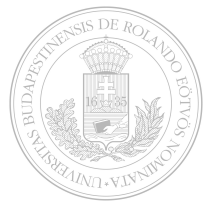
The necessity to parallelize

- Algorithms can easily exceed any reasonable amount of runtime.
  - Example from the slides: *"Algorithms more complex than $\mathcal{O}(n \log(n))$ are hopeless to trace."*
  - Solution: parallelization of subtasks!
- Problems
  - There are task that can be solved only sequentially (eg. reading from disk).
  - There are task which requires a lot of effort and work to parallelize.
- We expect - or at least hope - that parallelization shortens runtime significantly.



Amdahl's law

- Quantifies the theoretical magnitude of speedup due to parallelization.

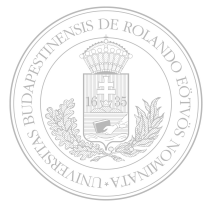


Amdahl's law

- Quantifies the theoretical magnitude of speedup due to parallelization.
- We can split a whole arbitrary algorithm (denoting with 1) into to parts:

$$1 = S + P, \tag{1}$$

where S denotes the fraction of runtime of the sequentially and P the parallel solved part.

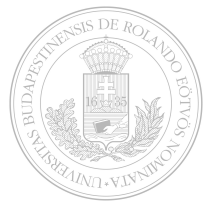


Amdahl's law

- Quantifies the theoretical magnitude of speedup due to parallelization.
- We can split a whole arbitrary algorithm (denoting with 1) into to parts:

$$1 = S + P, \tag{1}$$

  where S denotes the fraction of runtime of the sequentially and P the parallel solved part.
- Sequential part (S)
  - Takes a lot of time in every case, because this part is not parallelizable.

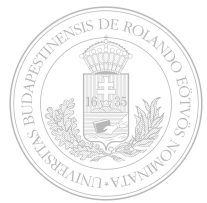


Amdahl's law

- Quantifies the theoretical magnitude of speedup due to parallelization.
- We can split a whole arbitrary algorithm (denoting with 1) into to parts:

$$1 = S + P, \tag{1}$$

  where S denotes the fraction of runtime of the sequentially and P the parallel solved part.
- Sequential part (S)
  - Takes a lot of time in every case, because this part is not parallelizable.
- Parallel part (P)
  - Doing the same operations in parallel on a lot of batch of data.
  - Should be always a lot faster than S.

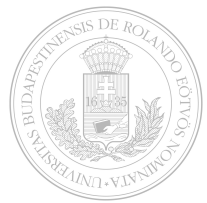


Amdahl's law

- Amdahl's law now can be formulated as the following:

$$Q_{\text{speedup}}(N) = \frac{1}{S + \frac{P}{N}} = \frac{1}{S + \frac{1-S}{N}}, \tag{2}$$

Where N is the number of parallel threads, and the runtime P was expressed in the denominator using equation (1).



Amdahl's law

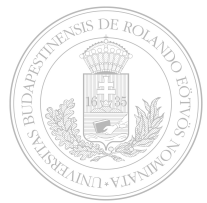
- Amdahl's law now can be formulated as the following:

$$Q_{\text{speedup}}(N) = \frac{1}{S + \frac{P}{N}} = \frac{1}{S + \frac{1-S}{N}}, \tag{2}$$

  Where N is the number of parallel threads, and the runtime P was expressed in the denominator using equation (1).

- The behaviour of the Q speedup's value can be expressed as

$$\begin{cases} Q_{\text{speedup}} = \frac{1}{S} & \text{if } N \to \infty \\ Q_{\text{speedup}} < \frac{1}{S} & \text{else} \end{cases} \tag{3}$$



Amdahl's law

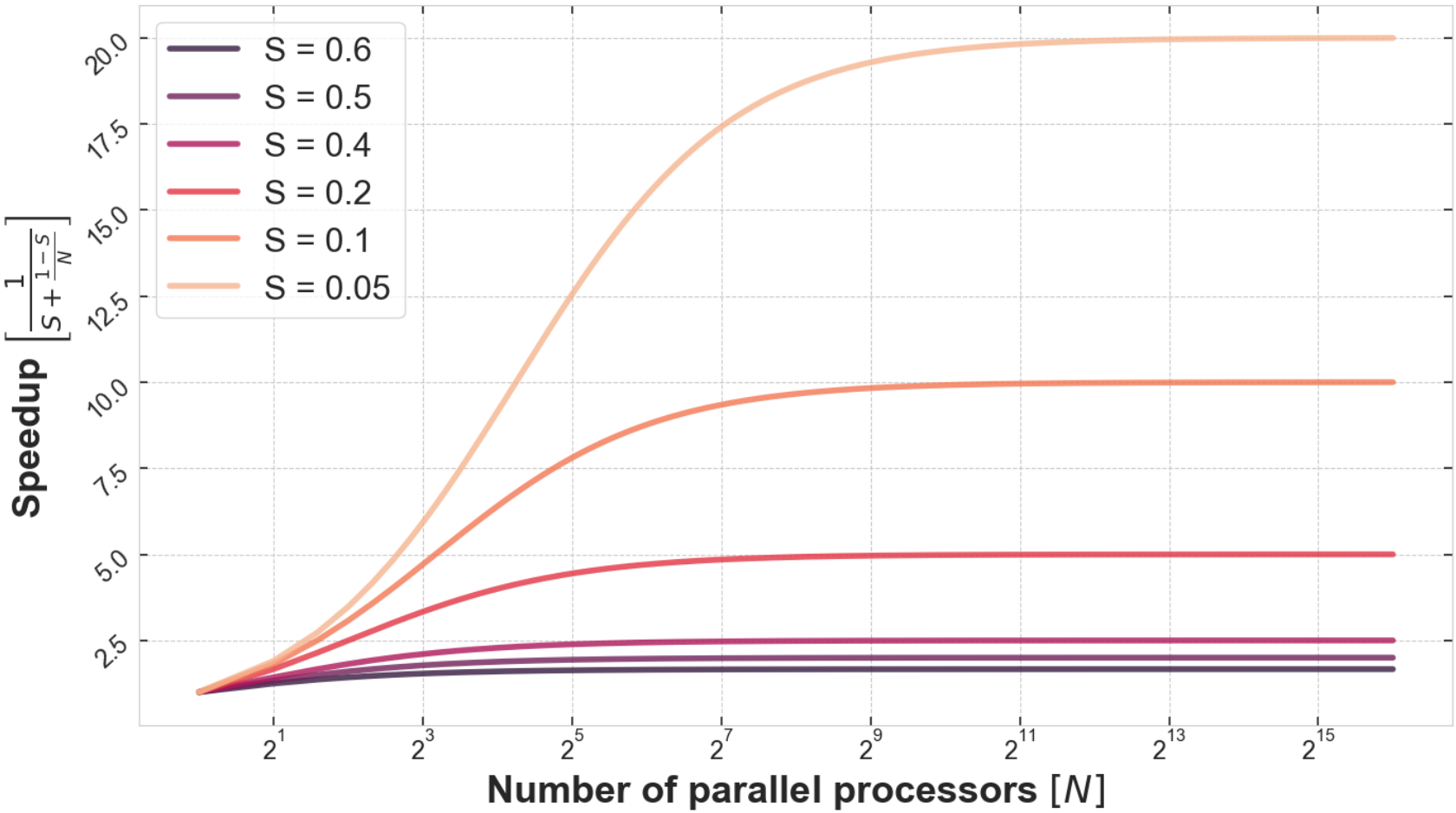


Figure 1: Visualization of the speedup with increasing number of parallel processors for different values of S. The figure shows that when the sequential part of the task is just the small portion of the whole algorithm, then the speedup is the highest. It can also seen, that the limit for the speedup is $\lim_{N\to\infty} = \frac{1}{S}$