Concepts and Models of Parallel and Data-centric Programming

Foundations III

Lecture, Summer 2020

Dr. Christian Terboven <terboven@itc.rwth-aachen.de>

High Performance Computing

RWTH AACHEN UNIVERSITY

Outline

Harsh Realities: Amdahl's Law

Performance

- Definition of Speedup (According to Amdahl)
  - Ratio between serial and parallel Execution of a Program
  - Indicator for relative performance improvement

$$\textit{Speedup } S_p(N) = \frac{T(1)}{T(N)}$$

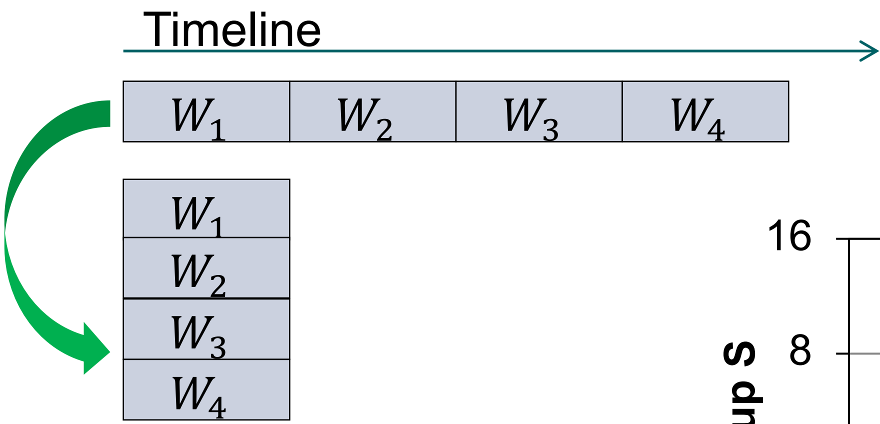
  - With T(N): runtime of a (parallel) program with N Processors

- Efficiency:

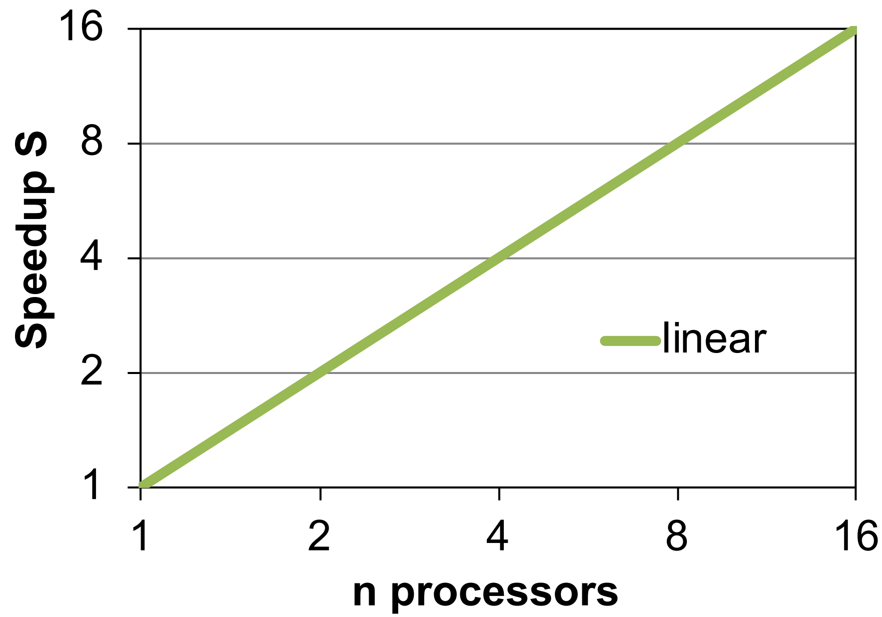
$$\textbf{Efficiency } E_p(N) = \frac{S_p(N)}{N} = \frac{T(1)}{N \cdot T(N)}$$

Linear Speedup

- Ideal situation: All work is perfectly parallelizable: linear Speedup
  – In general: upper bound for parallel execution of programs

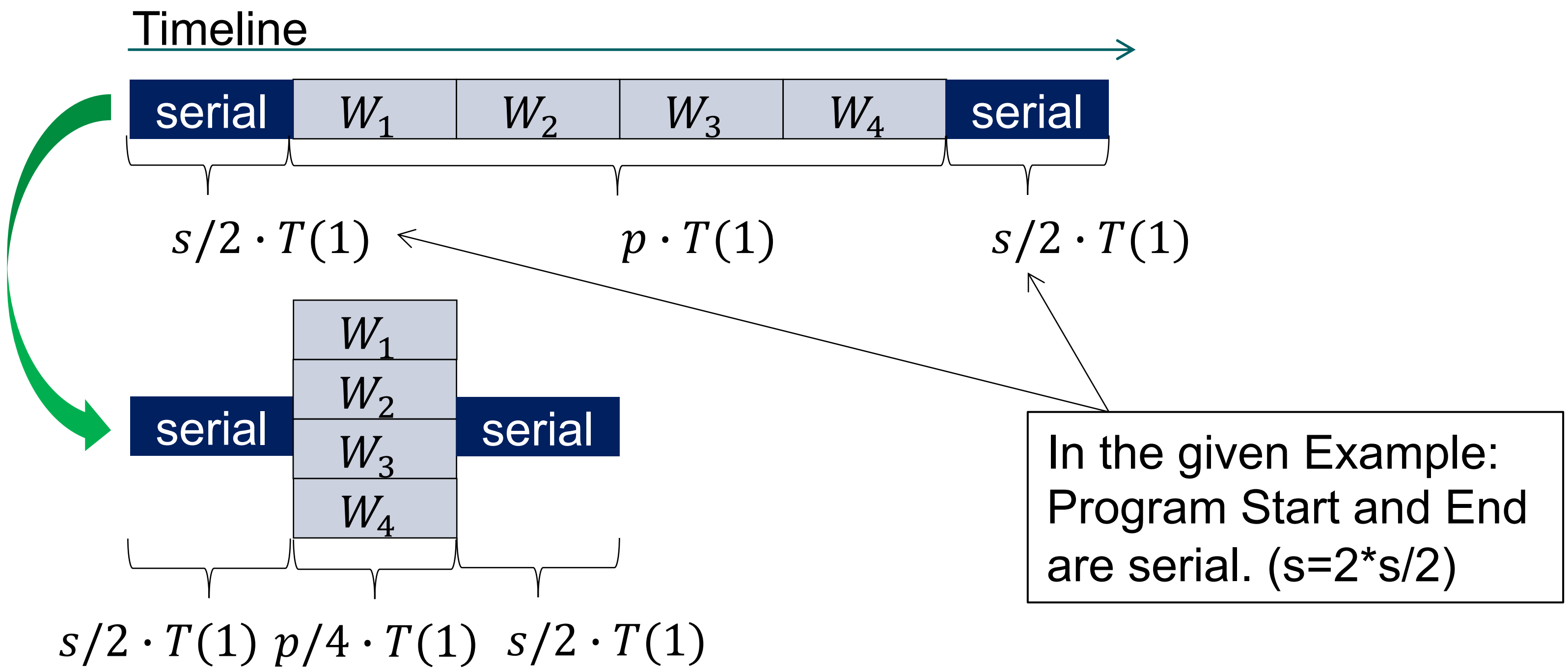


$$\mathbf{S}(\boldsymbol{N}) = \frac{\boldsymbol{T}(\mathbf{1})}{\boldsymbol{T}(\boldsymbol{N})} = \boldsymbol{N}$$



Limitations of scalability

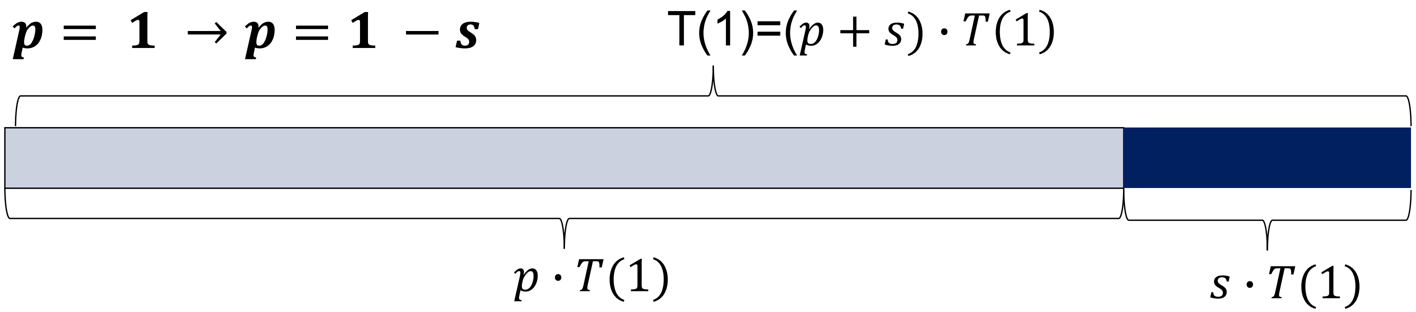
- A model more close to reality: There are serial parts which limit the maximum speedup



- Amdahl's law assumes the program is dividable into an ideal parallelizable fraction *p* and a serial fraction *s* (non-parallelizable)

Amdahl's Law

- $s + p = 1 \rightarrow p = 1 - s$



- The parallelized program's execution time is then assumed to be (with N processors):
  - $\mathbf{T(N)} = (s + \frac{p}{N}) \cdot T(1)$

- The speedup thus resembles to:

$$\boxed{S_p(N) = \frac{T(1)}{T(N)} = \frac{1}{s + \frac{1 - s}{N}}}$$

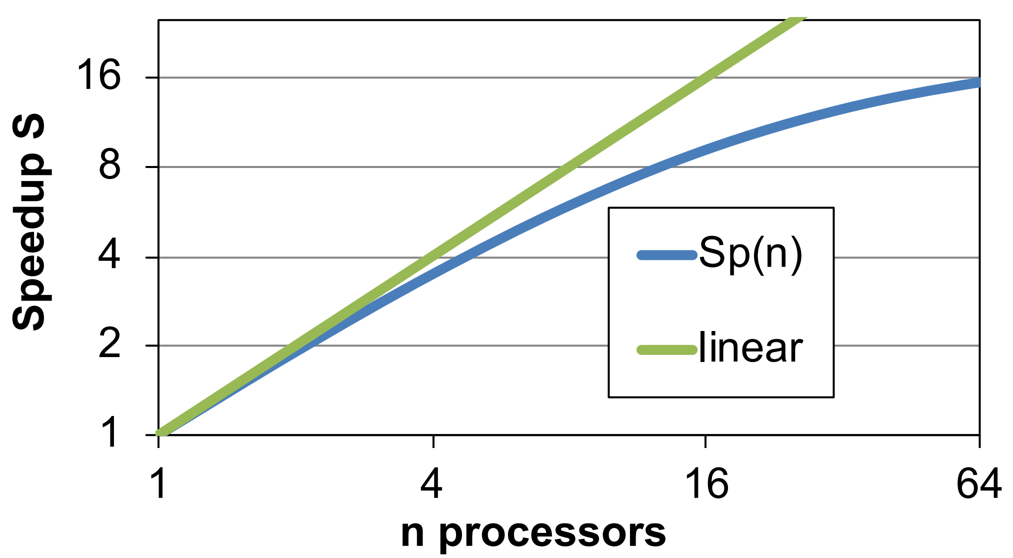
Amdahl's Law (1967)
or "<u>strong scaling</u>"

Further reading: Gene Amdahl: Validity of the Single Processor Approach to Achieving Large-Scale Computing Capabilities. In: AFIPS Conference Proceedings. 30, 1967, S. 483–485

Amdahl's Law (cont.)

- Example: Program with 5% serial and 95% parallel fraction
  – Speedup according to Amdahl:

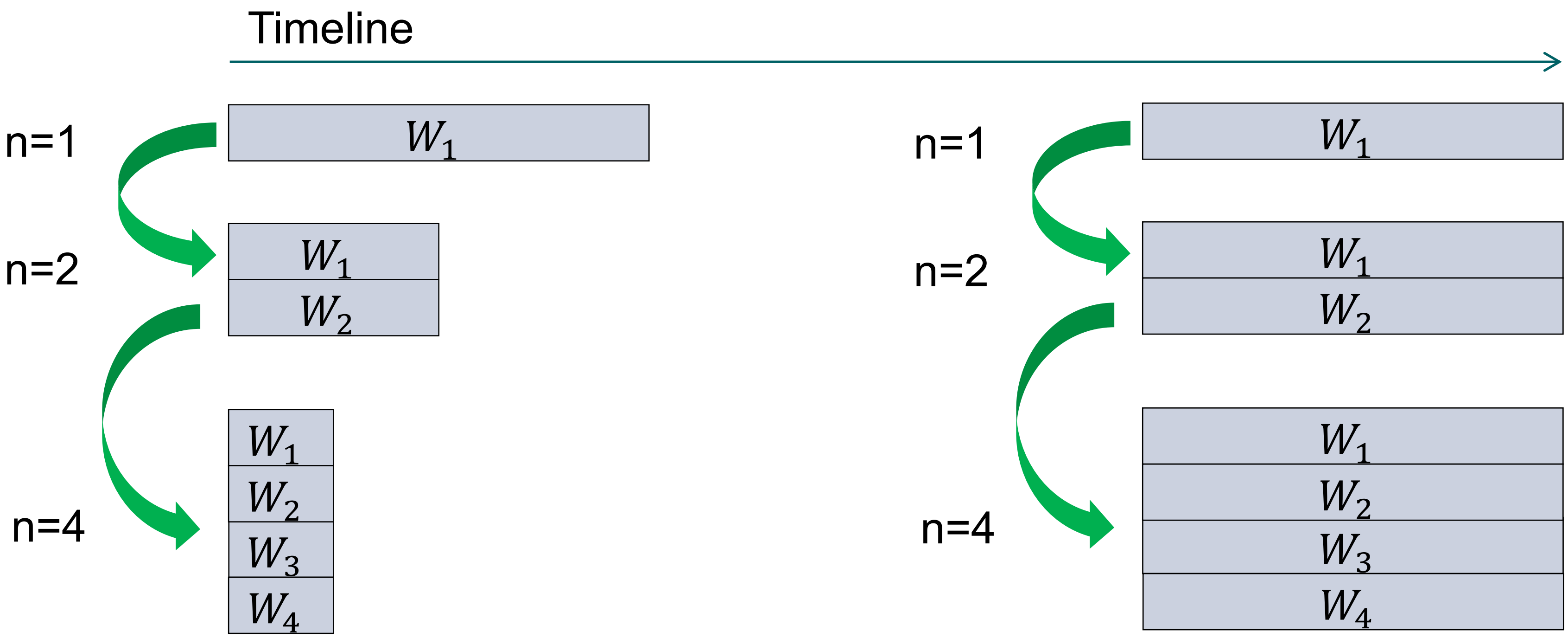
$$S_{0.95}(N) = \frac{T(1)}{T(N)} = \frac{1}{0.05 + \frac{1-0.05}{N}}$$



Harsh Realities: Gustafson's Law

Weak Scaling vs. Strong Scaling

- Gustafson's law
  - Addresses the assumption of a fixed data set, which Amdahl's law is based on
  - Problem size changes for weak scaling, fixed runtime (sketched below)



Strong scaling (p*T(1)=const)

Weak scaling (T(N)=const)

Gustafson's Law / 1

- Gustafson's law
  - Computations involving an arbitrarily large data set can be efficiently parallelized
  - Counterpoint to Amdahl's law, which puts a limit on the speedup of computations involving a fixed-size data set
- Assumption: execution time of a program can be decomposed into

$$\boldsymbol{s + p = (1 - p) + p = 1}$$

1 - p:	serial execution time
p:	parallel time for any of N processors

  - The key assumption here is that the work that needs to be done varies linearly with the number of involved processors

- Time that a single processor needs: $\boldsymbol{(1 - p) + Np = p(N - 1) + 1}$

Gustafson's Law / 2

- Speedup with Gustafson's law:

$$\boxed{S_p(N) = \frac{T(1)}{T(N)} = \frac{(1-p)+Np}{(1-p)+p} = Np + s}$$

"weak scaling"

- And efficiency: $$\varepsilon_p(N) = \frac{S_p(N)}{N} = \frac{(1-p)}{N} + p$$
- Implications
  - p (=fraction of time that the program executes in parallel) is held fixed
  - (while) the number of processors N is varied
  - With increasing number of processors, the speedup grows linearly
  - The serial part becomes more and more unimportant with more processors involved.

What you have learnt

What you have learnt

- Limits to scalability
  - Amdahl’s Law
  - Gustafson’s Law
- See Moodle for quiz!