Seminarvortrag
Amdahlsches und Gustafsonsches Gesetz

Volker Grabsch Yves Radunz

26. Mai 2008

Übersicht

Übersicht

Amdahlsches Gesetz

N	Anzahl der Prozessoren
P	Anteil des parallelisierbaren Programmcodes (an Laufzeit)
$(1 - P)$	Anteil des sequenziellen Programmcodes

$$\text{Zeitgewinn} = \frac{1}{(1 - P) + \frac{P}{N}}$$

- auch: speedup, Geschwindigkeitsgewinn

Amdahlsches Gesetz

N	Anzahl der Prozessoren
P	Anteil des parallelisierbaren Programmcodes (an Laufzeit)
$(1-P)$	Anteil des sequenziellen Programmcodes

$$\text{Zeitgewinn} = \frac{1}{(1-P) + \frac{P}{N}}$$

- auch: speedup, Geschwindigkeitsgewinn

Amdahlsches Gesetz

N	Anzahl der Prozessoren
P	Anteil des parallelisierbaren Programmcodes (an Laufzeit)
$(1-P)$	Anteil des sequenziellen Programmcodes

$$\text{Zeitgewinn} = \frac{1}{(1-P) + \frac{P}{N}}$$

- auch: speedup, Geschwindigkeitsgewinn

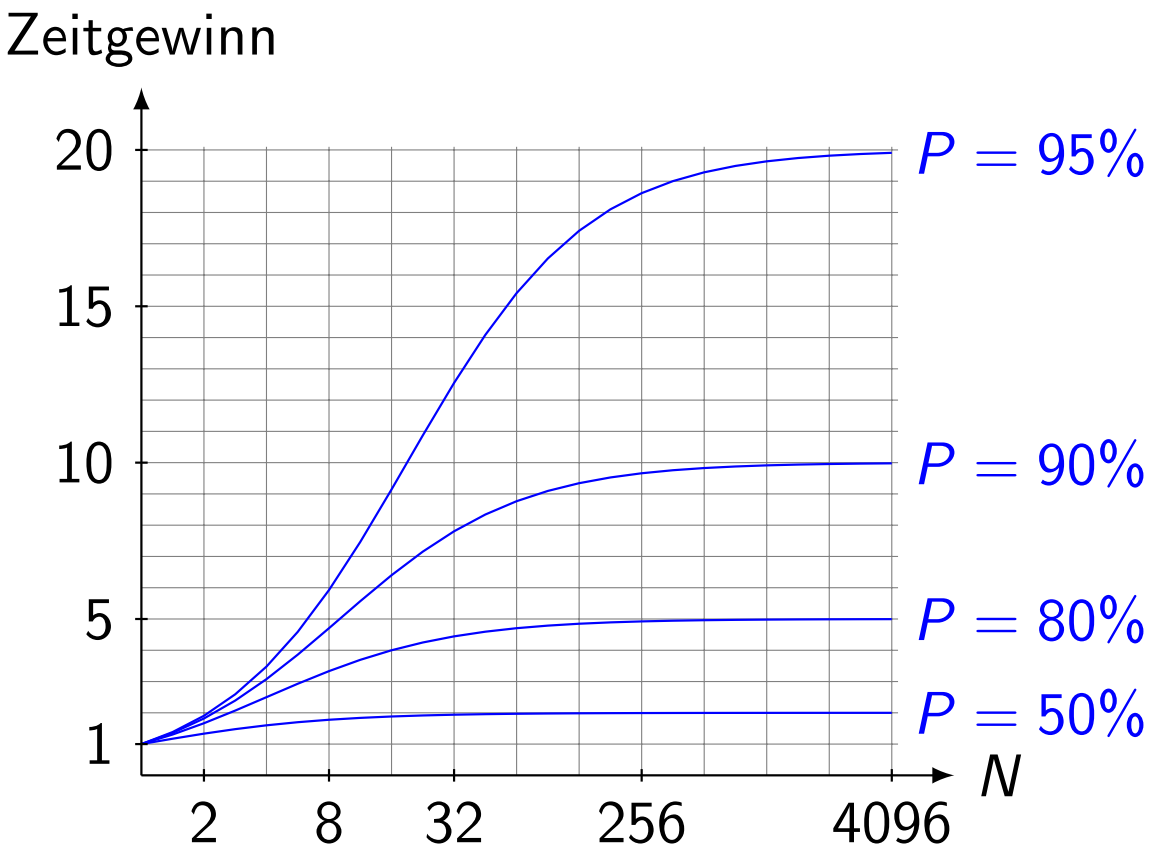
Amdahlsches Gesetz

N	Anzahl der Prozessoren
P	Anteil des parallelisierbaren Programmcodes (an Laufzeit)
$(1-P)$	Anteil des sequenziellen Programmcodes

$$\text{Zeitgewinn} = \frac{1}{(1-P) + \frac{P}{N}}$$

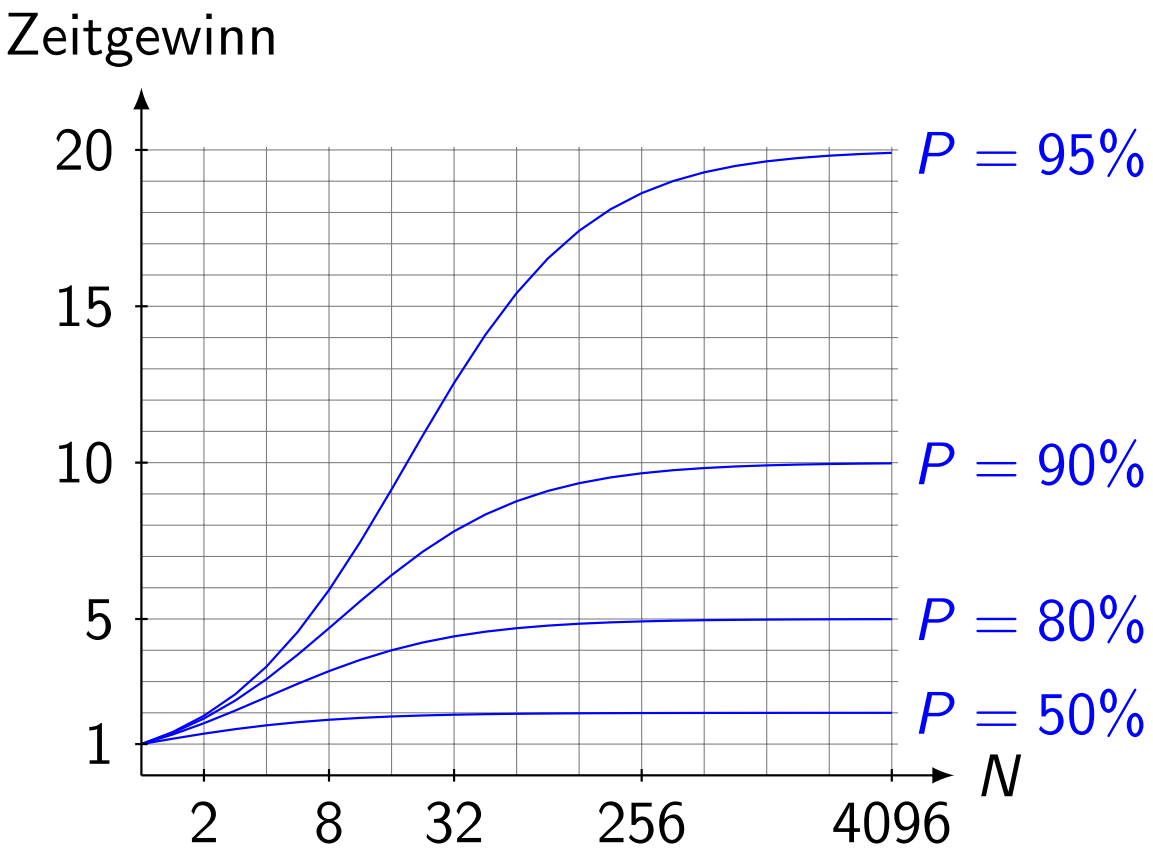
- auch: speedup, Geschwindigkeitsgewinn

Höchstgeschwindigkeit



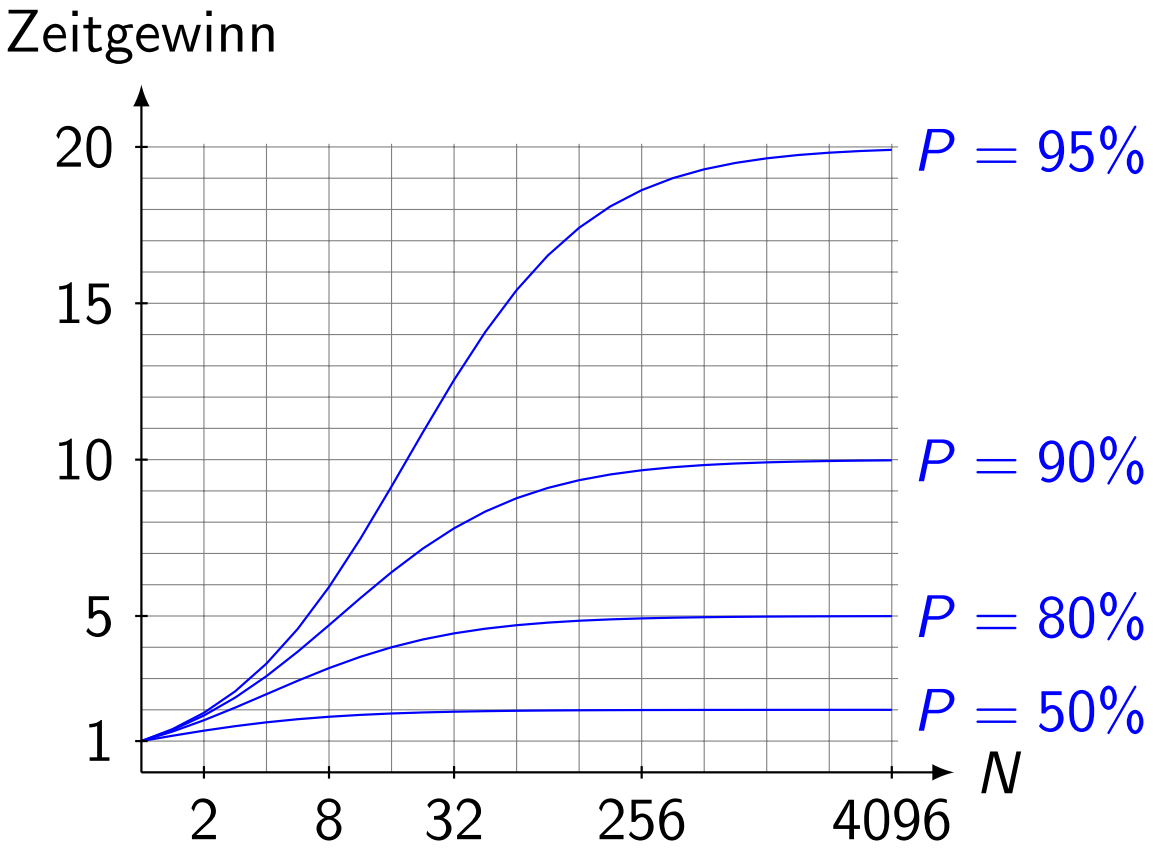
- Zeitgewinn $< \frac{1}{(1-P)}$
- folgt direkt aus Definition von „sequenzieller Programmcode“
- alternativ durch $N \to \infty$ herleitbar

Höchstgeschwindigkeit



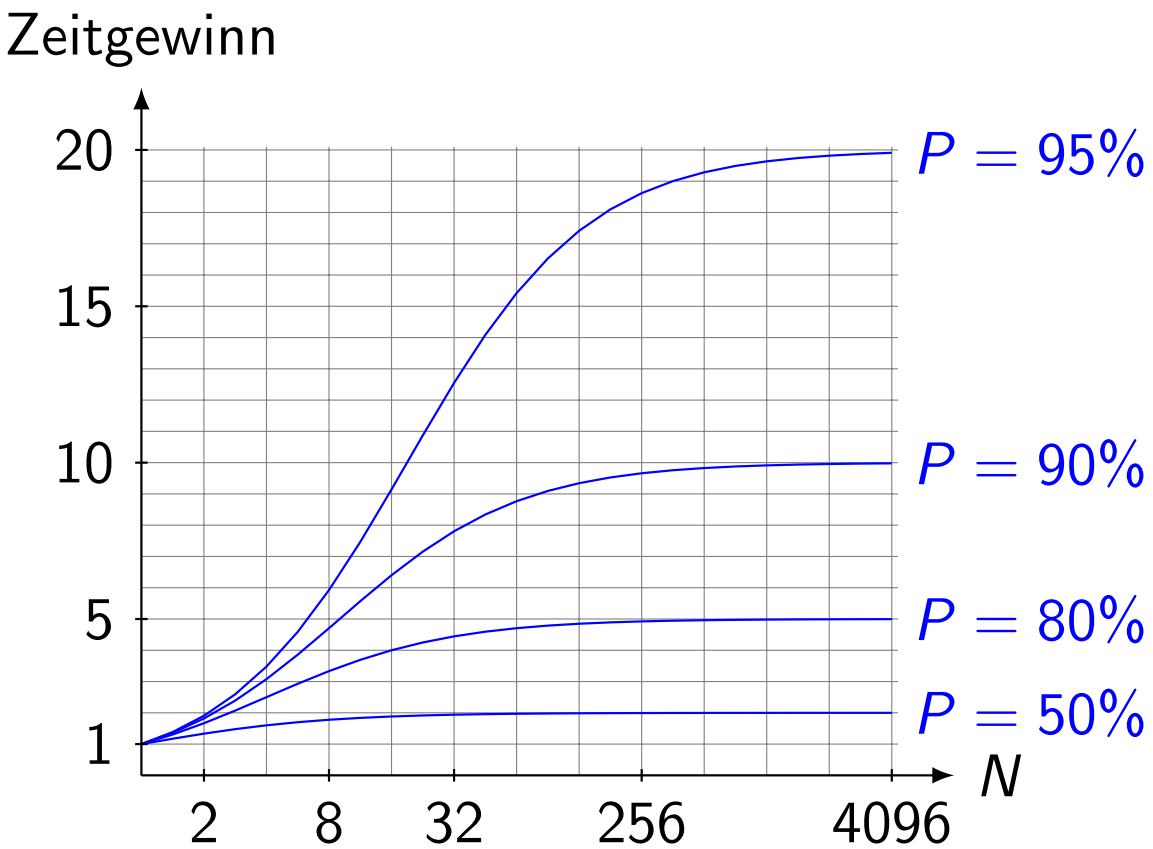
- Zeitgewinn $< \frac{1}{(1-P)}$
- folgt direkt aus Definition von „sequenzieller Programmcode“
- alternativ durch $N \to \infty$ herleitbar

Höchstgeschwindigkeit



- Zeitgewinn $< \frac{1}{(1-P)}$
- folgt direkt aus Definition von „sequenzieller Programmcode“
- alternativ durch $N \to \infty$ herleitbar

Höchstgeschwindigkeit



- Zeitgewinn $< \frac{1}{(1-P)}$
- folgt direkt aus Definition von „sequenzieller Programmcode“
- alternativ durch $N \to \infty$ herleitbar

Grenzen des Amdahlschen Gesetzes

Das Amdahlsche Gesetz ignoriert:

- wachsende Resourcen
  - Prozessor-Cache
  - RAM
- Verwaltungsaufwand für Parallelisierung
- wachsende Problemgröße ($\longrightarrow$ Gustafsonsches Gesetz)

Grenzen des Amdahlschen Gesetzes

Das Amdahlsche Gesetz ignoriert:

- wachsende Resourcen
  - Prozessor-Cache
  - RAM
- Verwaltungsaufwand für Parallelisierung
- wachsende Problemgröße ($\longrightarrow$ Gustafsonsches Gesetz)

Grenzen des Amdahlschen Gesetzes

Das Amdahlsche Gesetz ignoriert:

- wachsende Resourcen
  - Prozessor-Cache
  - RAM
- Verwaltungsaufwand für Parallelisierung
- wachsende Problemgröße ($\rightarrow$ Gustafsonsches Gesetz)

Übersicht

Übertragung auf sequenzielle Programme

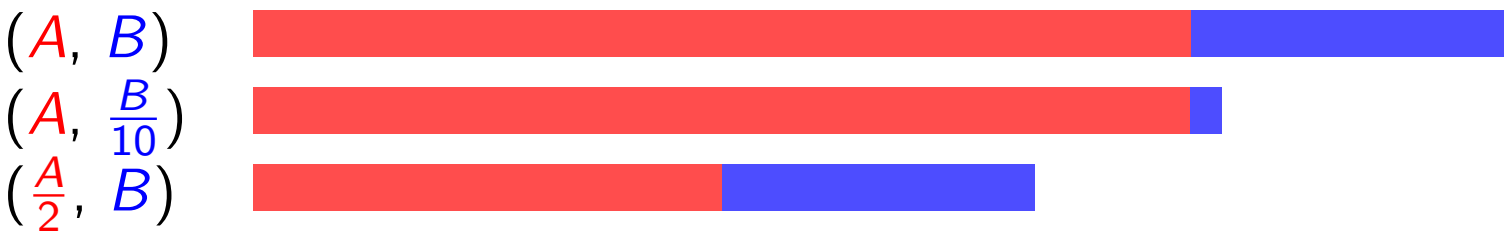
N Geschwindigkeits-Anstieg des optimierten Programmteils
P ursprünglicher Anteil des optimierten Programmteils

$$\text{Zeitgewinn} = \frac{1}{(1-P) + \frac{P}{N}}$$

Übertragung auf sequenzielle Programme

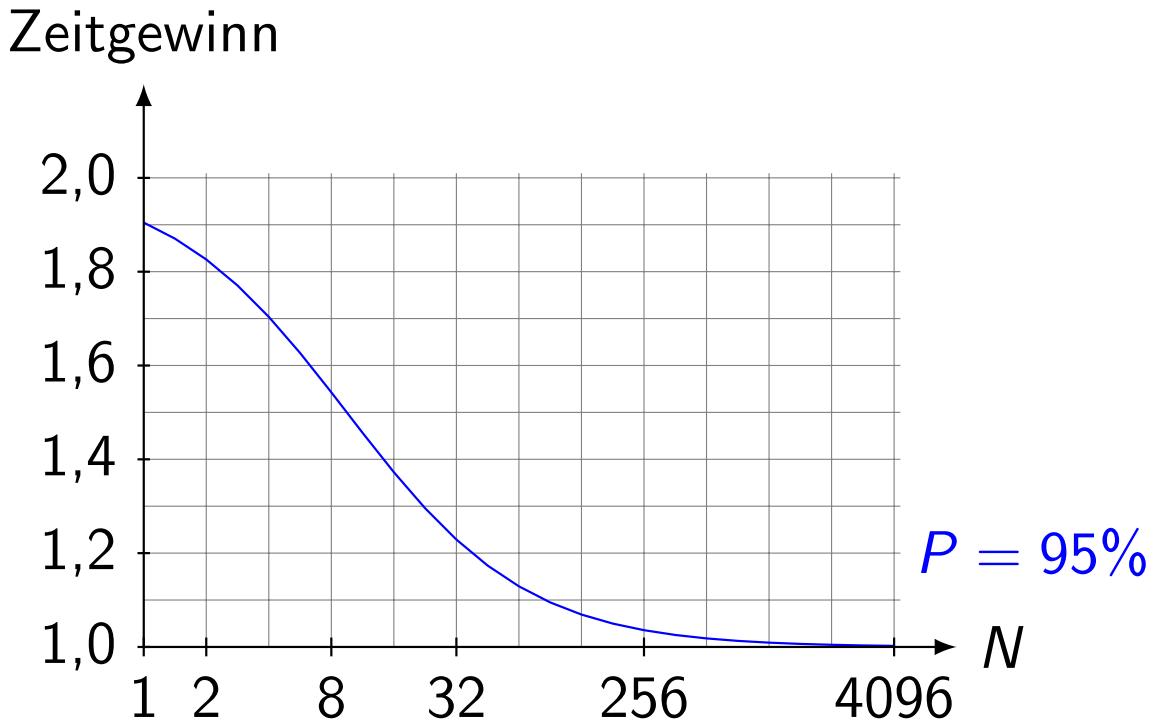
N Geschwindigkeits-Anstieg des optimierten Programmteils
P ursprünglicher Anteil des optimierten Programmteils

$$\text{Zeitgewinn} = \frac{1}{(1-P) + \frac{P}{N}}$$



Gesetz des abnehmenden Gewinns

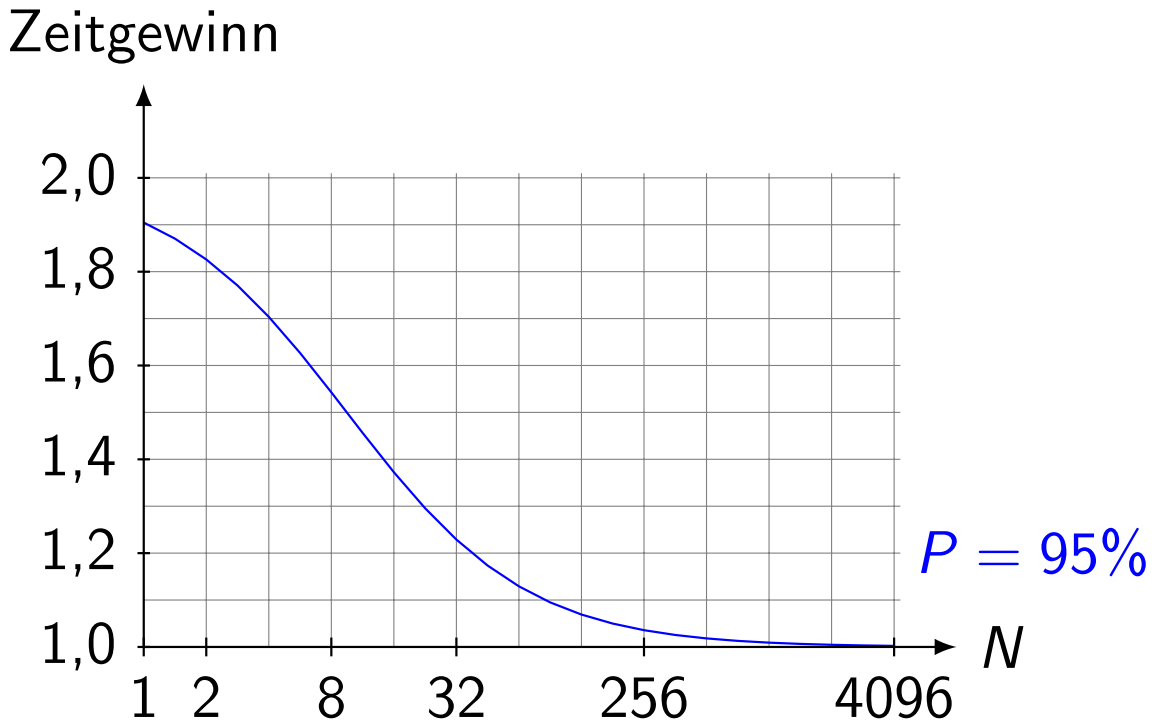
- Leistungssteigerung bei Verdoppelung von N:



- Verdoppelung bringt immer weniger

Gesetz des abnehmenden Gewinns

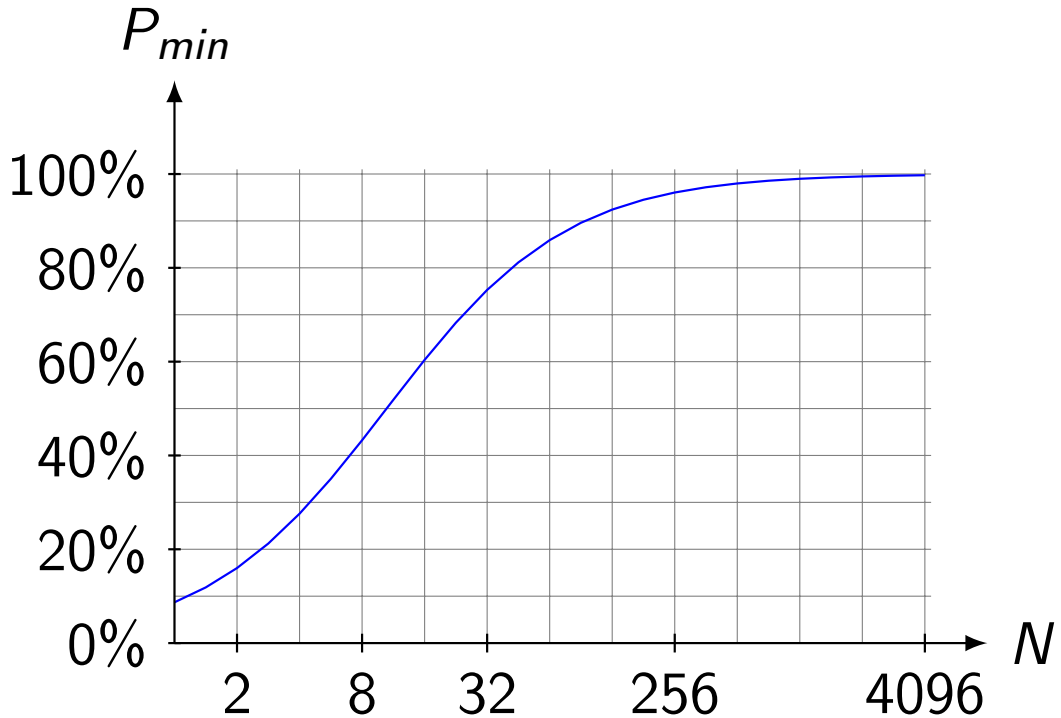
- Leistungssteigerung bei Verdoppelung von N:



- Verdoppelung bringt immer weniger

Höchster vertretbarer Aufwand

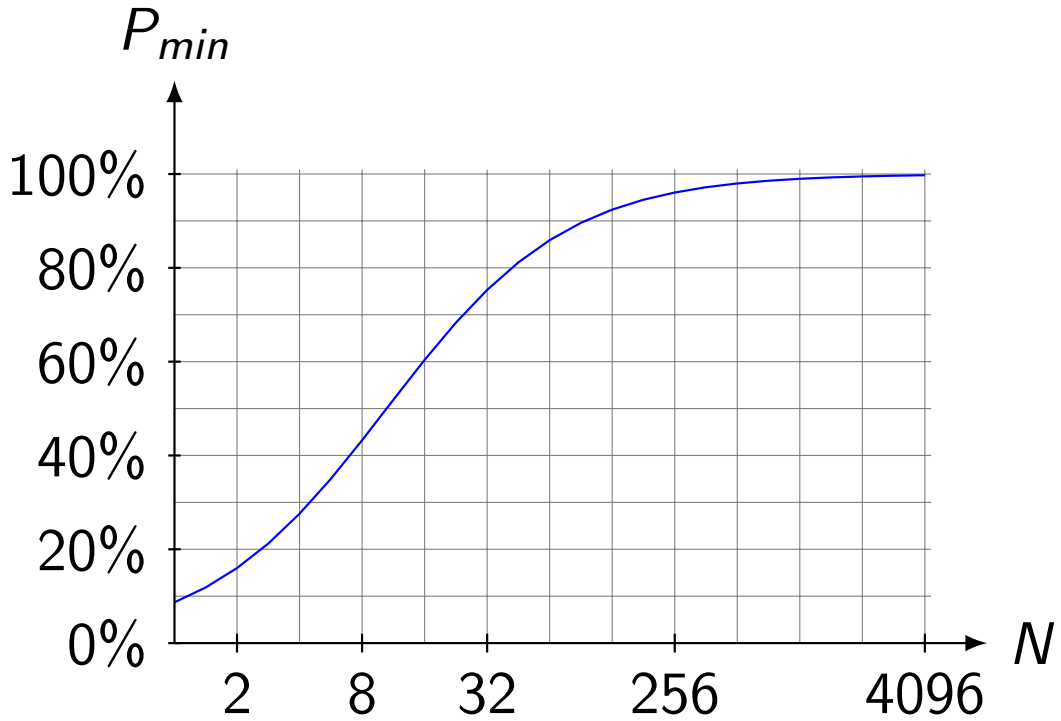
- Erforderlicher P-Anteil, damit sich Verdoppelung noch lohnt:



- „Verdoppelung lohnt sich“ = Leistungssteigerung um $\geq 5\%$
- Verdoppelung lohnt sich schnell nicht mehr

Höchster vertretbarer Aufwand

- Erforderlicher P-Anteil, damit sich Verdoppelung noch lohnt:



- „Verdoppelung lohnt sich“ = Leistungssteigerung um $\geq 5\%$
- Verdoppelung lohnt sich schnell nicht mehr

Übersicht

Einbeziehung der Problemgröße (nach Gustafson)

N Anzahl der Prozessoren
K Problemgröße
P' Anteil des parallelisierbaren Programmcodes bei $K = 1$

$$\text{Zeitgewinn} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}}$$

Herleitbar aus dem Amdahlschen Gesetz:

$$P = \frac{K \cdot P'}{(1 - P') + K \cdot P'}$$

Einbeziehung der Problemgröße (nach Gustafson)

N Anzahl der Prozessoren
K Problemgröße
P' Anteil des parallelisierbaren Programmcodes bei $K = 1$

$$\text{Zeitgewinn} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}}$$

Herleitbar aus dem Amdahlschen Gesetz:

$$P = \frac{K \cdot P'}{(1 - P') + K \cdot P'}$$

Einbeziehung der Problemgröße (nach Gustafson)

N Anzahl der Prozessoren
K Problemgröße
P' Anteil des parallelisierbaren Programmcodes bei $K = 1$

$$\text{Zeitgewinn} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}}$$

Herleitbar aus dem Amdahlschen Gesetz:

$$P = \frac{K \cdot P'}{(1 - P') + K \cdot P'}$$

Einbeziehung der Problemgröße (nach Gustafson)

N Anzahl der Prozessoren
K Problemgröße
P' Anteil des parallelisierbaren Programmcodes bei $K = 1$

$$\text{Zeitgewinn} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}}$$

Herleitbar aus dem Amdahlschen Gesetz:

$$P = \frac{K \cdot P'}{(1 - P') + K \cdot P'}$$

Gustafsonsches Gesetz – Gute Idee . . .

- „Hence, it may be most realistic to assume that run time, not problem size, is constant.“

$$1 = (1 - P') + \frac{K \cdot P'}{N}$$

$$\Rightarrow \quad K = N$$

- kein Zeitgewinn, aber Kapazitätsgewinn!

Gustafsonsches Gesetz – Gute Idee ...

- „Hence, it may be most realistic to assume that run time, not problem size, is constant.“

$$1 = (1 - P') + \frac{K \cdot P'}{N}$$

$$\Rightarrow \quad K = N$$

- kein Zeitgewinn, aber Kapazitätsgewinn!

Gustafsonsches Gesetz – Gute Idee ...

- „Hence, it may be most realistic to assume that run time, not problem size, is constant.“

$$1 = (1 - P') + \frac{K \cdot P'}{N}$$

$$\Rightarrow \quad K = N$$

- kein Zeitgewinn, aber Kapazitätsgewinn!

. . . aber schlechte Umsetzung

- Eigentliches Gesetz:

$$K = N$$

- Kompliziertere Umschreibung:

$$\begin{aligned}\text{Zeitgewinn} &= \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}} \\ &= (1 - P') + N \cdot P'\end{aligned}$$

- Kapazitätsgewinn als Zeitgewinn dargestellt
- „Sparen Sie Geld! Kaufen Sie 2 Stück zum Preis von einem!“

. . . aber schlechte Umsetzung

- Eigentliches Gesetz:

$$K = N$$

- Kompliziertere Umschreibung:

$$\text{Zeitgewinn} = \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}}$$
$$= (1 - P') + N \cdot P'$$

- Kapazitätsgewinn als Zeitgewinn dargestellt
- „Sparen Sie Geld! Kaufen Sie 2 Stück zum Preis von einem!“

. . . aber schlechte Umsetzung

- Eigentliches Gesetz:

$$K = N$$

- Kompliziertere Umschreibung:

$$\begin{aligned}\text{Zeitgewinn} &= \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}} \\ &= (1 - P') + N \cdot P'\end{aligned}$$

- Kapazitätsgewinn als Zeitgewinn dargestellt
- „Sparen Sie Geld! Kaufen Sie 2 Stück zum Preis von einem!“

. . . aber schlechte Umsetzung

- Eigentliches Gesetz:

$$K = N$$

- Kompliziertere Umschreibung:

$$\begin{aligned} \text{Zeitgewinn} &= \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}} \\ &= (1 - P') + N \cdot P' \end{aligned}$$

- Kapazitätsgewinn als Zeitgewinn dargestellt
- „Sparen Sie Geld! Kaufen Sie 2 Stück zum Preis von einem!“

. . . aber schlechte Umsetzung

- Eigentliches Gesetz:

$$K = N$$

- Kompliziertere Umschreibung:

$$\begin{aligned} \text{Zeitgewinn} &= \frac{(1 - P') + K \cdot P'}{(1 - P') + \frac{K \cdot P'}{N}} \\ &= (1 - P') + N \cdot P' \end{aligned}$$

- Kapazitätsgewinn als Zeitgewinn dargestellt
- „Sparen Sie Geld! Kaufen Sie 2 Stück zum Preis von einem!“

Übersicht

Quellen von Verwirrungen

- Massive Parallelisierung lohnt sich nicht?
  - Amdahlsches Gesetz ignoriert Problemgröße
  - $N \to \infty$: kein Zeitgewinn, aber Kapazitätsgewinn
- Gustafsonsches Gesetz widerspricht Amdahlschem Gesetz?
  - Gustafsonsches Gesetz spricht auch von Zeitgewinn
  - lediglich anderer Aspekt betrachtet

Quellen von Verwirrungen

- Massive Parallelisierung lohnt sich nicht?
  - Amdahlsches Gesetz ignoriert Problemgröße
  - $N \to \infty$: kein Zeitgewinn, aber Kapazitätsgewinn
- Gustafsonsches Gesetz widerspricht Amdahlschem Gesetz?
  - Gustafsonsches Gesetz spricht auch von Zeitgewinn
  - lediglich anderer Aspekt betrachtet

Quellen von Verwirrungen

- Massive Parallelisierung lohnt sich nicht?
  - Amdahlsches Gesetz ignoriert Problemgröße
  - $N \to \infty$: kein Zeitgewinn, aber Kapazitätsgewinn
- Gustafsonsches Gesetz widerspricht Amdahlschem Gesetz?
  - Gustafsonsches Gesetz spricht auch von Zeitgewinn
  - lediglich anderer Aspekt betrachtet

Quellen von Verwirrungen

- Massive Parallelisierung lohnt sich nicht?
  - Amdahlsches Gesetz ignoriert Problemgröße
  - $N \to \infty$: kein Zeitgewinn, aber Kapazitätsgewinn
- Gustafsonsches Gesetz widerspricht Amdahlschem Gesetz?
  - Gustafsonsches Gesetz spricht auch von Zeitgewinn
  - lediglich anderer Aspekt betrachtet

Algorithmen ändern sich durch Parallelisierung

- Amdahlsches Gesetz empirisch widerlegt?
  - Einige Algorithmen ändern sich durch Parallelisierung
  - Paralleles und sequenzielles Programm nur vergleichbar, wenn

$$\sum_{\text{Prozessoren}} \#\text{Befehle}_{par} \quad \geq \quad \#\text{Befehle}_{seq}$$

- Beispiel
  - sortierte Liste
  - lineare Suche mit Randprüfung
  - Parallelisierung $\longrightarrow$ Binärsuche

Algorithmen ändern sich durch Parallelisierung

- Amdahlsches Gesetz empirisch widerlegt?
  - Einige Algorithmen ändern sich durch Parallelisierung
  - Paralleles und sequenzielles Programm nur vergleichbar, wenn

$$\sum_{\text{Prozessoren}} \#\text{Befehle}_{par} \quad \geq \quad \#\text{Befehle}_{seq}$$

- Beispiel
  - sortierte Liste
  - lineare Suche mit Randprüfung
  - Parallelisierung $\longrightarrow$ Binärsuche

Algorithmen ändern sich durch Parallelisierung

- Amdahlsches Gesetz empirisch widerlegt?
  - Einige Algorithmen ändern sich durch Parallelisierung
  - Paralleles und sequenzielles Programm nur vergleichbar, wenn

$$\sum_{\text{Prozessoren}} \#\text{Befehle}_{par} \quad \geq \quad \#\text{Befehle}_{seq}$$

- Beispiel
  - sortierte Liste
  - lineare Suche mit Randprüfung
  - Parallelisierung $\longrightarrow$ Binärsuche

Algorithmen ändern sich durch Parallelisierung

- Amdahlsches Gesetz empirisch widerlegt?
  - Einige Algorithmen ändern sich durch Parallelisierung
  - Paralleles und sequenzielles Programm nur vergleichbar, wenn

$$\sum_{\text{Prozessoren}} \#\text{Befehle}_{par} \quad \geq \quad \#\text{Befehle}_{seq}$$

- Beispiel
  - sortierte Liste
  - lineare Suche mit Randprüfung
  - Parallelisierung $\longrightarrow$ Binärsuche

Algorithmen ändern sich durch Parallelisierung

- Amdahlsches Gesetz empirisch widerlegt?
  - Einige Algorithmen ändern sich durch Parallelisierung
  - Paralleles und sequenzielles Programm nur vergleichbar, wenn

$$\sum_{\text{Prozessoren}} \#\text{Befehle}_{par} \quad \geq \quad \#\text{Befehle}_{seq}$$

- Beispiel
  - sortierte Liste
  - lineare Suche mit Randprüfung
  - Parallelisierung $\rightarrow$ Binärsuche

Wirkliche Widerlegung

- Amdahlsches Gesetz dennoch empirisch widerlegt?
  - Ja, denn es ignoriert die wachsenden Resourcen
  - mehr Prozessoren $\longrightarrow$ mehr Cache
  - Geschwindigkeitszuwachs, wenn Teilprobleme in Cache passen

Wirkliche Widerlegung

- Amdahlsches Gesetz dennoch empirisch widerlegt?
  - Ja, denn es ignoriert die wachsenden Resourcen
  - mehr Prozessoren $\longrightarrow$ mehr Cache
  - Geschwindigkeitszuwachs, wenn Teilprobleme in Cache passen

Wirkliche Widerlegung

- Amdahlsches Gesetz dennoch empirisch widerlegt?
  - Ja, denn es ignoriert die wachsenden Resourcen
  - mehr Prozessoren $\longrightarrow$ mehr Cache
  - Geschwindigkeitszuwachs, wenn Teilprobleme in Cache passen

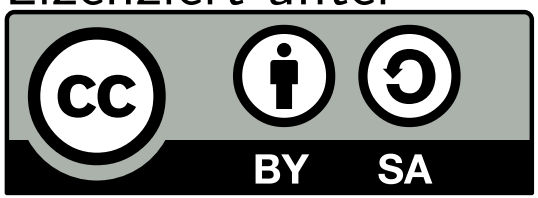
Ende

Vielen Dank für die Aufmerksamkeit!

URL und Lizenz

- Veröffentlicht unter
  `http://www.profv.de/uni/`

- Lizenziert unter



Creative Commons BY-SA 3.0