

Übung 2 - Kennzahlen

1. - Rechenzeit abschätzen (wobei Kennzahlen aber nur bedingt in Praxis übertragbar)
 - Theoretisch obere Schranke (Optimum)
 - Vergleichbarkeit unter* Rechensystemen und Algorithmen
 - Änderungen statistischer Wert über die Lebenszeit eines Rechensystems können z.B. als Health-Check eingesetzt werden

* / von

2.

$p \hat{=}$ Anzahl Prozesse

Idealer Speedup: $s_p = p$

Speedup: $s_p(n) = \frac{T(n)}{T_p(n)}$

 \nwarrow Laufzeit des sequentiellen Programms

 \nearrow Laufzeit des parallelen Programms

Speedup für Problemgröße n

Effizienz: $E_p(n) = \frac{s_p(n)}{p}$

 \nwarrow Speedup

 \nearrow

Effizienz

 Anzahl an Prozessen (Recheneinheiten)

Testreihe 1

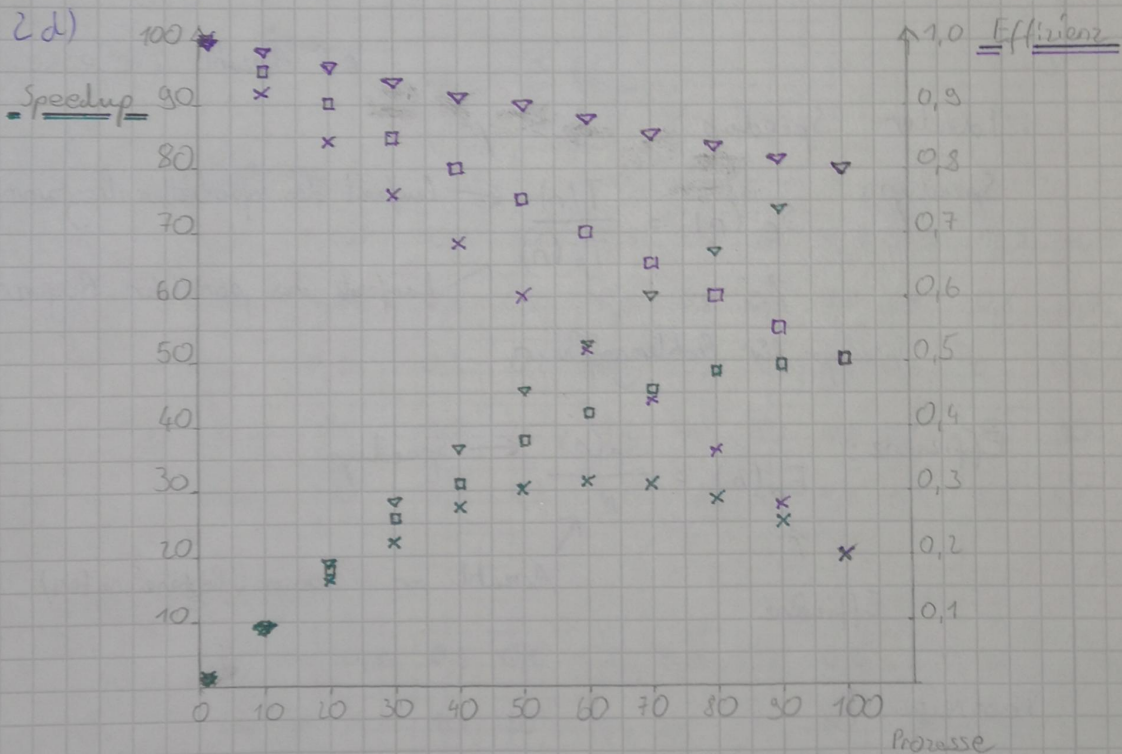
Id. Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	9,5	18,0	25,5	31,9	37,6	42,0	45,7	48,1	49,5	50,0
Effizienz	1,00	0,95	0,90	0,85	0,80	0,75	0,70	0,65	0,60	0,55	0,50

Testreihe 2

Id. Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	9,2	16,8	22,8	27,2	30,0	31,2	30,8	28,8	15,2	20,0
Effizienz	1,00	0,92	0,84	0,76	0,68	0,60	0,52	0,44	0,36	0,28	0,20

Testreihe 3

Id. Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	9,8	13,2	28,2	36,8	45,0	52,9	60,2	67,1	73,5	80,0
Effizienz	1,00	0,98	0,96	0,94	0,92	0,90	0,88	0,86	0,84	0,82	0,80



- Messreihe 1
- × Messreihe 2
- ▽ Messreihe 3

- e) Messreihe 1: asymptotische Annäherung an Speedup 50
 ↳ $P > 100$ nicht sinnvoll für Skalierung
- Messreihe 2: Speedup Maximum bei 60 Prozessern
 ↳ $P > 60$ bringt keinen Performancegewinn

Messreihe 3: Am ehesten dem idealen Speedup, am effizientesten

↳ Weitere Prozesse können weiteren (nahezu linearen) Speedup bringen

3. a)

~~$$\text{seq. Laufzeit } T_{\text{seq}} = T_{\text{ges}} - T_{\text{para}}$$~~

$$T_{\text{ges}} = T_{\text{para}} + T_{\text{seq}}$$

$$t_{\text{para}} = \frac{T_{\text{para}}}{T_{\text{ges}}}$$

$$t_{\text{seq}} = \frac{T_{\text{seq}}}{T_{\text{ges}}}$$

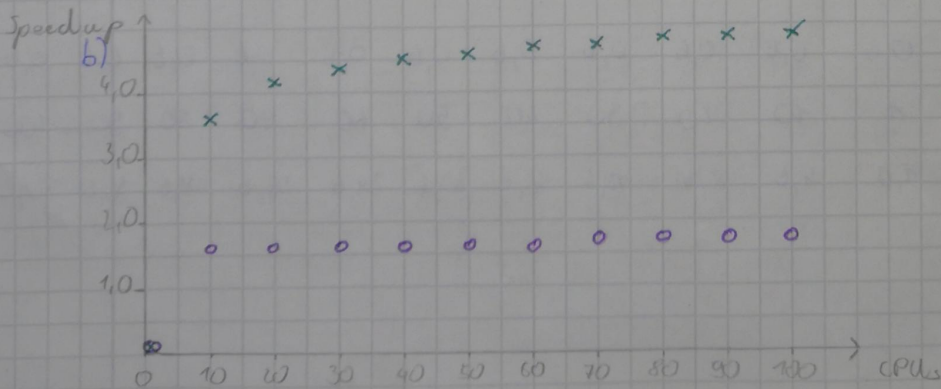
$$\text{Speedup} = \frac{1}{t_{\text{seq}} + \frac{1}{p} t_{\text{para}}}$$

Testreihe 1

T_{ges}	1000	100	50	33,4	25	20	16,6	14,3	12,5	11,1	10
t_{para}	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
t_{seq}	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
Id-Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	3,6	4,2	4,4	4,5	4,6	4,7	4,7	4,8	4,8	4,8

Testreihe 2

t_{seq}	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6
Id-Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	1,6	1,6	1,6	1,6	1,6	1,6	1,7	1,7	1,7	1,7



Messreihe 1

Messreihe 2

c) Messreihe 1: Höherer paralleler Laufzeitanteil resultiert in einem höheren Speedup

↳ stagniert ab 30 CPU's → paralleler Anteil bleibt gleich und zunehmende Anzahl von Prozessen erreicht kaum Verbesserung

Messreihe 2: Niedriger paralleler Laufzeitanteil → geringer Speedup

↳ stagniert ab 10 CPU's → da der parallele Anteil so gering ist, wird schon ab 10 Prozessen kein weiterer Speedup erreicht

↳ Mehr Prozesse bringen ~~zusätzlich~~ Overhead, welcher zusätzlichen Speedup verhindert

4.

a)

$$t_{seq} = 1 - t_{para}$$

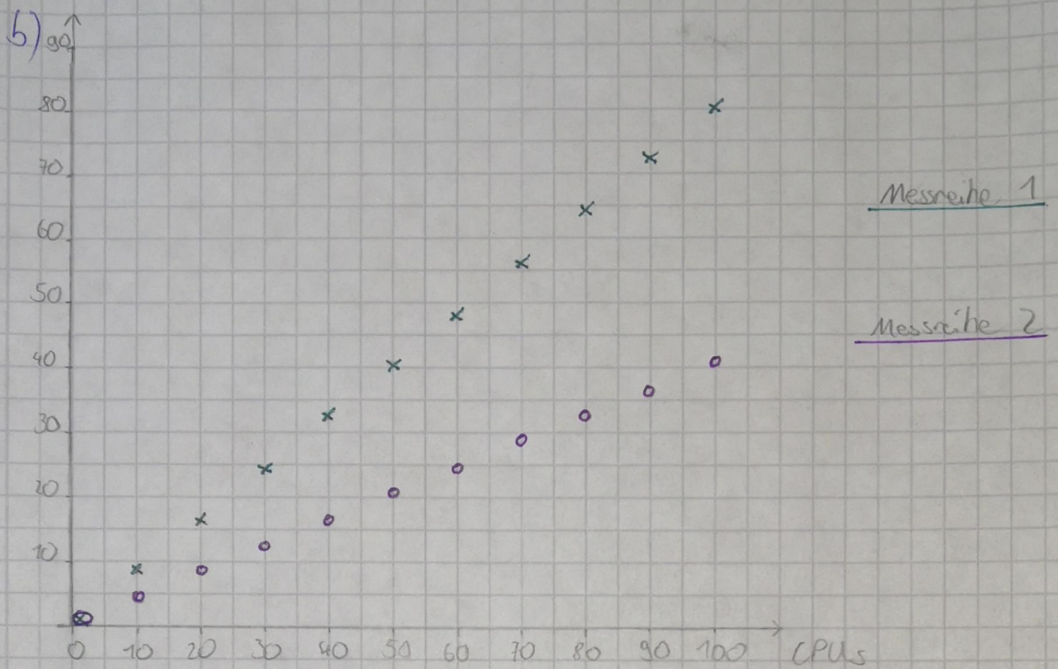
$$Speedup = t_{seq}(n) + p \cdot t_{para}(n)$$

Testreihe 1

seq. Laufzeit	200	20	10	6,6	5	4	3,4	2,9	2,5	2,2	2
t_{para}	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8	0,8
t_{seq}	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2	0,2
Id. Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	8,2	16,2	24,3	32,2	40,2	48,0	56,0	64,2	71,4	80,2

Testreihe 2

t_{seq}	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6	0,6
Id. Speedup	1	10	20	30	40	50	60	70	80	90	100
Speedup	1,0	4,6	8,6	12,6	16,6	20,6	24,6	28,6	32,6	36,6	40,6



- c) Messreihe 1: Höherer ~~Anteil~~ parallelisier Laufzeitanteil, resultiert in einem höheren Speedup ~~Anteil~~
- beide Messreihen zeigen lineares Wachstum
- ↳ hinzufügen von CPUs lohnt sich in beiden Fällen

Amdahl → pessimistisch \Rightarrow gleichbleibende Problemgröße

Lawson → optimistisch \Rightarrow wachsende Problemgröße