



Karlsruher Institut für Technologie
Institut für Technische Informatik
Prof. Dr. Wolfgang Karl

Klausur Rechnerstrukturen
Sommersemester 2011
Musterlösung

Aushang der Ergebnisse: ab Mitte September 2011

Musterlösung 1: Parallelverarbeitung und Verbindungsstrukturen

9P

Quantitative Maßzahlen

4P

- a) Amdahls Gesetz (Je 0,5P für Formel und Erklärung):

1P

$$T(n) = \underbrace{\frac{T(1)}{n} * (1 - a)}_1 + \underbrace{T(1) * a}_2$$

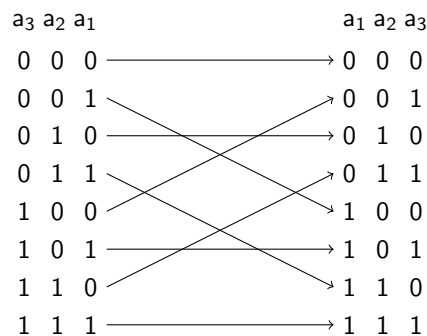
Die Formel zerfällt in die Ausführungszeit des parallel ausführbaren Programmteils 1 und den rein sequentiell ausführbaren Programmteil 2. Es gilt: a mit $(0 \leq a \leq 1)$ ist der Anteil des Programms, der nur sequentiell ausgeführt werden kann.

- b) • (0,5P) $U(n) = \frac{I(n)}{n} = R(n) * E(n) = \frac{P(n)}{n * T(n)}$ 1P
 • (0,5P) $R(n) = \frac{P(n)}{P(1)}$
- c) Die absolute Beschleunigung und absolute Effizienz erhält man, indem der beste sequentielle Algorithmus mit dem besten parallelen Algorithmus verglichen wird. 1P
- d) Der Zusatzaufwand für die Parallelisierung durch Kommunikation und Synchronisation „verfälscht“ das Ergebnis bei der sequentiellen Ausführung. 1P

Verbindungsstrukturen

5P

- e) • Fähigkeit, die wesentlichen Eigenschaften des Verbindungsnetzes auch bei beliebiger Erhöhung der Knotenzahl beizubehalten. 1P
 • Vergrößerung möglich ohne die wesentlichen Eigenschaften des Netzwerks zu verlieren.
- f) Kreuzpermutation: 1P



- g) • Durchmesser: 6 *1P*
 • Minimale Bisektionsbreite: 4
- h) Bei einem fehlertoleranten Netz muss zwischen jedem Paar von Knoten (0,5P) mindestens ein weiterer, redundanter Weg vorhanden sein (0,5P). *1P*
- i) • Anzahl der Knoten $N = K^n$ *1P*
 • Verbindungsgrad: $2n$

Musterlösung 2: Parallele Architekturen

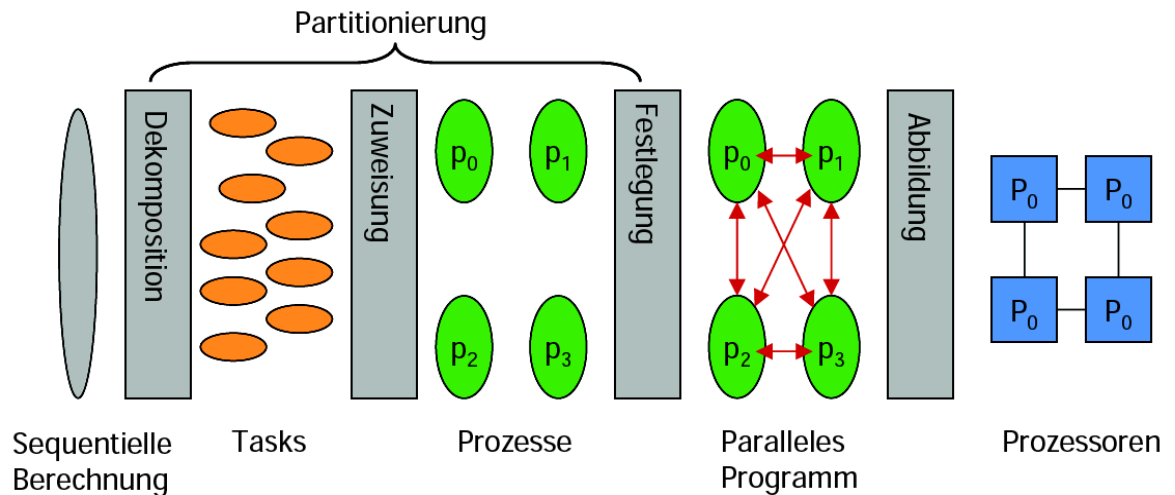
8P

Parallele Architekturen und Parallelisierung

6P

a)

2P

b) *Das Problem selbst* muss skalieren (Problemgröße, Parallelisierbarkeit).

1P

c)

3P

	Gemeinsamer Speicher	Physikalisch verteilter Speicher
Globaler Adressraum	<p>SMP Symmetrischer Multiprozessor</p> <p>UMA: Uniform Memory Access</p>	<p>DSM Distributed-shared-memory Multiprozessor</p> <p>NUMA: Non-Uniform Memory Access</p>
Getrennte Adressräume	<p>leer</p>	<p>Nachrichtengekoppelter (Shared-nothing) Multiprozessor / Cluster</p> <p>NORMA: No Remote Memory Access</p>

Vektorverarbeitung**2P**

- d) • (0,5 P) Vektorprozessoren (Vektorrechner): SIMD
 • (0,5 P) Nachrichtengekoppelte Parallelrechner: MIMD

1P

e) Verkettung (0,5 P)

1P

- Verkettung mehrerer (spezialisierter) Pipelines
- Die Ergebnisse einer Pipeline werden sofort der nächsten Pipeline zur Verfügung gestellt.

(0,5 P für Erklärung)

Musterlösung 3: Fragen des Rechnerentwurfs

8P

Leistungsaufnahme

4P

a) $P_{switching} = C_{eff} * U^2 * f$

0,5 P

b) C_{eff} bleibt unverändert.

2 P

$$Reduction = \frac{Power_{new}}{Power_{old}} = \frac{U_{new}^2 * f_{new}}{U_{old}^2 * f_{old}} = \frac{(0.8 * U_{old})^2 * (0.5 * f_{old})}{U_{old}^2 * f_{old}} = 0.8^2 * 0.5 = 0.32$$

D.h. die elektrische Leistungsaufnahme kann auf 32% gesenkt werden.

c) Reduktion von C_{eff} , dadurch Reduktion von $P_{switching}$ ($\frac{1}{2}$ P)

1,5P

Reduktion von $P_{shortcircuit}$ ($\frac{1}{2}$ P)

Keine Beeinflussung von P_{static} und $P_{leakage}$ ($\frac{1}{2}$ P)

Low-Power-Entwurf

4P

d) Formel: $\mathbb{P}_{Schalt} = 2 * \mathbb{P}(1) * (1 - \mathbb{P}(1))$

0,5 P

e) Signalwahrscheinlichkeit linkes ODER-Gatter:

3,5 P

$$\mathbb{P}_{links}(1) = 1 - \mathbb{P}(0) = 1 - \mathbb{P}_{Eingang1}(0) * \mathbb{P}_{Eingang2}(0) = 1 - \frac{1}{2} * \frac{3}{4} = \frac{5}{8}$$

Schaltwahrscheinlichkeit linkes ODER-Gatter:

$$\mathbb{P}_{Schalt links} = 2 * \frac{5}{8} * (1 - \frac{5}{8}) = \frac{15}{32}$$

Signalwahrscheinlichkeit rechtes ODER-Gatter:

$$\mathbb{P}_{rechts}(1) = 1 - \frac{3}{8} * \frac{1}{2} = \frac{13}{16}$$

Schaltwahrscheinlichkeit rechtes ODER-Gatter:

$$\mathbb{P}_{Schalt rechts} = 2 * \frac{13}{16} * (1 - \frac{13}{16}) = \frac{13}{8} * \frac{3}{16} = \frac{39}{128}$$

Gesamt Schaltwahrscheinlichkeit:

$$\mathbb{P}_{Schalt} = \frac{15}{32} + \frac{39}{128} = \frac{60}{128} + \frac{39}{128} = \frac{99}{128}$$

Musterlösung 4: Rechnerarchitektur

15P

- a) Dynamische Parallelisierung mittels Hardware (Superskalartechnik) vs. statische Parallelisierung durch den Compiler (VLIW-Technik). *1P*
- b) *5P*

Feld	R1	R2	R3	R4
Value	-	-	-	-
Valid	0	0	0	0
RS	Mul 1	Int 2	Mul 2	Int 1

Unit	Empty	InFU	Op	Dest	Src1	Vld1	RS1	Src2	Vld2	RS2
Int 1	0	0	add	R4	(R2)	1	-	-	0	Mul 2
Int 2	0	0	sub	R2	-	0	Mul 2	-	0	Int 1
Mul 1	0	1	mul	R1	(R2)	1	-	(R3)	1	-
Mul 2	0	0	mul	R3	-	0	Mul 1	(R2)	1	-

- c) Statische Sprungvorhersage ist für einen Befehl immer gleich. Bei dynamischen Sprungvorhersagetechniken ist die Vorhersagerichtung abhängig von der Vorgeschichte der Verzweigung. *1P*
- d) Tabelle: *4P*

Richtung	Aktuelle Vorhersage		
	Historie	Prädiktor	Vorh.
NT	T	(T , T)	T
T	NT	(T , NT)	T
NT	T	(T , NT)	NT
NT	NT	(T , NT)	T
T	NT	(NT , NT)	NT
NT	T	(T , NT)	NT
NT	NT	(T , NT)	T
T	NT	(NT , NT)	NT

- e) Tabelle: *4P*

Prozessor	Aktion	Prozessor 1		Prozessor 2		Prozessor 3	
		Line 1	Line 2	Line 1	Line 2	Line 1	Line 2
	init	-	-	-	-	-	-
2	rd 4			4/E			
1	rd 1	1/E					
1	rd 3		3/E				
3	rd 4			4/S		4/S	
1	wr 1	1/M					
2	rd 3		3/S		3/S		
3	wr 1	1/I					1/M
3	rd 3		3/S		3/S	3/S	
3	rd 4			4/S			4/S
1	wr 3		3/M		3/I	3/I	
2	wr 4			4/M			4/I
2	wr 2				2/M		

Musterlösung 5: Fehlertoleranz

10P

Ausfallverhalten

2P

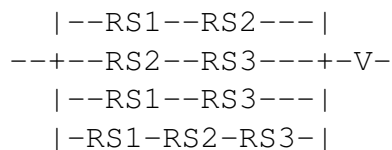
- a) Spätphase 1P
 b) Alterungseffekte führen zu altersbedingten Ausfällen 1P

Redundanzsysteme

5P

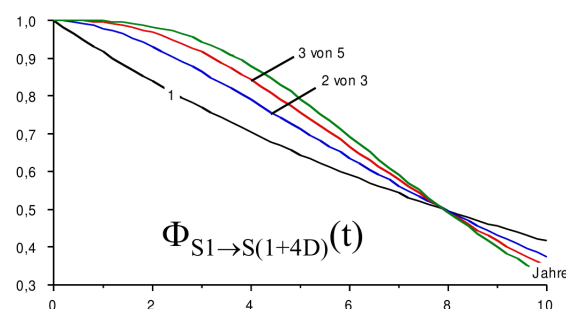
- c) Zuverlässigkeitsblockdiagramm:

1P



- d) Systemfunktion: $SF = (RS_1 \wedge RS_2 \vee RS_2 \wedge RS_3 \vee RS_1 \wedge RS_3 \vee RS_1 \wedge RS_2 \wedge RS_3) \wedge V$ 1P
 e) Funktionswahrscheinlichkeit für ein 2-aus-3 System mit Mehrheitsentscheider: 1,5P
 $\varphi_3^2 = \varphi(V) * \sum_{k=2}^3 \binom{3}{k} * \varphi(RS)^k * (1 - \varphi(RS))^{(3-k)}$
 f) Das 3-aus-5-System zeigt in der Anfangszeit eine höhere Funktionswahrscheinlichkeit als das 2-aus-3-System, nach einiger Zeit (Schnittpunkt der beiden Graphen im Bild) kehrt sich dieser Effekt jedoch um. 1,5P

■ Statisch redundantes System



Berechnungen zur Verfügbarkeit

3P

- g) $V = \frac{MTTF}{MTBF} = \frac{MTTF}{MTTF+MTTR}$ 1P
 h) Eingesetzt: $0.98 = \frac{MTTF}{MTTF+10h} \iff 0.98 = 1 - \frac{10h}{MTTF+10h} \iff 0.002 = \frac{10h}{MTTF+10h} \iff 500h = MTTF + 10h \iff MTTF = 490h$ 1P
 i) Es gilt $z(t) \sim \frac{d}{dt} F_L(t)$. Die Ausfallrate ist proportional zur ersten Ableitung der Fehlerwahrscheinlichkeit. Richtig ist auch: $z(t) \sim \frac{1}{1-F_L(t)}$. Die Ausfallrate ist antiproportional zur Überlebenswahrscheinlichkeit (welche als Gegenwahrscheinlichkeit zur Fehlerwahrscheinlichkeit definiert ist). 1P

Musterlösung 6: Quantifizierung

10P

Analytische/empirische Leistungsbewertung

3P

- a) Gesetz von Little & Rechnung: $Q = W * D \Leftrightarrow D = \frac{Q}{W}$, für $W \neq 0$, Q: Anzahl von Aufträgen in der Warteschlange, W: Wartezeit, D: Durchsatz 1P

Mit $Q_A = 45$ und $W_A = 90 \text{ Minuten} = 3/2 \text{ Stunden}$: $\Rightarrow D_A = \frac{45 \frac{\text{Jobs}}{\text{Stunde}}}{3/2 \text{ Stunden}} = 30 \frac{\text{Jobs}}{\text{Stunde}}$

verglichen mit $D_B = 20 \frac{\text{Jobs}}{\text{Stunde}}$ entscheiden Sie sich für System A, da $D_A > D_B$. Der Durchsatz ist für den Rechenzentrumsleiter von zentraler Bedeutung (Systemsicht!).

- b) Software-Monitore: Einfach zu realisieren. Bedingt durch die Integration in die Anwendung / Betriebssystem beeinflussen Software-Monitore die normalen Betriebsverhältnisse 2P

Hardware-Monitore: Aufwändiger in der Realisierung, aufwändige Hardwareentwicklung notwendig. Keine Beeinflussung des Systems.

Modelltheoretische Verfahren

3P

- c) Modelltheoretische Verfahren sind unabhängig von der Existenz des Rechners 1P
- d) Modellbildung ($\frac{1}{2}$ P): Annahmen über die Struktur und Betrieb des Rechners ($\frac{1}{2}$ P), Darstellung der relevanten Merkmale des Systems ($\frac{1}{2}$ P), Abstrahierung komplexer Systeme ($\frac{1}{2}$ P) 2P

Leistungsbewertung mit Benchmarks

4P

- e) SPEC ermöglicht einen fairen Vergleich ($\frac{1}{2}$ P) von unterschiedlichen Systemen inklusive Compiler und OS ($\frac{1}{2}$ P). 1P
- f) $Spec_{ratio x} = \frac{Referenzzeit_x}{Laufzeit_x \text{ auf Testsystem}}$ für einen Benchmark x . (jeweils $\frac{1}{2}$ P). 1P
- g) $Spec_{int base} = \sqrt[n]{\prod_{x=1}^n Spec_{ratio x}}$ für alle n Integer-Benchmarks übersetzt und ausgeführt mit konservativen Optimierungseinstellungen (base). 1P
- h) Die faire Vergleichbarkeit ($\frac{1}{2}$ P). Man könnte die "neuen" Messwerte nicht mehr direkt mit den "alten" Vergleichen, sondern müsste die bisherigen erneut bestimmen ($\frac{1}{2}$ P). 1P