

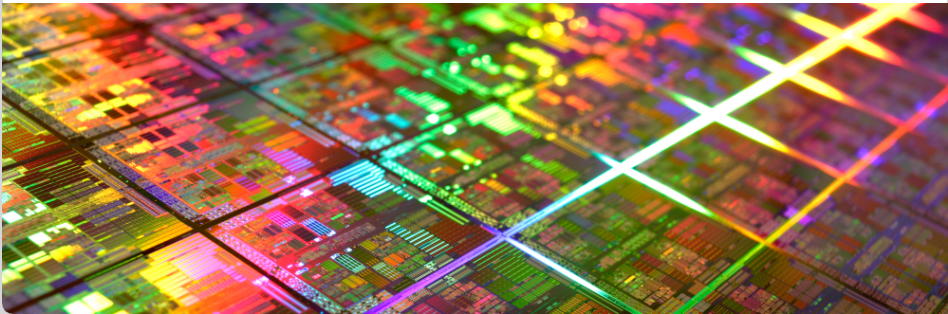
Zentralübung Rechnerstrukturen im SS2011

Leistungsbewertung und Fehlertoleranz

Martin Schindewolf, Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

5. Mai 2011



Leistungsfähigkeit

- **Quantifizierung der Leistungsfähigkeit?**
 - Was ist Leistungsfähigkeit?
 - Was bedeutet “schneller”?
 - Wie sind Systeme zu vergleichen?

- **Entscheidung bei Entwurf, Auswahl und Veränderung von Rechenanlagen**
 - Objektive Quantifizierung
 - Erfassen des gesamten Systems, nicht nur von Teilaspekten

Fehlertoleranz

Zuverlässigkeit/Verfügbarkeit eines Systems bestimmen

■ **Anwendersicht:** Reduzierung von

- Antwortzeit (response time)
 - Latenzzeit
 - CPU Time (User, System)
- Ausführungszeit (execution time)

■ **Betriebssicht:** Erhöhung von

- Anzahl durchgeführter Jobs
- Durchsatz
- Energieeffizienz (Betriebskosten)

→ **Auswertung benötigt Bewertungsverfahren**

■ Auswertung von HW-Eigenschaften

- Einfacher Vergleich
- Bewertung sehr spezieller Aspekte (Takt)
- Angabe einer hypothetischen Maximalleistung (MIPS)
- Meist nicht/selten aussagekräftig
- Alltagsbeispiel: GHz-Manie → QuantiSpeed

■ Mixe

- Theoretische Berechnung einer mittleren Operationszeit T aus den Operationszeiten und Auftrethäufigkeiten von n Befehlen
- $T = \sum_{i=1}^n p_i * t_i$ mit $\sum_{i=1}^n p_i = 1$ wobei $p_i \leq 1$

■ Kernprogramme

- Typische Anwenderprogramme, für den zu bewertenden Rechner geschrieben
- Berechnung der Ausführungszeit anhand der Ausführungszeiten der Befehle

■ Benchmarks

- Programmsammlungen im Quellcode
- Übersetzung & Messung der Ausführungszeiten
- Problem: Einfluss von OS und Compiler
- Synthetische Benchmarks (Whetstone, Dhrystone), Quasi-Simulation von Anwenderprogrammen
- Kernels (LINPACK)
- Standardisierte Benchmarks (SPEC, TPC, EEMBC...)

■ Ziele

- Stellt fairen Vergleich sicher
- Ermöglicht Angabe einer Maximalleistung

■ SPEC-Benchmark

- Integer vs. Fließkomma: SPECint / SPECfp
- Geschwindigkeit vs. Durchsatz (rate)
- Optimierung: Konservativ (base) vs. aggressiv
- $SPECratio_x = \frac{t_{ref_x}}{t_{exec}}$

Übungsblatt Aufgabe V a):

Die Ergebnistabelle der SPEC-Seite für die Xeon X5677-Architektur gliedert sich in die Spalten **Base** und **Peak**. Erklären Sie den Laufzeitunterschied für 400.perlbench. Vergleichen Sie dies mit den Ergebnissen für 483.xalanbmk.

- **Peak** erlaubt aggressive Optimierungen im Gegensatz zu Base. → **Laufzeitunterschied** durch Optimierung
 - Kaum Laufzeitunterschiede für 483.xalanbmk:
 - Entweder waren die durchgeführten Optimierungen nicht wirkungsvoll,
 - oder weitere Optimierungen wurden nicht angestrebt.
- Sektion **Peak Optimization Flags** zeigt, dass Compileroptimierungen **nicht** verwendet wurden.

Übungsblatt Aufgabe V b):

Es gilt $SPEC_{ratio} = \frac{Referenzzeit_x}{Laufzeit_x \text{ auf Testsystem}}$ für einen Benchmark x.

Die Tabelle auf der angegebenen Webseite enthält die **Laufzeiten der Benchmarks** auf dem Testsystem und die $SPEC_{ratio}$.

Rechenbeispiel Leistungsbewertung (forts.)

Results Table				
Benchmark	Base		Peak	
	Seconds	Ratio	Seconds	Ratio
400.perlbench	344	28.4	291	33.6
401.bzip2	443	21.8	442	21.8
403.gcc	303	26.6	269	29.9
429.mcf	198	46.0	174	52.3
445.gobmk	389	26.9	358	29.3
456.hmmer	177	52.8	172	54.4
458.sjeng	426	28.4	408	29.7
462.libquantum	33.8	613	33.8	613
464.h264ref	529	41.8	486	45.5
471.omnetpp	271	23.1	214	29.2
473.astar	310	22.6	297	23.6
483.xalancbmk	168	41.1	168	41.1

Quelle: <http://www.spec.org/cpu2006/results/res2010q2/cpu2006-20100329-10254.html>

Somit ergibt sich nach dem Umstellen und Einsetzen:

$$\text{Referenzzeit}_{462.\text{libquantum}} = 613 * 33,8 \text{ s} = 20719,4 \text{ s}$$

Übungsblatt Aufgabe V c):

Welches der unter

<http://www.spec.org/cpu2006/results/cpu2006.html>
aufgeführten Systeme entspricht am ehesten dem Referenzsystem?

Antwort: die Suche ergibt: **Ultra Enterprise 2** von Sun Microsystems.

Begründung:

- Die unter b) errechnete Referenzlaufzeit für den ausgewählten Benchmark stimmt annähernd überein:

Benchmark	<i>Referenzzeit_{errechnet}</i>	<i>Laufzeit_{Ultra Enterprise 2}</i>
462.libquantum	20719,4	20704

- Es wird der $SPEC_{int_base}^{2006} = 1.00$ angegeben.

Übungsblatt Aufgabe V c):

Welches der unter

<http://www.spec.org/cpu2006/results/cpu2006.html>
aufgeführten Systeme entspricht am ehesten dem Referenzsystem?

Antwort: die Suche ergibt: **Ultra Enterprise 2** von Sun Microsystems.

Begründung:

- Die unter b) errechnete Referenzlaufzeit für den ausgewählten Benchmark stimmt annähernd überein:

Benchmark	<i>Referenzzeit</i> _{errechnet}	<i>Laufzeit</i> _{Ultra Enterprise 2}
462.libquantum	20719,4	20704

- Es wird der $SPEC_{int_base}^{2006} = 1.00$ angegeben.

■ Monitore

- Gezielte Abfrage und Akkumulation von HW-Ereignissen
- Software-Monitore
- Werkzeug zur Optimierung, weniger zur Klassifizierung
- Nachteil: Beeinträchtigung des Systemverhaltens (SW)

■ Analytische Methoden

- Deterministisch (feste Werte)
- Stochastisch (statische Verteilung)
- Operationell (gemessen in festem Zeitintervall)

■ Simulationen

- Modellbildung
- Deterministische, stochastische oder aufzeichnungsgesteuerte Simulation

- **Prozessortakt** gibt lediglich den Arbeitstakt (min/typ/max) des Prozessors an.
 - Kein Maß für Leistungsfähigkeit, da keine Aussage über Effizienz, Güte des Befehlssatzes etc.
 - Beispiele: Pentium4 vs. Pentium-M
- **CPI** ist ein Maß für die Effizienz einer Architektur.
 - Unterschied zwischen maximalen CPI unter Idealbedingungen und realen, programmabhängig gemessenen CPI
 - Zur Leistungsbewertung als **alleinige** Maßzahl nicht ausreichend: Effizienz $\neq$ Geschwindigkeit!
- **MIPS** auf den ersten Blick ideal, weil zwei Maßzahlen (Takt, CPI) zusammengeführt werden.
 - Aufgrund des CPI-Einflusses jedoch ebenfalls vom ausgeführten Programm abhängig
 - Nur unter gleichen Bedingungen (Sourcecode, Compiler, OS) direkt vergleichbar.

- **CPI** (Zyklen pro Instruktion)

$$CPI = \frac{c}{i}$$

- **MIPS** (Million Instructions per Second)

$$MIPS = \frac{f}{CPI * 10^6}$$

- **Taktrate** (Frequenz)

$$f = \frac{c}{t} = \frac{i * CPI}{t} \text{ [Hz]}$$

- **CPU-Zeit**

$$t_{cpu} = c * t$$

Beachten Sie auch mögliche Umformungen!

Frage: Welche wichtigen **Architekturparameter** beeinflussen jeweils die Zykluszeit, die Anzahl der Instruktionen und den CPI-Wert?

Antwort:

- Die **Zykluszeit** hängt von der **Organisation** und der **Technologie** ab.
- Die **Anzahl der Instruktionen** ist bedingt durch die **Befehlssatzarchitektur** und die Güte des **Compilers**.
- Die **Zyklen pro Instruktion** werden durch die **Organisation** und die **Befehlssatzarchitektur** beeinflusst.

Prozessor A arbeitet ein Problem in 2ms ab. Er hat eine CPI von 7/5 und benötigt 3.500.000 Instruktionen für die Abarbeitung der Problemstellung. Prozessor B arbeitet dieses Problem ebenfalls in 2ms ab. Er hat eine CPI von 3/2 und benötigt 1.500.000 Instruktionen.

- **Welcher Prozessor ist für dieses Problem zu wählen und warum?**

$$f = \frac{i * CPI}{t}, MIPS = \frac{f}{CPI * 10^6}$$

$$f_A = \frac{3.5 * 10^6 * \frac{7}{5}}{2 * 10^{-3}} = 2450 MHz$$

$$MIPS_A = \frac{f_A}{CPI_A * 10^6} = \frac{2.45 * 10^9}{\frac{7}{5} * 10^6} = 1750 MIPS$$

$$f_B = \frac{1.5 * 10^6 * \frac{3}{2}}{2 * 10^{-3}} = 1125 MHz$$

$$MIPS_B = \frac{f_B}{CPI_B * 10^6} = \frac{1.125 * 10^9}{\frac{3}{2} * 10^6} = 750 MIPS$$

Prozessor A arbeitet ein Problem in 2ms ab. Er hat eine CPI von 7/5 und benötigt 3.500.000 Instruktionen für die Abarbeitung der Problemstellung. Prozessor B arbeitet dieses Problem ebenfalls in 2ms ab. Er hat eine CPI von 3/2 und benötigt 1.500.000 Instruktionen.

- **Welcher Prozessor ist für dieses Problem zu wählen und warum?**

$$f = \frac{i * CPI}{t}, MIPS = \frac{f}{CPI * 10^6}$$

$$f_A = \frac{3.5 * 10^6 * \frac{7}{5}}{2 * 10^{-3}} = 2450 MHz$$

$$MIPS_A = \frac{f_A}{CPI_A * 10^6} = \frac{2.45 * 10^9}{\frac{7}{5} * 10^6} = 1750 MIPS$$

$$f_B = \frac{1.5 * 10^6 * \frac{3}{2}}{2 * 10^{-3}} = 1125 MHz$$

$$MIPS_B = \frac{f_B}{CPI_B * 10^6} = \frac{1.125 * 10^9}{\frac{3}{2} * 10^6} = 750 MIPS$$

Prozessor A arbeitet ein Problem in 2ms ab. Er hat eine CPI von 7/5 und benötigt 3.500.000 Instruktionen für die Abarbeitung der Problemstellung. Prozessor B arbeitet dieses Problem ebenfalls in 2ms ab. Er hat eine CPI von 3/2 und benötigt 1.500.000 Instruktionen.

- **Welcher Prozessor ist für dieses Problem zu wählen und warum?**

$$f = \frac{i \cdot CPI}{t}, MIPS = \frac{f}{CPI \cdot 10^6}$$

$$f_A = \frac{3.5 \cdot 10^6 \cdot \frac{7}{5}}{2 \cdot 10^{-3}} = 2450 MHz$$

$$MIPS_A = \frac{f_A}{CPI_A \cdot 10^6} = \frac{2.45 \cdot 10^9}{\frac{7}{5} \cdot 10^6} = 1750 MIPS$$

$$f_B = \frac{1.5 \cdot 10^6 \cdot \frac{3}{2}}{2 \cdot 10^{-3}} = 1125 MHz$$

$$MIPS_B = \frac{f_B}{CPI_B \cdot 10^6} = \frac{1.125 \cdot 10^9}{\frac{3}{2} \cdot 10^6} = 750 MIPS$$

Prozessor A arbeitet ein Problem in 2ms ab. Er hat eine CPI von 7/5 und benötigt 3.500.000 Instruktionen für die Abarbeitung der Problemstellung. Prozessor B arbeitet dieses Problem ebenfalls in 2ms ab. Er hat eine CPI von 3/2 und benötigt 1.500.000 Instruktionen.

- Welcher Prozessor ist für dieses Problem zu wählen und warum?

$$f = \frac{i * CPI}{t}, MIPS = \frac{f}{CPI * 10^6}$$

$$f_A = \frac{3.5 * 10^6 * \frac{7}{5}}{2 * 10^{-3}} = 2450 MHz$$

$$MIPS_A = \frac{f_A}{CPI_A * 10^6} = \frac{2.45 * 10^9}{\frac{7}{5} * 10^6} = 1750 MIPS$$

$$f_B = \frac{1.5 * 10^6 * \frac{3}{2}}{2 * 10^{-3}} = 1125 MHz$$

$$MIPS_B = \frac{f_B}{CPI_B * 10^6} = \frac{1.125 * 10^9}{\frac{3}{2} * 10^6} = 750 MIPS$$

Rechenbeispiel II (forts.)

$$i_A = 3.500.000, CPI_A = \frac{7}{5}$$
$$f_A = 2450MHz, MIPS_A = 1750MIPS$$

$$i_B = 1.500.000, CPI_B = \frac{3}{2}$$
$$f_B = 1125MHz, MIPS_B = 750MIPS$$

$$t = 2ms$$

■ Welcher Prozessor ist für dieses Problem zu wählen und warum?

■ Prozessor B, weil

- ohne Berechnung: Gleich schnell in der Abarbeitung bei wesentlich weniger Instruktionen (1.5 vs. 3.5 Mio Instruktionen)
- halbe Taktfrequenz ($P \sim U^2 * f$, Fertigung)

Rechenbeispiel II (forts.)

$$i_A = 3.500.000, CPI_A = \frac{7}{5}$$
$$f_A = 2450MHz, MIPS_A = 1750MIPS$$

$$i_B = 1.500.000, CPI_B = \frac{3}{2}$$
$$f_B = 1125MHz, MIPS_B = 750MIPS$$

$$t = 2ms$$

- **Welcher Prozessor ist für dieses Problem zu wählen und warum?**
- **Prozessor B**, weil
 - ohne Berechnung: Gleich schnell in der Abarbeitung bei wesentlich weniger Instruktionen (1.5 vs. 3.5 Mio Instruktionen)
 - halbe Taktfrequenz ($P \sim U^2 * f$, Fertigung)

Rechenbeispiel III Benchmarks

Benchmarks sind eine verlässliche Methode zur Leistungsbewertung. Auf einem 4GHz-Prozessor wird ein solcher Benchmark abgearbeitet. Nachfolgende Tabelle listet die auftretenden Befehlstypen mit Häufigkeit und jeweiliger Zyklenzahl.

Befehlstyp	Anzahl in 10^3	Zyklenzahl
Integer-Arithmetik	300	1
Fließkomma-Arithmetik	75	2
Speicherzugriff	150	3
Kontrollflusstransfer	25	4

Zu bestimmen sind die Werte für Ausführungszeit, CPI, MIPS und MFLOPS.

Rechenbeispiel III (forts.)

Befehlstyp	Anzahl in 10^3	Zyklenzahl
Integer-Arithmetik	300	1
Fließkomma-Arithmetik	75	2
Speicherzugriff	150	3
Kontrollflusstransfer	25	4

■ Anzahl Instruktionen

$$i = \sum i_{\text{typ}} = (300 + 75 + 150 + 25) * 10^3 = 550.000$$

■ Taktzyklen

$$\begin{aligned} c &= \sum i_{\text{typ}} * c_{\text{typ}} \\ &= (300 * 1 + 75 * 2 + 150 * 3 + 25 * 4) * 10^3 = 1.000.000 \end{aligned}$$

■ Zykluszeit bei 4GHz Taktfrequenz

$$t = \frac{1}{f} = \frac{1}{4\text{GHz}} = 0.25 * 10^{-9} \text{s} = 0.25 \text{ns}$$

Rechenbeispiel III (forts.)

Befehlstyp	Anzahl in 10^3	Zyklenzahl
Integer-Arithmetik	300	1
Fließkomma-Arithmetik	75	2
Speicherzugriff	150	3
Kontrollflusstransfer	25	4

■ Anzahl Instruktionen

$$i = \sum i_{\text{typ}} = (300 + 75 + 150 + 25) * 10^3 = 550.000$$

■ Taktzyklen

$$c = \sum i_{\text{typ}} * c_{\text{typ}} \\ = (300 * 1 + 75 * 2 + 150 * 3 + 25 * 4) * 10^3 = 1.000.000$$

■ Zykluszeit bei 4GHz Taktfrequenz

$$t = \frac{1}{f} = \frac{1}{4\text{GHz}} = 0.25 * 10^{-9}\text{s} = 0.25\text{ns}$$

Rechenbeispiel III (forts.)

Befehlstyp	Anzahl in 10^3	Zyklenzahl
Integer-Arithmetik	300	1
Fließkomma-Arithmetik	75	2
Speicherzugriff	150	3
Kontrollflusstransfer	25	4

■ Anzahl Instruktionen

$$i = \sum i_{\text{typ}} = (300 + 75 + 150 + 25) * 10^3 = 550.000$$

■ Taktzyklen

$$c = \sum i_{\text{typ}} * c_{\text{typ}} \\ = (300 * 1 + 75 * 2 + 150 * 3 + 25 * 4) * 10^3 = 1.000.000$$

■ Zykluszeit bei 4GHz Taktfrequenz

$$t = \frac{1}{f} = \frac{1}{4\text{GHz}} = 0.25 * 10^{-9}\text{s} = 0.25\text{ns}$$

Rechenbeispiel III (forts.)

Befehlstyp	Anzahl in 10^3	Zyklenzahl
Integer-Arithmetik	300	1
Fließkomma-Arithmetik	75	2
Speicherzugriff	150	3
Kontrollflusstransfer	25	4

■ Anzahl Instruktionen

$$i = \sum i_{\text{typ}} = (300 + 75 + 150 + 25) * 10^3 = 550.000$$

■ Taktzyklen

$$\begin{aligned} c &= \sum i_{\text{typ}} * c_{\text{typ}} \\ &= (300 * 1 + 75 * 2 + 150 * 3 + 25 * 4) * 10^3 = 1.000.000 \end{aligned}$$

■ Zykluszeit bei 4GHz Taktfrequenz

$$t = \frac{1}{f} = \frac{1}{4\text{GHz}} = 0.25 * 10^{-9} \text{s} = 0.25 \text{ns}$$

Rechenbeispiel III (forts.)

■ Angaben und bisherige Berechnungen:

$$f = 4\text{GHz} \rightarrow t_c = 0.25\text{ns}$$

$$i = 550.000, c = 1.000.000 = 1000 * 10^3$$

■ Ausführungszeit

$$t_{\text{exec}} = c * t_{\text{cyc}}$$

$$= 1000 * 10^3 * 0.25 * 10^{-9} = 250 * 10^{-6}\text{s} = 250\mu\text{s}$$

■ CPI

$$\text{CPI} = \frac{c}{i} = \frac{1000 * 10^3}{550 * 10^3} = \frac{100}{55} = \frac{20}{11} \approx 1.82$$

■ MIPS

$$\text{MIPS} = \frac{i}{t * 10^6} = \frac{550.000}{250} = 2200$$

■ MFLOPS

- Wie MIPS, wobei Anzahl der Befehle und Ausführungszeit nur für Fließkommaberechnung

$$\text{MFLOPS} = \frac{75.000}{(75.000 * 2) * (0.25 * 10^{-9}) * 10^6} = \frac{1}{0.5 * 10^{-3}} = 2000$$

Rechenbeispiel III (forts.)

■ Angaben und bisherige Berechnungen:

$$f = 4GHz \rightarrow t_c = 0.25ns$$

$$i = 550.000, c = 1.000.000 = 1000 * 10^3$$

■ Ausführungszeit

$$t_{exec} = c * t_{cyc}$$

$$= 1000 * 10^3 * 0.25 * 10^{-9} = 250 * 10^{-6}s = 250\mu s$$

■ CPI

$$CPI = \frac{c}{i} = \frac{1000 * 10^3}{550 * 10^3} = \frac{100}{55} = \frac{20}{11} \approx 1.82$$

■ MIPS

$$MIPS = \frac{i}{t * 10^6} = \frac{550.000}{250} = 2200$$

■ MFLOPS

- Wie MIPS, wobei Anzahl der Befehle und Ausführungszeit nur für Fließkommaberechnung

$$MFLOPS = \frac{75.000}{(75.000 * 2) * (0.25 * 10^{-9}) * 10^6} = \frac{1}{0.5 * 10^{-3}} = 2000$$

Rechenbeispiel III (forts.)

■ Angaben und bisherige Berechnungen:

$$f = 4GHz \rightarrow t_c = 0.25ns$$

$$i = 550.000, c = 1.000.000 = 1000 * 10^3$$

■ Ausführungszeit

$$t_{exec} = c * t_{cyc}$$

$$= 1000 * 10^3 * 0.25 * 10^{-9} = 250 * 10^{-6}s = 250\mu s$$

■ CPI

$$CPI = \frac{c}{i} = \frac{1000 * 10^3}{550 * 10^3} = \frac{100}{55} = \frac{20}{11} \approx 1.82$$

■ MIPS

$$MIPS = \frac{i}{t * 10^6} = \frac{550.000}{250} = 2200$$

■ MFLOPS

- Wie MIPS, wobei Anzahl der Befehle und Ausführungszeit nur für Fließkommaberechnung

$$MFLOPS = \frac{75.000}{(75.000 * 2) * (0.25 * 10^{-9}) * 10^6} = \frac{1}{0.5 * 10^{-3}} = 2000$$

Rechenbeispiel III (forts.)

■ Angaben und bisherige Berechnungen:

$$f = 4GHz \rightarrow t_c = 0.25ns$$

$$i = 550.000, c = 1.000.000 = 1000 * 10^3$$

■ Ausführungszeit

$$t_{exec} = c * t_{cyc}$$

$$= 1000 * 10^3 * 0.25 * 10^{-9} = 250 * 10^{-6}s = 250\mu s$$

■ CPI

$$CPI = \frac{c}{i} = \frac{1000 * 10^3}{550 * 10^3} = \frac{100}{55} = \frac{20}{11} \approx 1.82$$

■ MIPS

$$MIPS = \frac{i}{t * 10^6} = \frac{550.000}{250} = 2200$$

■ MFLOPS

- Wie MIPS, wobei Anzahl der Befehle und Ausführungszeit nur für Fließkommaberechnung

$$MFLOPS = \frac{75.000}{(75.000 * 2) * (0.25 * 10^{-9}) * 10^6} = \frac{1}{0.5 * 10^{-3}} = 2000$$

Rechenbeispiel III (forts.)

■ Angaben und bisherige Berechnungen:

$$f = 4GHz \rightarrow t_c = 0.25ns$$

$$i = 550.000, c = 1.000.000 = 1000 * 10^3$$

■ Ausführungszeit

$$t_{exec} = c * t_{cyc}$$

$$= 1000 * 10^3 * 0.25 * 10^{-9} = 250 * 10^{-6}s = 250\mu s$$

■ CPI

$$CPI = \frac{c}{i} = \frac{1000 * 10^3}{550 * 10^3} = \frac{100}{55} = \frac{20}{11} \approx 1.82$$

■ MIPS

$$MIPS = \frac{i}{t * 10^6} = \frac{550.000}{250} = 2200$$

■ MFLOPS

- Wie MIPS, wobei Anzahl der Befehle und Ausführungszeit nur für Fließkommaberechnung

$$MFLOPS = \frac{75.000}{(75.000 * 2) * (0.25 * 10^{-9}) * 10^6} = \frac{1}{0.5 * 10^{-3}} = 2000$$

Leistungsbewertung IV

(vergl. Hennessy and Patterson, Computer Architecture A Quantitative Approach, 4.Auflage, S. 43-44.)

Frage: Sie haben für Ihre neue Rechnerarchitektur die folgenden Werte experimentell bestimmt:

Befehlstyp	CPI	Anteil
Fließkomma-Arithmetik	4,0	25%
Restliche Befehle	1,33	75%

Die Häufigkeit der Instruktion FPSQR beträgt 2% und der $CPI_{FPSQR} = 20$. Es gibt zwei Entwurfsmöglichkeiten:

- (a) senken des CPI_{FPSQR} auf 2
- (b) senken des CPI-Wert der Gleitkommaoperationen auf 2,5.

Berechnen Sie den jeweiligen Gesamtgewinn der Alternativen und begründen Sie die Entscheidung.

Leistungsbewertung IV

(vergl. Hennessy and Patterson, Computer Architecture A Quantitative Approach, 4.Auflage, S. 43-44.)

Frage: Sie haben für Ihre neue Rechnerarchitektur die folgenden Werte experimentell bestimmt:

Befehlstyp	CPI	Anteil
Fließkomma-Arithmetik	4,0	25%
Restliche Befehle	1,33	75%

Die Häufigkeit der Instruktion FPSQR beträgt 2% und der $CPI_{FPSQR} = 20$. Es gibt zwei Entwurfsmöglichkeiten:

- (a) senken des CPI_{FPSQR} auf 2
- (b) senken des CPI-Wert der Gleitkommaoperationen auf 2,5.

Berechnen Sie den jeweiligen Gesamtgewinn der Alternativen und begründen Sie die Entscheidung.

Leistungsbewertung IV (forts.)

Antwort: Es ändern sich nur die Zyklen pro Instruktion, Taktrate und Anzahl der Instruktionen (i) bleiben gleich.

Der unoptimierte CPI-Wert errechnet sich nach:

$$CPI_{\text{base}} = \sum_{i=1}^n CPI_i * Anteil_i = (1,33 * 75\%) + (4 * 25\%) \approx 2,0.$$

Die Zyklen pro Instruktion mit neuem FPSQR-Befehl: $CPI_{(a)}$ kann durch Abziehen der gesparten Zyklen erfolgen:

$$\begin{aligned} CPI_{(a)} &= CPI_{\text{base}} - 0,02 * (CPI_{\text{old FPSQR}} - CPI_{\text{new FPSQR}}) \\ &= 2,0 - 0,02 * (20 - 2) = 1,64. \end{aligned}$$

Alternative (b) errechnet sich analog zum CPI_{base} :

$$CPI_{(b)} = (1,33 * 75\%) + (2,5 * 25\%) \approx 1,62.$$

Aufgrund des **geringeren** CPI-Werts bietet sich die **Alternative (b)** mit den verbesserten Zyklen pro Gleitkommaoperation an.

Leistungsbewertung IV (forts.)

Antwort: Es ändern sich nur die Zyklen pro Instruktion, Taktrate und Anzahl der Instruktionen (i) bleiben gleich.

Der unoptimierte CPI-Wert errechnet sich nach:

$$CPI_{\text{base}} = \sum_{i=1}^n CPI_i * Anteil_i = (1,33 * 75\%) + (4 * 25\%) \approx 2,0.$$

Die Zyklen pro Instruktion mit neuem FPSQR-Befehl: $CPI_{(a)}$ kann durch Abziehen der gesparten Zyklen erfolgen:

$$\begin{aligned} CPI_{(a)} &= CPI_{\text{base}} - 0,02 * (CPI_{\text{old FPSQR}} - CPI_{\text{new FPSQR}}) \\ &= 2,0 - 0,02 * (20 - 2) = 1,64. \end{aligned}$$

Alternative (b) errechnet sich analog zum CPI_{base} :

$$CPI_{(b)} = (1,33 * 75\%) + (2,5 * 25\%) \approx 1,62.$$

Aufgrund des **geringeren** CPI-Werts bietet sich die **Alternative (b)** mit den verbesserten Zyklen pro Gleitkommaoperation an.

Leistungsbewertung IV (forts.)

Antwort: Es ändern sich nur die Zyklen pro Instruktion, Taktrate und Anzahl der Instruktionen (i) bleiben gleich.

Der unoptimierte CPI-Wert errechnet sich nach:

$$CPI_{\text{base}} = \sum_{i=1}^n CPI_i * Anteil_i = (1,33 * 75\%) + (4 * 25\%) \approx 2,0.$$

Die Zyklen pro Instruktion mit neuem FPSQR-Befehl: $CPI_{(a)}$ kann durch Abziehen der gesparten Zyklen erfolgen:

$$\begin{aligned} CPI_{(a)} &= CPI_{\text{base}} - 0,02 * (CPI_{\text{old FPSQR}} - CPI_{\text{new FPSQR}}) \\ &= 2,0 - 0,02 * (20 - 2) = 1,64. \end{aligned}$$

Alternative (b) errechnet sich analog zum CPI_{base} :

$$CPI_{(b)} = (1,33 * 75\%) + (2,5 * 25\%) \approx 1,62.$$

Aufgrund des **geringeren** CPI-Werts bietet sich die **Alternative (b)** mit den verbesserten Zyklen pro Gleitkommaoperation an.

Leistungsbewertung IV (forts.)

Berechnung des Gewinns (Speedup) durch die Verwendung der Alternative (b) gegenüber dem vorherigen System (base):

$$\begin{aligned} \text{Speedup}_{(b)} &= \frac{\text{CPU time}_{base}}{\text{CPU time}_{(b)}} \\ &= \frac{i * \text{Taktrate} * \text{CPI}_{base}}{i * \text{Taktrate} * \text{CPI}_{(b)}} \\ &= \frac{\text{CPI}_{base}}{\text{CPI}_{(b)}} \end{aligned}$$

Eingesetzt ergibt sich:

$$\text{Speedup}_{(b)} = \frac{2,00}{1,62} \approx 1,23$$

→ Alternative (b) ist 1,23-mal schneller als bisheriges System.

Für eine Rechenanlage soll ein geeigneter Plattenspeicher angeschafft werden. Mithilfe eines Warteschlangenmodells sollen hierzu der **Durchsatz D** und die **Auslastung U** der Plattensysteme berechnet werden unter der Annahme, die durchschnittliche **Ankunftsrate A** von Schreib-/Leseaufträgen im System liegt bei 40/s. Zur Auswahl stehen Festplatten mit folgenden Daten:

- Platte 1: Zugriffszeit 12ms, Datenrate 6MByte/s
- Platte 2: Zugriffszeit 10ms, Datenrate 7,5MByte/s

Leistungsbewertung VI (forts.)

- Berechnen Sie für die 2 Festplatten die Bedienzeit X_i , wenn der Schreib-/Leseauftrag im Schnitt 100kB groß ist.
- Platte 1: Zugriffszeit 12ms, Datenrate 6MByte/s
- Platte 2: Zugriffszeit 10ms, Datenrate 7,5MByte/s
- **Bedienzeiten:** $X_i = t_{\text{Zugriff}} + t_{\text{Übertragung}}$

$$X_1 = 12ms + \frac{100kB}{6000kB/s} = 28,67ms$$

$$X_2 = 10ms + \frac{100kB}{7500kB/s} = 23,33ms$$

Leistungsbewertung VI (forts.)

- Berechnen Sie für die 2 Festplatten die Bedienzeit X_i , wenn der Schreib-/Leseauftrag im Schnitt 100kB groß ist.
- Platte 1: Zugriffszeit 12ms, Datenrate 6MByte/s
- Platte 2: Zugriffszeit 10ms, Datenrate 7,5MByte/s
- **Bedienzeiten:** $X_i = t_{\text{Zugriff}} + t_{\text{Übertragung}}$

$$X_1 = 12ms + \frac{100kB}{6000kB/s} = 28,67ms$$

$$X_2 = 10ms + \frac{100kB}{7500kB/s} = 23,33ms$$

Leistungsbewertung VI (forts.)

- Berechnen Sie für die 2 Festplatten die Bedienzeit X_i , wenn der Schreib-/Leseauftrag im Schnitt 100kB groß ist.
- Platte 1: Zugriffszeit 12ms, Datenrate 6MByte/s
- Platte 2: Zugriffszeit 10ms, Datenrate 7,5MByte/s
- **Bedienzeiten:** $X_i = t_{\text{Zugriff}} + t_{\text{Übertragung}}$

$$X_1 = 12ms + \frac{100kB}{6000kB/s} = 28,67ms$$

$$X_2 = 10ms + \frac{100kB}{7500kB/s} = 23,33ms$$

- Wie groß sind die Durchsätze D_i der einzelnen Festplatten?
Welche Festplatten wären aufgrund der Berechnung im System einsetzbar?

- **Maximaler Durchsatz:** $D_{imax} = \frac{1}{X_i}$

$$D_{1max} = \frac{1}{28,67ms} = 34,88/s$$

$$D_{2max} = \frac{1}{23,33ms} = 42,86/s$$

Nur Platten mit $D_{max} > A$ können eingesetzt werden, da sonst die Festplatte nicht genügend Zeit hat, um alle Aufträge rechtzeitig zu bedienen.

Aufgrund von $A = 40/s$ ist somit nur die Platte 2 einsetzbar.

- Wie groß sind die Durchsätze D_i der einzelnen Festplatten?
Welche Festplatten wären aufgrund der Berechnung im System einsetzbar?

- **Maximaler Durchsatz:** $D_{imax} = \frac{1}{X_i}$

$$D_{1max} = \frac{1}{28,67ms} = 34,88/s$$

$$D_{2max} = \frac{1}{23,33ms} = 42,86/s$$

Nur Platten mit $D_{max} > A$ können eingesetzt werden, da sonst die Festplatte nicht genügend Zeit hat, um alle Aufträge rechtzeitig zu bedienen.

Aufgrund von $A = 40/s$ ist somit nur die Platte 2 einsetzbar.

- Wie groß sind die Durchsätze D_i der einzelnen Festplatten?
Welche Festplatten wären aufgrund der Berechnung im System einsetzbar?

- **Maximaler Durchsatz:** $D_{imax} = \frac{1}{X_i}$

$$D_{1max} = \frac{1}{28,67ms} = 34,88/s$$

$$D_{2max} = \frac{1}{23,33ms} = 42,86/s$$

Nur Platten mit $D_{max} > A$ können eingesetzt werden, da sonst die Festplatte nicht genügend Zeit hat, um alle Aufträge rechtzeitig zu bedienen.

Aufgrund von $A = 40/s$ ist somit nur die Platte 2 einsetzbar.

- Wie groß sind die Durchsätze D_i der einzelnen Festplatten?
Welche Festplatten wären aufgrund der Berechnung im System einsetzbar?

- **Maximaler Durchsatz:** $D_{imax} = \frac{1}{X_i}$

$$D_{1max} = \frac{1}{28,67ms} = 34,88/s$$

$$D_{2max} = \frac{1}{23,33ms} = 42,86/s$$

Nur Platten mit $D_{max} > A$ können eingesetzt werden, da sonst die Festplatte nicht genügend Zeit hat, um alle Aufträge rechtzeitig zu bedienen.

Aufgrund von $A = 40/s$ ist somit nur die Platte 2 einsetzbar.

- Wie groß ist die Auslastung der einsetzbaren Festplatten?

- **Auslastung:** $U_i = D/D_{imax} = D * X_i$, hier $D = A$

$$U_2 = D * X_2 = 40/s * 23,33ms = 0,93,$$

d.h. 93% Auslastung

- Wie groß ist die Auslastung der einsetzbaren Festplatten?

- **Auslastung:** $U_i = D/D_{imax} = D * X_i$, hier $D = A$

$U_2 = D * X_2 = 40/s * 23,33ms = 0,93$,
d.h. 93% Auslastung

- Wie groß ist die Auslastung der einsetzbaren Festplatten?

- **Auslastung:** $U_i = D/D_{imax} = D * X_i$, hier $D = A$

$$U_2 = D * X_2 = 40/s * 23,33ms = 0,93,$$

d.h. 93% Auslastung

Analyse von Architekturen

- Graphische Repräsentation einer Architektur durch **Zuverlässigkeitsblockdiagramm**
- Abbildung auf gleichwertige **Strukturformel**
- Transformierung in Berechnungsformel

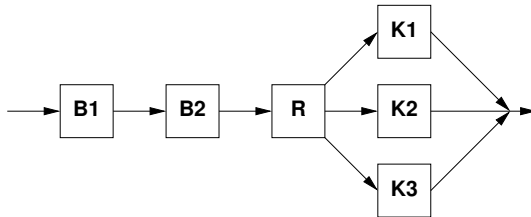
Aufgabe 7

Gegeben sei ein portables Rechnersystem bestehend aus zwei Batterien B_1 und B_2 , der eigentlichen Recheneinheit R und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3 . Zum fehlerfreien Betrieb des Systems sind beide Batterien, die Recheneinheit und mindestens eine Kommunikationskomponente erforderlich.

Erstellen Sie Zuverlässigkeitsblockdiagramm und Strukturformel.

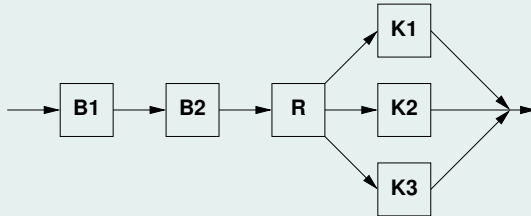
Aufgabe 7

Gegeben sei ein portables Rechnersystem bestehend aus zwei Batterien B_1 und B_2 , der eigentlichen Recheneinheit R und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3 . Zum fehlerfreien Betrieb des Systems sind beide Batterien, die Recheneneinheit und mindestens eine Kommunikationskomponente erforderlich. Erstellen Sie Zuverlässigkeitsblockdiagramm und Strukturformel.



Zuverlässigkeitsblockdiagramm

Zuverlässigkeitsblockdiagramm



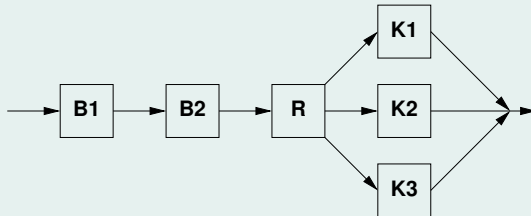
Strukturformel

$$S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$$

oder

$$S = B_1 \text{ and } B_2 \text{ and } R \text{ and } (K_1 \text{ or } K_2 \text{ or } K_3)$$

Zuverlässigkeitsblockdiagramm



Strukturformel

$$S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$$

oder

$$S = B_1 \text{ and } B_2 \text{ and } R \text{ and } (K_1 \text{ or } K_2 \text{ or } K_3)$$

Aufgabe 7 (forts.)

Berechnen sie die Funktionswahrscheinlichkeit.

$$S = B_1 \wedge B_2 \wedge R \wedge (K_1 \vee K_2 \vee K_3)$$

Strukturformel

- Gegeben seien die Funktionswahrscheinlichkeiten $\varphi(B)$, $\varphi(R)$ und $\varphi(K)$.

Umformung in Formel zur Berechnung:

- Funktionswahrscheinlichkeit eines **Seriensystems**:
 $\varphi(\wedge_{K \in \Lambda}) = \prod_{K \in \Lambda} \varphi(K)$, also $\varphi = \varphi(B) * \varphi(B) * \varphi(R) * \dots$
- Funktionswahrscheinlichkeit eines **Parallelsystems**:
 $\varphi(\vee_{K \in \Lambda}) = \sum_{\emptyset \neq A \in \Lambda} (-1)^{1+\#A} * \varphi(\wedge_{K \in A} K)$

Wie mit 1-aus-n umgehen?

- Betrachtung der **Ausfallwahrscheinlichkeit**
 - Umformung in Seriensystem gemäß boolescher Logik
 $(K_1 \vee K_2 \vee K_3) \rightarrow \neg(\neg K_1 \wedge \neg K_2 \wedge \neg K_3)$
 - $K \rightarrow \neg K$, entsprechend $\varphi(K) \rightarrow 1 - \varphi(K)$
 - Ausfallwahrscheinlichkeit für K-System damit: $(1 - \varphi(K))^3$
 - Anschließend: **Retransformation** in Funktionswahrscheinlichkeit

■ somit: $\varphi = \underbrace{\varphi(B) * \varphi(B) * \varphi(R)}_{\text{Seriensystem}} * \underbrace{(1 - (1 - \varphi(K))^3)}_{\text{Parallelsystem}}$

■ Zuverlässigkeitsblockdiagramm und Strukturformel:

Erfassung aller Funktionszustände

- Beispiel: 2-aus-3-System
- System funktionsfähig, wenn 1&2, 1&3, 2&3, 1&2&3 funktionsfähig
- System nicht funktionsfähig, wenn nur 1, 2, oder 3 funktionsfähig.

■ Zuverlässigkeitsberechnung direkt über:

$$\varphi_m^n = \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- Beispiel: 2-aus-3-System, n=2, m=3

$$\varphi_3^2 = \sum_{k=2}^3 \binom{3}{k} * \varphi(K)^k * (1 - \varphi(K))^{(3-k)}$$

■ Systeme mit Mehrheitsentscheider:

$$\varphi_m^n = \varphi(V) * \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- $\varphi(K)$: Funktionswahrscheinlichkeit der Komponente
- $\varphi(V)$: Funktionswahrscheinlichkeit des Mehrheitsentscheiders (Voter)
- Entscheider ist **single point of failure!**
 - $\varphi(V)$ idealerweise $\rightarrow 1$
 - Voter vergleichsweise einfache Einheit, daher geringe Fehleranfälligkeit
 - Ggf. seinerseits Redundanzsystem (Teilauswertungen)

Teilaufgabe a)

Ein RAID2-System besteht per Definition aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu Datenverlust kommt. Unter der Annahme, die Funktionswahrscheinlichkeit pro Festplatte betrage $\varphi(F) = 0,99$, wie hoch ist die Chance auf Datenverlust?

- allgemein:

$$\varphi_m^n = \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- $n=8, m=10, \varphi(K) = \varphi(F) = 0,99$ – also:

$$\varphi_{10}^8 = \sum_{k=8}^{10} \binom{10}{k} * 0.99^k * 0.01^{(10-k)} = 0.999886$$

Chance auf Datenverlust somit: $1 - 0.999886 = 0.000114$.

Teilaufgabe a)

Ein RAID2-System besteht per Definition aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu Datenverlust kommt. Unter der Annahme, die Funktionswahrscheinlichkeit pro Festplatte betrage $\varphi(F) = 0,99$, wie hoch ist die Chance auf Datenverlust?

- allgemein:

$$\varphi_m^n = \sum_{k=n}^m \binom{m}{k} * \varphi(K)^k * (1 - \varphi(K))^{(m-k)}$$

- $n=8, m=10, \varphi(K) = \varphi(F) = 0,99$ – also:

$$\varphi_{10}^8 = \sum_{k=8}^{10} \binom{10}{k} * 0.99^k * 0.01^{(10-k)} = 0.999886$$

Chance auf Datenverlust somit: $1 - 0.999886 = 0.000114$.

- Mean Time to Failure (**MTTF**): mittlere Funktionszeit (**E(L)**)
- Mean Time to Repair (**MTTR**): mittlere Reparaturzeit (**E(B)**)
- Mean Time between Failures (**MTBF**): mittlere Zeit zwischen zwei Ausfällen, $MTBF = MTTF + MTTR$

(für $MTTR \ll MTTF$ gilt somit: $MTBF \sim MTTF$)

- **Punktverfügbarkeit** eines Systems (V):
Wahrscheinlichkeit, ein System zu einem beliebigen Zeitpunkt fehlerfrei anzutreffen, unabhängig davon, ob es bis zu diesem Zeitpunkt bereits ausgefallen ist oder nicht.

$$V = \frac{MTTF}{MTTF + MTTR} = \frac{MTTF}{MTBF}$$

- Für über die Zeit konstante Ausfallraten gilt außerdem:

$$\text{Ausfallrate } \lambda = \frac{1}{MTTF}$$

Teilaufgabe b)

Eine Festplatte habe eine MTTF von 2 Jahren im Dauerbetrieb. Die Reparaturzeit (MTTR) setze sich zusammen aus der Zeit für das Herunterfahren des Rechners (2 Minuten), Austausch der Festplatte (10 Minuten) und anschließendes Hochfahren des Rechners (2 Minuten).

Berechnen Sie die Punktverfügbarkeit V .

$$\blacksquare MTTF = 2a = (2 * 365 * 24 * 60)min = 1051200min$$

$$\blacksquare MTTR = (2 + 10 + 2)min = 14min$$

$$\blacksquare V = \frac{MTTF}{MTTF + MTTR} = \frac{1051200}{1051214} = 0,999997$$

Teilaufgabe b)

Eine Festplatte habe eine MTTF von 2 Jahren im Dauerbetrieb. Die Reparaturzeit (MTTR) setze sich zusammen aus der Zeit für das Herunterfahren des Rechners (2 Minuten), Austausch der Festplatte (10 Minuten) und anschließendes Hochfahren des Rechners (2 Minuten).

Berechnen Sie die Punktverfügbarkeit V .

$$\blacksquare MTTF = 2a = (2 * 365 * 24 * 60)min = 1051200min$$

$$\blacksquare MTTR = (2 + 10 + 2)min = 14min$$

$$\blacksquare V = \frac{MTTF}{MTTF + MTTR} = \frac{1051200}{1051214} = 0,999997$$

- **Konstante Ausfallrate ist vereinfachtes Modell**
- **Reale Systeme: variable Ausfallwahrscheinlichkeit** über Zeit

Badewannenkurve

1 Frühphase

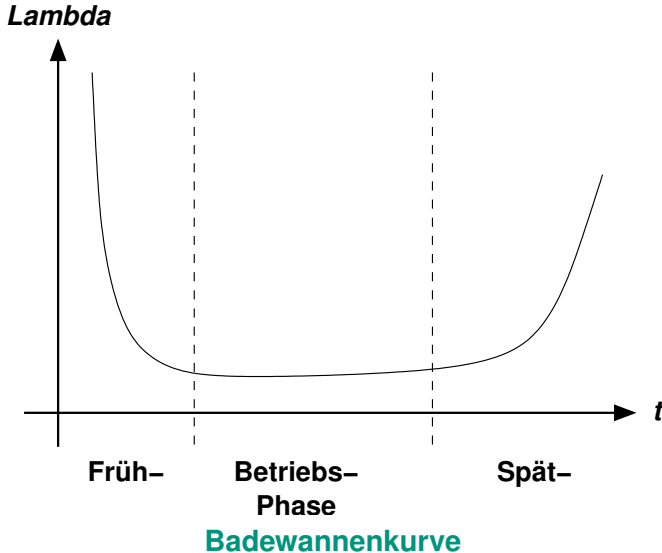
- Initialausfälle
- Fertigungsfehler, Bauteildefekte
- Ausfallrate exponentiell abfallend

2 Betriebsphase

- Nahezu konstante Ausfallrate

3 Spätphase

- Alterungseffekte
- Ausfallrate exponentiell ansteigend



Teilaufgabe d)

Gegeben sei ein 2-aus-3-System, dessen Komponenten zufallsverteilt mit gleicher Rate ausfallen. Die Überlebenswahrscheinlichkeit einer Komponente wird durch die Formel $R(t) = e^{-\lambda \cdot t}$, $t > 0$ beschrieben.

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?
- 2 Bestimmen Sie die Zeitintervalle, in denen das 2-von-3-System eine größere Überlebenswahrscheinlichkeit als eine einzelne Komponente aufweist.
- 3 Bestimmen Sie λ derart, dass die mittlere Lebensdauer für das gegebene 2-von-3-System $\frac{5}{6}$ beträgt.

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?

allgemein: $z(t) = \frac{f_L(t)}{R(t)} = \frac{1}{R(t)} * \frac{d}{dt}(F_L(t)) = \frac{1}{R(t)} * \frac{d}{dt}(1 - R(t))$

somit: $z(t) = \frac{1}{R(K, t)} * \frac{d}{dt}(1 - R(K, t)) = \frac{1}{e^{-\lambda t}} * (\lambda e^{-\lambda t}) = \lambda.$

- 2 Bestimmung der Zeitintervalle, in denen das 2-von-3-System eine größere Überlebenswahrscheinlichkeit als eine einzelne Komponente aufweist.

Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?

allgemein: $z(t) = \frac{f_L(t)}{R(t)} = \frac{1}{R(t)} * \frac{d}{dt}(F_L(t)) = \frac{1}{R(t)} * \frac{d}{dt}(1 - R(t))$

somit: $z(t) = \frac{1}{R(K, t)} * \frac{d}{dt}(1 - R(K, t)) = \frac{1}{e^{-\lambda t}} * (\lambda e^{-\lambda t}) = \lambda.$

- 2 Bestimmung der Zeitintervalle, in denen das 2-von-3-System eine größere Überlebenswahrscheinlichkeit als eine einzelne Komponente aufweist.

Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.

- 1 Wie groß ist die Ausfallrate für eine einzelne Komponente?

allgemein: $z(t) = \frac{f_L(t)}{R(t)} = \frac{1}{R(t)} * \frac{d}{dt}(F_L(t)) = \frac{1}{R(t)} * \frac{d}{dt}(1 - R(t))$

somit: $z(t) = \frac{1}{R(K, t)} * \frac{d}{dt}(1 - R(K, t)) = \frac{1}{e^{-\lambda t}} * (\lambda e^{-\lambda t}) = \lambda.$

- 2 Bestimmung der Zeitintervalle, in denen das 2-von-3-System eine größere Überlebenswahrscheinlichkeit als eine einzelne Komponente aufweist.

Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.

- ② Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.

2-von-3-System:

$$R(S_{2v3}, t) = \sum_{k=2}^3 \binom{3}{k} R(t)^k [1 - R(t)]^{3-k} = 3 * R(t)^2 - 2 * R(t)^3$$

Einzelkomponente: $R(K, t) = R(t)$

somit gilt:

$$R(K, t) = R(S_{2v3}, t)$$

$$\Leftrightarrow R = 3 * R^2 - 2 * R^3$$

$$\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$$

Wegen $R(K, t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörigen Werte $t_2 = 0$ und $t_3 = \frac{\ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{\ln(2)}{\lambda})$.

- ② Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.
2-von-3-System:

$$R(S_{2v3}, t) = \sum_{k=2}^3 \binom{3}{k} R(t)^k [1 - R(t)]^{3-k} = 3 * R(t)^2 - 2 * R(t)^3$$

Einzelkomponente: $R(K, t) = R(t)$

somit gilt:

$$R(K, t) = R(S_{2v3}, t)$$

$$\Leftrightarrow R = 3 * R^2 - 2 * R^3$$

$$\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$$

Wegen $R(K, t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörigen Werte $t_2 = 0$ und $t_3 = \frac{\ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{\ln(2)}{\lambda})$.

- ② Gesucht: t mit $R(K, t) < R(S_{2v3}, t)$, d.h. Bestimmung der Schnittpunkte der beiden Überlebenswahrscheinlichkeiten.
2-von-3-System:

$$R(S_{2v3}, t) = \sum_{k=2}^3 \binom{3}{k} R(t)^k [1 - R(t)]^{3-k} = 3 * R(t)^2 - 2 * R(t)^3$$

Einzelkomponente: $R(K, t) = R(t)$

somit gilt:

$$R(K, t) = R(S_{2v3}, t)$$

$$\Leftrightarrow R = 3 * R^2 - 2 * R^3$$

$$\rightarrow R_1 = 0, R_2 = 1, R_3 = 0.5$$

Wegen $R(K, t) = e^{-\lambda t}$ ergeben sich für R_2 und R_3 die dazugehörigen Werte $t_2 = 0$ und $t_3 = \frac{\ln(2)}{\lambda}$, d.h. das gesuchte Intervall ist $[t_2, t_3) = [0, \frac{\ln(2)}{\lambda})$.

- 3 Bestimmen Sie λ derart, dass die mittlere Lebensdauer für das gegebene 2-von-3-System $\frac{5}{6}$ beträgt.

Es gilt:

$$MTTF = \int_0^{\infty} R(S, t) dt, \quad R(K, t) = e^{-\lambda t}$$

$$MTTF = \int_0^{\infty} R(S_{2v3}, t) dt = \frac{5}{6} \rightarrow \lambda = 1$$

Zentralübung Rechnerstrukturen im SS2011

Leistungsbewertung und Fehlertoleranz

Martin Schindewolf, Wolfgang Karl

Lehrstuhl für Rechnerarchitektur und Parallelverarbeitung

5. Mai 2011

