Zentralübung Rechnerstrukturen im SS2009

Fehlertoleranz

Dr. Rainer Buchty
buchty@ira.uka.de

Universität Karlsruhe (TH) – Forschungsuniversität
Institut für Technische Informatik (ITEC)
Lehrstuhl für Rechnerarchitektur

23.07.2009

Warum Disziplin der Rechnerarchitektur?

- **Fehlertoleranzaspekte betreffen Entwurf und Betrieb**
  - Minimierung der Entwicklungskosten bei für die Anwendung ausreichender Zuverlässigkeit
  - Maximierung der Ausfallsicherheit
- **Maßzahlen**
  - Funktionswahrscheinlichkeit φ
  - Ausfallwahrscheinlichkeit $1 - \varphi$

Was bedeutet Verfügbarkeit?

Verfügbarkeit/Jahr	Ausfallzeit/Jahr
99%	87,6h (3,65d)
99,5%	43,8h (1,825d)
99,9%	8,76h
99,99%	52,56m
99,999%	5,26m

Hochverfügbarkeit: 99,999% („Five Nines“)

Definitionen

Begriffsbildungen

- Eminent wichtig
- Ähnlich lautende Begriffe bezeichnen **nicht** unbedingt identische Dinge
- 2 Beispiele aus der Klausur vom Sommersemester 2008

Definitionen (forts.)

Beispielaufgabe (1 Punkt)

Fehler werden unterschieden nach **Dauer und Ort**. Wie lässt sich die Fehler**dauer** genauer spezifizieren? (1 Punkt)

Definitionen (forts.)

Beispielaufgabe (1 Punkt)

Fehler werden unterschieden nach **Dauer und Ort**. Wie lässt sich die Fehler**dauer** genauer spezifizieren? (1 Punkt)

- **Textanalyse:** Was ist gefragt? (Dauer, nicht Ort!)
  - Es gibt keine Extrapunkte für Beantwortung der nicht gefragten Punkte.
  - Keine „Streubombentaktik“ nach dem Motto „irgendeiner der folgenden Begriffe wird schon passen“
- **Keine Zeit verschwenden:**
  Knappe, **präzise** Antwort ist ausreichend.

Definitionen (forts.)

Beispielaufgabe (1 Punkt)

Fehler werden unterschieden nach **Dauer und Ort**. Wie lässt sich die Fehler**dauer** genauer spezifizieren? (1 Punkt)

- **Textanalyse:** Was ist gefragt? (Dauer, nicht Ort!)
  - Es gibt keine Extrapunkte für Beantwortung der nicht gefragten Punkte.
  - Keine „Streubombentaktik“ nach dem Motto „irgendeiner der folgenden Begriffe wird schon passen“
- **Keine Zeit verschwenden:**
  Knappe, **präzise** Antwort ist ausreichend.
- **Antwort:** Temporärer und permanenter Fehler.

Definitionen (forts.)

Beispielaufgabe (3 Punkte)

Das System**ausfallverhalten** lässt sich in drei Kategorien einteilen. Welche sind dies und wie sind sie zu charakterisieren?

Definitionen (forts.)

Beispielaufgabe (3 Punkte)

Das System**ausfallverhalten** lässt sich in drei Kategorien einteilen. Welche sind dies und wie sind sie zu charakterisieren?

- **Textanalyse**
  - Was ist gefragt? (Begriffsbildung!)
  - Korreliert der Beantwortungsaufwand mit der Punktzahl, d.h. erscheint die Antwort angemessen?
- **Hinterfragen der eigenen Antwort**
  - Stimmt meine Definition?
  - Beispiel hier: Ausfallverhalten != Ausfallhäufigkeit

Definitionen (forts.)

Beispielaufgabe (3 Punkte)

Das System**ausfallverhalten** lässt sich in drei Kategorien einteilen. Welche sind dies und wie sind sie zu charakterisieren?

- **Textanalyse**
  - Was ist gefragt? (Begriffsbildung!)
  - Korreliert der Beantwortungsaufwand mit der Punktzahl, d.h. erscheint die Antwort angemessen?
- **Hinterfragen der eigenen Antwort**
  - Stimmt meine Definition?
  - Beispiel hier: Ausfallverhalten != Ausfallhäufigkeit
- **Beantwortung**
  - **Fail-stop-System:** Ausfälle sind nur Anhalteausfälle
  - **Fail-silent-System:** Ausfälle sind nur Unterlassungsausfälle
  - **Fail-safe-System:** Ausfälle sind nur unkritische Ausfälle

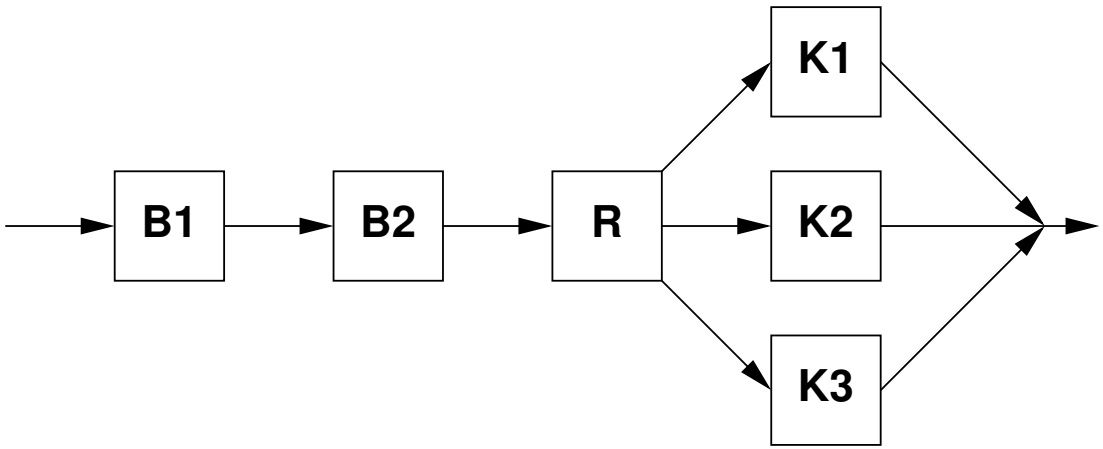
Analyse von Architekturen

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

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.

Analyse von Architekturen (forts.)

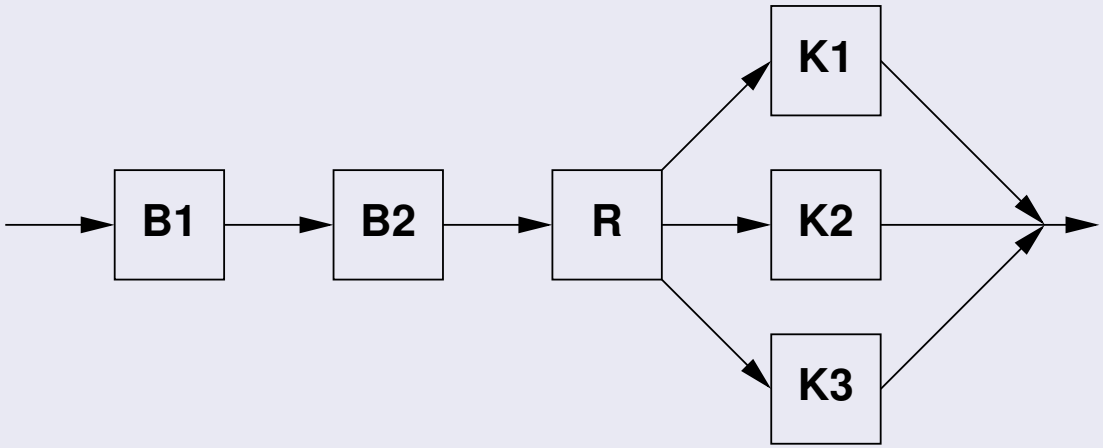
Gegeben sei ein portables Rechnersystem bestehend aus zwei Batterien B_1 und B_2, der eigentlichen Recheneinheit R und einer redundant ausgelegten Kommunikation über die Komponenten K_1 bis K_3. Zum fehlerfreien Betrieb des Systems sind beide Batterien, die Recheneneinheit und mindestens eine Kommunikationskomponente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



Zuverlässigkeitsblockdiagramm

Analyse von Architekturen (forts.)

Zuverlässigkeitsblockdiagramm



Strukturformel

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

oder

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

Analyse von Architekturen (forts.)

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

Strukturformel

- Gegegeben 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 $\varphi = \varphi(B) * \varphi(B) * \varphi(R) * ...$

- Funktionswahrscheinlichkeit eines **Parallelsystems**: $\varphi(\vee_{K \in \Lambda}) = \sum_{\emptyset \neq A \in \Lambda} (-1)^{1+\#A} * \varphi(\wedge_{K \in A} K)$

Analyse von Architekturen (forts.)

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

Redundanzsysteme

Ziel: **Erhöhung der Systemsicherheit**

- **Simpel, aber teuer:**
  Vervielfachung der kritischen Komponenten

- **Hot Standby**
  - Backup-Komponenten rechnen immer parallel, jedoch wird nur ein Ergebnis verwendet
  - Hoher Energiebedarf
  - Verschleiß der Backup-Komponenten
  - Schnelle Umschaltzeit

- **Cold Standby**
  - Backup-Komponenten werden bei Bedarf aktiviert
  - Verbrauch (fast) wie Normalsystem
  - Kein Verschleiß (außer normaler Alterung)
  - Benötigt Zeit zum Hochfahren und Umschalten

Redundanzsysteme (forts.)

- Kompromissentwurf: **Graceful Degradation**
  - System wird auf durchschnittlich benötigte Leistung hin aufgebaut mit Sicherheitszulage
  - Im Normalbetrieb: typischerweise Parallelbetrieb aller Einheiten bei gleichmäßiger Auslastung
  - Im Fehlerfall: Verteilung der verbleibenden Ressourcen
  - Im Fehlerfall weiterhin funktionsfähig, aber verringerte Gesamtleistung
- Problem: Wie **Fehlerzustand** erkennen?
- **Mehrfachsysteme mit Mehrheitsentscheider**
  - Mehrfache, unabhängige Durchführung von Systemaufgaben
  - Definition des korrekten Zustands über Mehrheitsentscheider
  - Identifikation fehlerhafter Berechnung bzw. von Fehlerzuständen
  - Klassische Ausführung:
    Triple Modular Redundancy (TMR) *Tres faciunt collegium.*

Redundanzsysteme (forts.)

- **Alltagsimplementationen**
  - Batterien (mehr Spannung/Strom als zum Betrieb benötigt)
  - RAID-Systeme
    - Mirroring (RAID-1): hot standby
    - Parität (RAID3-6): Entscheider-System
    - Erhöhung der Ausfallsicherheit

RAID	Anzahl Festpl.	Netto-kapazität	Ausfall-sicherheit
0	≥ 2	n	0
1	$\geq 2 * 1$	$\frac{n}{2}$	$\frac{n}{2}$
2	10	$\frac{8n}{10}$	2
3	≥ 2	n-1	1
4	≥ 2	n-1	1
5	≥ 3	n-1	1
6	≥ 4	n-2	2
DP	≥ 3	n-2	2

Redundanzsysteme (forts.)

- **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

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

Redundanzsysteme (forts.)

- **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)

Beispielaufgabe

Ein RAID2-System besteht per Definition aus 10 Festplattenspeichern. Hiervon dürfen zwei ausfallen, ohne dass es zu Datenverlust kommt. Unter der Annahme, die Verfügbarkeit 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:

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

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

Weitere Taxonomien

- Mean Time to Failure (**MTTF**): mittlere Funktionszeit
- Mean Time to Repair (**MTTR**): mittlere Reparaturzeit
- 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:

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

Beispielaufgabe

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.

Beispielaufgabe

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.

- $MTTF = 2a = (2 * 365 * 24 * 60)min = 1051200min$
- $MTTR = (2 + 10 + 2)min = 14min$
- $V = \frac{MTTF}{MTTF+MTTR} = \frac{1051200}{1051214} = 0,999997$

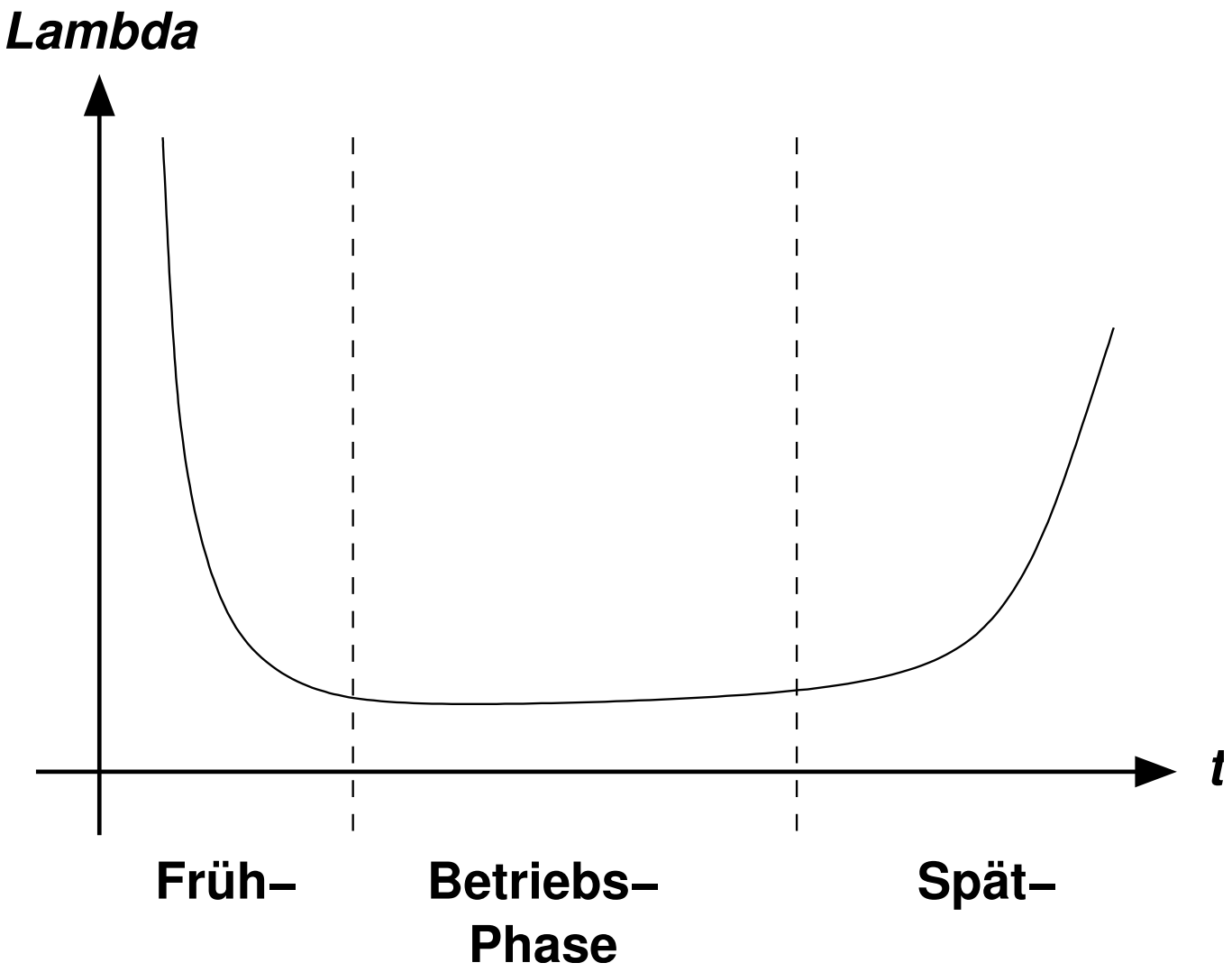
Alterungseffekte

- **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

Alterungseffekte



Badewannenkurve

Rechnen mit zeitlich variabler Ausfallrate

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.

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

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

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

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

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

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

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.

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

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

   Einzelkomponente: $\mathbf{R(K,t) = R(t)}$

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

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

Einzelkomponente: $\mathbf{R(K,t) = R(t)}$

somit gilt:
$\mathbf{R(K,t) = R(S_{2v3},t)}$
$\leftrightarrow \mathbf{R = 3 * R^2 - 2 * R^3}$
$\rightarrow R_1 = 0, R_2 = 0.5, R_3 = 1$

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

Rechnen mit zeitlich variabler Ausfallrate (forts.)

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

   Es gilt:

$$\mathbf{MTTF} = \int_{\mathbf{0}}^{\infty} \mathbf{R(S,t)dt}, \quad \lambda = \frac{1}{MTTF}, \quad R(K,t) = e^{-\lambda t}$$

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

Weiterführende Veranstaltungen im WS2009/10

Vorlesungen

- **Heterogene parallele Rechensysteme**
  - Parallele Programmierkonzepte
  - Architekturen
  - Rekonfigurierbare und Selbstkonfigurierende Systeme
- **Mikroprozessoren II**
  - Vertiefung des Themengebietes von MP I
  - Verbindungstechniken
  - Prozessorarchitektur
  - Techniken zur Parallelverarbeitung

Weiterführende Veranstaltungen im WS2009/10

Praktika und Seminare

- **Praktikum „Multicore-Programmierung“**
  - Anwendung und Vertiefen paralleler Programmiertechniken (OpenMP und MPI)
  - Leistungsanalyse und Optimierung
- **Praktikum „Multicore-Technologien“**
  - Aufbau von Multicore-Systemen in Hardware unter Verwendung aktueller FPGA-Technologie und FPGA-Entwurfswerkzeuge (Xilinx ISE und XPS)
  - Implementierung und Programmierung von Beispielanwendungen
- **Seminare**
  - Transactional Memory
  - Virtualisierung

Anmeldung über `http://ca.itec.uka.de/anmeldung/`

Studien- und Diplomarbeiten: Ansprechpartner

Anwendungsbeschleuniger

Fabian Nowak (nowak@ira.uka.de)

- Anwendungsbeschleuniger für numerische Verfahren
- Programmphasenerkennung, -vorhersage und -interpretation
- Entwicklungsumgebung für die parallele Programmierung heterogener, dynamisch veränderlicher Knoten
- Partiell-dynamische Hardwarerekonfiguration

Abstrahierung und Laufzeitsysteme

Mario Kicherer (kicherer@ira.uka.de)

- Laufzeitsysteme für heterogene, veränderliche Systeme
- Einheitliche Softwarebeschreibung durch Hardwareabstraktion
- Kontrolle der Zuweisung von Anwendungsteilen zu Ausführungseinheiten
- Kompatibilitätsaspekte

Studien- und Diplomarbeiten: Ansprechpartner (forts.)

Self-aware Memory

Oliver Mattes (mattes@ira.uka.de)

- Kommunikation und Synchronisation
- Reorganisation, Kohärenz und Konsistenz
- Systemsimulation und HW-Prototypen
- Benchmarking

Transactional Memory / Programmierung

Martin Schindewolf (schindew@ira.uka.de)

- Portierung einer Umbegung für Thread-level Speculation auf Software Transactional Memory
- Entwurf und Implementierung einer DLO-Umgebung für OpenMP in GCC
- Statische Erkennung spezieller Zugriffsmuster in GCC
- Modellierung und Modellbildung von Knoten und Topologien

Studien- und Diplomarbeiten: Ansprechpartner (forts.)

Systembewertung und -optimierung

David Kramer (kramer@ira.uka.de)

- Weiterentwicklung der vorhandenen Werkzeuge und Eclipse-Integration
- Bewertungsmetriken
- Interpretationsverfahren
- Echtzeitevaluation und Trenderkennung (Proaktivität)
- Bio-inspirierte Verfahren zur Systemverwaltung (Self-X)

Studien- und Diplomarbeiten: Ansprechpartner (forts.)

Systembewertung und -optimierung

David Kramer (kramer@ira.uka.de)

- Weiterentwicklung der vorhandenen Werkzeuge und Eclipse-Integration
- Bewertungsmetriken
- Interpretationsverfahren
- Echtzeitevaluation und Trenderkennung (Proaktivität)
- Bio-inspirierte Verfahren zur Systemverwaltung (Self-X)

Weitere Informationen:

- **Direktes Anschreiben** der jeweiligen Betreuer
- **Übersicht über ausgeschriebene Arbeiten** auf `http://ca.itec.uka.de/diploma/`
- **Abonnieren unseres Newsletters** via `https://www.lists.uni-karlsruhe.de/sympa/info/studinfo-karl`

Zur Klausur

- **11. August 2009, 14 Uhr**
- Verteilt auf bis zu 4 Hörsäle
  - Audimax A/B (30.95)
  - HSaF (50.35)
  - Neue Chemie (30.46)
  - Architektur HS37 (20.40)
- Hörsaaleinteilung wird nach

  Anmeldeschluss (02. August 2009)

  bekanntgegeben
- Sitzplatzeinteilung direkt am Klausurtermin per Aushang
- **Anmeldung über QISPOS**, ebenso Abmeldung
  - **Keine Nachmeldung** bei Versäumnis der Anmeldefrist
  - Elektronische **Ab**meldung bis 09. August 2009

Zur Klausur (forts.)

- Dauer der Klausur: **60 Minuten** (reine Bearbeitungszeit)
- Klausur besteht aus **6 Aufgaben** zu thematischen Schwerpunkten **aus Vorlesung und Übung**
  - Klausuren-Archiv auf RS-Website
  - Bei Vorbereitung: Fragen zu den einzelnen Aufgaben bitte an die jeweiligen Tutoren
- 60 Punkte erzielbar, **20 Punkte zum Bestehen** notwendig
- Faustregel: pro Punkt eine Minute Bearbeitungszeit
- **Keine Hilfsmittel zugelassen** (auch keine Wörterbücher)

Zur Klausur (forts.)

- Dauer der Klausur: **60 Minuten** (reine Bearbeitungszeit)
- Klausur besteht aus **6 Aufgaben** zu thematischen Schwerpunkten **aus Vorlesung und Übung**
  - Klausuren-Archiv auf RS-Website
  - Bei Vorbereitung: Fragen zu den einzelnen Aufgaben bitte an die jeweiligen Tutoren
- 60 Punkte erzielbar, **20 Punkte zum Bestehen** notwendig
- Faustregel: pro Punkt eine Minute Bearbeitungszeit
- **Keine Hilfsmittel zugelassen** (auch keine Wörterbücher)

Don’t panic!