Universität Stuttgart

RSS Reliable Software Systems

Übungsblatt 2

Sichere und zuverlässige Softwaresysteme (WiSe 2018/2019)

Abgabe: Fr. 16.11.2018, 23:55 Uhr — Besprechung: Montag, 19.11.2018

- Bitte lösen Sie die Übungsaufgabe in **Gruppen von 4 Studenten** und wählen EINEN Studenten aus, welcher die Lösung im ILIAS (Ordner `Gruppe für Teilnahme am Übungsbetrieb (inkl. Abgaben)/Abgaben/Übungsblatt 2/`) als **Gruppenabgabe** (unter Angabe aller Gruppenmitglieder) einstellt.
- Für schriftliche Aufgaben erstellen Sie EINE **PDF-Datei**, die Projekte für Programmieraufgaben fügen Sie als **ZIP-Datei** hinzu. Geben Sie beide Dateien ab und verzichten Sie darauf, diese in einer weiteren ZIP-Datei zusammenzuführen. Geben Sie bitte nicht nur den Sourcecode, sondern stets das komplette Projekt ab!
- Erstellen Sie ein **Titelblatt**, welches die Namen der Studenten, die Matrikelnummern, und die E-Mail-Adressen enthält. Im Quellcode fügen Sie diese Informationen bitte als Kommentar hinzu.
- Benennen Sie die Dateien nach dem folgenden Schema:
SZS[**Blattnummer**]**-**[**Nachnamen der Teammitglieder, alphabetisch**]**.**[**pdf oder zip**].

Aufgabe 1 Standby-Systemkonfiguration

Die Standby-Systemkonfiguration ist eine typische Strategie für dynamische Redundanz zur Erhöhung der Fehlertoleranz eines Systems. Es lassen sich drei Typen unterscheiden:

- *Cold-Standby* — In diesen Fall wird davon ausgegangen, dass ein Ersatzknoten nicht ausfällt.
- *Warm-Standby* — In einigen Situationen wird der Ersatzknoten nicht aktiv verwendet, und es ist sinnvoll anzunehmen, dass die Ausfallrate des Ersatzknotens niedriger ist als die des aktiven Knotens
- *Hot-Standby* — Wenn der Ersatzknoten mit der gleichen Ausfallrate wie der aktive Knoten ausfällt, wird er in einem Hot-Standby-Zustand betrachtet

Um den Unterschied zwischen den Ausfallraten von C*old-, Warm- und Hot-Standby-Zuständen* zu verdeutlichen, verwenden wir einen Parameter (bzw. Faktor) $\alpha = [0, 1]$. Im Falle von *Cold-Standby* gilt $\alpha = 0$, was vernünftig ist, weil die Ausfallwahrscheinlichkeit 0 ist. Im Falle von *Warm-Standby* gilt $0 < \alpha < 1$ — was eine reduzierte Ausfallrate bedeutet. Im Falle von *Hot-Standby* gilt $\alpha = 1$, was bedeutet, dass die Ausfallrate der Standby-Knoten gleich ist.

Angenommen, wir haben einen Hauptknoten A und einen warmen Standby-Knoten B. Im Falle eines Ausfalls des Knotens A werden alle Operationen auf den Knoten B umgeleitet (die Umleitung erfolgt automatisch und ohne Fehler). Angesichts von $\alpha = 0,6$ sind wir daran interessiert, die Zuverlässigkeit und MTTF der warmen Standby-Konfiguration von zwei Knoten mit folgenden Metriken unter Verwendung des gegebenen CTMC-Modells zu berechnen.

- Die Time to Failure (TTF) des Knotens A ist exponentiell verteilt. Hierin gehen wir von einer Ausfallrate von 0,3 pro Stunde aus, d.h. Ausfallrate(A) $= 0,3$.
- Die Time to Failure (TTF) des Knotens B ist exponentiell verteilt. Hierin gehen wir von einer Ausfallrate von 0,2 pro Stunde aus, d.h. , d.h. Ausfallrate(B) $= 0,2$.
- Knoten B weist eine reduzierte Ausfallrate von $\alpha \times$ Ausfallrate(B) auf.

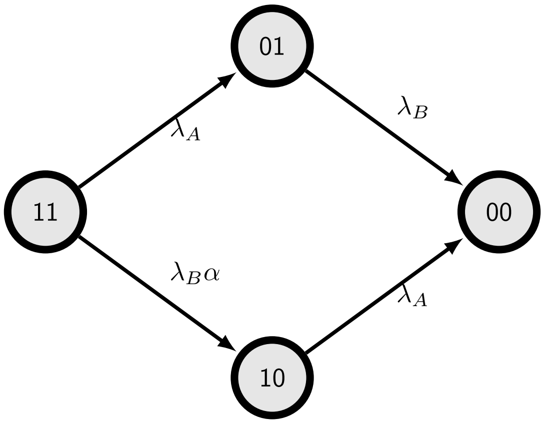

Abbildung 1: CTMC für die Warm-Standby-Konfiguration

Abbildung 1 zeigt ein Markov-Modell (CTMC), das das Systemverhalten für die Warm-Standby-Konfiguration modelliert.

- Zustand (s_1) 11 bedeutet, dass sowohl Knoten A als auch B funktionieren.
- Zustand (s_2) 01 bedeutet, dass Knoten A ausgefallen ist, aber Knoten B funktioniert.
- Zustand (s_3) 10 bedeutet, dass der Knoten A funktioniert, aber der Knoten B ausgefallen ist.
- Zustand (s_4) 00 bedeutet, dass der Knoten A und B beide fehlgeschlagen sind.

Die numerische Lösung zur Suche nach MTTF ist wie folgt:

$$\int_0^\infty R(t)dx = \frac{\lambda_A\lambda_B + \lambda_A^2 + \alpha\lambda_B^2}{\lambda_A\lambda_B(\lambda_A + \alpha\lambda_B)}$$

(a) Bestimmen Sie die Transitionsratenmatrix R

(b) Modellieren Sie die CTMC in PRISM (werfen Sie einen Blick auf das PRISM-Tutorial, um die PRISM-Syntax zu lernen: `http://www.prismmodelchecker.org/tutorial/die.php`).

(c) Wie hoch ist die Wahrscheinlichkeit, dass unsere redundante Konfiguration nach genau 30 Stunden ausgefallen ist?

(d) Ermitteln Sie die MTTF dieser Systemkonfiguration numerisch über die angegebene Formel.

(e) Erstellen Sie eine neue `property`, indem Sie die festen 30 Zeiteinheiten in einen Zeitparameter T ändern. Erstellen Sie ein Experiment für Ihre Eigenschaft im Zeitrahmen von 0 bis 30 Stunden. Können Sie anhand der Grafik erkennen, wie viele Stunden es dauert, bis die CTMC einen stationären Zustand erreicht?

(f) Erstellen Sie eine neue `property` und finden Sie die erwartete Zeitspanne, die vergeht, bis der erste Systemausfall eintritt. Vergleichen Sie den Wert von MTTF, den Sie numerisch gefunden haben, mit dem Wert, den PRISM gelöst hat.

Aufgabe 2 Systemredundanz vs. Komponentenredundanz

Hier untersuchen wir die Wirksamkeit der Systemkonfiguration in Bezug auf die Systemzuverlässigkeit. Betrachten wir ein System, das aus zwei Serienkomponenten A und B besteht. Das Zuverlässigkeitsblockdiagramm (RBD) ist in Abbildung 2 gegeben.

Wenn wir uns entscheiden, die Zuverlässigkeit des Systems durch Redundanz mit einer einzigen Kopie zu verbessern, sind zwei Lösungen möglich. In Abbildung 3 wird die komplette Sequenz repliziert, was *Systemredundanz* bedeutet; in Abbildung 4 wird jede Komponente einzeln repliziert, was *Komponentenredundanz* bedeutet.

Abbildung 2: RBD für die serielle Komposition

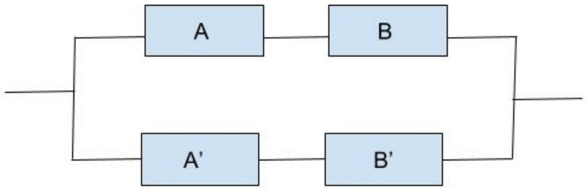

Abbildung 3: RBD für die Systemredundanz

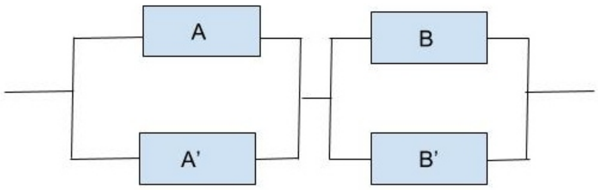

Abbildung 4: RBD für die Komponentenredundanz

(a) Berechnen Sie die Zuverlässigkeit der Konfiguration aus den Abbildungen 2, 3 und 4 (verwenden Sie Symbole von $R_A, R_B, R_{A'}, R_{B'}$ anstelle von Istwerten).

(b) Vergleichen Sie die Zuverlässigkeit der Systemredundanz und der Komponentenredundanz. Welche ist zuverlässiger?

Aufgabe 3 Duane-Modell

Für ein Softwaresystem ist bekannt, dass nach 20 Tagen bereits insgesamt 7 Fehler und nach 100 Tagen insgesamt 25 Fehler registriert wurden. Gehen Sie davon aus, dass für dieses System das Duane-Modell anwendbar ist.

(a) Eine Gerade kann durch eine Geradengleichung der Form $y(x) = m \cdot x + c$ beschrieben werden. In der Vorlesung haben Sie erfahren, dass bei logarithmischer Achsenskalierung (jeweils Logarithmus zur Basis 10) die kumulative MTBF ($MTBF_c(t) = \kappa \cdot t^{\alpha}$) beim Duane-Modell eine Gerade darstellt. Stellen Sie die Verbindung der Parameter der Geradengleichung mit denen des Duane-Modells her. Runden Sie Zwischen- und Endergebnisse auf jeweils zwei Dezimalstellen.

(b) Berechnen Sie die Parameter α und κ des Duane-Modells für das gegebene System.

(c) Berechnen Sie die kumulative $MTBF_c$ nach 200 Tagen und die derzeitige Ausfallrate λ_i für das System nach 100 Tagen.

Aufgabe 4 Bewertung von Anomalie-Erkennungs-Algorithmen

Ein Datensatz D enthält 100 Datenpunkte und hat eine Teilmenge von fünf Datenpunkten ($\{A, B, C, D, E\}$), von denen bekannt ist, dass diese anormal sind. Zwei Anomalie-Erkennungs-Algorithmen X und Y werden angewendet. Die beiden Algorithmen weisen jedem Datenpunkt einen *Anomaliewert* im Bereich von 1 bis 100 zu. Je niedriger der *Anomaliewert*, desto höher ist die Wahrscheinlichkeit, dass ein Datenpunkt eine Anomalie ist. Ein Schwellwert t dient dazu, den zugewiesenen *Anomaliewert* in ein *Anomalie-Label* zu konvertieren, d.h. wenn ein *Anomaliewert* unter einem bestimmten Schwellwert liegt, wird er als Anomalie gemeldet.

Die folgende Tabelle zeigt die Ergebnisse, die die beiden Algorithmen X und Y den 5 anomalen Datenpunkten zugeordnet haben. Darüber hinaus zeigt die Tabelle das ideale Ranking der anomalen Datenpunkte:

	$\{A, B, C, D, E\}$
Algorithm X	1, 5, 9, 16, 20
Algorithm Y	4, 7, 12, 13, 15
Ideal Algorithm	1, 2, 3, 4, 5

(a) Für jeden Algorithmus X und Y zeichnen und vervollständigen Sie bitte die Wahrheitsmatrix (Confusion Table).

(b) Zeichnen Sie die ROC-Kurven für den Algorithmus X und Y mit Hilfe von R. Welcher ist besser?

(c) Was ist der Trade-Off bei der Definition des Schwellwertes t für die jeweiligen Algorithmen?

(d) Bitte zeichnen Sie die ROC-Kurve eines Algorithmus, der dem angegebenen Datensatz zufällig einen *Anomaliewert* zuordnet.

Hinweise zu den Übungen:

- Durch die Teilnahme am Übungsbetrieb können Sie sich bis zu **3 Bonuspunkte für die Klausur** verdienen.
- Bedingungen:
  - Während des Semesters darf maximal eine Abgabe im ILIAS ausgelassen werden.
  - Während des Semesters darf pro Person max. eine Hörsaalübung ausgelassen werden.
  - Jede Gruppe muss im Laufe des Semesters eine Aufgabe in der Hörsaalübung präsentieren.
  - Jede Präsentation wird mit folgender Skala bewertet:
    * 3 Punkte: Korrekt
    * 2 Punkt: Sinnvoll aber fehlerhaft
    * 1 Punkte: Sinnlos/falsch oder fehlt (Basispunkt für Präsentation)

Viel Erfolg!