entwickler.press
shortcuts

Resilience

Wie Netflix sein System schützt

Uwe Friedrichsen, Stefan Toth,
Eberhard Wolff

Uwe Friedrichsen, Stefan Toth, Eberhard Wolff

Resilience

Wie Netflix sein System schützt

ISBN: 978-3-86802-561-3

Ein Imprint der Software & Support Media GmbH

1 Eine kurze Einführung in Resilient Software Design

In letzter Zeit hört man immer häufiger den Begriff „Resilience“ und manchmal auch etwas vollständiger „Resilient Software Design“. Irgendwie scheint es mit dem Umgang mit Fehlern zur Laufzeit zusammenzuhängen. Aber was daran ist so neu und anders, dass man dafür einen neuen Begriff prägen muss? Oder hat man wieder nur alten Wein in neue Schläuche gefüllt? Zeit für eine kurze Einführung: Worum geht es? Was ist anders? Und wie fühlt es sich an?

Wo fängt man am besten an, wenn man eine kurze Einführung in Resilient Software Design schreiben will? Wahrscheinlich am besten beim Wert von Software: Der primäre Zweck aller Geschäftsprozesse und der sie implementierenden und unterstützenden IT-Systeme ist es, Geld zu erwirtschaften bzw. Kundenbedürfnisse zu befriedigen – am besten beides. Das funktioniert aber nur, solange die IT-Systeme zuverlässig und – von außen betrachtet – fehlerfrei laufen.

Sind die Systeme nicht verfügbar oder fehlerhaft, sind die Kunden unzufrieden, und man verdient kein Geld mit ihnen – kurzum: Sie sind wertlos. Nur zuverlässig laufende Systeme haben Wert. Die Verfügbarkeit von Systemen in Produktion ist also essenziell für den Wert der Software. (Bei manchen Systemen wie z. B. eingebetteten Systemen geht es häufig sogar um viel mehr als nur einen monetären Wert, aber das wollen wir an dieser Stelle nicht näher betrachten.)

Verfügbarkeit und Fehlertypen

Was aber ist *Verfügbarkeit* genau? Die Verfügbarkeit *A* (für *Availability*, den englischen Begriff für Verfügbarkeit) ist definiert als [1] $A := MTTF / (MTTF + MTTR)$. Darin bedeuten:

- *MTTF (Mean Time To Failure):* die durchschnittliche Zeit vom Beginn des ordnungsgemäßen Betriebs eines Systems bis zum Auftreten eines Fehlers
- *MTTR (Mean Time To Recovery):* die durchschnittliche Zeit vom Auftreten eines Fehlers bis zur Wiederherstellung des ordnungsgemäßen Betriebs des Systems

Während der Nenner also die gesamte Zeit beschreibt, beschreibt der Zähler den Teil der Zeit, in dem das System ordnungsgemäß funktioniert. Damit kann die Verfügbarkeit Werte zwischen 0 für „gar nicht verfügbar“ und 1 für „immer verfügbar“ annehmen. Wenn man den Wert mit 100 multipliziert, erhält man die vertrautere Darstellung als Prozentwert.

Bevor wir uns der Frage zuwenden, wie man die Verfügbarkeit maximieren kann, ist es sinnvoll, einen kurzen Blick auf die möglichen Arten von Fehlern zu werfen, die die Verfügbarkeit kompromittieren können. Die eingängige Literatur (siehe z. B. [2]) unterscheidet fünf Fehlertypen:

1. *Crash Failure* (Absturzfehler): Ein System antwortet permanent nicht mehr, hat bis zum Zeitpunkt des Ausfalls aber korrekt gearbeitet.
2. *Omission Failure* (Auslassungsfehler): Ein System reagiert auf (einzelne) Anfragen nicht, sei es, dass es die Anfragen nicht erhält oder keine Antwort sendet.
3. *Timing Failure* (Antwortzeitfehler): Die Antwortzeit eines Systems liegt außerhalb eines festgelegten Zeitintervalls.
4. *Response Failure* (Antwortfehler): Die Antwort, die ein System gibt, ist falsch.
5. *Byzantine Failure* (byzantinischer/zufälliger Fehler): Ein System gibt zu zufälligen Zeiten zufällige Antworten („es läuft Amok“).