

Resilienz und Fehlertoleranz in verteilten Systemen

Modul „Software Engineering“

14. Januar 2025
Derhachov, Schmidt, Westholt

HTWK Leipzig, FIM

Gliederung

- 1 Resilienz und Fehlertoleranz
- 2 Strategien
- 3 Pattern und Konzepte
- 4 Fazit
- 5 Diskussion
- 6 Quellen

Resilienz und Fehlertoleranz

Begriffsklärung

Begriffe

- Resilienz
 - ▶ Fähigkeit eines Systems, trotz Störungen, Angriffen oder Ausfällen funktionsfähig zu bleiben oder sich schnell davon zu erholen
- Fehlertoleranz
 - ▶ Fähigkeit eines Systems, trotz vorhandener Fehler oder Störungen korrekt zu funktionieren

Resilienz und Fehlertoleranz

Motivation

Motivation

- Schutz vor Ausfällen in geschäftskritischen Anwendungen
- Reduktion betrieblicher Kosten und Steigerung der Nutzerzufriedenheit
- Anforderungen an Verfügbarkeit in regulierten Branchen

Resilienz und Fehlertoleranz

Motivation

- Verbreitung von Microservices und Cloud-Technologien
- Herausforderungen durch Kommunikationsausfälle und Spitzenlasten
- Bedarf an innovativen Resilienzmustern

Strategien

Resilienzstrategien

- Redundanz
- Partitionierung
- Skalierung

Strategien

Fehlertoleranzstrategien

- Fehlerbehandlung und -isolierung
- Nutzung von Fallback-Mechanismen
- Vermeidung kaskadierender Fehler

Pattern und Konzepte

Circuit-Breaker

- Isoliert fehlerhafte Dienste
- Unterbricht Anfragen bei wiederholtem Fehler.
- Verhindert kaskadierende Ausfälle

Pattern und Konzepte

Circuit-Breaker: Zustandsdiagramm

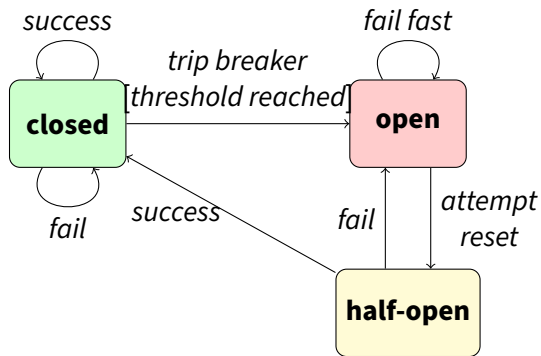


Abbildung 1: Circuit Breaker Zustandsdiagramm

Pattern und Konzepte

Circuit-Breaker: Vorteile

- Verhindert kaskadierende Ausfälle in verteilten Systemen.
- Verbesserte Systemstabilität durch Isolierung fehlerhafter Dienste.
- Bessere Benutzererfahrung durch Fallback-Mechanismen.
- Unterstützt Resilienz und Wiederherstellung in kritischen Systemen.

Pattern und Konzepte

Circuit-Breaker: Nachteile

- Erhöhte Komplexität in der Implementierung und Wartung.
- Risiko von Fehlkonfiguration (z. B. falsche Schwellenwerte).
- Zusätzlicher Overhead durch Überwachung und Statusverwaltung.
- Fallback-Daten können veraltet oder ungenau sein.

Pattern und Konzepte

Retry-Muster

- Automatisches Wiederholen fehlgeschlagener Operationen
- Nutzung von Exponential Backoff
- Verbesserung der Resilienz bei temporären Fehlern

Pattern und Konzepte

Retry-Muster: Sequenzdiagramm

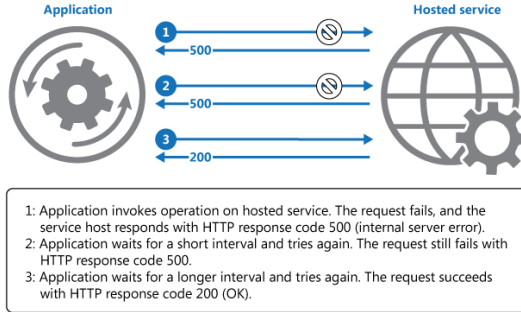


Abbildung 2: Sequenzdiagramm des Retry Patterns

Pattern und Konzepte

Retry-Muster: Vorteile

- Reduziert die Wahrscheinlichkeit eines vollständigen Anwendungsabsturzes bei vorübergehenden Fehlern.
- Verbessert die Zuverlässigkeit, indem kurzfristige Probleme (z. B. Netzwerkprobleme) automatisch überwunden werden.
- Ermöglicht ein einheitliches Fehlerbehandlungsmodell in einer Anwendung.

Pattern und Konzepte

Retry-Muster: Nachteile

- Erhöhte Komplexität in der Implementierung und Wartung.
- Verzögert die Gesamtverarbeitung, wenn ein Vorgang wiederholt fehlschlägt.
- Kann echte, dauerhafte Fehler verschleiern, wenn nur wiederholt wird, ohne die Ursache zu analysieren.
- Nicht jeder Fehler ist vorübergehend (z. B. Authentifizierungsfehler), was zu unnötigen Wiederholungen führt.

Pattern und Konzepte

Kombination mit Circuit-Breaker

- Stoppt Wiederholungen bei permanenten Fehlern
- Ermöglicht Systemen, sich zu erholen
- Optimiert Ressourcennutzung

Pattern und Konzepte

Load Balancing

- Verteilung der Last auf mehrere Server
- Strategien: Round Robin, Least Connection, Resource Based
- Verbesserung von Leistung und Ausfallsicherheit

Pattern und Konzepte

Load Balancer: Architekturdiagramm

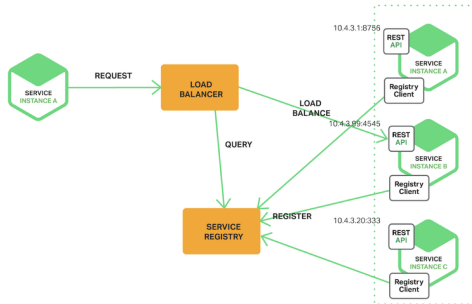


Abbildung 3: Architekturdiagramm mit einem zentralen Load Balancer

Pattern und Konzepte

Health Checks bei Load Balancing

- Überwachung der Zustände von Diensten
- Vermeidung von Überlastungen
- Umgang mit fehlerhaften Knoten

Fazit

Was wurde gemacht?

- Analyse und Implementierung von Resilienzstrategien
- Fallstudien und Praxisbeispiele
- Implementierungen in Python

Fazit

Fallstudie: Netflix

- Nutzung von Hystrix für Circuit-Breaker
- Dynamisches Load Balancing
- Skalierung und Fehlertoleranz

Diskussion

- Erweiterung der Strategien auf andere Szenarien
- Entwicklung neuer Muster zur Resilienzsteigerung
- Untersuchung ökonomischer Auswirkungen

Quellen

