

Replikation

Motivation

Grundlagen

Aktive Replikation

Passive Replikation



- Zielsetzungen
 - Tolerierung **permanenter Server-Ausfälle**
 - Hohe Verfügbarkeit von Diensten

- Replikation der Server-Seite
 - Gruppe von Replikaten statt einzelner Server
 - Replikatgruppengröße abhängig von der Anzahl zu tolerierender Ausfälle
 - Problem: **Redundante Auslegung zustandsbehafteter Dienste**
 - Oftmals gewünschte Eigenschaft: *Starke Konsistenz*
 - Zustandsänderung eines Clients ist nach ihrer Bestätigung für alle sichtbar
 - Replikate vollziehen kausal abhängige Änderungen in derselben Reihenfolge

- Herausforderungen
 - Wie interagiert ein Client mit einer Replikatgruppe?
 - Wie kann Fehlerunabhängigkeit zwischen Replikaten erreicht werden?
 - Wie lassen sich die fehlerfreien Replikate einer Gruppe konsistent halten?



- Zugriff auf replizierten Dienst per **Gruppenreferenz**
 - Kollektion von Replikadressen
 - Aktualisierung bei Änderung der Replikatgruppenszusammensetzung
- Kommunikation mit Replikatgruppe (Varianten)
 - Nutzung eines **Kontaktreplikats**
 - Auswahlmechanismus abhängig von Replikationsarchitektur und -ansatz
 - *Failover* bei (vermutetem) Ausfall des bisherigen Kontaktreplikats
 - * Kontaktierung eines anderen Replikats
 - * Replikatwechsel lässt sich für die Client-Anwendung transparent gestalten
 - Interaktion mit mehreren / allen Replikaten
 - Paralleles Senden derselben **Anfragen an verschiedene Replikate**
 - Reaktion bei eintreffenden Antworten (Alternativen)
 - * Verwendung der schnellsten Antwort
 - * Verifizierung des Ergebnisses durch Vergleich von Antworten
 - Vergleichskriterien
 - Kommunikationsaufwand
 - Komplexität bei der Behandlung von Replikatausfällen



- Austausch von Replikaten
 - **Rekonfigurierung der Replikatgruppe** erforderlich
 - Initialisierung des neuen Replikats mittels Zustandstransfer
- **Fehlerunabhängige Replikate**
 - Ausfälle verschiedener Replikate dürfen nicht dieselbe Ursache haben
 - Techniken zur Reduzierung des Grads der Fehlerabhängigkeit
 - Platzierung von Replikaten in verschiedenen *Fehlerdomänen*
 - * Unterschiedliche Stromansbindungen
 - * Redundante Netzwerkverbindungen
 - Verteilung über mehrere geografische Standorte (*Georeplikation*)
 - Einsatz **heterogener Replikatimplementierungen** (*N-Version Programming*)

■ Literatur



Liming Chen, Algirdas Avižienis

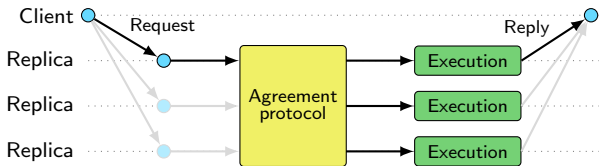
N-version programming: A fault-tolerance approach to reliability of software operation

Proceedings of 8th International Symposium on Fault-Tolerant Computing (FTCS-8), S. 3–9, 1978.



■ Grundprinzip

- **Bearbeitung aller Anfragen durch alle Replikate**
- Erstellung einer totalen Ordnung auf den Anfragen per *Einigungsprotokoll*



■ Charakteristika

- Hoher Ressourcenverbrauch zur Wahrung der Replikatkonsistenz
- Geringer Einfluss von Replikatausfällen auf die **Verfügbarkeit des Diensts**

■ Literatur



Fred B. Schneider

**Implementing fault-tolerant services using
the state machine approach: A tutorial**

ACM Computer Survey, 22(4):299–319, 1990.



■ Einigungsprotokoll

[Nähere Details in der nächsten Vorlesung.]

- **Totale Ordnung von Anfragen aller Clients**
- Zuverlässige Übertragung von Anfragen
- Uniforme Einigung

Wird eine Anfrage auf irgendeinem Replikat zugestellt, muss sie letztendlich auf allen fehlerfreien Replikaten zugestellt werden

→ *Totalgeordneter, zuverlässiger, uniformer Multicast* erforderlich

■ Deterministische Replikate

- Implementierung einer **deterministischen Zustandsmaschine**

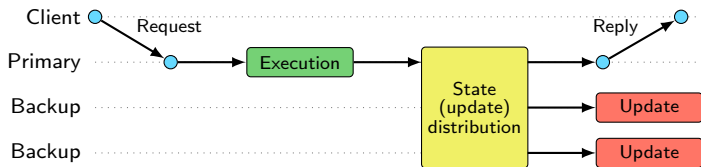
Ausgehend vom selben Zustand führt die Ausführung derselben Anfragen in derselben Reihenfolge zu denselben Antworten und Zustandsänderungen

- Quellen von Nichtdeterminismus (Beispiele)
 - Zufallszahlen
 - Zeit
 - Nebenläufigkeit
 - Externe Aufrufe



■ Grundprinzip

- **Bearbeitung aller Anfragen durch ein Primärreplik**
- Bereitstellung zusätzlicher Replikate zur Behandlung von Ausfällen



■ Vergleich zur aktiven Replikation

- **Niedrigerer Ressourcenverbrauch** im fehlerfreien Fall
- Komplexere Fehlerbehandlung bei Ausfall des Primärreplikats

■ Literatur



Navin Budhiraja, Keith Marzullo, Fred B. Schneider, Sam Toueg
The primary-backup approach
Distributed Systems (2nd Edition), Addison-Wesley, S. 199–216, 1993.



- Bei **Zustandsverteilung übertragene Informationen** (Alternativen)
 - Sicherungspunkt
 - Zustandsänderungen
 - Sicherungspunkt + neueste Zustandsänderungen
 - Zeitpunkt der Zustandsaktualisierung (Alternativen)
 - In regelmäßigen Intervallen (*Warm passive replication*)
 - Bei Ausfall des Primärreplikats (*Cold passive replication*)
- Dauer der **Ausfallbehandlung ist abhängig vom gewählten Ansatz**
- Realisierung starker Konsistenz
 - Problem
 - Primärreplikant ist den anderen Replikanten im Allgemeinen voraus
 - Vorsprung darf für Clients nicht unmittelbar sichtbar werden
 - **Sendezeitpunkt der Antwort ist entscheidend** (Beispiele)
 - Ausführung ist durch einen Sicherungspunkt abgedeckt
 - Zustandsänderung wurde per Einigungsprotokoll an andere Replikate verteilt
 - Anfrage ist Bestandteil einer persistenten Log-Datei
- [Je nach Ansatz können auch bei passiver Replikation deterministische Replikate erforderlich sein.]

