Aspekte der Hochverfügbarkeit in Servern für Telekommunikationsdienste

Dr. Thomas Wolff

TFH Berlin, 19. Nov. 2004

Übersicht

Anforderungen
Was ist Verfügbarkeit?
Ziele
Systemarchitektur
Verfügbarkeitsstrategie
Softwareaspekte

Verfügbarkeitsanforderungen

verschiedener Applikationen

Applikationsbereich	Availability	Down Time p.a.
Personal computer	90.0%	36 d 12 h
Kleinbetriebe	99.0%	87 h 36 m
ISPs, Großbetriebe	99.9%	8 h 46 m
Daten-Center	99.99%	52 m 33 s
Carrier-grade Tel., Med., Bank	99.999%	5 m 15 s
Militär, CG Ziel	99.9999%	31.5 s

Kostenaspekte

- **Kosten wachsen exponentiell für jede 9**
- **Bedarf am Use Case bemessen (z.B. Notruf)**

Diese Folie stellt anhand postulierter Verfügbarkeitszahlen dar, welche Bedeutung das Thema Verfügbarkeit für verschiedenenartige Einsatzbereiche hat.

Was ist High Availability?

Kombination aus

Verlässlichkeit / Reliability

- **Fehlerhäufigkeit MTBF**

Wartungsfreundlichkeit / Reparability

- **Wie lange dauert Reparatur?**
- **Müssen andere Komponenten unterbrochen werden?**
- **Muss neu gebootet werden?**
- **Lässt sich die Reparatur als Hotswap ausführen?**

Redundanz

- **Kapazitätsreserven / Hot standby**
- **Failover: Wie lange dauert die Serviceübernahme von der Standby-Komponente?**

Im Gegensatz zur vorigen Folie, wo es um die Größenordnung der Verfügbarkeit ging, wird hier angedeutet, unter welchen verschiedenen Umständen Verfügbarkeit gefordert wird.

Zur Verfügbarkeit im Fehlerfall tritt noch hinzu, dass weitgehend unterbrechungsfreier Betrieb im Update-Fall (z.B. neue Software-Features, schnellere Hardware) gefordert wird (Deutsche Telekom 4s, USA 0s).

Zielsetzung

Primäres Ziel: kontinuierliche Service-Verfügbarkeit

- **aber was bedeutet das?**

Ebenen der Fehlertransparenz

(Beispiel Telefonvermittlung)

- **Keine Wiederwahl erforderlich**
- **Verbindungen bleiben bestehen**
- **Verbindungsaufbau wird fortgeführt**
- **Keine hörbaren Störgeräusche oder Unterbrechungen**
- **Kein Datenverlust**

(Folie im Vortrag ausgelassen.)

Am Beispiel Telefonvermittlung wird hier dargestellt, dass Verfügbarkeit auch bezüglich des erwarteten Serviceniveaus verschieden interpretiert bzw. gefordert werden kann. Z.B. ist es deutlich schwieriger, Verbindungen, die sich gerade erst im Aufbau befinden (Wählvorgang), unterbrechungsfrei weiter aufzubauen, als lediglich bereits hergestellte Sprachleitungen aufrechtzuerhalten.

Besonders sensibel ist der Bereich Abrechnung; im Fehlerfall sollen weder Daten verloren gehen (Verlust von Geld für den Betreiber, der die Gespräche nicht abrechnen kann) noch dürfen Abrechnungen doppelt vorgenommen werden (massive Kundenbeschwerden, Verlust von Image und Kunden).

Systemplanung zur Schadensbegrenzung

Fehlerauswirkung hängt von vielen Details ab

Beispiel: Ausfall eines Leitungsvermittlung-Kabels

Direkte Folge: Failover einer FRU (field replaceable units) zu einer Standby-FRU (andere Verbindung)

Annahmen:

- **Vermittlungseinheit für 672 Leitungen**
- **Failoverzeit 50 ms**
- **15 min Fehlersuche und -reparatur**

Fall a) Leitungen auf einem „T3“-Kabel (28*24 Ports)

- **unterbrochene Gespräche: max. 672**
- **Ausfallminuten: 10080**
- **99,997% Uptime/Jahr**

Fall b) Leitungen auf 28 „T1“-Kabel verteilt (à 24 Ports)

- **unterbrochene Gespräche: max. 24**
- **Ausfallminuten: 360**
- **99,99989% Uptime/Jahr**
- **kann 6000 mal passieren für 99,999% Uptime/Jahr**

An einem speziellen Beispiel stellt diese Folie dar, dass die Planung der Systemarchitektur (hier ggf. Verzicht auf eigentlich ökonomischere größere Leitungen) starken Einfluss auf die Verfügbarkeitsbilanz und die Fehlerauswirkungen haben kann.

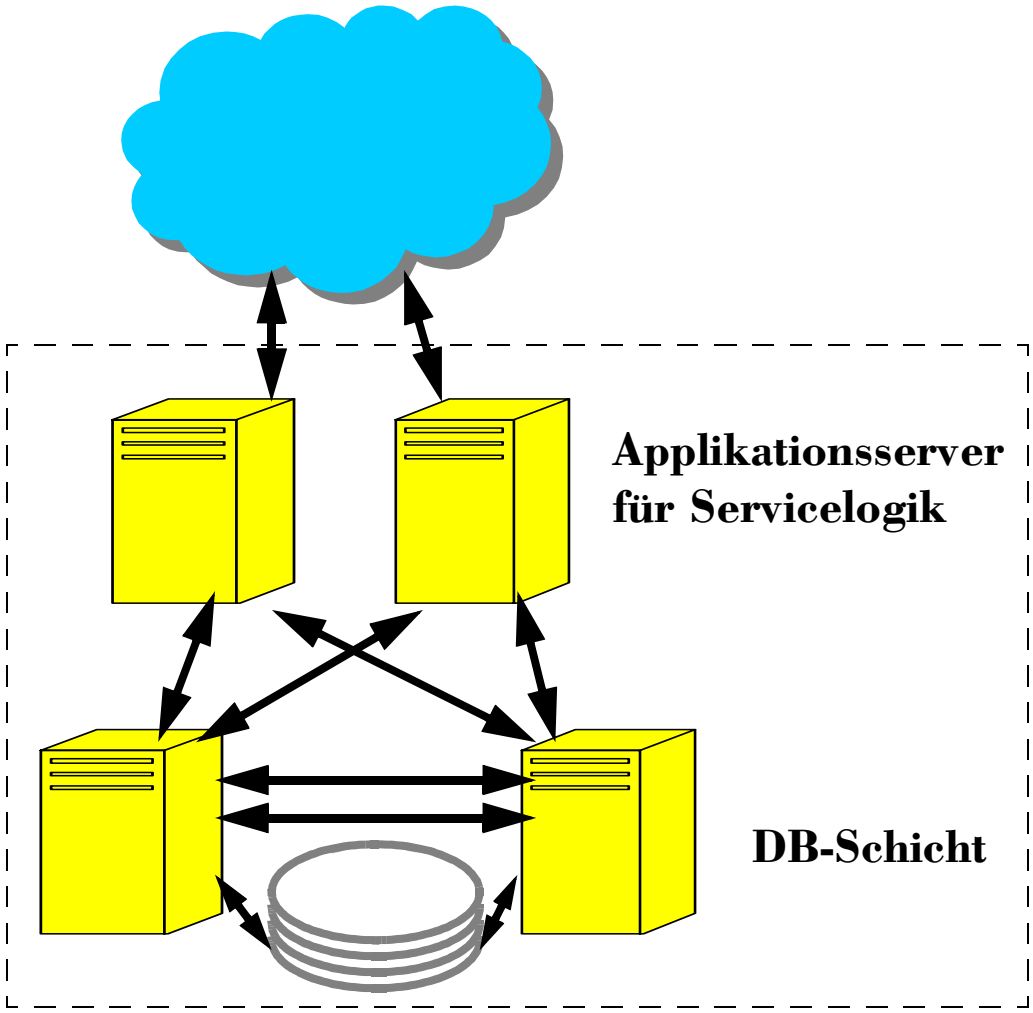
Qualitätsstrategie

Ausfallbehandlungsstrategie des Herstellers

Systemfunktion zügig wieder herstellen

- **Redundante Pfade durch alle Systemkomponenten**
- **Hardware-Failover, Hotswap**
- **Software-Failover**
- **Erkennen und Auflösen von Softwareblockaden**

Der Ausfall einer Systemkomponente muss erkannt werden, eine andere Komponente muss die Funktion übernehmen, und natürlich muss zu diesem Zweck jeder Workflow-Pfad durch das System redundant sein.



Wie kann bei einem Software-Failover der laufende Betrieb unterbrechungsfrei fortgeführt werden?

Die Übernahme der Funktion einer Softwarekomponente durch eine andere lässt sich noch relativ leicht vorstellen, wenn das Umschalten im Workflow des System erstmal erfolgt ist.

Darüber hinaus ist es bei session-orientierten Services erforderlich, auch bereits laufende Sessions fortzusetzen. Die folgende Folie deutet an, mit welchen softwaretechnischen Mitteln dies ermöglicht werden kann. Die gesamte Information über den Zustand einer Session wird in einem Objekt gespeichert und dieses wird vom Basissystem redundant verfügbar gehalten, so dass alle Server darauf zugreifen können.
Gleichzeitig wird damit das Problem behandelt, dass die Rechenressource „Thread" bzw. „Prozess" aufgrund ihres Overheads (Anlegen, Verwalten, Speicherbedarf) nicht in der Größenordnung bereitgehalten werden kann, wie es gleichzeitige Vorgänge im System gibt.

Hochverfügbarkeitstechnik

exemplarisch

Session State / Persistent State (Pattern)

Eine Plattform bietet u. A. die Möglichkeit, den Applikationszustand zwischen den Bearbeitungsschritten redundant zu sichern.

Session kann fortgeführt werden, unabhängig vom Betriebssystemprozess oder auch der Hardware

Architektur und Organisation