# Hochverfügbarkeit versus Disaster Recovery

**Hartmut Streppel**
**Oracle Deutschland B.V. & Co. KG**
**München**

**Schlüsselworte**
Hochverfügbarkeit, Disaster Recovery, Cluster, Business Continuity, Recovery Point Objective, Recovery Time Objective

## Einleitung

Vor allem deutsche Unternehmen verteilen IT-Komponenten, die für unternehmenskritische Anwendungen genutzt werden, auf mehrere Rechenzentren. Damit soll sicher gestellt werden, dass selbst beim Ausfall eines RZs - zumindest aus Sicht der IT - ein unterbrechungsfreier Betrieb gewährleistet ist. Ist das so? Gibt es nicht viele Fehlersituationen, in denen auch ein Ausweich-RZ nicht weiterhilft? Braucht man nicht eine dedizierte Disaster Recovery Lösung, um Katastrophen effektiv zu begegnen? Dieser Artikel soll die Unterschiede zwischen einer „einfachen" Hochverfügbarkeits- und einer Disaster Recovery Lösung erklären.

Wie man sich gegen das Eintreten von Katastrophen schützt, ist ein anderes Thema.

## Hochverfügbarkeit

Ein System ist hochverfügbar, wenn es auch beim Auftreten von Fehlern eine Verfügbarkeit von mindestens 99,99% erreichen kann (s.a. de.wikipedia.org/wiki/Hochverfügbarkeit). 99,99%, auch vier Neunen genannt, bedeuten eine maximale Nichtverfügbarkeit von ca. 52 Minuten pro Jahr oder weniger als 5 Minuten pro Monat. Die Frage, ob geplante Unterbrechungen, z.B. für Wartungsarbeiten, zur maximal verfügbaren Zeit zählen oder nicht, soll hier nicht weiter betrachtet werden.

Es ist allgemein anerkannt, dass eine Verfügbarkeit von 99,99% nur mit hoher Redundanz und dem Einsatz von Clustersystemen erreicht werden kann. Solche Clustersysteme überwachen Komponenten des Systems und sind in der Lage, in Fehlerfällen entsprechend zu reagieren und im schlimmsten Fall hochverfügbare Dienste auch zwischen Rechnern hin- und her zu schwenken.

Betrachtet man die Anforderungen an RPO (Recovery Point Objective – maximale Dauer, für die Daten verloren gehen dürfen) und RTO (Recovery Time Objective – maximal erlaubte Nichtverfügbarkeit eines Dienstes im Fehlerfall), so gelten für Hochverfügbarkeitslösungen typischerweise:

- RTO < 5 Minuten und
- RPO = 0, d.h. kein Datenverlust.

## Disaster Recovery

Die Definition, die auf en.wikipedia.org/wiki/Disaster_Recovery zu finden ist und die ich sehr gut finde, ist: „**Disaster recovery** (DR) is the process, policies and procedures that are related to preparing for recovery or continuation of technology infrastructure which are vital to an organization after a natural or human-induced disaster."

Die Frage ist nur, was ist eine Katastrophe? Diese Frage ist nicht eindeutig zu beantworten. Die allgemein verwendete Definition: Feuer, Erdbeben, Hochwasser greift auf jeden Fall zu kurz. Es gibt viele „kleinere“ Fehler, die als Katastrophe betrachtet werden müssen. Noch sinnvoller ist es, mögliche Fehlersituationen, die eine Bedrohung für den IT-Betrieb bedeuten könnten (nach einer Bedrohungsanalyse(Business Impact Analysis)), in Klassen einzuteilen, und dann zu entscheiden, wo die Katastrophe beginnt, und wie man sich gegen Fehler dieser Klassen absichern kann.

Die Anforderungen an RPO und RTO im Falle einer Katastrophe (K-Fall) sind natürlich erheblich geringer, vor allem wegen der Nichtplanbarkeit einer Katastrophe. Typische in SLAs zu findende Werte sind:

- RTO < 24 Stunden
- RPO < 1 Stunde.

Kann der IT-Betrieb eine Lösung implementieren, die RTO auf wenige Minuten drückt und garantieren, dass kein Datenverlust passieren kann, wird das Management dies mit Freude zur Kenntnis nehmen. Ob letztendlich das Geld für eine solche Infrastruktur bereit gestellt wird, ist eine andere Frage. Technisch möglich sind heutzutage extrem gute Lösungen.

**Fehlerklassen**

Im Folgenden werden vier Fehlerklassen definiert und mögliche Abwehrmaßnahmen gegen die in ihnen enthaltenen Fehler erklärt.

K1. beschreibt einfache Fehler, die vor allem die Hardware betreffen: Ausfälle von Festplatten oder Netzteilen, und einfache Ausfälle von Software.

K2. beinhaltet schwerwiegende Softwarefehler, z.B. im Betriebssystem und komplexere Hardwarefehler, wie z.B. den Ausfall von kompletten Servern.

K3. Enthält Mehrfachfehler, z.B. gleichzeitiger Ausfall von zwei Rechnern, aber auch den gleichzeitigen Ausfall unterschiedlicher Komponenten. Dazu kommen Fehler in kritischen Komponenten wie z.B. in einer Cluster-Software, die ja gerade Redundanz über mehrere Systeme hinweg koordinieren soll. Vor allem enthält K3 komplexe Fehlersituationen, die in ihrer Art nicht vorhersehbar sind.

K4. schließlich enthält Datenkorruption und Datenverluste aller Art, vor allem aber durch administrative Fehler.

**Absicherung gegen Fehler in den unterschiedlichen Fehlerklassen**

Fehler der Fehlerklasse K1 werden heutzutage entweder durch Redundanz der Hardwarekomponenten abgefedert, oder durch einfache Betriebssystemmechanismen, die in der Lage sind, die in der Hardware vorhandenen Redundanzen effektiv auszunutzen. Typische Technologien sind Multipathing Treiber für Zugriffe auf Geräte im SAN und im Netz.

Fehler der Klasse K2 werden durch Redundanz auf Systemebene und darauf aufbauende Cluster-Systeme abgesichert. Damit können komplette Systemausfälle auf der Hardware- und Software-ebene abgefedert werden.

Bei Fehlern der Klasse K3 helfen die „normalen“ Redundanzen nicht mehr. Hier handelt es sich in der Regel um eine Katastrophe, die nur mit einer getrennten Umgebung und geeigneten

Umschaltprozeduren überlebt werden kann.

Fehler der Klasse K4 benötigen einen Mechanismus, Daten verzögert zu replizieren oder mit Hilfe von Snapshot-technologien schnell und sicher auf ältere Stände der Software zurückschalten zu können. Eine typische Technologie aus diesem Bereich ist Oracle Flashback.

**Was ist eine Katastrophe?**

Auf Grund der Klassen-Einteilung ergeben sich zunächst zwei Fragen: Wie wird der Ausfall eines kompletten Rechenzentrums einsortiert? Und: wo ist die Grenze zwischen „normalen“ Fehlern und einer Katastrophe?

Ein RZ-Ausfall ist in der Regel eine Katastrophe. In der Art und Weise, wie er aber typischerweise getestet wird, ist es eher ein relativ gut zu beherrschender Einfachfehler. Der auf einen Schlag ausgefallene Strom ist von den eingesetzten Clustersystemen in typischen Metro-Clustern, d.h. Clustern die sich über mehrere, in der Regel mindestens einige Kilometer voneinander entfernte Rechenzentren erstrecken, leicht und sicher zu erkennen. Ob eine echte Katastrophe im RZ, die mit einem Brand in einem zentralen Router beginnt und bei der anschließender weitere Systeme durch Löschwasser überflutet werden, sicher erkannt wird, ist fraglich.

Auf jeden Fall sind Fehler der Klassen K3 und K4 als Katastrophe zu betrachten.

Die wesentliche Eigenschaft einer Katastrophe ist, dass ihr Ablauf nicht vorhersehbar und damit die Abwehrmaßnahmen nicht oder nur sehr schwer planbar sind.Daraus ergeben sich nun Anforderungen an eine Disaster Recovery Lösung, mit der die Folgen einer Katastrophe minimiert werden können.

**Anforderungen an eine Disaster Recovery Lösung**

Aus der Definition der Fehlerklassen wird klar, dass die wesentliche Anforderung an eine DR-Lösung ist, dass zwei (oder mehr) weitestgehend voneinander unabhängige Infrastrukturen zur Verfügung stehen müssen. Die Idee ist, dass ein Fehler, so komplex und katastrophal er auch sein mag, sich nicht in die unabhängige DR-Infrastruktur ausbreiten kann.

Mit geeigneten Switchover- und Failover-Verfahren ist es nun möglich, im K-Fall einen Schwenk von Anwendungen vom durch eine Katastrophe betroffenen primären RZ ins Ausweich-RZ durchzuführen. Natürlich können auch beide RZs produktiv genutzt werden. Nur müssen dann nach dem Auftreten der Katastrophe und vor dem Switchover/Failover von Anwendungen unter Umständen Umkonfigurationen durchgeführt werden.

**Weitestgehende Unabhängigkeit in einer DR-Umgebung**

Das Ziel dieser Anforderung ist es, die Ausweich-Umgebung von der Produktionsumgebung weitestgehend abzuschotten, damit Fehler sich nicht ausbreiten können. Zu dieser Forderung gehören, dass zwischen Produktions- und Ausweich-Rechenzentrum:

- keine Storage Area Networks,
- keine Broadcast Domains,
- keine „stretched“ Cluster

konfiguriert werden sollten. Am sinnvollsten ist, wenn nur eine IP-Verbindung besteht, über die die Kommunikation inkl. der Datenreplikation zwischen den Rechenzentren läuft.

Sind diese Forderungen umgesetzt, wird es immer noch Single Points of Failures geben, die zu beseitigen zum Teil kaum möglich ist. So kann es bei Verwendung identischer Software-Versionen, dazu gehört auch Firmware von Netz- und SAN-Komponenten, dazu kommen, dass ein Fehler in einer Version in beiden Umgebungen gleichzeitig auftritt. Diesem Problem kann mit dem Betrieb unterschiedlicher Versionen in den beiden Umgebungen begegnet werden. Gerade Technologien, die im Umfeld von DR-Lösungen eingesetzt werden, sind häufig in der Lage, mit unterschiedlichen Versionen zusammenzuarbeiten.

**Datenreplikation**

Die wichtigste Komponente einer DR-Lösung ist sicherlich die Datenreplikation. Mit ihr werden die Daten, die benötigt werden, um im K-Fall die unternehmenskritischen IT-Prozesse wieder anlaufen zu lassen, vom Produktions-RZ in das bzw. die Ausweich-RZs repliziert. Andere Komponenten einer DR-Lösung, z.B. automatisierte Umschaltprozeduren und sichere Change Management Prozesse sind allerdings nicht zu unterschätzen.

Das Thema Replikation ist zu komplex, als dass man es in einem Absatz erläutern kann. Trotzdem sollen hier die beiden wesentlichsten Aspekte zusammengefasst werden: welche Arten gibt es und wie kann „zero data loss“ gewährleistet werden.

Zunächst unterscheidet man zwischen block-, anwendungs- und snapshot-basierter Replikation. Anwendungs-basierte hat den Vorteil, dass sie die Daten kennt, die repliziert werden, und tatsächlich nur die Daten repliziert, die repliziert werden müssen. Beide Eigenschaften fehlen der block-basierten Replikation.

Bei asynchroner Replikation kann es passieren, dass Daten auf dem lokalen System zwischengespeichert werden müssen, da z.B. die Bandbreite zum Ausweich-RZ nicht ausreicht. Fällt das System jetzt aus, sind diese Daten verloren.

Bei synchroner Replikation kann dies nicht passieren, da erst, nachdem die Daten sich im Ausweich-RZ auf „stable storage“ befinden, die Anwendung weiter arbeiten darf. Daraus ist aber ersichtlich, dass die Latenz zwischen den Rechenzentren der maßgebliche Faktor sind, ob synchron oder asynchron repliziert wird.

Ein weiteres Problem existiert mit synchroner Replikation: was soll geschehen, wenn z.B. die Kommunikation zwischen den RZs gestört ist. Hält die Anwendung an oder läuft sie nach einer gewissen Zeit weiter, ohne die Daten zu replizieren. Die meisten Technologien unterstützen heute beide Varianten. Die erste garantiert, dass Daten immer zweimal vorhanden sind. Allerdings ist das Verfahren in vielen Fällen nicht akzeptabel, da die Verfügbarkeit des Gesamtsystems jetzt auch von der Verfügbarkeit der Leitungen und der Zielsysteme abhängt. Und aus den Formeln für Verfügbarkeit ist klar, dass die Verfügbarkeit sinkt, je mehr Komponenten beteiligt sind. Die zweite Lösung, nach einer konfigurierbaren Wartezeit weiter zu arbeiten, erscheint in den meisten Fällen die sinnvollere.

Aufwändige Infrastrukturen vermeiden dieses Problem durch mehrfache synchrone Replikation im Nahbereich und von dort asynchroner Replikation in ein weiter entferntes Ziel-RZ.

**Replikation virtueller Maschinen**

Auf den ersten Blick scheint es eine tolle Lösung zu sein, einfach komplette virtuelle Maschinen z.B. von Frankfurt nach Las Vegas zu replizieren, um dann im K-Fall die VM einfach in Las Vegas starten zu können. Das mag manchmal funktionieren, ist aber keinesfalls eine sicher Disaster Recovery Strategie. Die Gefahr, dass durch fehlerhafte Software oder falsche Bedienung die VM schon im

primären RZ nicht mehr sauber funktioniert, und damit natürlich auch nicht die replizierte VM im Ausweich-RZ ist weit größer als die Wahrscheinlichkeit, dass eine Katastrophe ein RZ lahmlegt.

Viel sinnvoller ist es, nur die sich verändernden Daten der wichtigen Anwendungen zu replizieren, ansonsten aber eine zweite unabhängige VM im Ausweich-RZ zu verwalten. Damit ist sichergestellt, dass auf jeden Fall die VM funktioniert, wenn im K-Fall umgeschaltet werden muss. Dies erfordert natürlich ein exzellentes Change Management!

**Zusammenfassung: Hochverfügbarkeit versus Disaster Recovery**

Was ist nun der Unterschied zwischen einer typischen Hochverfügbarkeitslösung und einer Disaster Recovery Lösung.

Hochverfügbarkeitslösungen sind in der Lage, typische Einfachfehler automatisch zu erkennen und so zu reagieren, dass ein hochverfügbarer Dienst fast ohne Unterbrechung weiter betrieben werden kann. Zu diesen Einfachfehlern können einfach zu diagnostizierende RZ-Ausfälle gehören.

Hochverfügbarkeitslösungen sind in der Regel nicht ausreichend, um Mehrfachfehler oder komplexe Einzelfehler abzufedern. Hierzu sind Disaster Recovery Lösungen notwendig, die in möglichst weitgehender Unabhängigkeit von der produktiven Umgebung betrieben werden. Um sich gegen Ausfälle durch Datenkorruption oder -verlust zu wappnen, sind zusätzliche Vorkehrungen zu treffen, um entweder die Datenreplikation mit zeitlichem Versatz zu betreiben oder aber in der Lage zu sein, auf konsistente Datenbestände der Vergangenheit zugreifen zu können.

**Kontaktadresse:**
Hartmut Streppel
Oracle Deutschland B.V. & Co. KG
Riesstr. 25
D-80992 München

| | |
|---|---|
| Telefon: | +49 (0) 89-1430 2588 |
| Fax: | +49 (0) 89-1430 1150 |
| E-Mail | Hartmut.Streppel@oracle.com |
| Internet: | www.oracle.de |