Wieviele Neunen hätten Sie denn gerne? - Oracles Lösungen für hochverfügbaren SAP-Betrieb im Überblick

Jan Brosowski

Oracle Deutschland B.V. & Co KG
Walldorf

Schlüsselworte

HA, High Availability, DR, Disaster Recovery, SAP, Enqueue Server, Replicated Enqueue, Oracle RAC, Solaris Cluster, Oracle Grid Infrastructure

Wieviele Neunen dürfen es denn sein?

Die „vier Neunen", im Sinne von 99,99% Verfügbarkeit, gelten als der Einstieg in den Bereich der hochverfügbaren Systeme. Auch für SAP-Systeme wird diese sehr häufig gefordert, oder auch eine höhere Verfügbarkeit mit weiteren „Neunen". Hierbei ist der Systembegriff sehr weit gefasst, denn für ein SAP-System ist weit mehr erforderlich als nur eine zuverlässige Hardware oder Datenbank. In diesem Überblick werden die verschiedenen Komponenten einer hochverfügbaren SAP-Infrastruktur diskutiert und vorgestellt, wie diese orchestriert werden.

Die schon erwähnten 99,99% Verfügbarkeit bedeuten, im Jahr mit weniger als 9 Stunden Ausfallzeiten auskommen zu können[1]. Die Angabe der Neunen umfasst aber keine Angabe, wie sich diese 9 Stunden später zusammensetzen dürfen, und welche Verfahren zur Sicherstellung der Verfügbarkeit zur Erreichung des Wertes geeignet sind.

Wie kann man Verfügbarkeit bestimmen?

Es ist relativ einfach, im Nachhinein zu bestimmen, wie lange ein System in einem Zeitraum verfügbar war. Im Vorhinein ist dies allerings sehr schwierig, wenn nicht gar unmöglich. Ursächlich hierfür ist die Komplexität eines SAP-Systems, welches aus einer großen Anzahl miteinerander verbundener Komponenten besteht. [2]

Ein pragmatischer Ansatz zum Bestimmen einer Verfügbarkeitsklasse für ein komplexes System wie ein SAP-System ist daher der umgekehrte Ansatz. Hier greifen Verfahren wie Failure Modes and Effects Analysis (FMEA), welches aber üblicherweise nur Einfachfehler und deren Wahrscheinlichkeiten betrachtet. Man errechnet, wie lange ein System für ein Wiederanfahren nach einem Ausfall maximal benötigen darf, und trifft Maßnahmen, um dies sicher zu stellen.

Zunächst muss man hierzu zwei unterschiedliche Ausfallszenarien unterscheiden: Geplante Ausfälle, beispielsweise um Wartungsarbeiten zu erledigen, und ungeplante Ausfälle, beispielsweise beim Absturz des Betriebssystems eines Servers.

---

[1] Bei durchschnittlich 365,25 Tagen pro Jahr ergeben sich für 99,99% eine maximale Gesamt-Downtime von 8,76 Stunden, bei 99,999% sind es nur noch 52 Minuten.

[2] Das Verfahren des Reliability Block Diagram and Reliability Modeling hat sich im Bereich der HA zwar für kleine Systeme etabliert, doch ist es bei größeren, komplexen Systemen unpraktikabel. Teilweise sind auch Daten, wie beispielsweise Verfügbarkeitsangaben für Prozesse, nicht verfügbar

Insbesondere innerhalb der ungeplanten Ausfälle unterscheidet man verschiedene Fehlerklassen, auf die unterschiedlich reagiert wird. Gängig sind folgende vier Fehlerklassen:

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.

Üblicherweise sind Fehler der Klasse K1 einfach aufzufangen. Als bekannte Maßnahmen seien hier redundante Festplatten, redundante Netzteile oder redundante Lüfter genannt. Sollte eine dieser Komponenten ausfallen, übernehmen die verbliebenen Komponenten direkt und ohne Ausfall des Gesamtsystems. Die so abgesicherten Komponenten sind im Regelfall mit dem Makel einer hohen Ausfallwahrscheinlichkeit behaftet, und aufgrund des häufigen Bedarfs nach einerAbsicherung haben sich preiswerte und zuverlässige Verfahren etabliert.

Fehler der Klasse K2 fängt man mit komplexeren Parallelschaltungen von Systemen ab, beispielsweise in Clustersystemen. Diese können je nach Ausprägung sofort ausgefallene Komponenten ersetzen, oder es gibt einen (automatisierten) Prozesse des Wiederanfahrens. Beispiele hier sind Failover-Cluster oder parallele Datenbanksysteme wie Real Application Clusters.

Fehler der Klassen K3 und K4 werden nicht mehr automatisiert mit normalen Redundanzen abgefangen. Fehler der Klasse K3 werden üblicherweise als Katastrophen[3] angesehen. Hier greifen Verfahren des Disaster Recvovery, um in einer getrennten Umgebung das SAP-System wieder verfügbar zu machen. Fehler der Klasse K4 hingegen benötigen Mechnismen, das SAP-System auf einen älteren Stand zurückzusetzen.

Für alle Fehlerklassen sollte man drei Fragen beantworten:

- Wie wahrscheinlich ist es, dass der Fehler im Betrieb auftritt?
- Wie lange darf der Ausfall im Falle des Eintretens eines Fehlers sein?

---

[3] Die Frage, wo ein „Einzelfehler“ aufhört und wo eine Katastrophe anfängt, ist nicht allgemein zu beantworten. Fehler der Klassen 3 und 4 werden im Allgemeinen als Katastrophen angesehen, da sie sich meist aus mehreren Fehlern ergeben. Auch sollte man einen Ausfall eines Rechenzentrums als Katastrophe betrachten. Er wird zwar typischerweise als ein relativ gut zu beherrschender Einfachfehler getestet. In der Relatität sind Ausfälle von Rechenzentren auf Mehrfachfehler zurückzuführen, beispielsweise durch einen Brand eines zentralen Routers, in dessen Folge dann Teile der Stromversorgung zusammenbrechen und weitere Systeme durch Löschwasser in Mitleidenschaft gezogen werden. Es ist fraglich, ob derartige Katastropfen durch klassische Cluster sicher erkannt werden.

- Darf beim Auftreten des Fehlers ein Datenverlust eintreten, und wenn, wie viele Daten dürfen verloren gehen?

Im folgenden liegt der Fokus auf den Fehlerklassen K1 bis K3 und welche Maßnahmen ergriffen werden, um diese zu kompensieren. K4 wird am Ende kurz diskutiert, da diese Klasse von Fehlern ein deutlich verschiedenes Herangehen als die übrigen erfodert.

Was muss man bei SAP-Systemen verfügbar halten?

SAP-Systeme bestehen aus einer Vielzahl an Komponenten. Doch nicht alle verursachen bei ihrem Versagen gleich einen Ausfall des gesamten SAP-Systems. Beispielsweise skalieren SAP-Applikationsserver nicht nur über mehrere Instanzen, um eine höhere Performance zu erreichen. Sie können auch einander ersetzen und so den Ausfall von Applikationsservern kompensieren.

Es bleiben letztendlich zwei kritische Komponenten, sogenannte „Single Point of Failures“ in der SAP-Architektur übrig, welche abzusichern sind:

- Die erste Komponente ist die Datenhaltung für das SAP-System, in den meisten Fällen ist dies eine Datenbank.

- Die zweite Komponente sind die sogenannten System Central Services (häufig auch verkürzt als Central Services bezeichnet). Diese stellen mit dem Enqueue Server und dem Message Server zwei für das SAP-System kritische Komponenten bereit.

Im folgenden werden verschiedene Methoden vorgestellt, um diese beiden Komponenten hochverfügbar zu halten.

Technologien für SAP-Hochverfügbarkeit

Für die Datenbank, im konkreten Fall eine Oracle-Datenbank, gibt es drei verschiedene Varianten, wie diese für SAP hochverfügbar gehalten werden kann.

Das klassische Verfahren ist ein sogenannter Failover-Cluster. Hierzu wird die Datenbank im Falle des Versagens auf neuer Hardware neu gestartet. Dieser Prozess erfordert eine Orchestrierung verschiedener aufeinander aufbauender Teilschritte. Die verschiedenen Filesysteme, auf der die Datenbank ihre unterschiedlichen Daten abgelegt hat, müssen auf dem neuen Server gemounted werden, die Datenbank muss gegebenenfalls neu parametrisiert werden, sie muss starten und ein Crash-Recovery durchführen. Gegenenfalls sind auch Teile des SAP-Systems anzupassen.

Auch ist der automatisierte Vorgang des Erkennens eines Fehlerfalls[4] nicht trivial, weshalb der Einsatz spezieller Cluster-Software wie Oracle Solaris Cluster mit entsprechenden Agenten empfohlen wird. Während des Neustarts der Datenbank ist diese kurzzeitig nicht verfügbar, wodurch auch das SAP-System nicht verfügbar ist.

Eine zweite, im deutschsprachigen Raum eher unübliche Variante ist der Einsatz von Oracle Dataguard als Hochverfügbarkeitslösung. Hierbei wird auf einem zweiten HW-System eine zweite

---

[4] Es gibt verschiedene Ereignisse, die zur fehlerhaften Annahme führen können, dass ein Failover eingeleitet werden muss, obwohl dies nicht notwendig ist. Exemplarisch genannt sei hier das sogeannte Split-Brain-Phänomen, bei dem beide Cluster-Seiten annehmen, die jeweils andere sei ausgefallen. Ohne ein geeignetes Verfahren zur Vermeidung dieses Zustandes, beispielsweise über ein sogenanntes Quorum, kann dies zu einem komplexen Mehrfachfehler führen.

Datenbank mittels Oracle Dataguard synchron gehalten. Im Falle des Versagens wird dies durch den Observer-Prozess automatisiert festgestellt, und die Datenbank auf dem zweiten Hardware-System zur neuen primären Datenbank umgeschaltet. Es handelt sich also auch um eine Failover-Lösung, die eine kurze Downtime während des Umschalt-Vorgangs bedeutet.

Ohne eine Downtime während des Umschalt-Vogangs kommt die letzte Variante für die Datenbank aus, Oracle Real Application Clusters. Diese Option der Datenbank ermöglicht es, auf verschiedenen Hardware-Systemen simultan die identische Datenbank zur Verfügung zu stellen. Sollte eines der HW-Systeme ausfallen, wird der Betrieb auf den verbliebenen Systemen fortgesetzt. Im Falle von SAP würden sich die vorgelagerten Applikationsserver auf die verbliebenen Datenbankserver neu verbinden, so dass nur einzelne Benutzer kurze Verlangsamungen des Systems bemerken würden.

Für die Central Services stellt SAP selbst nur einen Replikationsmechanismus für den sogenannten Enqueue Server zur Verfügung. Ein Verfahren zum koordinierten Umschalten zwischen primärem Enqueue Server und seinen Replikaten fehlt. Die Aufgabe einer Hochverfügbarkeitslösung in diesem Zusammenhang ist es folglich, die Prozesse von Enqueue Server und Replikaten zu überwachen und ggf. diese untereinander zu koordinieren, Rollen zu vertauschen oder Neustarts durchzuführen.

Im Falle von Failover-Clustern bietet sich Solaris Cluster an, da dieser auch über entsprechende Agenten für die Central Services verfügt. Diese steuern die Verlagerung der Central Services inklusive der Koordination des Enqueue Servers und seiner Replikate. Dies bedeutet, dass außer der Zeit für den Neustart der Datenbank keine weitere Downtime beim Einsatz von Solaris Cluster entsteht, und dass Anwender nach der Verlagerung der Datenbank ohne weitere Unterbrechungen weiterarbeiten können.

Im Falle von RAC wird mit SAPCTL eine Integration der Enqueue-Replikationsmechanismen der SAP vorgenommen. Es bietet sich insbesondere dann an, wenn man die SAP Central Services auf einem der Knoten des RAC-Verbunds betreiben möchte. SAPCTL basiert auf der Oracle Grid Infrastructure und sorgt dafür, dass im Falle des Ausfalls des Knotens, auf dem die Central Services betrieben werden, der Replicated Enqueue Server auf einem anderen Knoten zum neuen Enqueue Server wird.

Wichtig kann zudem eine Integration der verschiedenen Technologien für Hochverfügbarkeit in Technologien für Disaster Recovery sein. Für das Disaster Recovery werden Systeme vorgehalten, die durch möglichst umfassende Trennung von den primären Systemen weitgehend vor der Replikation von Fehlern geschützt sind. Dies kann auf verschiedene Arten erfolgen, die eine eigene Abhandlung ausfüllen könnten.

Die nachfolgende Tabelle gibt einen Überblick über die Funktionsweise der verschiedenen Methoden und zeigt erste Anknüpfungspunkte für eine weiterführende Diskussion der DR-Fragen.

	Klassischer Fail-Over-Cluster	Replizierender Fail-Over-Cluster	Active-Active-Cluster
Beispielhaftes Oracle-Produkt	Oracle Solaris Cluster	Oracle Dataguard	Oracle Real Application Clusters (RAC)
Beschreibung Datenbank	Datenbank wird auf getrenntem Server neu gestartet	Replikat der Datenbank wird auf getrenntem Server zur primären Datenbank	Datenbank läuft parallel auf mehreren Knoten, die alle vollständigen Zugriff auf alle Daten haben
Notwendige Storage-Infrastruktur	Hochverfügbares Storage, auf das alle Knoten zugreifen können	Getrenntes Storage für beide Knoten mit Schutz gegen K1-Fehler	Hochverfügbares Storage, auf das alle Knoten simultan zugreifen können.
Integration SAP Central Services	Integriert in Solaris Cluster Infrastruktur (Netweaver Agent)	manuell oder skript-gesteuert	Integrataion in Grid Infrastructure mit SAPCTL
Typische Failoverzeiten	15-30 Minuten (DB + SAP CS)	5-10 Minuten (DB only)	0 Minuten (Datenbank) 5-10 Minuten (SAP CS)
DR-Integration	Unterstützt in der sog. Geographic Edition verschiedene Replikationsverfahren zum DR-Standort . Dies umfasst die Integration der SAP Central Services	Kann in DR-Verfahren integriert werden (bspw. Oracle Solaris Cluster, Oracle Siteguard)	Kann mit Replikationsverfahren kombiniert und in DR-Verfahren integriert werden (bspw. Oracle Solaris Cluster, Oracle Siteguard)

Kombination von Technologien

Oracle Solaris Cluster besitzt augenscheinlich die umfassendste Integration in SAP der drei oben vorgestellten Technologien. Zudem bietet der diebeste Unterstützung von Disaster Recovery. Allerdings bietet er als klassischer Failover-Cluster beispielsweise nicht die Verfügbarkeit eines RAC.

Die Lösung hier liegt in der Kombination der verschiedenen Oracle-Technologien zu einer Art Baukasten. Solaris Cluster bildet hierbei das Grundmodul, welches eine für viele Zwecke ausreichende Verfügbarkeit bereitstellt.

Sollte zusätzlich eine Replikation an einen DR-Standort gewünscht werden, so kann Solaris Cluster entsprechend erweitert werden. Hierzu wird die Orchestrierung von Solaris Cluster Geographic Edition genutzt in Kombination mit der Replikation von Oracle Dataguard.[5]

Sollte die Verfügbarkeit eines Failover aufgrund der angestrebten Recovery-Zeiten nicht ausreichen, kann Oracle RAC eingesetzt werden. In diesem Fall ergänzen sich die Funktionalitäten der Produkte.

Beide Erweiterungen können auch kombiniert werden.

Fehlerklasse K4

Fehler aufgrund von Fehlbedienungen sind gerade in komplexen SAP-Systemen schwierig zu korrigieren.

Sind die Systeme sauber voneinander getrennt, können Fehler – sofern die Fehlbedienung ausreichend schnell bekannt wird – durch Technologien wie Flashback Database korrigiert werden. In diesem Fall wird die Datenbank und somit das gesamte SAP-System auf einen Zustand vor dem Fehlerfall zurückgesetzt. Dies bedeutet, dass sämtliche Ändernungen nach dem Fehler zusammen mit dem Fehler verloren gehen. Dies bietet sich beispielsweise an, wenn unmittelbar nach dem Fehlerereignis dafür gesorgt wurde, dass keine weiteren wichtigen Transaktionen auf dem System vorgenommen wurden.

[5] Oracle Solaris Geographic Edition ist bereits in der Oracle Solaris Cluster Lizenz enthalten. Oracle Dataguard ist in der für SAP notwendigen Oracle Datenbank Enterprise Edition ebenfalls enthalten

Theoretisch ist es zwar ebenfalls möglich, eine einzelne Datenbank-Transaktion zurückzunehmen. Dies ist allerdings im Falle von SAP-Systemen nur dann sinnvoll möglich, wenn die fehlerbehaftete Datenbank-Transaktion eindeutig identifiziert werden kann, und spätere Datenbank-Transaktionen durch die fehlerbehaftete Datenbank-Transaktion keine Folgefehler erzeugt haben.[6] Beispiele hierfür sind versehentlich gelöschte Datenbanken oder Tabellen, insbesondere wenn der Fehler dem entsprechenden Mitarbeiter unmittelbar auffällt und er die notwendigen Maßnahmen zur Fehler-Korrektur umgehend umsetzt.

Gerade in komplexen SAP-Systemen, in denen verschiedene Systeme bei Transaktionen aufwändig zusammenarbeiten, ist ein gemeinsames Zurücksetzten auf einen bestimmten Stand praktisch unmöglich, ohne die Integrität der Systeme zueinander zu gefährden. Hier greifen dann Verfahren,die auf Applikationsebene korrigierende Maßnahmen durchführen.

Planned Downtime

Der Umgang mit ungeplanten Ausfällen ist nur ein Teil einer Hochverfügbarkeitsstrategie. Der zweite Teil ist der Umgang mit geplanten Ausfällen, bzw. deren Vermeidung oder Verkürzung.

Geplante Ausfälle sind im Allgemeinen für Wartungsarbeiten notwendig, sofern diese nicht während des normalen Betriebs ausgeführt werden können oder deren Ausführung riskant erscheint. Wartungsarbeiten können sowohl ein Verändern von Parametern oder Konfigurationen, als auch die Installation von neuen Versionen oder Patches sein. Im Falle eines SAP-Systems sind folgende Komponenten Kandidaten für Patches und Konfigurationsänderungen:

- Hardware: beispielsweise Firmware-Updates für Server oder deren Komponenten etc.
- Virtualisierung: sowohl HW-integrierte Hypervisoren als auch Software-Hypervisoren
- Betriebssystem und andere systemnahe Software wie eine Cluster-Infrastruktur
- Datenbank
- SAP-Software

Wartungsarbeiten werden üblicherweise in sogenannten Wartungsfenstern durchgeführt, für die Ausfälle des Systems angekündigt werden. Meist sind diese Wartungsfenster über das Jahr verteilt, sodass je nach angestrebter Systemverfügbarkeit beispielsweise zwei Fenster zu je 2 Stunden zur Verfügung stehen.

Man benötigt also Verfahren zur Minimierung der notwendigen Aktivitäten innerhalb der Wartungsfenster. Beispiele hierfür sind:

- Rolling Patching/Upgrades: Hier werden redundante Komponenten nacheinander gewartet, während die jeweils verbleibenden die Last übernehmen. Im Idealfall entstehen hierbei keine oder nur sehr kurze Downtimes. Das Patchen eines Oracle RAC Clusters oder der Storage Cells eines Engineered Systems sei hier als Beispiel angeführt.
- Wartung einer Kopie: Hier werden die Daten des zu wartenden Systems kopiert, an der Kopie werden die Wartungsarbeiten durchgeführt während das originale System weiterläuft, und

[6] Eine fehlerbehaftete SAP-Transaktion kann mehr als eine Datenbank-Transaktion verursacht haben.

dann wird in einer kurzen Downtime das gewartete System gestartet. Beispiele sind hier das Patching eines Solaris-Betriebssystems oder das Patchen eines ungenutzen Oracle Homes.

- Wartung eines laufenden Systems: Bestimmte Änderungen bedürfen keines Neustarts und können daher im laufenden Betrieb durchgeführt werden. Dies kann beispielsweise die Installation neuer systemnaher Software sein, die für den Betrieb nicht permanent gebraucht wird, oder das Verändern von Parametern in der Datenbank, deren Änderung im laufenden Betrieb vorgesehen ist.

Auf ein weiteres Verfahren zur Minimierung von Wartungsarbeiten soll allerdings auch noch kurz eingegangen werden: Der Verzicht auf eben diese. Getreu dem alten Motto “Never touch a running system” wird häufig für kritische Systeme bewusst auf jegliche regelmäßige Wartungsarbeiten wie Patching oder kleinere Upgrades verzichtet, sondern statt dessen nach relativ langen Zeitfenstern große Upgrade-Projekte durchgeführt (häufig verbunden mit einem Wechsel der Hardware).

Generell erscheint dies eine valide Lösung, doch birgt sie im Falle von SAP-Systemen Risiken. Es mag zwar sein, dass ein SAP-System als Insel-System ausgeführt wird ohne direkten Kontakt zur Außenwelt, doch gibt es bedingt durch die Notwendigkeit, mit Clients auf das System zuzugreifen, zumindest indirekte Angriffspfade. Somit sind Sicherheits-bedingte Wartungsarbeiten notwendig. Auch kann es Wartungsarbeiten geben, die die Leistungsfähigkeit des Systems verbessern oder die auftretende Fehler beseitigen. Gerade bei komplexen Systemen wie einem SAP-System sind derartige Fehler und Verbesserungen mit hoher Wahrscheinlichkeit zu erwarten, und daher sollten entsprechende Wartungsfenster vorgesehen werden.

Zusammenfassung

Es gibt eine Reihe von Technologien, welche im SAP-Umfeld für hohe Verfügbarkeit sorgen können. Ausgehend von einer FME-Analyse können diese zu einer situationsspezifischen Lösung orchestriert werden. Es empfielt sich für größere Installationen einer Art Baukastenprinzip zu folgen. Das hier vorgestellte Prinzip auf Basis von Solaris Cluster kann mit einer Replikation mit Dataguard für den DR-Fall und einem Active-Active-Cluster auf Basis von Oracle RAC ergänzt werden und bietet so umfassenden Schutz für alle gängigen Verfügbarkeitsanforderungen.

Kontaktadresse:

Jan Brosowski

Oracle Deutschland B.V. & Co. KG

Altrottstraße 31

D-69190 Walldorf

Telefon: +49 (0) 6227 - 356 201

E-Mail jan.brosowski@oracle.com

Internet: www.oracle.com