Real Application Cluster (RAC) allein reicht nicht aus, um eine Oracle-Datenbank-Umgebung hochverfügbar bereitzustellen. Vielmehr ist die Einrichtung einer Standby-Datenbank dringend zu empfehlen, um den Ausfall eines RACs oder eines Single-Servers abzusichern. Dazu bietet Oracle unter anderem Tools wie den Data Guard, der jedoch leider nicht mit Produkten der Oracle Standard Edition verfügbar ist. Dieser Artikel beschreibt einen Weg, auch mit der Oracle Standard Edition Standby-Datenbanken zu erstellen und zu betreiben. Damit muss nicht zwangsläufig mehr Aufwand oder weniger Komfort verbunden sein.

Standby-Datenbanken als Teil einer hochverfügbaren Oracle-Datenbank auf Basis der Standard Edition

Thilo Solbrig, ASPICON GmbH

Eine Standby-Datenbank ist unverzichtbarer Bestandteil einer ausgereiften Hochverfügbarkeitslösung (HA-Lösung). Das Aufsetzen eines RACs bietet neben dem Skalierungseffekt und der möglichen Downtime-Reduzierung für geplante Wartungen eine hervorragende Vorsorge gegen Server-Ausfall. RAC ist bereits ohne Aufpreis in der Standard Edition enthalten. Das allein ist jedoch keine Absicherung gegen Site Failures beziehungsweise Fehler im Shared Storage (Blockfehler, Lost write, Storageausfall etc.). Defekte im Platten-Subsystem wirken sich gleichzeitig auf alle RAC-Nodes aus und können auch zum Komplettausfall einer geclusterten Datenbank führen. Es ist also ausschließlich mit einer RAC-Lösung noch kein schnelles Desaster-Recovery möglich. An dieser Stelle kommt das Konzept der Standby-Datenbank zum Tragen.

Im einfachsten Fall einer physischen Standby-Datenbank wird dabei an einem entfernten Standort eine blockidentische Kopie der primären Datenbank (oder eben des RACs) vorgehalten. Die Änderungen auf der Primärseite werden über die ohnehin anfallenden Archivelogs über einen Redo-Apply-Mechanismus auf die Standby-Seite synchronisiert. Bedingt durch die räumliche Trennung und das eigene Storage schützt eine Standby-Datenbank auf ideale Weise vor den oben genannten und nicht vom RAC abgedeckten Fehlerszenarien. Zusätzlich kann eine physische Standby-Datenbank für Ad-hoc-Reporting vorübergehend „read-only“ zum Einsatz kommen. Auch die aus der Oracle Enterprise Edition bekannte Flashback-Technologie ist ansatzweise mit einer Standby-Datenbank nachbildbar – vorausgesetzt, die Archivelogs werden zeitlich verzögert angewandt. Ein weiterer wichtiger Aspekt neben der Desaster-Absicherung ist die Möglichkeit, in Vorbereitung auf Wartungsarbeiten am Primärsystem einen Switchover auf das Standby-System durchzuführen, sodass auch während der Wartung weiter mit der Datenbank gearbeitet werden kann.

Oracle Data Guard als kostenfreier Bestandteil der Oracle Enterprise Edition bietet dafür bereits eine leistungsfähige Unterstützung. Nutzern der Oracle Standard Edition / Standard Edition One steht ein solches oder ähnliches Feature seitens Oracle jedoch nicht zur Verfügung. Diese Lücke wird von mehreren Drittanbietern geschlossen. Dieser Artikel stellt die Standby-Lösung „Dbvisit Standby“ der neuseeländischen Firma Avisit Solutions Ltd vor.

Der Failover

Prinzipiell unterscheidet man im Kontext der Standby-Architektur zwischen Failover und Switchover. Ein Failover wird eingeleitet, wenn die Primärseite unplanmäßig ausfällt. Er ist insbesondere dadurch gekennzeichnet, dass in der Regel nicht mehr alle Redo-Informationen auf die Standby-Seite übertragen werden können und deshalb die Standby-Datenbank nur zum maximal noch möglichen Stand hochgefahren wird. Failover-Szenarien sind daher nahezu sicher mit Datenverlust verbunden. Der Verlustumfang hängt dabei sehr eng mit der Logswitch-Frequenz zusammen und kann durch häufige Log-Switches, unter Beachtung der daraus resultierenden negativen Performance-Auswirkungen, minimiert werden. Auch die Ablage von Archivelog-Kopien auf einem physisch getrennten System beugt im Falle des Server-Crashs einem Archivelog-Verlust vor. Bei einem Failover versucht die Standby-Seite prinzipiell keine Kommunikation mit der Primär-Datenbank mehr.

Der Switchover

Switchover ist der geplante Rollentausch und läuft ohne Datenverlust ab, da sich die beteiligten Instanzen untereinander koordinieren können. Er wird typischerweise als Vorbereitung von Wartungsarbeiten oder Patch-Installationen am Primärsystem genutzt. Dbvisit Standby arbeitet nach

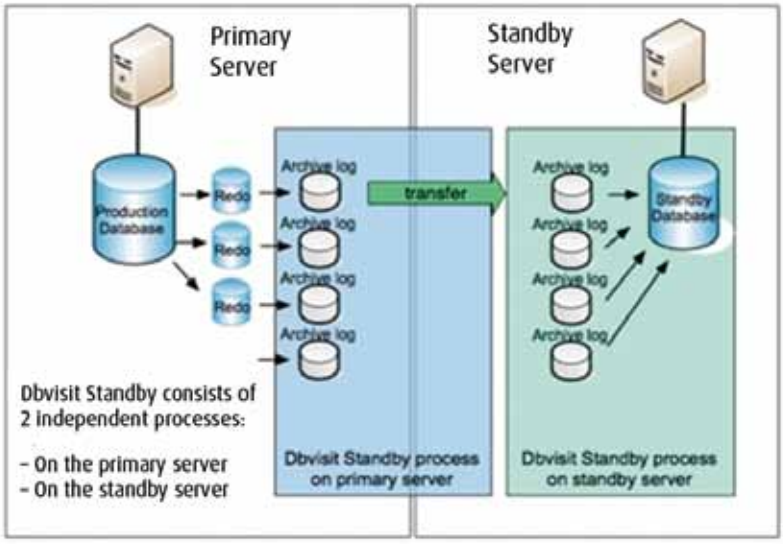

Abbildung 1: Dbvisit High-Level-Architektur

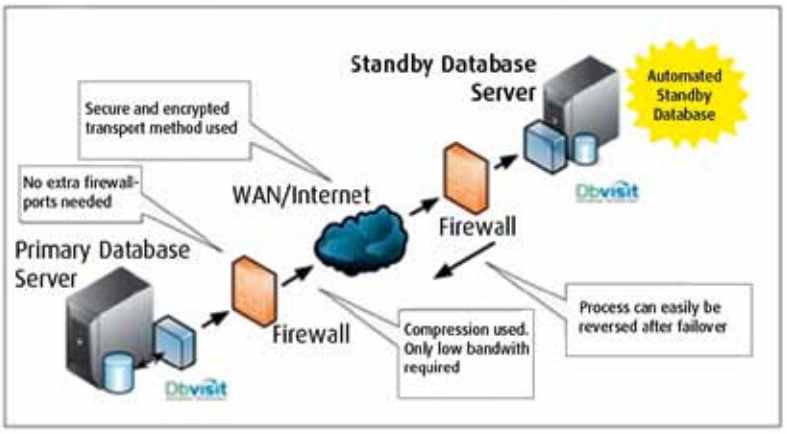

Abbildung 2: Verschlüsselter, komprimierter Logtransfer über Standard-SSH-Ports

demselben Grundprinzip wie Oracle Data Guard, indem es die auf der Primärseite anfallenden Archivelogs auf die blockidentische Standby-Datenbank anwendet (siehe Abbildung 1).

Neben einigen Einschränkungen gegenüber Data Guard kann Dbvisit Standby allerdings auch mit Vorteilen aufwarten. So werden die Archivelogs per se verschlüsselt und komprimiert zum Standby-Server übertragen, da hierfür eine SSH-Verbindung aufgebaut wird. Zudem erlaubt der Einsatz von „rsync" als einer optionalen Übertragungsvariante neben „scp" und „sftp" das Wiederaufnehmen des Archivelog-Transports im Falle einer Netzwerk-Unterbrechung. So müssen bereits übertragene Fragmente von Archivelogs nicht erneut über das Netzwerk gesendet werden (siehe Abbildung 2).

Weitere interessante Vorzüge gegenüber Data Guard sind:

- Die vollautomatische Generierung der Standby-Datenbank – optional mit Offline-Übertragung der Datenbank – was insbesondere bei großen Datenbeständen oder niedriger Netzbandbreite von erheblichem Nutzen ist
- Ein Mailbenachrichtigungs-Mechanismus, der neben Fehlermeldungen auch einen regelmäßigen Archivelog-Gap-Report beinhaltet und somit zeitnah über eventuelle Lücken in der Archivelog-Versorgung informiert
- Ein Start-Stop-Tool, das selbstständig den Datenbank-Modus (primary, standby) erkennt und die entsprechende Datenbank in den korrekten Modus hochfährt – „open" für die Primär- und „mout" für die Standby-Datenbank

Dbvisit Standby kann ebenso wie Data Guard eine Primär-Datenbank auf mehrere Standby-Datenbanken spiegeln und unterstützt ASM, OMF, RAC-to-Single sowie RAC-to-RAC. Anders als bei Data Guard muss die Archivelog-Übertragung in Dbvisit Standby aber über einen externen Scheduler (cron, Windows Task oder Dbvisit Scheduler) organisiert werden. Da optional vor jeder Übertragung ein Log-Switch veranlasst werden kann, ist somit als angenehmer Nebeneffekt auch das bis einschließlich 10*g* R2 vakante Problem umgangen, dass der DBA die Datenbank nur schwer zu einem garantierten Logswitch-Rhythmus bewegen konnte (Stichwort „archive_lag_target").

Erweiterte Standby-Konzepte wie Logical Standby, Snapshot Standby oder Active Data Guard sind Dbvisit Standby allerdings fremd. Auch findet man hier keine Automatismen zur Einleitung eines Failover, wie sie etwa vom Data Guard Observer bekannt sind. Ein kompletter Switchover, also ein kontrollierter Rollentausch, dauert aufgrund der Implementierung in dokumentierten Einzelschritten, sogenannten „Checkpoints", im Vergleich zu einem Data Guard Switchover mittels Data Guard Broker auch ein wenig länger. Dbvisit ist in seiner Arbeitsweise auf das reine Archivelog-Apply eingeschränkt und entspricht damit dem Maximum-Performance-Mode von Data Guard. Hauptgrund dafür ist, dass Oracle die für Maximum-Availability oder Maximum-Protection nötige Schnittstelle zu den Redologs der Standby-Datenbank nicht offengelegt hat.

Insgesamt lässt sich sagen, dass sich Dbvisit Standby hinsichtlich des Feature-Umfangs nicht mit Oracle Data Guard messen kann und das auch sicher nicht will. In jedem Fall stellt es aber ein zuverlässiges und vollkommen ausreichend ausgestattetes Pen-

dant im Bereich von Standard Edition / Standard Edition One dar. Eine 30-Tage-Testversion steht auf der Dbvisit-Webseite nach kurzer Registrierung zum Download bereit: http://www.dbvisit.com/download.php

Praxiserfahrungen

Die ASPICON GmbH als Avisits Technologie-Partner für Deutschland, Österreich und die Schweiz hat bereits zahlreiche Projekte in verschiedenen Konstellationen und Betriebssystemen erfolgreich umgesetzt, als Single-to-Single, RAC-to-Single oder auch RAC-to-RAC, mit und ohne ASM-Unterstützung. Neben den häufiger anzutreffenden Alltagskonfigurationen waren dabei durchaus auch Herausforderungen wie etwa 4-Node-RAC-to-Single oder ein 2-Node-RAC-to-2-Node-RAC über eine 150 km lange WAN-Strecke zu meistern. Im letzteren Projekt bestand die Anforderung darin, einen dem Data Guard Observer vergleichbaren Mechanismus zu entwickeln, der bei Ausfall der Primär-Datenbank einen automatischen Failover einleitet. Dbvisit stellt selbst nichts Adäquates bereit. Das Unternehmen hat sich in dem Fall jedoch klar hinter Avisits Standpunkt gestellt und dem Kunden dringend von einer solchen Implementation abgeraten. Ein automatisches Failover sollte, wenn man diese Entscheidung denn überhaupt einer Maschine zubilligen will, nur im Maximum-Protection-Mode in Betracht gezogen werden. Denn nur dort lässt sich sicherstellen, dass ein versehentlicher Failover keinen Datenverlust nach sich zieht. Ist die Datenbank so hochkritisch, dass eine menschliche Bewertung eines Fehlerfalls nicht abgewartet werden kann, dann dürfte sie aller Wahrscheinlichkeit auch keinen Datenverlust tolerieren können. Wie bereits erwähnt, wird der Maximum-Protection-Mode von Dbvisit Standby aber nicht unterstützt.

Die „Single-to-Single"-Architektur ist erwartungsgemäß am einfachsten umzusetzen. Sie funktioniert quasi „out-of-the-box". Etwas diffiziler wird es dann aber im RAC-Bereich, da hier einige Einstellungen für die Koordinierung der Clusterknoten untereinander hinzukommen und sowohl Switchover als auch Failover mit einigen manuellen, wenngleich ausgezeichnet dokumentierten Nacharbeiten verbunden sind.

In jedem Fall sind aber clientseitig Vorkehrungen zu treffen, damit die Datenbank nach einem Switchover, respektive Failover, möglichst schnell und mit minimalem Konfigurationsaufwand wieder erreichbar ist. Ein Umstand, der natürlich gleichermaßen für eine Data-Guard-Umgebung zutrifft. Aus praktischer Erfahrung kristallisieren sich hierfür vor allem drei Wege heraus.

1. Änderung des Connect-Strings: Im Connect-String steht der Hostname des Primär-Servers. Nach einem Switchover/Failover wird er gegen den Hostnamen des neuen Primär-Servers ausgetauscht. Diese Variante bietet sich insbesondere bei zentralisierter Verwaltung der Connect-Strings, etwa im Oracle Internet Directory oder Active Directory beziehungsweise in der zentral abgelegten Datei tnsnames.ora an.
2. Virtueller Hostname: Anstelle des Hostnamens des Datenbank-Servers selbst wird im Connect-String ein DNS-Alias auf den jeweils aktuellen Primär-Server verwendet. Nach einem Switchover/Failover ändert man nicht den Host im Connect-String, sondern den im DNS-Alias hinterlegten Verweis. Diese Variante wäre bei zahlreichen oder heterogen verteilten Connect-Strings gegenüber der ersten zu bevorzugen. Da viele Betriebssysteme allerdings einen DNS-Cache verwalten, kann es hier einige Zeit dauern, bis die Clients die produktive Datenbank nach einer Umschaltung wieder erreichen. Damit kann bei kritischen Anwendungen unter Umständen wertvolle Zeit verloren gehen.
3. Connect-Failover: Eine dritte Möglichkeit ist die von uns favorisierte Nutzung des Connect-Failovers. Es werden hierbei beide Server im Connect-String angegeben, aber nur am jeweils produktiven Server läuft der Listener. Connect-Versuche auf die Standby-Seite scheitern damit und werden automatisch in Sekundenbruchteilen auf den produktiven Server umgeleitet. Insbesondere beim Einsatz von RAC oder Oracle Restart lässt sich diese Variante durch Erstellung entsprechender Cluster-Ressourcen sehr gut automatisieren. Aber auch ohne ein solches Hilfsmittel ist dieser Weg einfach umsetzbar. Im Gegenzug führt ein vergessener Listener jedoch zwangsläufig zu Connect-Fehlern am Client. Neben der Erstabsicherung von Datenbanken war für einige Kunden übrigens der Umstieg von Data Guard auf Dbvisit Standby vor allem unter dem Aspekt interessant, dass damit der Weg zurück von der Enterprise Edition auf die Standard Edition – verbunden mit einer entsprechenden Kosteneinsparung in den laufenden Supportkosten – bereitet wurde.

Fazit

Aufbau und Betrieb mindestens einer Standby-Datenbank ist neben Oracle Real Application Clusters ein unverzichtbarer Bestandteil einer zuverlässigen Hochverfügbarkeitslösung. Dafür ist das Unternehmen nicht zwingend auf die Lizenzierung einer Oracle Enterprise Edition angewiesen, sondern kann bereits mit der Standard Edition / Standard Edition One ohne Zusatzkosten RAC nutzen und unternehmenskritische Datenbanken mithilfe einer kostengünstigen Drittanbieter-Lösung um eine oder mehrere Standby-Datenbanken ergänzen. Solche Produkte sind ebenso komfortabel einzurichten und zu betreiben wie das Oracle-Enterprise-Edition-Feature Oracle Data Guard.

Thilo Solbrig
ASPICON GmbH
vertrieb@aspicon.de