Aufbau einer 12*c*-RAC- und Data-Guard-Umgebung mit NFS Storage bei der DEVK

Johannes Ahrends, Carajan DB, und Tim Hensel, DEVK Versicherungen

Im Rahmen eines Projekts wurden die vertriebsunterstützenden Systeme der DEVK-Versicherungen auf eine „always online"-Lösung ausgerichtet. Als Basis sollte eine neue, stabile sowie hochverfügbare Oracle-Infrastruktur dieser Anforderung entsprechen. Das Ziel-Design wurde zudem zugunsten einer schnellen Wiederherstellbarkeit und einer möglichst unterbrechungsfreien Wartbarkeit erarbeitet.

Vor der endgültigen Architektur-Entscheidung der zugrunde liegenden Infrastruktur wurden diverse Szenarien zur Realisierung detailliert betrachtet, gewichtet und anschließend bewertet. Dazu gehörten neben Virtualisierungs-Lösungen mit VMware oder Oracle VM auch die Oracle Engineered Systems ODA und Exadata.

Letztendlich fiel die Entscheidung – nicht zuletzt aufgrund der zu diesem Zeitpunkt teilweise unklaren Lizenz-Politik im VMware-Umfeld und der damit verbundenen Risiken – auf eine Hardware-Lösung mit Cisco UCS Blades.

Die anfängliche Idee, hierauf ein sich über beide Rechenzentren erstreckendes Oracle-Stretched-Cluster zu realisieren, wurde verworfen, da sich im Laufe des Projekts im Hinblick auf die Hochverfügbarkeit der Applikationsserver frühzeitig für ein NetApp-Metrocluster im Master-Slave-Modus entschieden wurde. Dies hätte im Fehlerfall ein manuelles Umschalten des zugrunde liegenden Storage bedeutet.

Somit sollte die Verfügbarkeit der drei benötigten Datenbanken durch die Oracle-eigenen HA-Komponenten RAC in Kombination mit Data Guard abgebildet werden (Maximum Availability Architecture). Realisiert wurde das Ganze in der Produktionsumgebung durch jeweils zwei physikalische Server im RAC-Verbund, wobei die beiden RZ-Standorte über Data Guard miteinander verbunden wurden *(siehe Abbildung 1)*. Von Active Data Guard und Fast Start Failover wurde abgesehen; von letzterem primär aufgrund des fehlenden dritten Standorts für den Observer.

Um einen ausgereiften Staging-Prozess realisieren zu können, gibt das Betriebskonzept der DEVK vor, einer produktiven Umgebung technisch gleichartige Systeme

in anderen Ebenen voranzustellen. Somit galt es, zusätzlich eine Vorproduktions-, eine Entwicklungs- und eine sogenannte „Maintenance"-Umgebung aufzubauen. Letztere kann als reine „Spielwiese" für Oracle-DBAs bezeichnet werden, auf der beispielsweise das Einspielen von Patches als Erstes vorgenommen wird, bis diese dann sukzessive von Stage zu Stage bis in die Produktion eingerichtet werden.

Einzig in der Entwicklung wurde ursprünglich auf die HA-Realisierung durch RAC und Data Guard verzichtet, um Lizenzen einzusparen. In Summe wurden also dreizehn Oracle-Datenbank-Server (vier für Maintenance, einer für Entwicklung, vier für Vorproduktion und vier für Produktion) mit jeweils mindestens drei Datenbanken aufgebaut, ohne hierbei auf die neue Multitenant-Architektur zurückzugreifen.

Bei der Auswahl des zugrunde liegenden Filesystems für die Real Application Cluster hat man sich bewusst nicht für das bei der DEVK bisher eingesetzte und bei den DBAs stets als sehr zufriedenstellend und stabil angesehene ASM, sondern für NFS entschieden. Grund ist die Vorgabe der geringen Wiederherstellungszeiten und die damit verbundene Anforderung, die bis zu 1TB großen Datenbanken im Bedarfsfall möglichst schnell recovern zu können. Dieser Herausforderung konnte man durch Einsatz der Snapshot-Technologie, die wiederum NFS-Shares voraussetzt, problemlos gerecht werden, was sich im späteren Verlauf des Projekts bestätigen sollte. Von der Alternative, ASM auf NFS zu implementieren, wurde abgesehen; man entschied sich hingegen für den Einsatz von direct NFS (dNFS) für Oracle12*c*.

Als Sicherungstool sollte das von NetApp Anfang dieses Jahres veröffentlichte Plug-in des SnapCenter für Oracle-Datenbanken dienen. Zu Beginn des Projekts nahm man an einem Beta-Programm dieser neuen Komponente für das SnapCenter teil, wobei selbige ausführlich geprüft wurde und anfängliche Fehler im direkten Austausch mit NetApp ausgemerzt werden konnten.

Filesystem-Architektur der Datenbanken

Auf den Datenbank-Servern wurde Oracle 12.1.0.2 auf SUSE Linux Enterprise Server 11 mit Service Pack 4 installiert, wobei diese Konstellation der Versionen primär den recht strengen Release-Vorgaben des SnapCenter geschuldet war. Zudem ist SUSE die strategische Linux-Distribution der DEVK.

Die Basis-Installation und -Konfiguration der Serversysteme erfolgte mithilfe der Plattform „Ansible". Durch diese Automatisierungsmethode ist sichergestellt, dass sich die Konfigurationen – angefangen von OS-Usern und -Gruppen über Kernel-Parameter bis hin zu Filesystemen – auch im kleinsten Detail nicht unterscheiden. Später wurden zusätzlich definierte Checks durch das Tool „ServerSpec" durchgeführt, um auch zukünftig Stage-übergreifend vor ungewünschten Änderungen bewahrt zu bleiben.

Bei der Verzeichnisstruktur orientierte man sich an der Oracle Flexible Architecture und entschied sich für die Installation der GI-

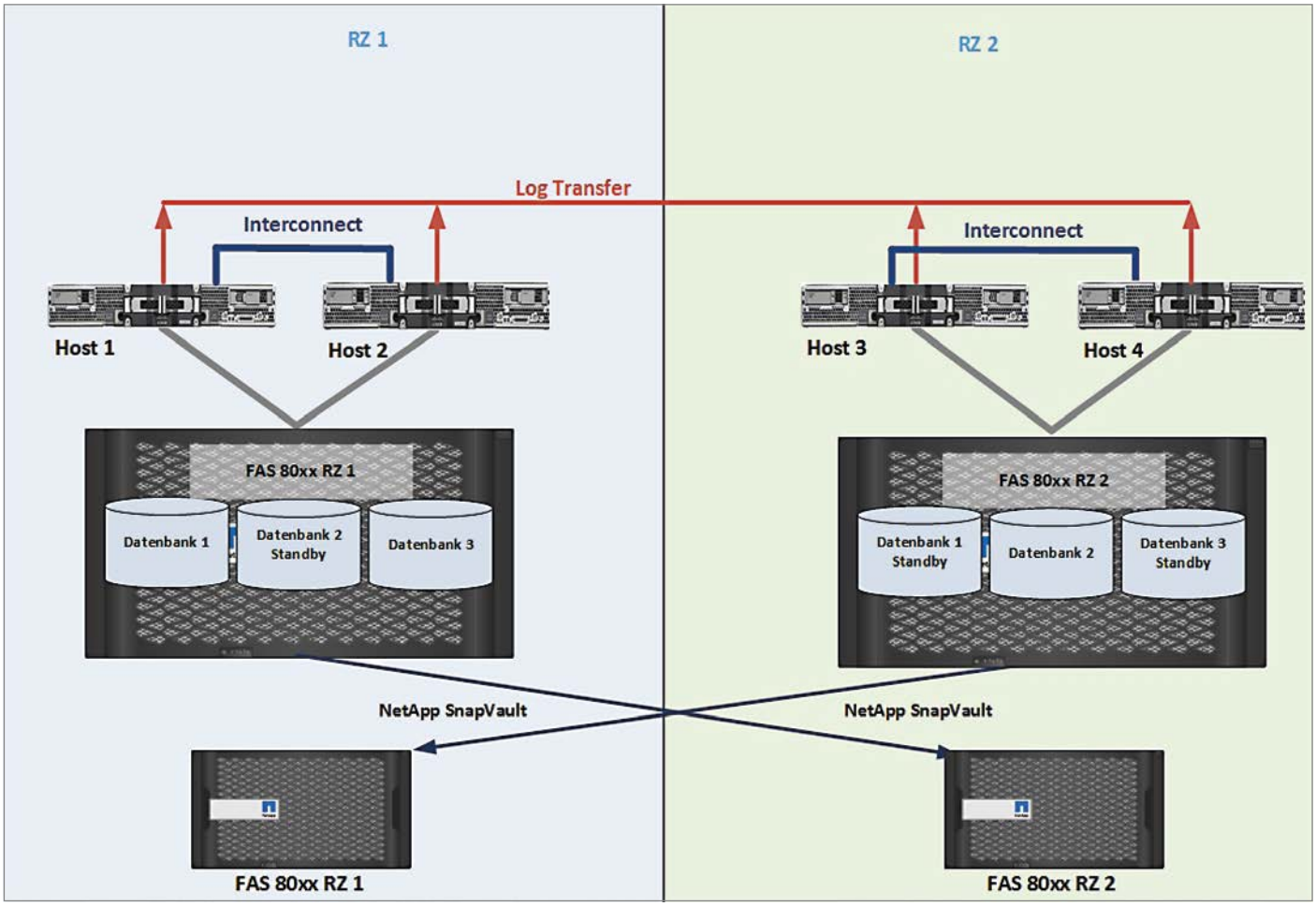

Abbildung 1: Jeweils zwei physikalische Server im RAC-Verbund

und Datenbank-Software auf einem gemeinsamen NetApp-Volume „/u01/app“. Hier ist bewusst auf ein Shared Oracle Home pro RAC verzichtet worden, um einen möglichen SPoF zu vermeiden und im Patching- und Upgrade-Verfahren flexibler zu sein.

Für jede Datenbank wurden jeweils sieben NetApp-Volumes eingerichtet. Dabei wurde zum einen auf Best Practices von NetApp beziehungsweise SnapCenter Rücksicht genommen, zum anderen ergab sich im Laufe des Projekts noch Optimierungspotenzial. So wurde zum Beispiel für die Flashback-Logs ein eigenes NFS-Volume eingerichtet, damit diese nicht unnötig Snapshot-Platz belegen. Das Sichern der Archive-Logs hingegen ist für das SnapCenter im Hinblick auf ein mögliches Recovery oder Cloning obligatorisch, wodurch diese also zwingend auf einem Volume abgelegt werden müssen, von dem Snapshots erzeugt werden. Aufgrund dieser Trennung und der gleichzeitigen Verwendung einer Fast Recovery Area wurden im Filesystem symbolische Links angelegt, die aus der FRA heraus auf ein separates Volume verweisen. Final stellen sich die Filesysteme für eine Datenbank wie in *Tabelle 1* dar, wobei lediglich für die ersten beiden Volumes Snapshots erzeugt werden.

File	Zweck
/u03/oradata/	Datenbank-Dateien, Spfiles und Password-Files
/u04/orabackup	Archive-Log-Dateien
/u03/redoA/	Redo- und Standby-Redo-Logs, Control-Files, Broker-Files
/u03/redoB/	Redo- und Standby-Redo-Logs, Control-Files, Broker-Files
/u03/temp/	Temp-Files
/u04/flashback	Flashback-Logs
/u04/rmanbackup	RMAN-Backups

Tabelle 1

Interessant ist dabei, dass die Control-Files nicht zwangsläufig auf einem Volume liegen müssen, von dem auch Snapshots generiert werden. Das SnapCenter holt sich zum Zeitpunkt des Sicherns der Datenbank die Information über die Lokation der Control-Files aus der Datenbank und sichert diese einmal als Kopie und einmal als Tracefile in den Snapshot der Daten-Dateien.

Damit die Datenbanken das Direct NFS verwenden, wurde die Datei „/etc/oranfstab“ erstellt. Entgegen der Vorgabe von Oracle „keep each „/etc/oranfstab“-file synchronized on all nodes“ (Oracle Grid Infrastructure Installation Guide 12.1) unterscheiden sich die „oranfstab“-Dateien auf den einzelnen RAC-Servern zwangsläufig, da durch redundante Storage-Anbindungen für die „local: Parameter“ jeweils vier unterschiedliche IP-Adressen pro Volume definiert werden müssen.

Das zuletzt aufgeführte Volume wird genutzt, um die Standby-Datenbanken mit dem RMAN zu sichern. Dies hat zum einen den Vorteil, dass Backups mit einem Oracle-eigenen Tool durchgeführt, und zum anderen − dem weitaus wichtigeren Aspekt −, dass hierdurch eventuelle korrupte Blöcke ausfindig gemacht werden können. Alternativ müsste man entsprechende Jobs wie DBVerify oder Analyze Validate Structure definieren, die die Primär-Datenbank belasten würden. So kann man davon ausgehen, dass auf der Standby-Seite keine korrupten Blöcke vorhanden sind und bei Bedarf wäre ein Block Media Recovery auf der Primär-Datenbank möglich. Voraussetzung ist ein vorhandener RMAN-Catalog.

Änderungen der Konfiguration

Nachdem die Infrastruktur bereits aufgebaut war, stieß man im laufenden Betrieb auf gewisse Fehler beziehungsweise Effekte, aufgrund derer die ursprünglichen Konfigurationen weniger Komponenten noch einmal angepasst wurden. Ursprünglich war eines der drei für die RAC-Cluster angelegten Volumes für die Voting Disks in dem jeweils entfernten Rechenzentrum angelegt. Die Voting Disks sind also von einem entfernten Storage gemountet worden, wodurch man sich eine höhere Ausfallsicherheit versprach. Die Praxis zeigte jedoch, dass dies bei Stromausfall oder Netzwerk-Unterbrechungen zum entfernten RZ auf dem hiesigen RAC unmittelbaren Einfluss hat. Zwar laufen die RAC-Cluster auch ohne die dritte Voting Disk, jedoch gibt es allein bei simplen Linux-Befehlen wie einem „df“ Probleme, da das gemountete NFS nicht zugreifbar ist. Infolgedessen wurde das Volume mit der dritten Voting Disk wieder auf das jeweils lokale Storage verlegt.

Entgegen der Oracle-Empfehlung für die OFA-Struktur wurden die Verzeichnisse für die Grid Infrastructure anders benannt: „/u01/app/grid/product/12.1.0.2/gridhome” für die Grid-Software und „/u01/app/gridbase“ für das Grid-Base. Das funktioniert auch ganz gut und ist nach Ansicht der Autoren so besser strukturiert. Allerdings führt dies gegebenenfalls dazu, dass das Grid Infrastructure Management Repository (GIMR) zunächst „falsch“ aufgebaut wird. Laut Oracle-Dokumentation (Oracle Grid Infrastructure Installation Guide 12.1) erkennt der Installer das „ORACLE_BASE“ selbst, wenn das Verzeichnis die Struktur „/u[0-9][1-9]/app/<osuser>“ hat. Da dies in der vorliegenden Konfiguration nicht der Fall war, wurde das GIMR fälschlicherweise mit dem „ORACLE_BASE /u01/app/oracle“ aufgebaut. Daraufhin wurde vorsichtshalber in der „.profile“ des Benutzers „grid“ die Variable „ORACLE_BASE“ auf „/u01/app/gridbase“ gesetzt.

Da für die Entwicklungsumgebung nur ein Server bereitgestellt wurde, installierte man hier eine Single-Instance-Konfiguration. Dies erwies sich allein bei der Ansible-Automatisierung durchweg als Spezialfall. Zudem hatte man sich einen Bruch in der Abbildung des Staging eingebaut. Später hat man die Konfiguration dieses einen Servers auf RAC geändert; zukünftig soll ein zweiter Server hinzukommen, um ein vollwertiges RAC, wenn auch kein Data Guard, in der Entwicklung zu haben.

Fazit

Nachdem die grundlegenden Planungen abgeschlossen waren, erstreckte sich der Aufbau der neuen Infrastruktur bis hin zum produktiven Betrieb der Datenbanken über einen Zeitraum von etwa einem halben Jahr. Hierzu sei aber gesagt, dass zwischenzeitlich jede der vier Umgebungen im Rahmen von streng definierten Abnahmetests umfangreich geprüft wurde. Hinzu kamen die im Artikel erwähnten an-

fänglichen Schwierigkeiten mit dem SnapCenter oder beispielsweise eine in ihrer Gänze neu implementierte Überwachung über den Oracle Enterprise Manager.

Die Zielsetzung der hohen Verfügbarkeit wurde erreicht. Bei einer geplanten Stromabschaltungs-Übung des gesamten Primary-Rechenzentrums beispielsweise wurden die Datenbanken und das Storage absichtlich nicht in den normalen Ablaufplan integriert. Dabei hätte man vorab einen Data Guard Switchover durchführen können; stattdessen konnte problemlos ein Failover initiiert werden, nachdem der Strom der gesamten Infrastruktur im ersten RZ gekappt wurde. Es sei erwähnt, dass zu diesem Zeitpunkt die Datenbanken noch keinen produktiven Status hatten. Durch die Aktivierung von Flashback auf allen Datenbanken ist man in der Lage, nach einem Failover durch ein Reinstate das Data Guard zeitnah wieder in einen konsistenten Zustand zu versetzen. Hierbei wird übrigens der Modus „Maximum Availability" eingesetzt.

Auch im Rahmen von Wartungsarbeiten zeigten sich die Vorteile des Data Guard insofern, als dass der Juli-PSU durch Standby First Apply nahezu ohne Downtime für die Applikationen implementiert werden konnte. Unschön ist weiterhin das leidige Thema der Java-Patches. Hier musste zwingend eine Downtime von etwa zehn Minuten her, um das Datapatch in die mit der OJVM betriebene Datenbank korrekt einzuspielen.

Das Backup- und Recovery-Verfahren durch das SnapCenter – wohlgemerkt das erste offizielle Release von NetApp – erledigt seine Aufgaben zuverlässig. Um die Datenbank per Snapshot zu sichern, wird diese kurzzeitig in den Backup-Modus („alter database begin backup;") gesetzt und die „Verpointerungen" auf Storage-Ebene gesetzt. Hat man sich einmal in der browserbasierten GUI zurechtgefunden, kann man schnell und einfach neben den regelmäßigen Sicherungsjobs auch Tasks wie Point-In-Time-Recovery oder gar einen Clone einer laufenden Datenbank erstellen. Dieser Clone kann sogar zeitlich in der Vergangenheit liegen, je nach Vorhaltezeit der Snapshots. Somit ist es durch SnapVault innerhalb weniger Minuten möglich, auch einen bis zu 35 Tage alten Snapshot für den Aufbau eines Clone zu verwenden. Wichtig in der Gesamt-Konstellation ist jeweils der Einsatz eines Secondary-Storage, worauf die aktuell erstellten Snapshots der Datenbank per SnapVault-Funktion kopiert werden, da ein Snapshot allein noch kein sicheres Backup darstellt.

Leider war es – entgegen den Angaben von NetApp – nicht möglich, die Standby-Datenbanken über das SnapCenter brauchbar zu sichern. Dies wäre nur über ein Offline-Backup möglich gewesen.

Alles in allem wurde eine stabile, zuverlässige Oracle-Infrastruktur geschaffen, die ihrer Bezeichnung „alway online" bisher voll gerecht wird.

Tim Hensel
tim.hensel@devk.de

Johannes Ahrends
johannes.ahrends@carajandb.com

Oracle VM und Virtual Shared Storage

Nico Henglmüller und Dr. Thomas Petrik, Sphinx IT Consulting GmbH

Die Architektur von Oracle VM sieht die Verantwortlichkeit der Speicherverwaltung bei externer Hardware. Eine solche ist für Klein- und Mittelbetriebe oft unerschwinglich. Dies führte die Autoren zur Frage: „Wie kann eine neuartige Infrastruktur konzipiert werden, die gleichzeitig hochausfallsicher, universell einsetzbar und einfach zu betreiben ist; noch dazu kostengünstiger als eine vergleichbare Enterprise-Lösung?" Eine ambitionierte Truppe machte sich auf die Suche und berichtet in diesem Artikel von der verwendeten Technologie und vom innovativen Ergebnis.

Zu Beginn stand die Idee, einen hochverfügbaren und leistbaren, Multipurpose Cluster mit Hard-Partitioning-Unterstützung von Oracle-Datenbanken zu entwickeln. Dieser Idee folgte ein Prototyp, bestehend aus zwei Oracle-VM-Knoten und einem Virtual Shared Storage. Die eingesetzten Knoten basieren auf Standard-x86-Architektur und verwenden ei-