

DB Hostwechsel leicht gemacht

Robert Ortel
Hypoport AG
Berlin

Schlüsselworte

Datenbank, Umzug, Hostwechsel, ASM, Diskgruppe, Cluster, Listener, URL, Service-Name

Einleitung

Eine Datenbank soll oder muss auf einen anderen Server umgezogen werden? Mit Datapump oder RMAN geht das leicht von der Hand, doch die Downtime steigt mit der Größe der DB. Dieser Teil gelingt deutlich schneller und einfacher mit ASM und einem Shared Storage und die Downtime reduziert sich auf wenige Minuten!

Doch nun müssen alle Anwendungen den neuen Server ansprechen. Das bedeutet häufig unterschiedlichste Konfigurationsänderungen. Doch man kann seine DB-Infrastruktur auch so verändern, das Hostnamen für die DBs im Unternehmen keine Rolle mehr spielen, sondern lediglich Service-Namen! Der Schlüssel ist ein Konstrukt, welches wir Listener Cluster nennen.

Motivation

Im Rahmen eines Oracle Lizenz Audits, bei dem die Hypoport AG von Beratern begleitet wurde, gab es die Herausforderung, quasi alle Datenbanken anders auf den verfügbaren Servern zu verteilen. Das war die Geburtsstunde der Überlegungen für einen einfachen Hostwechsel.

Zwei entscheidende Vereinfachungen halfen, die Umzüge mit hoher Akzeptanz und wenig Downtime durchzuführen: die Nutzung der Vorteile von ASM und Shared Storages sowie eines sog. Listener Clusters, womit URLs auch nach einem DB-Hostwechsel konstant bleiben.

Hostwechsel

Um eine Datenbank auf einem anderen Server zu nutzen, erscheint es als die einfachste Variante die Daten zu kopieren. Dafür kämen RMAN Backup & Restore oder Datapump Export und Import in Frage. Doch die schnellste Variante ist das ganz sicher nicht, bedenkt man wie groß DBs werden können. Kopiert man die Daten gar nicht, sondern macht sie einfach auf dem anderen Server verfügbar, ist dies die schnellste Methode.

Voraussetzung dafür ist ein Shared Storage, an das alle fraglichen Server angeschlossen sein müssen. Eine typische Möglichkeit das zu realisieren ist ein SAN (Storage Area Network) mit einem oder mehreren zentralen Storage(s), auf das alle relevanten Server Zugriff haben (siehe Abbildung 1). Die von einem SAN für Datenbanken in der Regel angebotenen Blockdevices sollten mit Oracle ASM (Automated Storage Management) zu einer oder mehreren Diskgruppen zusammengesetzt werden, welche jede Datenbank exklusiv als Speicherort für sämtliche Datendateien nutzt.

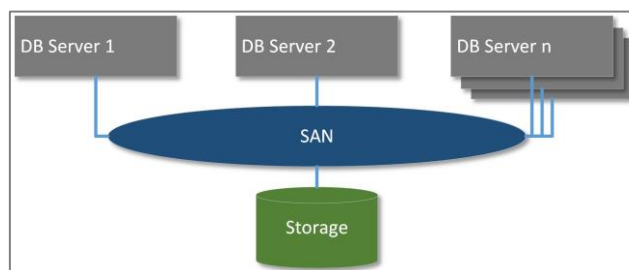


Abbildung 1, ein Storage Area Network (SAN) bei dem sich mehrere Server den Zugriff auf ein Storage teilen

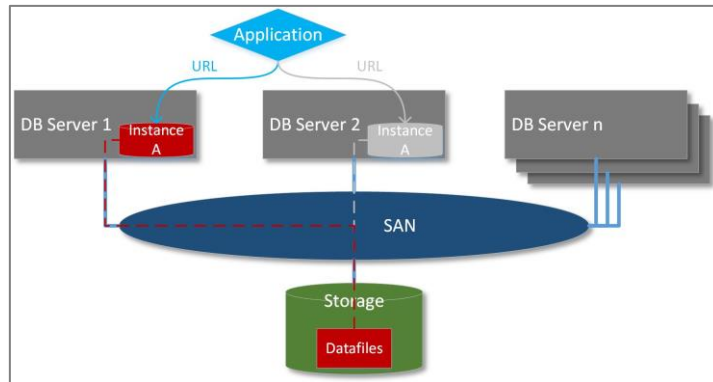


Abbildung 2, DB Instanz A vor einem Umzug auf einen anderen Server

Die den ASM-Diskgruppen zugrunde liegenden Blockdevices können im SAN und dem Storage so konfiguriert werden, dass mehr als ein Server Zugriff darauf hat. Das ist die Schlüssel-Voraussetzung für einen Hostwechsel einer Datenbank innerhalb sehr kurzer Zeit.

Alle anderen Voraussetzungen beziehen sich auf die nötigen Vorbereitungen auf dem Ziel-Server vor dem Umzug und variieren stark je nach Umgebung und Randbedingungen im jeweiligen Unternehmen. Dazu gehören Dinge wie die Installation der Oracle Software, Konfiguration von Listnern und des IO-Stacks (Sichtbarkeit der Blockdevices) bis hin zum Kopieren von Wallets, SPFILEs und der Einrichtung von Cron-Jobs. All diese Vorbereitungen erfordern jedoch keine Downtime der umzuziehenden DB und können während deren Betriebes vorgenommen werden. Sind diese Vorbereitungen abgeschlossen (siehe Abbildung 2) und steht eine Downtime von wenigen Minuten zur Verfügung, kann der Umzug durchgeführt werden. Er teilt sich in folgende Phasen ein:

1. Stoppen der DB(s) auf dem Quell-Server
2. Dismounting der zugehörigen Diskgruppe(n) der umzuziehenden DB(s) auf dem Quell-Server
3. Mounting der gleichen Diskgruppe(n) auf dem Ziel-Server
4. Einrichten der DB-Resource(n) in Grid Infrastructure auf dem Ziel-Server (Abhängigkeit der DB(s) von Diskgruppen, Umgebungsvariablen, Pfad zum SPFILE)
5. Starten der DB(s) auf dem Ziel-Server
6. Umstellung der Anwendungs-URLs

Nach diesen wenigen und einfachen Schritten ist der Umzug auf einen anderen Server bereits abgeschlossen (siehe Abbildung 3). Die Ausführungszeiten der einzelnen Schritte sind minimal, so dass sich die effektive Downtime für einen Umzug einer DB auf bis zu ca. 2 Minuten reduzieren lässt, sofern deren Runter- und Hochfahren schnell gelingt.

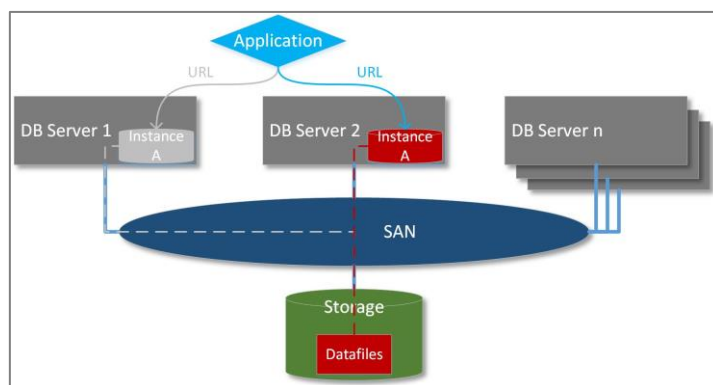


Abbildung 3, DB Instanz A nach einem Umzug auf einen anderen Server

Konstante URLs

Hat eine Datenbank den Server gewechselt, muss immer die zugehörige Anwendung umkonfiguriert werden, denn ihre Datenbank ist jetzt nur noch auf einem anderen Server erreichbar. Das ist bei einer umgezogenen DB, mit einer zugehörigen Anwendung, noch überschaubar, doch der Komplexität sind im Grunde keine Grenzen gesetzt und die nötigen Änderungen nach einem solchen Umzug können sehr vielfältig sein. Doch eine URL für eine DB kann von dem Server, auf dem die DB tatsächlich läuft, unabhängig gemacht werden. Allein der Service-Name identifiziert dann die gewünschte DB. Den Schlüssel dazu nennen wir Listener Cluster.

Ein solches Listener Cluster ist ein gewöhnliches Cluster auf Basis von Oracle CRS/Grid Infrastructure und wird auch genauso aufgesetzt. Es dient jedoch nicht dem Betrieb von Datenbanken (RAC), sondern allein dem Betrieb der SCAN-Infrastruktur, insbesondere des/der SCAN-Listener. Dieses Listener Cluster wird nach Umstellung aller URLs zu dem zentralen Element, das für jeden Verbindungsaufbau zu jeder Datenbank benötigt wird. Fällt es aus, ist keine Datenbank mehr erreichbar, ohne erneut die URLs zu ändern. Nur bereits existierende Verbindungen bleiben dann erhalten. Aus diesem Grund sind Maßnahmen zur Sicherstellung einer hohen Verfügbarkeit dringend geboten und das Vorhandensein eines Clusters sehr willkommen. So sind alle kritischen Ressourcen, mehr als einmal vorhanden und im Fehlerfall werden diese Ausweichressourcen automatisch verwendet.

Da vorhandene Datenbanken sich an Listnern einer höheren Version registrieren können, empfehle ich die Installation der aktuell höchsten Version: 12.1.0.2. Das bewahrt auch vorerst vor Update-Installationen.

Der SCAN-Hostname, welcher bei der Installation des Clusters angegeben wurde, und die SCAN-VIPs auflöst (hier: `lsnr-cluster-ks.scan.hypoport.local`) dient zusammen mit dem Port der SCAN-Listener (hier: 1521) für alle DBs, die von diesem Konstrukt profitieren sollen, nun als `remote_listener`. Der Parameter `local_listener` bleibt hinsichtlich der Einstellungen zur Nutzung des Listener Clusters unverändert und zeigt üblicherweise auf einen Listener Prozess, der auf dem entsprechenden DB Server läuft.

```
SQL> alter system set remote_listener='lsnr-cluster-ks.scan.hypoport.local:1521' scope=both;
```

Jede Datenbank mit diesem gesetzten Parameter wird daraufhin versuchen, all ihre Services (`db_unique_name` und `service_names`) jeweils mit dem `db_name` als zugehörige SID bei dem `remote_listener` zu registrieren. Das gelingt jedoch zunächst nicht, weil eine Funktion namens „Valid Node Checking For Registration“ (VNCR) nur Datenbanken, welche direkt Teil des

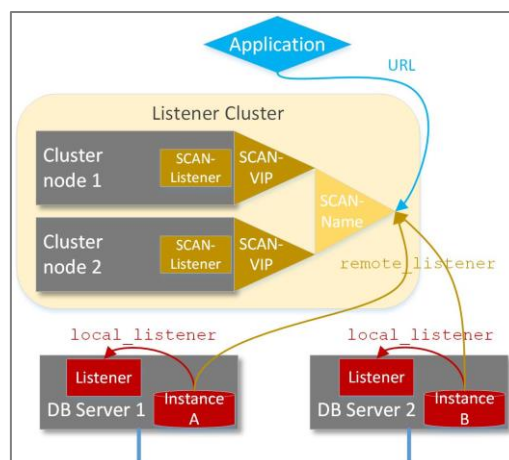


Abbildung 4, Umgebung mit Listener Cluster im Einsatz

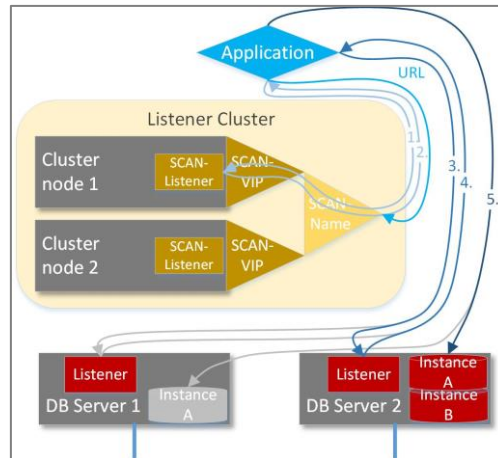


Abbildung 5, Verbindungsaufbau zur DB über das Listener Cluster und die Unabhängigkeit der URL vom DB Server

Clusters sind (RAC), für die Registrierung zulässt. Die Funktion kann nicht abgeschaltet werden, doch andere Server oder Netze können mit `srvctl` freigeschaltet werden (z.B. `srvctl modify scan_listener -update -invitednodes 192.168.100.*`).

Sind die gewünschten Services bei den SCAN-Listnern registriert, ermöglicht das die URLs der Anwendungen für diese DBs umzustellen. Während zuvor der echte Hostname noch Teil der URL war, sollte dieser jetzt durch den SCAN-Hostname ersetzt werden:

```
jdbc:oracle:oci:(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP) (HOST = lsnr-cluster-ks.scan.hypoport.local) (PORT = 1521) ) (CONNECT_DATA = (SERVER = DEDICATED) (SERVICE_NAME = PLAY_KS) ) )
```

Mit dieser Änderung der URL sind alle Schlüsselbedingungen geschaffen, um das Listener Cluster nutzen zu können. Abbildung 4 skizziert eine solche Umgebung.

Wird mit einer solchen URL eine Verbindung aufgebaut, antwortet der SCAN-Listener der Anwendung mit den Verbindungsdaten des `local_listener` der angefragten DB. So wendet sich die Anwendung an den `local_listener` und wird so wie üblich zur DB vermittelt. Der `local_listener` verliert nun nicht etwa an Bedeutung, auch wenn man das erwarten könnte. Läuft dieser nicht, kommt ebenfalls keine neue Verbindung zu den zugehörigen DBs mehr zustande. Abbildung 5 skizziert die genannten Schritte des Verbindungsaufbaus, welche beim Einsatz des Listener Clusters stattfinden.

Zieht eine DB nun auf einen anderen Server um, so brechen ohnehin alle vorhandenen Verbindungen zusammen. Eine neue Verbindung mit einer URL, welche direkt auf den alten Server zeigt, führt jetzt natürlich zum Fehler. Eine neue Verbindung mit der URL über das Listener Cluster hingegen verbindet erfolgreich zur Ziel-Datenbank. Denn nachdem die DB ihren Umzug auf den neuen Server abgeschlossen hat, registriert sie sich - wie bei jedem Start - beim `remote_listener`, den SCAN-Listnern des Listener Clusters, und so gelingen künftige Verbindungsversuche über das Listener Cluster, egal auf welchem Server die DB läuft.

Listener Cluster und Ausfälle

Zuvor wurde bereits darauf hingewiesen, welche zentrale Bedeutung dem Listener Cluster nun zukommt und wie wichtig die hohe Verfügbarkeit dieses Konstruktes ist. Fällt ein Knoten des Listener Clusters aus, werden der SCAN-Listener und die SCAN-VIP, die auf diesem Knoten liefen, auf einen der verbleibenden Cluster-Knoten umgezogen. Es kommt dadurch zu keinen Einschränkungen.

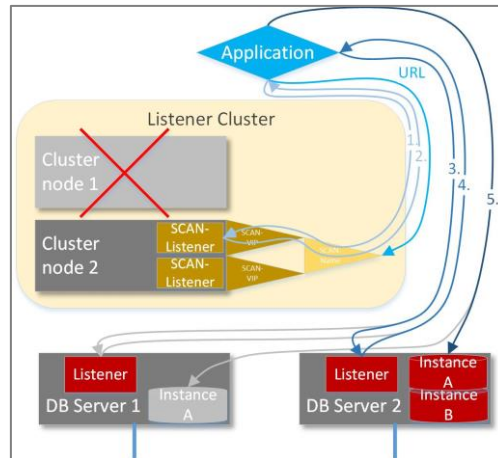


Abbildung 6, gesicherte Konnektivität auch bei Ausfall eines Listener Cluster Knotens

Abbildung 6 veranschaulicht den Verbindungsaufbau bei einem ausgefallenen Cluster Knoten. Bei einem Cluster aus 2 Knoten darf es aber zu keinem weiteren Ausfall eines Knotens kommen, bevor das Listener Cluster und damit die Konnektivität zu den DBs ausfällt. Zusätzliche Knoten empfehlen sich bei Bedarf nach höherer Hochverfügbarkeit. Solange mindestens ein Cluster Knoten aktiv bleibt, ist die Konnektivität gesichert.

Für eine hohe Verfügbarkeit bietet es sich auch an, die Diskgruppe des Listener Clusters, welche die Voting Disks und die Cluster Registry enthält, mit einer höheren Redundancy zu versehen (Normal oder High), um Ausfälle einzelner zugehöriger Disks abzufangen, sofern im SAN keine anderen Mechanismen deren Hochverfügbarkeit sichern. Jeder Cluster Knoten, auf dem diese Diskgruppe nicht mehr verfügbar ist, scheidet aus dem Cluster aus. Das gilt es zu vermeiden.

Lizenzrechtliches

Die Installation eines Oracle Clusters geschieht üblicherweise als Vorbereitung für ein RAC (Real Application Cluster) und erfordert dann auch die entsprechenden Lizenzen. Die Installation der Cluster-Software, ohne die Verwendung mit Datenbanken für ein RAC, erfordert jedoch überraschenderweise keine Lizenz. Oracle findet dazu in Ihrer eigenen Dokumentation klare Worte (http://docs.oracle.com/cd/E50790_01/doc/license.121/e17614/editions.htm#BABHIAAF).

„... Oracle Clusterware may be used to protect any application (restarting or failing over the application in the event of a failure) on any server, free of charge. ...“

Auch ohne vorhandene Lizenzen für RAC kann das hier beschriebene Listener Cluster installiert und verwendet werden. Für den Support seitens Oracle gelten dann jedoch Einschränkungen (siehe auch Dokumentation), die im Einzelfall abzuwägen sind.

Das öffnet die Tür für die Installation des Listener Clusters in VMs, wo die Lizenzierungspolitik von Oracle sonst schnell hohe Lizenzkosten verursacht.

Versionswechsel

Früher oder später ist für jede Datenbank ein Update auf eine höhere Version fällig. Das Listener Cluster bleibt da nicht außen vor, denn diese Listener auf einer alten Version zu belassen führt zu Kompatibilitätsproblemen. Ein Listener darf jedoch eine höhere Version als die Datenbank haben und so nimmt das Listener Cluster im Update-Plan einen der vorderen Plätze ein.

Aus meiner Sicht ist die ideale Update-Methode aber nicht, das Update des vorhandenen Listener Clusters, sondern das Ersetzen durch ein neues Listener Cluster. Der Einsatz von VMs (siehe vorheriger Abschnitt) nimmt dem den Schrecken.

Das neue Cluster kann, sobald bereit, in einer kurzen Downtime von wenigen Minuten den SCAN-Namen des alten Clusters übernehmen (`srvctl modify scan -n <scan_name>`) und damit zum aktiven Listener Cluster werden. Änderungen an den URLs werden bei diesem Update ebenfalls vermieden. Gleichzeitig bleiben so vor der Aktivierung Spielräume für Tests. Im Anschluss nimmt man das alte Listener Cluster außer Betrieb und gibt die freigewordenen IP Adressen wieder frei.

Schlusswort

Natürlich bedeutet ein solches Listener Cluster auch zusätzliche Aufwände für Hardware, Installation und Wartung. Doch die gewonnene Flexibilität und Verkürzung von Downtimes, beim Umzug von DBs auf andere Server, durch beide Techniken überwiegt diese Aufwände deutlich. Als einziger DBA bei Hypoport möchte ich diese Möglichkeiten nicht mehr missen.

Kontaktdaten:

Robert Ortel
Hypoport AG
Klosterstraße, 71
D-10179 Berlin

Telefon: +49 (0) 30-42086 2206
Fax: +49 (0) 30-42086 1999
E-Mail: robert.ortel@hypoport.de
Internet: www.hypoport.de