Von der Clusterware zur Grid Infrastructure
Upgrade von Oracle RAC

Dr. Frank Haney
Consultant
Jena

Schlüsselworte:

Oracle Database, Clusterware, Grid Infrastructure, RAC, Upgrade

Einleitung

Releasewechsel im Datenbankumfeld sind eine ziemlich komplexe Aufgabe. Häufig sind damit ein kompletter Wechsel der Infrastruktur und ein Upgrade der Applikation verbunden. Ziel ist, einerseits ein möglichst reproduzierbares und strukturiertes Verfahren zu haben, und andererseits die Ausfallzeit so weit als möglich zu minimieren. Die Spezifik von RAC-Umgebungen besteht darin, daß bei Upgrades des immer zuerst das Cluster auf den neuen Releasestand gebracht werden muß, ehe an das Upgrade der Datenbanken gegangen werden kann. Im folgenden Beitrag soll das Clusterupgrade näher betrachtet werden. Auf das Datenbankupgrade wird nur im Rahmen der Besonderheiten von RAC-Umgebungen eingegangen. Der Schwerpunkt liegt dabei auf den Erfahrungen beim Upgrade der Clusterware zur Grid Infrastructure 11*g* Release 2. Einige Reminiszenzen an das Upgrade von 9 nach 10 und von 10 nach 11 Release 1 sollen aber nicht ausgespart werden.

Allgemeine Herausforderungen und Szenarien

Bei der Planung des Upgrade steht zunächst die Frage, ob neue Hardware zur Verfügung steht oder nicht. Wenn neue Hardware zur Verfügung steht, dann hat das den Vorteil, daß die neue Umgebung ohne jede Beeinträchtigung der laufenden Applikationen implementiert werden kann. Die neue Clusterware bzw. Grid Infrastructure kann sauber separat installiert und getestet werden, bevor die Datenbanken nach und nach auf das neue Release gebracht werden. Nachteile sind die mit separater Hardware verbundenen Kosten und der zusätzlich im Storage benötigte Platz. Außerdem erfordert die eigentliche Datenbankmigration auf die neue Plattform relativ viel Zeit, die dann auch Ausfallzeit für die Applikation bedeutet, weil die zu bevorzugende Methode Export/Import bzw. seit Oracle 10*g* Datapump vergleichsweise langsam ist.
Wenn keine neue Hardware zur Verfügung steht, dann muß sowohl das Upgrade des Clusters als auch der Datenbanken am Ort erfolgen. Das hat den Nachteil, daß ein fehlgeschlagenes Upgrade des Clusters die Verfügbarkeit der Applikation dramatisch beeinträchtigen kann. Außerdem kann es Konflikte geben, wenn nicht alle Datenbanken gleichzeitig auf den neuen Releasestand gebracht werden können. Der Vorteil besteht in einer deutlich reduzierten Ausfallzeit, wenn das Upgrade der Datenbanken nicht mit Export/Import, sondern als Upgrade der Datenbank selber durchgeführt wird, sei es manuell oder mit dem DBUA.
Im vorliegenden Fall stand keine neue Hardware zur Verfügung, und zusätzlich warf die Infrastruktur ein paar besondere Probleme für die Gestaltung des Upgrade auf:

- Die Server haben keine lokalen Platten. Die Installation wird für alle Knoten in den gleichen Volume eines NetApp-Filers vorgenommen. Es geht also zunächst um die Schaffung identischer lokaler Pfade für die Installationen auf den Servern.

- Alle Server eines Clusters teilen sich eine OS-Installation (Linux x86-64), d.h. schreiben in das gleiche root-Dateisystem.
- Jede Installation (CRS, Datenbank, Agent etc.) gehört einem eigenen OS-User.
- Jede Installation hat eigenes `/etc/oraInst.loc` d.h. globales Inventory. Die Installationen haben keine „Kenntnis“ voneinander.

Vom Cluster Manager zur Clusterware

Das Upgrade des Oracle 9*i* Cluster Managers zur 10*g* Clusterware ist ein einschneidender Architekturwechsel. Die Clusterware ist eine völlig neue Softwareschicht. Deswegen war das Upgrade „in place“ auch ein sehr komplexer Prozeß, der ausgiebige Tests erforderte, ehe man das Upgrade der produktiven Umgebung wagen konnte. Der entscheidende Punkt war das Upgrade der Konfigurationsdatei des 9*i*-Clusters zur Cluster Registry 10*g*, wenn `$ORA_CRS_HOME/root.sh` auf dem ersten Knoten läuft. Das schlug regelmäßig fehl. In `ocrconfig.log` stand:

```
[ OCRCONF][2546082016]ocrconfig starts...
[ OCRCONF][2546082016]Upgrading OCR data
[ OCRRAW][2546082016]propriogid:1: INVALID FORMAT
[ OCRRAW][2546082016]ibctx:1:ERROR: INVALID FORMAT
[ OCRRAW][2546082016]proprinit:problem reading the bootblock or superbloc 22
```

Auch ein Service Request konnte keine Abhilfe schaffen. Insgesamt ist das Upgrade so schlecht dokumentiert, daß zweifelhaft ist, ob das je einer wirklich „in place“ durchgeführt hat.
Es wurde dann eine völlig autonome Installation der 10*g* Clusterware vorgenommen. Das 9i-Cluster wurde heruntergefahren und seine Konfigurationsdatei zunächst „versteckt“. Die Installation hatte keine “Kenntnis” davon, daß auf den Knoten bereits der Cluster Manager 9*i* installiert und konfiguriert war. Danach konnte man das 9i-Cluster bei laufenden 10g-Cluster neu starten. Beide liefen stabil parallel. Die Datenbanken konnten jetzt nach Bedarf nacheinander auf das neue Release gebracht werden. Beim Upgrade des Clusters entstand einmalig eine Ausfallzeit für alle Datenbanken von ca. 2 Stunden, danach nur noch für die Datenbank die gerade auf das neue Release gebracht wurden. Dieser Zeitbedarf war dann relativ unabhängig von der Datenbankgröße. Da noch zusätzliche Einstellungen für den Betrieb der eBusiness-Suite vorgenommen werden mußten, betrug dieser etwas 4 Stunden.

Nur ein kleiner Schritt: Das Upgrade der Clusterware nach 11*g* Release 1

Nach den vorangegangenen Erfahrungen hatte ich für den nächsten Schritt, das Upgrade von 10.2 nach 11.1 das Schlimmste erwartet, wurde aber angenehm enttäuscht. Nach nicht mal einer ganzen Stunde war der Vorgang beendet. Das liegt natürlich hauptsächlich daran, daß der Sprung von der Clusterware 10.2 nach 11.1 technologisch eher klein ist. Am Ende der Installation wird das Skript `$ORA_CRS_HOME/install/rootupgrade` nacheinander auf den Knoten ausgeführt. Dieses führt das eigentliche Upgrade durch. Dadurch ist es ein Rolling Upgrade ohne Ausfallzeit für Cluster, Datenbanken und Applikationen. Ausfallzeit entsteht erst dann, wenn man auch die Datenbanken auf den Stand von 11*g* Release 1 bringen will.

Oracle Grid Infrastructure 11*g* Release 2: Neue Features aus der Sicht des Upgrade

Der Schritt von 11.1 nach 11.2 ist deutlich größer als der vorherige nach 11.1, weil nicht nur etliche neue Features hinzugekommen sind, sondern sich Architektur und Prozeßstruktur einschneidend verändert haben. Das schlägt sich auch im Upgrade nieder, das deutlich mehr Überlegungen im Vorfeld zur Beherrschung der Komplexität erfordert und ausgiebig getestet sein will. Diese Überlegungen betreffen auch den Einsatz der neuen Features. Das soll im folgenden etwas näher beleuchtet werden:

- Es gibt jetzt einen einzigen init-Prozeß, den *Oracle High Availability Daemon* (OHASD), gestartet vom Skript `/etc/init.ohasd`, von dem aus kaskadierend alle anderen Prozesse gestartet werden. Das sind deutlich mehr als in vorherigen Releases. Näheres dazu in der Support Note 1053147.1.
- *Automatic Storage Management* (ASM) wird nicht mehr gebündelt mit der Datenbanksoftware ausgeliefert und bei Bedarf mit dieser installiert, sondern mit der Grid-Infrastructure-Software. Hintergrund ist, daß einerseits ASM systematisch eher zum Grid als zur Datenbank gehört, und andererseits die für den Betrieb des Clusters wesentlichen Dateien Voting und OCR jetzt in ASM gespeichert werden können. ASM muß also unabhängig von der Datenbankinstallation verfügbar sein. Das ist auch wichtig für die Verwendung des neuen ASM Cluster File System (ACFS), das die Speicherung von Nichtdatenbankdateien in ASM ermöglicht. Ungünstig ist, daß ASM beim Upgrade nicht abgewählt werden kann, auch wenn man es nicht verwendet und in der Zukunft nicht verwenden will. Nach dem Upgrade läuft auf jedem Knoten eine ASM-Instanz.
- Es gibt die Möglichkeit, das Cluster im Netz über einen einzigen Namen anzusprechen, der auf bis zu 3 virtuelle IP-Adressen aufgelöst werden kann. Dieser *Single Cluster Access Name* (SCAN) muß im DNS aufgelöst werden. Mit dem SCAN ist ein SCAN-Listener verbunden, der die Connect Requests an die Listener der einzelnen Knoten verteilt. Der SCAN-Listener ist optional, kann aber beim Upgrade nicht abgewählt werden, man braucht also mindestens eine freie IP-Adresse im öffentlichen Netz. Da aber aus dem öffentlichen Netz auch die Adressen der physischen und virtuellen Knoten stammen, könnte es schnell eng mit Adressen werden, wenn man in dem gleichen Netz mehrere Cluster betreiben will, besonders wenn dieses Netz vielleicht auch noch segmentiert ist. Deswegen sollte das Upgrade nach Grid Infrastructure 11.2 mit grundsätzlicheren Überlegungen zum Design der IT-Landschaft einhergehen. Warum muß man für jede Applikation bzw. für jeden Mandanten ein eigenes Cluster betreiben, wenn sich der gleiche Effekt viel dynamischer mit Server Pools in einem größeren, übergreifenden Cluster realisieren läßt?
- Auch beim Upgrade erfolgt die Installation in ein neues Oracle Home. Dieses darf weder ein Unterverzeichnis des Home-Verzeichnis des OS-Users noch das Unterverzeichnis irgendeines Oracle-Base-Verzeichnisses. Wenn in 11.1 diesbezüglich nur eine Warnung bei der Ausführung von `$ORA_CRS_HOME/root.sh` angezeigt wurde, läßt sich jetzt die Installation nicht fortsetzen, wenn der OUI das feststellt. Wenn es gelungen ist, dem Installer diese Tatsache z.B. dadurch zu verbergen, daß man einen Volume zweimal unter verschiedenen Pfaden mountet, wird man hinterher feststellen müssen, daß dem User sein eigenes Home-Verzeichnis oder das Oracle-Base-Verzeichnis nicht mehr gehören, was entsprechende Seiteneffekte haben kann.
- Die Grid Infrastructure bringt eine eigene Zeitsynchronisation mit, den *Cluster Time Synchronization Service* (CTSS). Dieser ist aber nur aktiv, wenn der Installer keine andere Zeitsynchronisation (z.B. mittels NTP) findet. Ansonsten verbleibt er im Observer-Status. Wenn nun NTP nicht richtig konfiguriert ist, dann hat man gar keine Zeitsynchronisation.
- Backups der Voting Disk werden jetzt automatisch erstellt. Entsprechende eigene Routinen kann man also entfernen.
- Das *Cluster Verification Utility* (CVU) ist jetzt in den Installer integriert und prüft während der Installation die Voraussetzungen.

Ablauf und Probleme des Upgrade

Die Installation wird normal gestartet, man wählt Upgrade und macht die geforderten Angaben, so die Eingabe des Installationspfades, Namen und Port des SCAN-Listeners. Dann prüft das CVU die Voraussetzungen für die Installation. Wenn möglich, stellt es sogar Skripte zur Verfügung, die fehlende bzw. fehlerhafte Einstellungen korrigieren. Danach folgt die eigentliche Installation. Bei dieser läuft das Cluster zunächst völlig unbeeinträchtigt im alten Release. Am Ende der Installation muß das Skript `$GRID_HOME/rootupgrade` ausgeführt werden. Das läuft unter den gegebenen Umständen

auf dem ersten Knoten auch sauber durch, schlägt aber auf dem nächsten (zweiten) Knoten fehl. Die Ursache ist, daß das Skript die alten init-Skripte löschen will, aber nicht mehr findet, weil sie schon vom ersten Knoten aus gelöscht worden sind. (Bei einem Downgrade werden die Skripte aus dem alten $ORA_CRS_HOME zurückgeschrieben.) Das Problem entsteht dadurch, daß alle Knoten in das gleiche root-Dateisystem schreiben. (Aus diesem Grunde mußte schon bei der Installation der Clusterware 10g das Verzeichnis `/etc/oracle` auf einen lokalen Pfad verlinkt werden.) Die Lösung ist hier, die Skripte zu sichern und jeweils vor Start von `$GRID_HOME/rootupgrade` zurückzukopieren. Danach läuft das Skript sauber durch.
Es trat aber noch ein weiteres Problem auf: Das Upgrade ist erst abgeschlossen, wenn das Skript auf dem letzten Knoten gelaufen ist. Erst dann zeigt der Befehl `crsctl query crs activeversion` die Ausgabe 11.2.0.1.0. Im vorliegenden Fall war das Cluster aber auf dem ersten Knoten heruntergefahren und ließ sich auch nicht mehr starten, nachdem das Skript auf dem zweiten Knoten gelaufen war. Was war die Ursache? Auch hier war es wieder das für alle Knoten gleiche root-Dateisystem. Im Unterschied zu den alten init-Skripten ist das neue `/etc/init.ohasd` spezifisch für jeden Knoten, enthält z.B. den jeweiligen Knotennamen. Es ist klar, daß sich das Cluster nicht mehr starten läßt, wenn der Eintrag dort nicht mehr mit dem Resultat von `hostname` übereinstimmt. Das Skript mußte also auf einen lokalen Pfad verlinkt werden. (Das mußte schon bei der Installation der Clusterware 10g für das Verzeichnis `/etc/oracle` getan werden.) Genauso muß man auch mit der Konfigurationsdatei `/etc/ohasd` und den Verzeichnissen `/opt/oracle` und `/opt/ORCLfmap` verfahren.
Nachdem das Skript `$GRID_HOME/rootupgrade` auf dem letzten Knoten gelaufen ist, erfolgt die Konfiguration des Clusters. Dabei traten drei Fehler auf (Screenshots dazu im Vortrag):

1. Der *Net Configuration Assistent* bringt einen Fehler beim Upgrade des Listeners

   ```
   INFO: Oracle Net Services Configuration:
   INFO: Listener "LISTENER10" will be migrated to Oracle home: <$GRID_HOME>
   INFO: Failed migrating Oracle Net Services configuration. The exit code is 1.
   ```

   Ursache: Das Upgrade versucht, den unter dem Eigentümer der Datenbank laufenden Listener LISTENER10 auf das neue Release zu bringen. Das schlägt fehl, weil der Listener nicht LISTENER heißt. Ein in der Grid Infrastructure laufender Listener muß aber LISTENER heißen.
   Lösung: Listener mit 10.2-NETCA entfernen und mit 11.2-NETCA neu anlegen.

2. Der Installer zeigt den zunächst wenig aussagekräftigen Fehler:
   ```
   [INS 20802] The Oracle Cluster Verification Utility failed
   The plug-in failed in it's perform method
   ```
   Bei einer Durchsicht des Installer-Log stellte sich dann heraus, daß keine Zeitsynchronisation stattfindet, weil der CTSSD nicht läuft, obwohl NTP abgeschaltet wurde.

   ```
   INFO: CTSS is in Observer state. Switching over to clock synchronization checks
   using NTP
   INFO: Starting Clock synchronization checks using Network Time Protocol(NTP)...
   INFO: NTP Configuration file check started...
   INFO: NTP Configuration file check passed
   INFO: Checking daemon liveness...
   INFO: Liveness check failed for "ntpd"
   INFO: Check failed on nodes:
   INFO:   node1,node2
   INFO: PRVF-5415 : Check to see if NTP daemon is running failed
   INFO: Clock synchronization check using Network Time Protocol(NTP) failed
   INFO: PRVF-9652 : Cluster Time Synchronization Services check failed
   ```

Lösung: Wenn der CTSSD für die Zeitsysnchronisation sorgen soll, darf kein anderer Dienst dafür konfiguriert sein! Also müssen `/etc/ntp.conf` oder `/etc/xntp.conf` entfernt werden.

3. Nach dem Upgrade ist die Konfiguration des öffentlichen Interface inkorrekt. Das tritt in segmentierten Netzen auf (Subnetz z.B. 255.255.255.64). Der Befehl `oifcgf getif` zeigt nur den Interconnect. Der Befehl `srvctl config nodeapps –a` zeigt kein Interface zu den virtuellen IP-Adressen.

   Lösung: Das virtuelle Interface muß entsprechend Support Note 276434.1 "Modifying the VIP or VIP Hostname of a 10g or 11g Oracle Clusterware Node" neu konfiguriert werden. Dann ist die Ausgabe korrekt:

```
>oifcfg getif -global
eth3  192.168.178.0  global  cluster_interconnect
eth1  10.10.10.64  global  public
>srvctl config nodeapps  -a
VIP exists.:node2
VIP exists.: /10.10.10.4/10.10.10.4/255.255.255.192/eth1
VIP exists.:node1
VIP exists.: /10.10.10.3/10.10.10.3/255.255.255.192/eth1
```

Danach sollte die Oracle 11g Release 2 Grid Infrastructure sauber laufen.

Besonderheiten des Datenbankupgrade

Das Datenbankupgrade wurde manuell ausgeführt. Meine Erfahrungen beim Upgrade von RAC-Datenbanken mit dem DBUA waren in der Vergangenheit nicht besonders gut. Das mag aber eher gegen mich als gegen den DBUA sprechen. Hier sollen drei Spezifika eines manuellen Upgrades von RAC-Datenbanken genannt werden:

- Die Installation der Software führt zu folgender, nicht ignorierbarer Fehlermeldung:

  ```
  [INS-35354] The system on which you are attempting to install Oracle RAC is not part of a valid cluster
  ```

  Ursache: Die Datenbankinstallation hat einen eigenen OS-User und ein eigenes `oraInst.loc`, dadurch hat sie keine Kenntnis von der vorhandenen Grid Infrastructure.
  Lösung: Man kann `<oraInst.loc>/ContentsXML/inventory.xml` editieren und das $GRID_HOME der HOME_LIST hinzufügen. Eleganter und außerdem supportet ist es, mit dem Installer das $GRID_HOME dem Inventory der Datenbankinstallation hinzuzufügen:

  ```
  ./runInstaller -silent -ignoreSysPrereqs -attachHome ORACLE_HOME=$GRID_HOME
  ORACLE_HOME_NAME="Ora11g_gridinfrahome1" LOCAL_NODE='node1'
  CLUSTER_NODES=node1,node2 CRS=true
  ```

- Das eigentliche Datenbankupgrade kann nicht als RAC-Datenbank durchgeführt werden. Die Datenbank muß exklusiv im Zugriff eines Knotens sein. Deswegen muß der Parameter CLUSTER_DATABASE auf False gesetzt und die Datenbank neu gestartet werden. Dann kann das Upgrade des Data Dictionary (Catalog Upgrade) erfolgen:

  ```
  ALTER SYSTEM SET CLUSTER_DATABASE=false SCOPE=spfile;
  host srvctl stop database –d LB10
  STARTUP UPGRADE
  ```

```
@?/rdbms/admin/catupgrd.sql
ALTER SYSTEM SET CLUSTER_DATABASE=true SCOPE=spfile;
SHUTDOWN IMMEDIATE
```

- Die Datenbank muß dann mit dem neuen $ORACLE_HOME in der Cluster Registry (OCR) registriert und in der neuen Umgebung gestartet werden:

```
host srvctl remove database –d DB10
host srvctl add database –d DB10 –o $ORACLE_HOME
host srvctl add instance –d DB10 –i DB10-01 –n node1
host srvctl add instance –d DB10 –i DB10-02 –n node2
host srvctl start database –d DB10
```

Dann sollte alles laufen. Das kann man überprüfen:
`crsctl stat res -t -init`
zeigt alle Ressourcen, die vom init-Prozeß gestartet werden.
`crsctl stat res -t`
zeigt alle weiteren globalen und lokalen Ressourcen im Cluster, insbesondere auch die Datenbanken.

Fazit

Der Zeitbedarf für das Upgrade der Clusterware zur Grid Infrastructure liegt bei maximal zwei Stunden, wobei diese Zahl für die Verfügbarkeit ohne Belang ist, weil es sich um ein sogenanntes Rolling Upgrade handelt. Weder für das Cluster als Ganzes noch für die Datenbanken ist eine Auszeit erforderlich. Eine solche betrifft immer nur den jeweils zu upgradenden Knoten und die auf ihm laufenden RAC-Instanzen. Das Upgrade der Datenbanken selber erfordert eine Downtime für die Applikation. Diese ist abhängig von der Größe des Data Dictionary und von Anpassungen an Datenbank und Applikation, die für deren Interoperabilität vorgenommen werden müssen. Zwei bis vier Stunden sollten hier aber auf jeden Fall reichen.

Insgesamt ist das Upgrade Von Cluster und RAC-Datenbank nach 11g Release 2 ein ziemlich komplexes Unternehmen, das je nach Umgebung und Plattform auch einige Stolpersteine bereithält. Wichtig ist auch, gerade wenn man das Upgrade „in place“ durchführt, daß man eine Fallback-Strategie hat. Es gilt also, nicht nur das Upgrade, sondern auch das Downgrade ausgiebig zu testen. Eine gute Anleitung dafür bietet Support Note 969254.1. Dort wird beschrieben, wie man einen Downgrade durchführt, wenn das Upgrade fehlgeschlagen ist, entweder vor der Ausführung `rootupgrade.sh`, danach, oder während der Konfigurationsphase. Das kann dann als Ausgangspunkt für einen eventuellen neuen Upgradeversuch dienen.

Kontaktadresse:

Dr. Frank Haney
Anna-Siemsen-Str. 5
D-07745 Jena

Telefon: +49(0)3641-210224
E-Mail: info@haney.it
Internet: http://www.haney.it