

Oracle Restart: Die Grid Infrastructure für die Single Instance

Dr. Frank Haney, Jena

Oracle stellt eine Vielzahl von Hochverfügbarkeitslösungen zur Verfügung. Es ist dabei nicht trivial, die auszuwählen, die den jeweiligen Anforderungen am besten entspricht. Das gilt sowohl hinsichtlich der Verfügbarkeitsanforderungen als auch für wirtschaftliche Aspekte. Für einige ist die Grid Infrastructure die Basis.

Seit Oracle 11g R2 heißt die Clusterware „Grid Infrastructure“. Einer der wichtigsten Unterschiede zu vorherigen Versionen ist, dass ASM jetzt gebündelt mit der Clusterware- und nicht mehr mit der Datenbank-Software installiert wird, was die neue Namensgebung rechtfertigt. Folgende Varianten an Hochverfügbarkeitslösungen lassen sich auf der Basis der Grid Infrastructure realisieren:

- **Real Application Cluster (RAC)**
Die Datenbank läuft aktiv mit mehreren Instanzen auf verschiedenen Servern. Dabei handelt es sich um eine kostenpflichtige Option der Enterprise Edition. In der Standard Edition kann man sie ohne Zusatzkosten nutzen, sofern man die Beschränkungen der Standard Edition für das komplette Cluster einhält.
- **RAC One Node**
Hier darf die Datenbank nur auf einem Knoten mit einer Instanz aktiv laufen. Es handelt sich um eine Aktiv-Passiv-Lösung, die nur zusätzlich zur Enterprise Edition lizenzierbar ist, allerdings zu geringeren Kosten als das volle RAC. Im Bedarfsfall startet die Clusterware die Instanz auf einem anderen Knoten neu.
- **Failover-Datenbank**
Auch das ist eine Aktiv-Passiv-Lösung, bei der durch Skripte der Übergang der Datenbank-Ressource auf einen anderen Knoten erreicht wird. Das ist eigentlich nur noch interessant, wenn man in der Standard Edition die Lizenzbedingungen für RAC nicht erfüllen kann. Die Variante ist aber bezüglich des Supports nicht unproblematisch.

- **Fremdapplikationen**

Diese können mit der Grid Infrastructure hochverfügbar gemacht werden, ohne dass im Cluster eine Datenbank betrieben wird. Zu denken wäre hier an Application Server oder Enterprise Manager.

- **Oracle Restart**

Die Grid Infrastructure wird auf einem Stand-Alone-Server installiert. Es besteht nicht die Absicht, diesen zu einem Cluster zu erweitern.

Oracle Restart im Überblick

Das neue Feature in Oracle 11g R2 ist allerdings nur eingeschränkt in 11.2.0.1 nutzbar. Eine Implementierung ist erst ab 11.2.0.2 sinnvoll. Diese ist mit und ohne Verwendung von ASM möglich. Es gilt auch hier, dass die Grid Infrastructure den höchsten Release-Stand haben muss.

Oracle Restart ermöglicht den automatischen Restart der Oracle-Instanz und aller mit ihr verbundenen Ressourcen (Listener, Services, ASM, ASM Diskgroups) unter Beachtung der Abhängigkeiten nach einem Crash beziehungsweise einem beliebigen Reboot. Dabei wird auch der Zustand vor dem Crash berücksichtigt.

Gestartet werden die Ressourcen nicht vom „init“-Prozess oder dem jeweiligen Windows-Dienst, sondern vom Oracle High Availability Service (OHAS). Dieser bezieht die Information für die Starteigenschaften der jeweiligen Ressource aus der Oracle Local Registry (OLR). Oracle Restart hat eine Anbindung an Oracle Data Guard, was bewirkt, dass eine Standby-Datenbank immer in der richtigen Rolle startet.

Die Einbindung von Services in die Restart-Architektur ist unter Berücksichtigung der Datenbank-Rolle möglich. Ein Client verbindet sich dann, eine entsprechende TNS-Konfiguration vorausgesetzt, in einer Data-Guard-Umgebung immer mit dem Server, der die primäre Rolle hat. Auf Hochverfügbarkeitsereignisse kann mittels Fast Application Notification (FAN) reagiert werden. Eine Verarbeitung der Events ist dann über den Oracle Notification Service (ONS) oder serverseitige Callout-Skripte möglich.

Auch andere (Oracle- und Nicht-Oracle)-Applikationen können als lokale Ressource in die Restart-Umgebung eingebunden werden. Dadurch lässt sich ihr Start automatisieren. Notwendige Abhängigkeiten werden berücksichtigt und FAN-Callouts sind auch für diese Anwendungen möglich. Besonders sinnvoll ist das beispielsweise für den Grid beziehungsweise Cloud Control Agent oder auch für Database Control.

In der Dokumentation von Oracle 12c R1 ist das Feature überraschend als „deprecated“ deklariert. Die folgenden Ausführungen beziehen sich erst einmal auf 11g. Die Spezifika der Verwendung unter 12c werden später erläutert.

Besonderheiten der Installation

Die Implementierung beginnt mit der Installation der Grid Infrastructure für einen Stand-Alone-Server. Aber Achtung, diese Installationsoption erwartet die Verwendung von ASM. Bei der Installation muss man wenigstens eine Disk für eine initiale ASM-Diskgroup angeben und es wird dann eine ASM-Instanz gestartet. Das ist

jedoch nicht zwingend, weil ein Stand-Alone-Server keinen Shared Storage erfordert. Wenn man kein ASM verwenden will, dann muss die Installations-Option „Software Only“ ausgewählt werden.

Die Installation läuft anschließend komplett durch. Zum Schluss muss wie bei den meisten Installationen das Skript „\$ORACLE_HOME/root.sh“ ausgeführt werden. Wenn man ohne ASM installiert, fordert dieses zu Ausführung eines Perl-Skripts auf, ein analoges Skript gibt es für die Cluster-Installation (siehe Listing 1).

Nach erfolgreichem Lauf des Skripts ist die Installation komplett und der High Availability Service (HAS) läuft. Das kann man mit dem Befehl „crsctl check has“ überprüfen. Anschließend kann man mit dem Net Configuration Assistant („netca“) einen Listener konfigurieren und starten. Dieser verwendet wie bei einer Cluster-Installation das Grid-Infrastructure-Home.

Dann werden die Datenbank-Software in ein eigenes Home installiert und eine Datenbank angelegt. Bei den Grid-Installa-

tionsoptionen muss man „Datenbankinstallation mit nur einer Instanz“ auswählen. Beim Anlegen der Datenbank mit dem DBCA wird diese zum Schluss als HA-Ressource in Oracle Restart registriert, was der einzige Unterschied zu einer Datenbank-Erzeugung ohne Restart ist. Wenn man das Resultat kontrolliert, erscheint folgende Ausgabe (siehe Listing 2).

Verwaltung der Restart-Ressourcen

Genauso wie beim Cluster sind die beiden wichtigsten Werkzeuge „crsctl“ für die Verwaltung der Grid Infrastructure und „srvctl“ für die Verwaltung der Ressourcen. Es ist nicht notwendig, hier die komplette Syntax zu beschreiben; das kann nachgelesen werden. Ein paar Besonderheiten seien jedoch erwähnt. So kennt zum Beispiel Oracle Restart bei „crsctl“ die Argumente „cluster“ und „crs“ gar nicht. Stattdessen wird das Argument „has“ verwendet. Es heißt also „crsctl check|start|stop has“.

Dieses Kommando überprüft den Status des High-Availability-Service, startet bezie-

hungsweise stoppt ihn und startet beziehungsweise beendet auch alle in der OLR registrierten Applikationen unter Berücksichtigung der dort hinterlegten Abhängigkeiten. Was nun, wenn Datenbank und Listener schon vor der Restart-Implementierung vorhanden sind? Dann kann man sie mit „srvctl add database“ beziehungsweise „srvctl add listener“ hinzufügen.

Oracle Restart und Data Guard

Es gibt eine Integration von Oracle Restart und Data Guard. Diese erfordert aber den Data Guard Broker. Oracle Restart startet die Datenbank immer in der registrierten Rolle („PRIMARY“, „PHYSICAL_STANDBY“, „LOGICAL_STANDBY“ oder „SNAPSHOT_STANDBY“) und berücksichtigt gegebenenfalls Abhängigkeiten. Hinzu kommt, dass bei einem Rollenwechsel alle Services gestartet beziehungsweise gestoppt werden, die für die entsprechende Rolle konfiguriert sind. Hier als Beispiel das Anlegen eines Service, der nur läuft, wenn die Standby-Datenbank in der Rolle der primären ist: „srvctl add service -d ora11r2sb -s servsb -l primary“. Der Service muss nach dem Anlegen einmal manuell mit „srvctl start service -d ora11r2sb -s servsb“ gestartet werden. Dabei gibt es jedoch zwei Probleme:

1. Bei einem Failover oder Switchover wird zwar die neue Rolle der Datenbank berücksichtigt, aber der Open-Modus in der Restart-Konfiguration nicht geändert. Das wird allerdings erst ein Problem, wenn es zu einem Reboot des Servers kommt. Für die Lösung gibt es drei Möglichkeiten:
 - Nach jedem Rollentausch (Switch- oder Failover) wird der Open-Modus manuell mit „srvctl“ geändert. Das ist für das Switchover eine akzeptable Lösung, für das Failover kaum.
 - Implementierung eines „FAN-Call-out“, der die aktuelle Rolle der Datenbank ausliest und automatisch den Open-Modus ändert.
 - Gelöst in Patch 12596559 (on Top of 11.2.0.3.9 beziehungsweise in 11.2.0.4 etc.)
2. Ein Service für die Rolle „PHYSICAL_STANDBY“ lässt sich nicht starten, wenn die Standby-Datenbank nur ge-

```
hostname:/oracle/oragrid/11.2.0/grid# /oracle/oragrid/11.2.0/grid/perl/bin/perl
-I/oracle/oragrid/11.2.0/grid/perl/lib
-I/oracle/oragrid/11.2.0/grid/crs/install /oracle/oragrid/11.2.0/grid/crs/install/roothas.pl
```

Listing 1

```
$GRID_HOME/bin/crsctl stat res -t
-----
NAME                TARGET STATE         SERVER          STATE_DETAILS
-----
Local Resources
-----
ora.LISTENER.lsnr
                   ONLINE ONLINE        ora11r264
...
Cluster Resources
-----
...
ora.ora11r2.db
 1                ONLINE ONLINE        ora11r264      Open
```

Listing 2

```
#!/bin/ksh
umask 022
FAN_LOGFILE=/tmp/`hostname`.log
echo $* "reported="`date` >> $FAN_LOGFILE &
```

Listing 3

mountet ist. Das ist ein erwartetes Verhalten. Es gibt zwei Lösungen:

- Verwendung von Active Data Guard (kostenpflichtige Option)
- Services nur für die Rolle „PRIMARY“ anwenden

Reaktion auf Hochverfügbarkeits-Ereignisse

Statusänderungen in der Restart-Umgebung können an die Clients (Applikationen) propagiert werden. Das ist das Prinzip von Fast Application Notification (FAN) und erfordert die Verwendung von Services. Beispiele für solche HA-Events sind:

- Service nicht erreichbar
- Ausfall des Servers
- Rollenwechsel der Datenbank

Es gibt verschiedene Wege zur Propagierung der Information an die Clients:

- Verwendung FAN-integrierter Oracle-Clients („JDBC“, „ODP.NET“, „OCI“)
- Kommunikation mithilfe des Oracle Notification Service (ONS)
- Server Side Callouts (benutzerdefinierte Callout-Skripte)

Letzteres soll kurz dargestellt werden, um das Prinzip von FAN zu verdeutlichen:

1. Das Skript erstellen. Das hier „callout.sh“ genannte Skript macht nichts anderes als die Events in ein Logfile zu schreiben (siehe Listing 3).
2. Skript nach „\$GRID_HOME/racg/usrco“ kopieren.
3. Testen, etwa mit „ps -ef | grep smon“ und „kill -9 <smon ospid>“.
4. Die generierten Events haben folgende Struktur, etwa für Crash und automatischen Restart der Instanz (siehe Listing 4).
5. Den Logfile parsen und eine Reaktion implementieren: E-Mail, Stopp der Applikation etc. Das kann natürlich auch direkt im Skript erfolgen.

Integration einer „Fremd“-Applikation

Oracle Restart bietet auch die Möglichkeit, andere Applikationen einzubeziehen. Am Beispiel Database Control sind dazu folgende Schritte notwendig:

1. Es muss ein Action-Skript erstellt werden und dieses die Funktionen „start“, „stop“, „check“ sowie „clean“ für die zukünftige Ressource enthalten.
2. Das Skript wird getestet und nach „\$GRID_HOME/crs/public“ kopiert.
3. Die Ressource wird in der OLR registriert (siehe Listing 5).
4. Die Ressource muss mit „[oracle@ora11r264 bin]\$ crsctl start resource dbcontrol“ gestartet werden.
5. Man kann die Funktionalität testen, indem man den Prozess killt und nachsieht, ob Oracle Restart Database Control wieder startet.

Quo vadis Oracle Restart?

In der Oracle-12c-Dokumentation steht: „Oracle Restart is deprecated in Oracle Database 12c. Oracle Restart is currently restricted to manage single-instance Oracle databases and Oracle ASM instances only, and is subject to desupport in future releases. Oracle continues to provide Oracle ASM as part of the Oracle Grid Infrastructure installation for Standalone and Cluster deployments.“ Da das zu

Missverständnissen geführt hat, wird es in der MOS-Note 1584742.1 „Support Impact of the Deprecation Announcement of Oracle Restart with Oracle Database 12c“ näher erläutert:

- „Deprecated“ bedeutet in diesem Fall nicht, dass es bereits einen Ersatz gibt. Oracle weicht hier von der üblichen Vorgehensweise deutlich ab, siehe „Streams“.
- „Deprecated“ ist nicht gleich „Desupported“ – Oracle Restart wird nach wie vor voll unterstützt.
- Es gibt keinerlei Hinweis darauf, was Restart ersetzen könnte.
- Es muss aber einen Ersatz geben, schon wegen ASM auf Stand-Alone-Implementierungen.

Bei näherer Betrachtung ist der Stand der Dinge in 12c folgender:

- Es gibt keine grundsätzlich neue Funktionalität gegenüber 11g.
- Restart ist nach wie vor voll in den Installer integriert.

```
INSTANCE VERSION=1.0 service=ora12r1 database=ora12r1 instance=ora12r1
host=ora12r164bit status=down reason=FAILURE timestamp=2014-06-03
14:37:30
INSTANCE VERSION=1.0 service=ora12r1 database=ora12r1 instance=ora12r1
host=ora12r164bit status=up reason=FAILURE timestamp=2014-06-03
14:37:51
```

Listing 4

```
[oracle@ora11r264 bin]$ crsctl add resource dbcontrol -type local_resource
-attr "ACTION_SCRIPT=$GRID_HOME/crs/public/dbcontrol.sh, <weitere
Attribute>"
```

Listing 5

```
[oracle@ol16r01 ~]$ lsnrctl status
...
Service "pdb1" has 1 instance(s).
Instance "ora12r1", status READY, has 1 handler(s) for this service...
```

Listing 6

```
[oracle@ol16r01 ~]$ srvctl add service -h
Adds a service configuration to be managed by Oracle Restart.
Usage: srvctl add service -db <db_unique_name> -service <service_name>
[-pdb <pluggable_database>]
```

Listing 7

- Alle Tools („srvctl“, „crsctl“, „dbca“ etc.) unterstützen Restart auch mit der in 12c veränderten Syntax.
- Meldungen, Ausschriften etc. sind an 12c angepasst.
- Oracle Restart unterstützt Pluggable Database.

Pluggable Database und Oracle Restart

Die Pluggable Database läuft intern als Service (siehe Listing 6). Die Grid Infrastructure kennt diesen Service nicht, man kann jedoch einen Service anlegen, der auf die PDB zeigt und sich mit dieser verbindet (siehe Listing 7). Die überraschende Konsequenz ist: Wenn für die PDB ein Service konfiguriert und gestartet ist, dann wird die PDB beim Start der Container-Datenbank nicht nur gemountet, sondern es wird automatisch „READ/WRITE“ geöffnet. Das ist eine wesentlich elegantere Mög-

lichkeit für ein automatisches Startup der PDB als beispielsweise ein Startup Trigger. Auf diese Weise trägt ein eigentlich abgekündigtes Feature zur besseren Verwaltung neuer 12c-Funktionalität bei.

Fazit

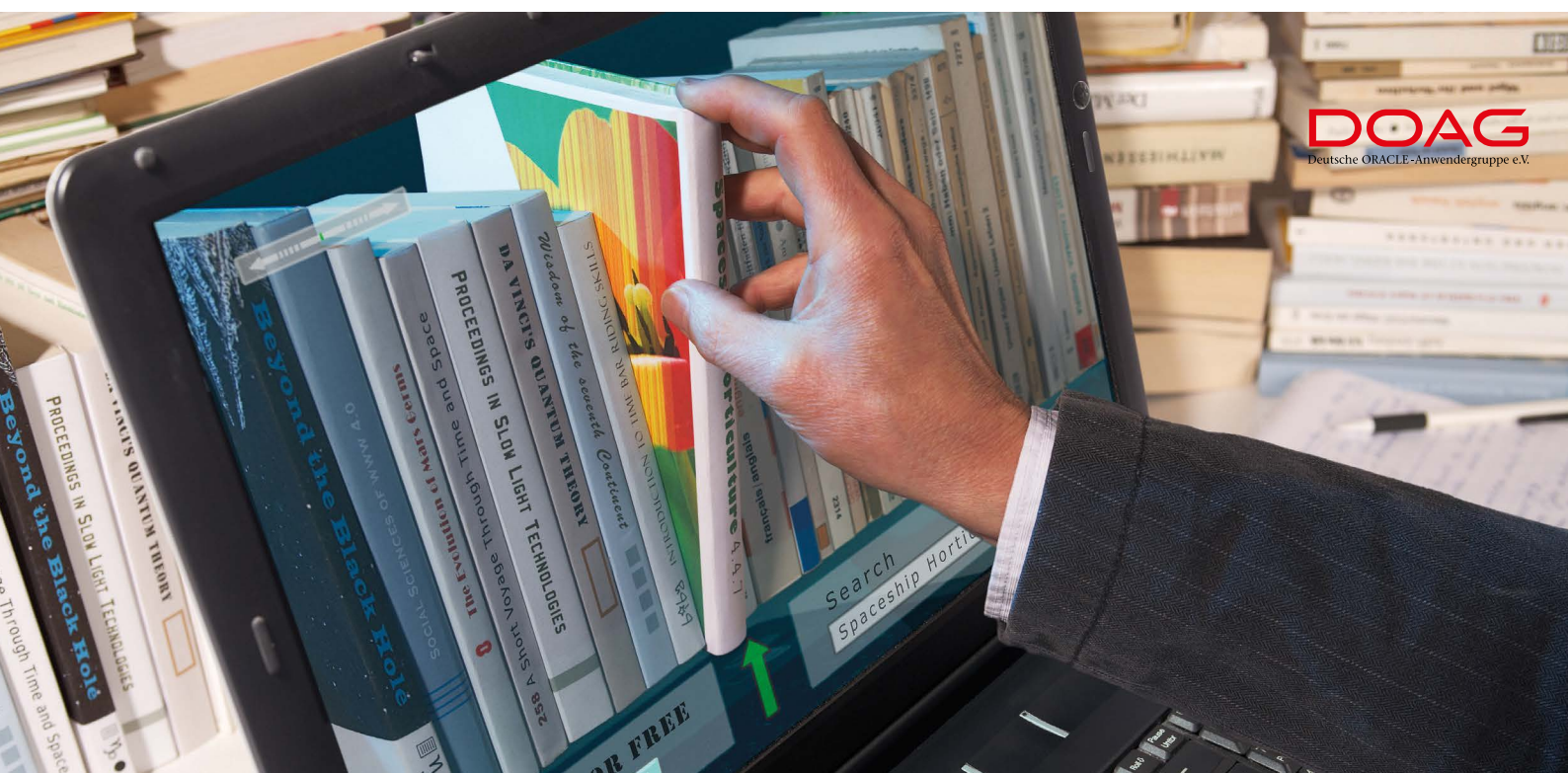
Es besteht kein Anlass, Oracle Restart in 12c nicht mehr zu verwenden. Oracle hätte sich vielleicht die Anmerkung in der Dokumentation verkneifen sollen, um Verwirrung zu vermeiden, zumal nicht erkennbar ist, wohin die Reise gehen soll.

Natürlich schafft Oracle Restart keine wirkliche Hochverfügbarkeit. Wie wir gesehen haben, hat aber der Einsatz der Grid Infrastructure für Single-Instance-Datenbanken durchaus Vorteile, nicht nur, weil damit der Einsatz von ASM möglich wird. Es sind vor allem die erleichterte Verwaltung, der automatische Neustart der Ressourcen, die Data-Guard-Integrati-

on und die übergreifenden Eigenschaften von FAN, die die Implementierung überlegenswert und sinnvoll machen.



Frank Haney
info@haney.it



DOAG
Deutsche ORACLE-Anwendergruppe e.V.

DOAG 2015
Datenbank

