

# Was leistet Oracle Restart?

**Dr. Frank Haney**  
**Consultant**  
**Jena**

## Schlüsselworte:

Oracle Database, Grid Infrastructure, Hochverfügbarkeit

## Zur Einordnung

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 wirtschaftlicher Aspekte. Eine besondere Vielfalt der Gestaltung von Hochverfügbarkeitslösungen bietet die *Oracle Grid Infrastructure*, wie seit der Version 11g Release 2 die Clusterware heißt. Einer der Hauptunterschiede zu vorherigen Versionen ist, daß ASM jetzt gebündelt mit der Clusterware- und nicht mehr mit der Datenbank-Software installiert wird, was die Namensgebung rechtfertigt. Folgende Varianten 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 RAC 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. RAC One Node kann ohne großen Aufwand zum vollwertigen RAC ausgebaut werden.
- **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** können mit der Grid Infrastructure hochverfügbar gemacht werden, ohne daß im Cluster eine Datenbank betrieben wird. Zu denken wäre hier auch an Application Server oder Enterprise Manager.
- **Oracle Restart:** Die Grid Infrastructure wird auf einem Standalone Server installiert. Es besteht nicht die Absicht, diesen zu einem Cluster zu erweitern.

Welche Vorteile die letzte Möglichkeit bietet, soll im Folgenden erläutert werden.

## Oracle Restart im Überblick

Das Feature ist neu in Oracle 11g Release 2, aber 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 bzw. 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 OHAS (Oracle High Availability Service). Dieser bezieht die Information für die Starteigenschaften der jeweiligen Ressource aus der OLR (Oracle Local Registry).

Oracle Restart hat eine Anbindung an Oracle Data Guard, was bewirkt, daß eine Standby-Datenbank immer in der richtigen Rolle startet.

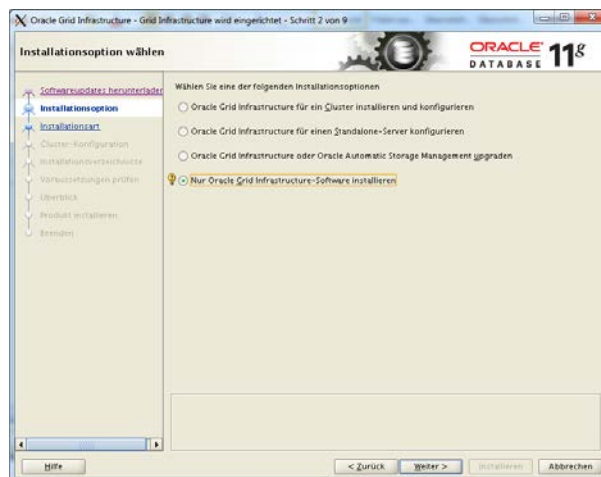
Die Einbindung von Services in die Restart-Architektur unter Berücksichtigung der Datenbankrolle ist möglich. Ein Client verbindet sich dann in einer Data Guard-Umgebung immer mit dem Server, der die primäre Rolle hat.

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 kann ihr Start automatisiert werden. Notwendige Abhängigkeiten werden berücksichtigt und FAN-Callouts auch für diese Anwendungen möglich. Besonders sinnvoll ist das z.B. für den Grid bzw. Cloud Control Agent oder auch Database Control.

## Besonderheiten der Installation

Die Implementierung beginnt mit der Installation der Grid Infrastructure für einen Standalone Server. Aber Achtung, diese Option erwartet die Verwendung von ASM. Das ist aber nicht zwingend, weil ein Standalone Server kein Shared Storage erfordert. Wenn man kein ASM verwenden will, dann muß wie in der folgenden Abbildung die Option „Software Only“ ausgewählt werden:



Die Installation läuft dann durch. Eventuelle Meldungen bei der Überprüfung der Voraussetzungen sollten nach Möglichkeit nicht ignoriert werden. Zum Schluß muß wie bei den meisten Installationen das Skript root.sh ausgeführt werden. Dieses fordert zu Ausführung eines Perl-Skripts auf (Ein analoges Skript gibt es für die Clusterinstallation):

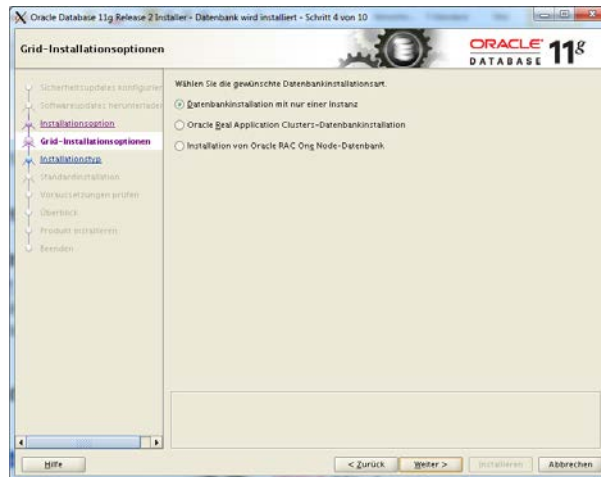
**To configure Grid Infrastructure for a Stand-Alone Server run the following command as the root user:**

```
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
```

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. (Hier sieht man schon, daß es syntaktische Abweichungen im Vergleich zur Clusteradministration gibt.)

Anschließend kann man mit dem Net Configuration Assistant (netca) einen Listener konfigurieren und starten. Dieser Listener verwendet das Grid Infrastructure Home.

Dann wird die Datenbanksoftware in ein eigenes Home installiert und eine Datenbank angelegt. Bei den Grid-Installationsoptionen 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, wie im folgenden Bild zu sehen:



Wenn man das Resultat kontrolliert, erscheint folgende Ausgabe:

```
$GRID_HOME/bin/crsctl stat res -t
```

```
-----  
NAME                TARGET  STATE          SERVER          STATE_DETAILS  
-----  
Local Resources  
-----  
ora.LISTENER.lsnr  
                ONLINE  ONLINE        ora11r264  
ora.ons  
                OFFLINE OFFLINE       ora11r264  
-----  
Cluster Resources  
-----  
ora.cssd  
    1            OFFLINE OFFLINE  
ora.diskmon  
    1            OFFLINE OFFLINE  
ora.evmd  
    1            ONLINE  ONLINE        ora11r264  
ora.ora11r2.db  
    1            ONLINE  ONLINE        ora11r264      Open
```

Die OFFLINE-Ressourcen sind optional und werden wie z.B. der `ocssd` nur bei der Verwendung von Oracle ASM benötigt.

## 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 nun hier nicht angezeigt, hier die komplette Syntax zu beschreiben. Das kann nachgelesen werden. Es sollen aber ein paar Besonderheiten erläutert werden. So kennt z.B. Oracle Restart bei `crsctl` die Argumente `cluster` und `crs` gar nicht. Stattdessen wird das Argument `has` verwendet. Es heißt stattdessen:

```
crsctl check|start|stop has
```

Dieses Kommando überprüft den Status des High Availability Service und startet bzw. stoppt ihn. Starten und Stoppen startet bzw. beendet auch alle in der OLR registrierten Applikationen unter Berücksichtigung der dort hinterlegten Abhängigkeiten. Wenn man alle Applikationen starten bzw. stoppen möchte, die aus einem Oracle Home laufen, kann man summarisch

```
srvctl start|stop home -o $ORACLE_HOME -s <state file>
```

benutzen. Im `state file` werden die zu stoppenden bzw. startenden Ressourcen benannt, z.B. Datenbank und Services im Falle des DB-Home oder Listener und ONS beim Grid Home. Auch einzelne Ressourcen aus dem jeweiligen Home werden mit `srvctl` verwaltet und konfiguriert. Die Restart-Konfiguration läßt sich wie folgt anzeigen und entsprechend auch modifizieren:

```
[oracle@ora11r264 bin]$ srvctl config database -d ora11r2 -a
Database unique name: ora11r2
Database name: ora11r2
Oracle home: /opt/oracle/oradb/11.2.0/db_1
Oracle user: oracle
Spfile:
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: AUTOMATIC
Database instance: ora11r2
Disk Groups:
Services:
Database is enabled
```

Was nun, wenn Datenbank und Listener schon vor der Restart-Implementierung vorhanden sind? Dann kann man Sie mit `srvctl add database` bzw. `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. Dazu kommt, daß bei einem Rollenwechsel alle Services gestartet bzw. 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:

Service anlegen

```
srvctl add service -d ora11r2sb -s servsb -l primary
```

Service starten (muß nach dem Anlegen einmal manuell gestartet werden)

```
srvctl start service -d ora11r2sb -s servsb
```

Dabei gibt es aber 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 aber 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.
  - Abwarten, ob der entsprechende Enhancement Request (Bug 9375220) Erfolg haben wird.
  - Implementierung eines FAN Callout, der die aktuelle Rolle der Datenbank ausliest und automatisch den Open Modus ändert. (Bevorzugte Lösung)
2. Ein Service für die Rolle Physical\_Standby läßt sich nicht starten, wenn die Standby-DB nur gemounted ist. Das ist ein erwartetes Verhalten. Es gibt zwei Lösungen:
  - Verwendung von Active Data Guard (kostenpflichtige Option)
  - Services nur für die Rolle PRIMARY verwenden

## Integration einer „Fremd“-Applikation am Beispiel von Database Control

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

- Es muß ein Action Sript erstellt werden. Dieses muß die Funktionen `start`, `stop`, `check` und `clean` enthalten.
- Das Skript wird dann getestet und nach `$GRID_HOME/crs/public` kopiert.
- Dann wird die Ressource in der OLR registriert:  

```
[oracle@ora11r264 bin]$ crsctl add resource dbcontrol -type local_resource -attr "ACTION_SCRIPT=$GRID_HOME/crs/public/dbcontrol.sh, DESCRIPTION=Local Resource for Database Control, DEGREE=1, ENABLED=1, AUTO_START=restore, START_TIMEOUT=120, UPTIME_THRESHOLD=1h, CHECK_INTERVAL=60, STOP_TIMEOUT=120, SCRIPT_TIMEOUT=120, RESTART_ATTEMPTS=3, OFFLINE_CHECK_INTERVAL=60, START_DEPENDENCIES=, STOP_DEPENDENCIES="
```
- Die Ressource muß dann gestartet werden:  

```
[oracle@ora11r264 bin]$ crsctl start resource dbcontrol
```
- Man kann dann die Funktionalität testen, indem man den Prozeß killt und nachsieht, ob Oracle Restart ihn wieder startet.
- Schließlich sollte man sich die Konfiguration anschauen und eventuell anpassen:  

```
[oracle@ora11r264 public]$ crsctl status resource dbcontrol -p NAME=dbcontrol TYPE=local_resource ACL=owner:oracle:rwx,pgrp:oinstall:rwx,other::r-- ACTION_SCRIPT=/opt/oracle/oragrid/11.2.0/grid/crs/public/dbcontrol.sh AGENT_FILENAME=%CRS_HOME%/bin/scriptagent AUTO_START=restore CHECK_INTERVAL=60 DESCRIPTION=Local DB Console Resource ENABLED=1 OFFLINE_CHECK_INTERVAL=60 RESTART_ATTEMPTS=3 SCRIPT_TIMEOUT=120 START_DEPENDENCIES= START_TIMEOUT=120 STOP_DEPENDENCIES= STOP_TIMEOUT=120
```

## Fazit

Oracle Restart ist sicher keine wirkliche Hochverfügbarkeitslösung. Es hilft aber, die Verfügbarkeit von Standalone Servern zu verbessern, von denen wesentliche Funktionalität der IT-Landschaft abhängt, ohne daß sie selber produktiv im engeren Sinne sind. Damit ist z.B. der Datenbankserver mit dem Recovery Catalog oder dem Grid (Cloud) Control Repository gemeint. Es ist auch sinnvoll, um den OEM Agent besser verfügbar zu machen. Speziell in Data Guard-Umgebungen macht Oracle Restart Sinn, weil es möglich ist, mit Services und Callout-Skripten die verschiedenen Szenarien zu automatisieren.

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

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