

# Einfach Einfach: 11gR2 Real Application Cluster

Sebastian Solbach / Ralf Durben  
ORACLE Deutschland B.V. & Co. KG

## Schlüsselwörter:

11g Release 2, 11gR2, Real Application Cluster, Clusterware, Grid Infrastruktur, Software, Installation

## Einleitung

Wer glaubt Real Application Cluster sei kompliziert, kennt 11gR2 RAC nicht. Durch Grid Plug and Play (GPnP) ist eine 11gR2 RAC Installation super einfach und schnell zu bewerkstelligen. Voraussetzung sind eigentlich nur 2 Rechner mit Oracle Enterprise Linux, ein Shared Storage und schon kann es losgehen.

## Was ist RAC – Einfach erklärt

Oracle Real Application Cluster ist ein Datenbanksystem bei dem mehrere Rechner (Instanzen) auf eine gemeinsame Datenbank zugreifen. Diese Datenbank liegt dabei in einem gemeinsamen Speicher (Shared Disk Prinzip). Die mehreren Datenbank Instanzen teilen sich einerseits die Arbeit, dienen gleichzeitig aber auch als gegenseitige Absicherung. Der Name Real Application Cluster soll verdeutlichen, dass jede Applikation, die auf einem SMP System skaliert, auch auf einem RAC Cluster läuft und mit der Anzahl der Knoten linear skaliert.

Eine wichtige Bedeutung nimmt hierbei die Netzwerkverbindung zwischen den Knoten ein, der sogenannte Interconnect. Über diesen läuft das von Oracle patentierte Cache Fusion Protokoll, mit dem die Datenbank Instanzen untereinander kommunizieren und über welches die gemeinsam benutzten Datenbank Blöcke ausgetauscht werden.

Letztendlich bietet diese geclusterte Oracle Datenbank für jede Applikation ein sehr hohes Maß an Hochverfügbarkeit und falls nötig auch entsprechende Skalierbarkeit, sollten die einzelnen Ressourcen nicht mehr ausreichen. Eigentlich eine Funktionalität die man jeder Datenbank bieten möchte, vorausgesetzt der Aufwand und die Kosten halten sich im Rahmen.

## Die Grid Infrastruktur

Oracle geht deswegen mit der 11gR2 Grid Infrastruktur einen Schritt weiter, als es noch die vorhergehenden Versionen der Clusterware taten. So will die Grid Infrastruktur nicht nur die grundlegende Basis für Insellösungen, d.h. wenige einzelne RAC Datenbanken sein, sondern legt auch den Grundstein für einen großen gemeinsam verwendeten Cluster. In diesem Cluster teilen sich mehrere Applikationen oder Datenbanken die gemeinsam benutzten Ressourcen auf, um deren Auslastung zu optimieren. Darum hat Oracle im neuesten Release großen Wert auf die vereinfachte Installation und Bedienung einer solchen Cluster Umgebung gelegt.

## Die letzte Herausforderung: Hardware Voraussetzungen

Für die Installation eines Real Application Cluster müssen bestimmte Hardware Voraussetzungen erfüllt sein: Minimum sind 2 Server mit jeweils 2 Netzwerkkarten und ein gemeinsam verwendetes Storage System. Jeder Server sollte über mindestens 1536 MB RAM und 2 CPUs verfügen (mehr ist natürlich immer empfohlen) und den Zugriff auf das gemeinsame Storage gewährleisten.

Die Bereitstellung und Konfiguration des gemeinsamen Storage ist bei der ganzen Cluster Installation auch der herausforderndste Teil. Denn hier kann leider auch die beste Installationsroutine nicht alle Möglichkeiten der Anbindung kennen, geschweige denn vorbereiten. Ist diese Hürde gemeistert und beide Systeme haben Zugriff auf dieselben Platten, LUNs steht der Installation nichts mehr im Wege.

Aber auch diese Hürde ist im Gegensatz zu den vorhergehenden Releases viel einfacher zu nehmen: Vorbei sind die Zeiten von unterschiedlichen Rechten und unterschiedlichen Größen der Platte für Clusterware Konfigurationsdateien und ASM. Nach unserer Erfahrung benötigen Sie einfach nur mindestens 8 gleich große LUNs/Disks:

- 4 für die Datenbankfiles (Daten Diskgruppe)
- 4 für die Flashback Recovery Area (FRA Diskgruppe).

Diese Devices müssen dem Grid Installation User und der Betriebssystemgruppe asmadmin gehören. Ob dies über ASMLib bewerkstelligt wird, oder über eigene UDEV Regeln, ist dabei gleich. ASMLib spielt seine Stärken erst bei sehr vielen LUNs und vielen Datenbanken im RAC aus. UDEV dagegen ist wahrscheinlich leichter und schneller konfiguriert:

Beispiel einer UDEV Regel, die die notwendigen /dev/sd\* Devices dem User Grid und der asmadmin Gruppe zuweist:

```
KERNEL=="sd[c-z]*", OWNER="grid", GROUP="asmadmin", MODE="0660"
```

Wir empfehlen auf jedem Device eine Partition anzulegen, auch wenn dieses nicht zwingend notwendig ist.

## Einfach zu erledigen: Software Voraussetzungen

Die Software Voraussetzungen für eine 11gR2 Grid Installation auf Oracle Enterprise Linux 5 lassen sich ganz einfach erfüllen. Oracle bietet schon seit geraumer Zeit, das sogenannte Oracle Validated Paket an, welches alle notwendigen Änderungen am System vornimmt. So werden nicht nur alle für die Oracle Datenbank notwendigen Softwarepakete (RPMs) installiert, sondern auch die Limits und fast alle Kernel Parameter gesetzt.

Sollten Sie nicht über einen Unbreakable Linux Support verfügen und damit keinen direkten Zugriff auf das oracle-validate und die aktuellen Software Pakete haben, können Sie das Paket auch direkt bei Oracle herunterladen (<http://oss.oracle.com/e15/oracle-validated/>):

```
# wget http://oss.oracle.com/e15/oracle-validated/oracle-validated-1.0.0-22.e15.i386.rpm
```

Die Abhängigkeiten der Pakete werden am leichtesten über den öffentlichen public-yum (<http://public-yum.oracle.com>) Server von Oracle aufgelöst:

```
# cd /etc/yum.repos.d
# wget http://public-yum.oracle.com/public-yum-el5.repo
```

Ist die eingesetzte Linux Version in der public-yum-el5.repo aktiviert, wird oracle-validated einfach mit yum installiert:

```
# yum install oracle-validated-1.0.0-22.el5.i386.rpm
...
Dependencies Resolved
```

```
=====
Package           Arch Version           Repository           Size
=====
Installing:
oracle-validated  i386 1.0.0-22.el5 /oracle-validated-1.0.0-22.el5.i386 43k
...
=====
```

Aber selbst ohne Oracle Validated braucht man auf die Konfiguration nicht zu achten. In den 11gR2 Installer wurde das alte Cluster Verification Utility (CLUVFY) integriert, welches die Systemvoraussetzungen prüft und Kernelparameter über ein sogenanntes FixUp Script einfach selber setzen kann. Nur fehlende Pakete müssten dann per Hand nachinstalliert werden.

Eines der wenigen Dinge, die noch nicht automatisch geprüft werden, ist eine sauber gepflegte Datei /etc/hosts und das Ausschalten der Firewall, bzw. von SELinux.

### **Ganz Einfach: Die Installation**

Während der Installation werden Sie sehr gut vom neuen Oracle Universal Installer unterstützt. Folgt man den Schritten im Installer, so findet man zu jeder Zeit nützliche Hilfe-Informationen und das Ganze geht recht einfach von der Hand. Hierbei kommt man schon bei einer Typical Installation schnell ans Ziel, wer etwas mehr Kontrolle haben möchte, wählt den Advanced Modus.

In der Typical Variante trägt man zuerst seine Knoten und deren Virtuelle IP Adressen (VIP) ein. Zusätzlich legt man den sogenannten Single Client Access Namen (SCAN) fest (dieser ist auch gleichzeitig der Clusternamen). Der SCAN ist ein virtueller Hostname der vom DNS Server auf 3 IP Adressen aufgelöst wird. Hat man im Moment keinen Netzwerkadministrator zur Hand, der dies für einen erledigt, genügt bei der Installation auch ein einfacher Eintrag in die Datei /etc/hosts als provisorische Übergangslösung.

Anhand der Hostnamen versucht der Installer automatisch die Netzwerke für den Interconnect und das Public Netzwerk zu identifizieren. Im Normalfall liegt er hiermit richtig (man kann dies aber selbstverständlich über „Identify Network Interfaces“ nachprüfen und ändern).

Während des Installationsprozesses werden Dateien auch auf die Remote Knoten transferiert. Dies geschieht, wie früher auch über secure shell (SSH). Allerdings kann nun der Installer das ganze Setup selbst übernehmen, was die Installation doch stark vereinfacht. „SSH Connectivity“ auswählen – Password angeben – Setup klicken – Fertig!

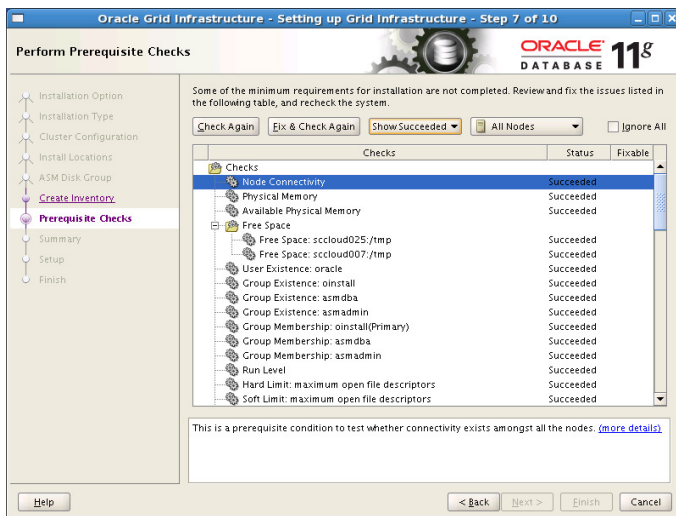


Abb. 1: CLUVFY Test im Installer

Nun noch die Installationsverzeichnisse angeben (die übrigens auch auf Vorhandensein und richtige Rechte überprüft werden), ASM für das Cluster auswählen, Diskgruppe angeben und man ist eigentlich fertig.

Dann prüft der Installer, wie oben schon erwähnt, mit Hilfe von CLUVFY, ob alle Voraussetzungen erfüllt sind. Wenn nicht, kann er bestimmte Einstellungsfehler sogar direkt über die FixUp Skripte beheben. Sind alle Voraussetzungen erfüllt, beginnt die Installation.

## Der „(un)kritische“ Punkt: Das root.sh Skript

Früher galt das root.sh Skript der Cluster-Installation als der Punkt, wo man die „Daumen gedrückt hielt“, da ja nichts schief laufen durfte. Denn sonst bedeutete dies, dass man mit der Installation komplett von vorne beginnen musste – inklusive manueller Arbeiten die Knoten zurückzusetzen.

Dieses gilt jetzt nicht mehr: Nicht nur, das viel weniger schief geht – es wird ja besser geprüft – sondern es gibt auch eine Möglichkeit nur das root.sh Skript neu aufzurufen.

Das root.sh Skript besteht aus 2 Bestandteilen: Einem herkömmlichen Teil, der die Verzeichnis-Berechtigungen setzt und dem Aufruf von rootcrs.pl welches die Clusterkonfiguration vornimmt.

Während der erste Teil eigentlich nie neu gestartet werden muss, kann das rootcrs.pl nun manuell aufgerufen werden. Sowohl zur Dekonfiguration als auch zur Konfiguration des Clusters. Hierzu führt man bei einem fehlgeschlagenen root.sh Skript einfach folgende Befehle aus:

```
$ORACLE_HOME/crs/install/rootcrs.pl -deconfig -force
```

Behebt die Ursache für den vorhergegangenen Abbruch und startet dann die Konfiguration neu mit:

```
$ORACLE_HOME/crs/install/rootcrs.pl
```

Hierbei enthält die Datei crsconfig\_params alle notwendigen Parameter für die Installation.

Auch für größere Cluster gibt es nun eine Erleichterung: Kann doch das root.sh Skript, nachdem es auf dem Installationsknoten erfolgreich durchgelaufen ist, nun parallel auf allen restlichen Knoten ausgeführt werden.

Die restlichen Konfigurationsassistenten sind danach eigentlich nur noch Formsache.

## **Einfache Client Konfiguration: Der SCAN (Single Client Access Name)**

Hinter der Einführung von SCAN steckt die simple Idee: Egal wie mein Cluster aussieht der Client verbindet sich immer gleich. Auch wenn der komplette Cluster sich verändert, die Verbindungsinformationen der Datenbank Clients ändern sich nicht. Sie bestehen immer aus dem sogenannten SCAN Namen und dem Service der Datenbank.

Alles Andere übernimmt der Server bzw. wird auf dem Server oder in der Datenbank konfiguriert:

- Loadbalancing
- Transparent Application Failover (TAF)
- Fast Connect Failover (FCF)

Durch diese vereinfachte Möglichkeit braucht ein 11gR2 Client auch den komplizierten Eintrag in der TNSNAMES.ora bzw. die komplizierte TNSNAMES.ora Syntax im Connect String nicht mehr, sondern kann sich ganz einfach mit Easy Connect bzw. der JDBC Thin Syntax mit dem Server verbinden:

```
sqlplus <username>/<password>@<scanname>:<port>/<service>
```

z.B.

```
sqlplus scott/tiger@scanname:1521/oltp
```

```
jdbc:oracle:thin:@scanname:1521/oltp
```

Vereinfacht beschrieben sieht der Verbindungsaufbau wie folgt aus:

- Client versucht den SCAN Namen aufzulösen und bekommt vom DNS Server eine von den 3 möglichen IP-Adressen zurück
- Client kontaktiert die SCAN VIP und verbindet sich mit dem SCAN Listener
- SCAN Listener informiert den Client, auf welchem Knoten der gewünschte Service läuft. Läuft der Service auf mehreren Knoten, wird der Knoten zurückgemeldet, der im Moment am wenigsten ausgelastet ist.
- Client verbindet sich mit dem lokalen Listener auf dem gemeldeten Knoten und erhält über diesen eine Verbindung zur Datenbank

Sollte innerhalb dieses Prozesses ein Fehler auftreten, versucht es der Client einfach erneut, mit einer anderen der übrigen 2 SCAN VIPs.

## Die einfachen Storage Tools: ASMCA

Nach der Installation gibt es zwar schon eine Daten Diskgruppe, die bis jetzt die Clusterware Konfigurationsfiles aufgenommen hat, aber es die Diskgruppe für die Flashback Recovery Area (FRA) muss noch erstellt werden. Diese kann der Storage Administrator einfach über den ASM Configuration Assistant (ASMCA) anlegen.

Falls auf der Grid Infrastruktur nur 11.2 Datenbanken laufen sollen, empfiehlt es sich die Kompatibilitätseinstellungen der Diskgruppen auf 11.2.0.0 anzupassen, damit die Datenbanken auch von den neuen 11.2 ASM Funktionalitäten profitieren können:

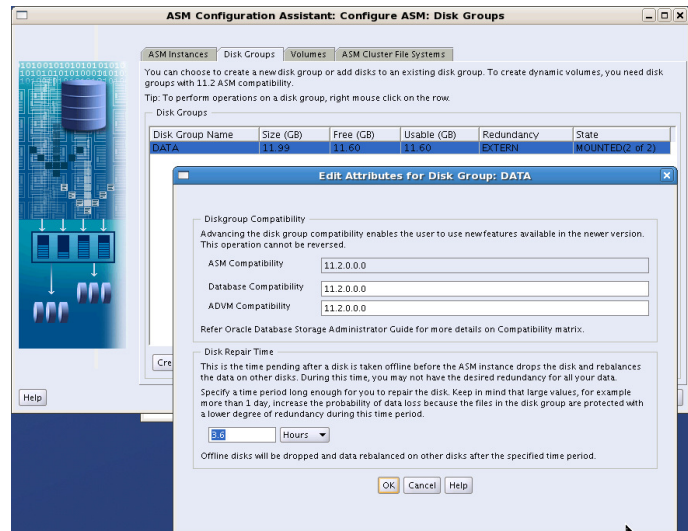


Abb. 2: ASMCA Compatibility Einstellungen

Daneben bietet der ASMCA (fast) alle Funktionalitäten, die man zur Verwaltung und Administration von ASM benötigt. Sei es das Anlegen der neuen dynamischen Volumes oder die Erstellung des ASM Cluster Filesystems (ACFS). Dies kann z.B. dazu verwendet werden um ein gemeinsam genutztes Oracle Home zu installieren.

Hat man ASM konfiguriert und sich für eine Speicherform für das Oracle Home entschieden (geshared oder lokal) kann man mit der Datenbank Software Installation weitermachen.

### Noch einfacher: Die Datenbank Software Installieren

Dieser Punkt war auch in der Vergangenheit immer der leichteste. Eine Software Only Installation ohne Datenbank durchzuführen, war nach einer erfolgreichen Clusterware Installation nie problematisch. Noch weniger mit 11gR2, denn auch hier werden alle notwendigen Voraussetzungen für eine Datenbank Installation geprüft. Falsch eingestellte Parameter können über die FixUp Skripte korrigiert werden.

Persönlich empfehle wir immer eine Software Only Installation mit anschließend separater Erstellung der Datenbank. Hauptgrund ist die empfohlene Installation von Patch Set Updates (PSUs). Wird die Datenbank bei der Software Installation gleich mit erzeugt, so erfordern diese Patches eine Inhaltliche Änderung des Data Dictionaries. Dieses erspart man sich, wenn die Datenbank erst nach dem Patchen erzeugt wird.

## Einfach Toll: Server Pools

Das Anlegen der Datenbank geschieht am leichtesten über den Datenbank Configuration Assistant kurz DBCA. Dieser erstellt auch alle notwendigen Skripte, falls man die zukünftigen Datenbanken lieber manuell anlegen möchte. Im Gegensatz zu früher, sind aber diesmal auch die Registrierung beim Cluster und die Konfiguration der Datenbank Console in den Skripten enthalten.

Ziemlich früh wird man beim Ablauf des Assistenten gefragt, ob man eine Administrator Managed oder Policy Managed Datenbank anlegen möchte.

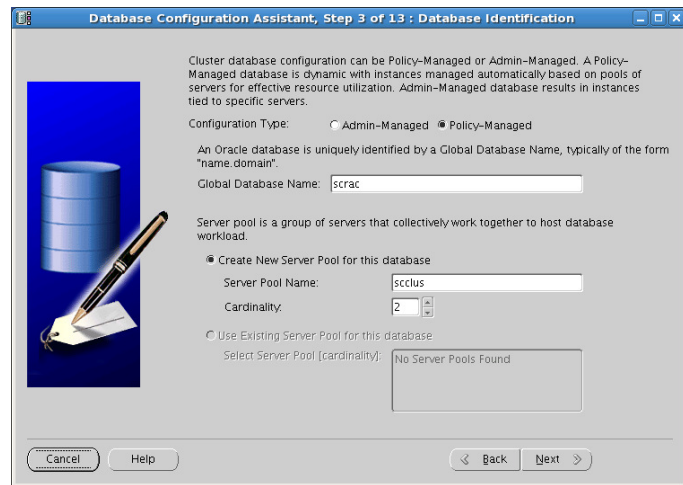


Abb. 3: Policy Managed vs. Administrator Managed

Während die bisherige Verwaltung von Datenbanken, nun Administrator Managed bezeichnet, erforderte, dass jede Datenbank und jeder Service explizit einem Knoten zugeteilt wurde, bietet die neue regelbasierende Administrationsform sehr viel Flexibilität und Vereinfachung.

Benutzt man die regelbasierte Administration ordnet man eine Datenbank einfach einem sogenannten Server Pool zu. Ein Server Pool ist eine logische Aufteilung des Clusters in einzelne kleinere Einheiten. Ein Server Pool wird durch ein Minimum und Maximum definiert und besteht aus dynamisch zugeordneten Knoten. Erweitert man den Cluster und wird dadurch ein Server Pool größer, so wachsen automatisch die RAC Datenbanken mit, ohne Eingriff des Administrators. Umgekehrt genauso: Wird ein Server Pool kleiner, weil ein Server ausfällt, werden auch automatisch weniger Datenbank Instanzen gestartet, ebenfalls ohne Zutun des Administrators. Des Weiteren legt man über die Gewichtung der Server Pools fest, welcher Server Pool höhere Priorität besitzt und eben beim Ausfall eines Knotens nicht „geschrumpft“ wird bzw. als erster neue Ressourcen erhält, wenn der Cluster erweitert wird.

Diese Art der Verwaltung steht für Oracle Datenbanken ab Release 11.2 zur Verfügung.

## Graphische Verwaltung der Grid Infrastruktur

Die Grid Infrastruktur läuft und RAC Datenbanken wurden auf dem Cluster angelegt. Die Verwaltung des Clusters wird über die graphische Benutzeroberfläche des Database Controls 11gR2 bzw. Grid Controls 11gR1 durchgeführt:

Hiermit können übrigens auch weitere Applikationen unter die Überwachung der Clusterware gestellt werden. Damit und mit der Verfügbarkeit des ASM Cluster Filesystems, bietet die Oracle Grid Infrastruktur alles, was für eine Hochverfügbarkeitsumgebung notwendig ist; ob mit Oracle RAC Datenbank oder ohne.

Select All	Name	Cardinality	Current State	Target State	Running Hosts	Resource Type	Owner
<input type="checkbox"/>	Show ora.gsd	Runs on all servers	⚠	⚠	n/a	ora.gsd type	oracle
<input type="checkbox"/>	Show ora.sgid	1	⚠	⚠	n/a	ora.sgid type	oracle
<input type="checkbox"/>	Show ora.sctsd	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd1	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd2	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd3	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd4	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd5	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd6	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd7	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd8	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd9	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd10	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd11	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd12	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd13	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd14	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd15	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd16	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd17	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd18	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd19	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd20	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd21	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd22	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd23	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd24	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd25	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd26	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd27	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd28	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd29	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd30	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd31	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd32	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd33	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd34	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd35	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd36	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd37	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd38	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd39	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd40	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd41	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd42	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd43	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd44	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd45	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd46	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd47	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd48	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd49	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd50	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd51	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd52	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd53	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd54	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd55	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd56	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd57	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd58	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd59	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd60	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd61	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd62	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd63	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd64	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd65	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd66	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd67	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd68	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd69	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd70	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd71	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd72	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd73	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd74	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd75	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd76	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd77	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd78	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd79	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd80	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd81	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd82	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd83	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd84	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd85	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd86	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd87	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd88	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd89	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd90	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd91	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd92	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd93	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd94	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd95	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd96	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd97	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd98	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd99	1	⚠	⚠	n/a	ora.sctsd type	oracle
<input type="checkbox"/>	Show ora.sctsd100	1	⚠	⚠	n/a	ora.sctsd type	oracle

Abb. 4: Cluster Administration im Database Control

## Nützliche Command Line Utilities – Clusterweite Kommandos

Auch wenn Database Control die komplette Administration der Clusterware bietet, gibt es einige nützliche Tools zur Administration der Grid Infrastruktur bzw. der RAC Datenbanken.

Interessant an 11gR2 ist nun, dass fast alle Kommandos nun clusterweit funktionieren. So lässt sich auch der komplette Cluster mit nur einem Befehl herunterfahren.

Zu den wichtigsten Befehlen zählt das `srvctl`. Das Server Control wird verwendet um alle `ora.*` Ressourcen die es im Cluster gibt zu verwalten. Hiermit werden Datenbanken gestoppt, Server Pools geändert, die Listener angepasst und Services angelegt. Es hat eine sehr ausführliche Hilfe, die man sich mit `-h` zu jedem Befehl anzeigen lassen kann.

Das Cluster Control (`crsctl`) dagegen ist für die darunterliegende Clusterware zuständig. Der wichtigste Befehl hier ist definitiv die Statusanzeige in tabellarischer Form:

```
# crsctl stat res -t
```

Über `crsctl` werden auch clusterspezifische Parameter abgefragt und der Cluster gestartet oder gestoppt. Hiermit kann auch veranlasst werden, dass nach einem Neustart des Servers die Clusterware nicht automatisch startet. Dies geschieht mit

```
# crsctl disable crs
# crsctl enable crs
```

Ein Hinweis zum Backup und Restore der Clusterware: Während RAC Datenbanken wie Single Instanz Datenbanken über RMAN gesichert werden, stellt sich die Frage wie mache ich ein Backup von den Clusterware spezifischen Informationen. Dies ist insbesondere die Oracle Cluster Registry (OCR), das sogenannte Herz des Clusters, in dem alle Informationen zum Cluster gespeichert sind.



Die OCR wird von Oracle alle 4 Stunden automatisch gesichert, erfordert also nicht unbedingt einer weiteren Betrachtung. Allerdings kann man ein Backup auch manuell anstoßen, mit Hilfe von:

```
# ocrconfig -manualbackup
```

Genau wie das `srvctl` sind auch `ocrconfig` und `crsctl` mit einer umfangreichen Hilfefunktion ausgestattet, die über `-h` aufgerufen werden.

### **Schnell den Cluster erweitern**

Mit wenigen Befehlen wird der Cluster um einen zusätzlichen Knoten erweitert. Dies ist absolut kein Hexenwerk und geht super schnell. Voraussetzung ist, der funktionierende Zugriff auf den Shared Storage. Damit kann mit der Erweiterung gestartet werden:

Hierzu wird zuerst einmal SSH auf dem neuen Knoten eingerichtet:

```
$ $CRS_HOME/opt/grid/oui/bin/runSSHSetup.sh -user oracle -hosts 'sccloud007
sccloud025 sccloud016' -advanced -exverify
```

Dann wird sicherheitshalber mit `cluvfy` geprüft, ob der Knoten zum Hinzufügen bereit ist.

```
$ cluvfy stage -pre nodeadd -n sccloud016 -verbose
```

Nun kann die Grid Infrastruktur „erweitert“ werden:

```
$ cd $CRS_HOME/oui/bin
$ ./addNode.sh -silent "CLUSTER_NEW_NODES={sccloud016}"
"CLUSTER_NEW_VIRTUAL_HOSTNAMES={sccloud03vip}"
```

Zum Abschluss die geforderten Root Skripte auf dem neuen Knoten ausführen:

```
# /opt/oraInventory/orainstroot.sh
# $CRS_HOME/root.sh
```

Fertig: Der Cluster ist erweitert. Eigentlich ist der neue Clusterknoten nun einsatzbereit und wurde dem Free Pool zugeordnet. Damit er aber eine aktive Datenbank Instanz übernehmen kann, brauchen wir noch die Datenbank Software auf dem Rechner. Diese wird ebenfalls über ein `addNode.sh` einfach kopiert, damit ist gewährleistet, dass auch alle installierten Patches gleich mit auf dem neuen Knoten sind.

```
$ cd $ORACLE_HOME/oui/bin
$ ./addNode.sh -silent "CLUSTER_NEW_NODES={sccloud016}"
```

Nun kann der Knoten wirklich am Cluster teilnehmen. Die Datenbank erweitert man nun dank Server Pool ganz einfach: Einfach den Server Pool vergrößern, den Rest übernimmt Oracle:

```
$ srvctl modify srvpool -g scclus -u 3
```

## Fazit

Mit 11gR2 wurden Installation und Betrieb der Grid Infrastruktur und RAC Datenbanken stark vereinfacht. Viele Automatismen nehmen der Datenbankadministration viele Handgriffe ab. Die Dynamik eines Clusters wird durch das einfache Hinzufügen weiterer Knoten, sowie die Verwendung von Server Pools und SCAN optimal unterstützt.

Kontaktadresse:

Sebastian Solbach  
ORACLE Deutschland B.V. & Co. KG  
Riesstr. 25  
D-80992 München

Telefon: +49 (0) 711-72840 239  
E-Mail [sebastian.solbach@oracle.com](mailto:sebastian.solbach@oracle.com)

Ralf Durben  
ORACLE Deutschland B.V. & Co. KG  
Rudolfstr. 5  
D-59556 Lippstadt

Telefon: +49 (0) 211-74839 461  
E-Mail [ralf.durben@oracle.com](mailto:ralf.durben@oracle.com)

Internet: [www.oracle.de](http://www.oracle.de)