RAC Prozessarchitektur

Thorsten Grebe
twg-it
Berlin

Schlüsselworte: RAC Prozesse

Einleitung

Die Sicht auf die Betriebssystemprozesse eines Real Application Clusters in der Version 11gR2 kann schwindelig machen. Um 100 Prozesse werden auf jedem RAC Knoten für den Betrieb einer Oracle Clusterdatenbank gestartet. Wie hängen diese Prozesse miteinander zusammen? Welcher Prozess reagiert auf welches Adminwerkzeug? Ist der Überblick noch zu retten?

Die Perspektive

Als DBA kann man sich einem Real Application Cluster auf zweierlei Weise nähern. Die erste, zu empfehlende, erfolgt aus der Vogelperspektive. Man verwaltet RAC über den Enterprise Manager: Alles fügt sich zu einem Ganzen, scheint einheitlich und spielerisch verwaltbar. Der DBA erlebt die einfache Leichtigkeit das RAC-Daseins (Vortrag Nr. 15: „*Einfach Einfach: 11gR2 Real Application Cluster"* und Vortrag Nr. 361: *„Die Leichtigkeit des Seins von 11gR2 Grid*", Sebastian Solbach und Ralf Durben). Auf diese Ebene gehören auch die großen Begriffe wie Grid Infrastruktur, Clusterware und Ressource. Dank ihrer immanenten Unschärfe lassen sich diese Begriffe kaum falsch verwenden.

Es gibt aber auch eine alternative Perspektive, aus der man einen Real Application Cluster betrachten kann, nämlich vom Boden aus, auf der Prozessebene, bewaffnet mit Konsolenwerkzeugen. Hier eröffnet sich dem DBA eine ganz andere Erfahrung: Ein Schneesturm aus über 100 RAC-Prozessen verhindert den Durchblick, eine Auswahl von mehr als zwei Dutzend Administrationswerkzeugen verhagelt den Überblick, eine Streuung von Logs und Tracedateien in alle Himmelsrichtungen vernebelt den Einblick. Nicht die beschwingte Leichtigkeit des Seins blickt dem DBA hier ins Visier, sondern das zermürbende Elend der Komplexität.

In 11gR2 lässt sich ein ACFS-Verzeichnis spielend einfach über den ASM Konfigurations-Assistenten (asmca) anlegen, das schafft ein Grundschüler. Von oben betrachtet kann die einfache Leichtigkeit, mit der ein Clusterdateisystem aufgesetzt ist, begeistern. Auf Prozessebene werden hier jedoch dem bereits im Debug-Sumpf ertrinkenden DBA noch einmal 20 weitere Prozesse entgegengeschleudert, deren Wechselwirkung zunächst neue Rätsel aufgibt.

Die Nomenklatur im Cluster fördert zudem nicht immer die Orientierung: Die ähnlich lautenden *Oracle Cluster Synchronization Services Daemon* (OCSSD), *Cluster Synchronization Services Daemon* (CSSD) und *Cluster Synchronization Services* (CSS) oder die in Beziehung stehenden Begriffe Clusterware, CRS und CRSD laden zur willkürlichen Verwendung ein. Erschwerend kommt hinzu, dass sich die Definition einiger Begriffe über die Versionen 10.1 bis 11.2 gewandelt hat. CRS beispielsweise bezeichnet *die Cluster Ready Services*, eingeführt in 10.1, offiziell umbenannt in 10.2 zu Clusterware, in 11.2 jedoch nur noch ein kleiner, abhängiger Bestandteil derselben, auf Betriebssystemebene im Prozess crsd.bin materialisiert und in der Ressource ora.crsd virtualisiert – RAC-Semantik für den kleinen Philosophen im DBA.

Die Prozesse
Perfekt wird die Verwirrung auf der Prozessebene nicht allein durch die schiere Anzahl an Prozessen, sondern auch durch Umbenennungen, Zugänge, Abgänge und durch das Einführen von Begriffen, durch die neue Abstraktionsebenen geschaffen werden. Längst kann man sich nicht mehr darauf verlassen, dass der Klarname eines Prozesses zu seinem Kürzel passt. So stehen LMD und LMS offiziell für *G*lobal *E*nqueue *S*ervice *D*aemon und *G*lobal *C*ache *S*ervice *P*rocess:

```
SQL>  select name, description
        from V$BGPROCESS
    where name in ('LMS0','LMD0')

NAME  DESCRIPTION
----- ----------------------------------------
LMD0  global enqueue service daemon 0
LMS0  global cache service process 0
```

Die alten Lock-Prozesse (LMS, LMD, LCK, LMON) stammen aus der frühen RAC-Entwicklung, als noch in Prozessen gedacht wurde. Der Servicegedanke und überhaupt das Denken im Globalen kamen erst später auf.

Mit jeder neuen RAC-Version wurden bisher nicht nur neue Begriffe, sondern stets auch neue Prozesse eingeführt, in 10g z.B. die Prozesse des Eventmanagements (EVM), des Nachrichtendienstes (ONS) oder die ASM-Prozesse. Auch 11.1 sah neue Prozesse einziehen, wie den *Space Management Coordinator* (SMCO) und den *Virtual Keeper of Time* (VKTM). Zu den Neuen in 11.2 zählen SCAN, MDNS und GSN. Was nutzt die einfache Leichtigkeit des Klickens im EM, wenn es die Konzepte sind, die Verdauungsprobleme bereiten. Bei allen Dreien gibt es übrigens sowohl Prozesse auf der Betriebssystemebene (`tnslsnr, mdnsd.bin, gsnd`) als auch CRS-Ressourcen, die nur mit *crsctl* sichtbar werden, oder *crs_stat*, falls man noch Tools verwenden möchte, die ab 11.2 als veraltet gelten. Die Zeit, die man früher mit dem Einrichten der ssh-Konnektivität, dem Zusammenstottern der cluvfy-Syntax und dem Korrigieren der Kernelparameter vergeudete, nimmt einem in 11.2 zwar der Installer ab, doch benötigt der DBA diese zurückgewonnene Zeit jetzt dringend, um eine Entscheidung für oder gegen GNS zu treffen, was zunächst einmal voraussetzt, dass er verstanden haben muss, wie SCAN, MDNS und GNS miteinander zusammenhängen.

Neu in 11.2 sind je 2 Root- und Oracleagenten. Nur scheinbar Dubletten, denn ein Root- und Oracle-Agentenpaar wird vom ebenfalls neuen OHASD, dem *Oracle High Availability Services Daemon* kontrolliert, das zweite Root- und Oracle-Agentenpaar vom CRSD. In 11.2 wird ferner der Auftritt eines zukünftigen Funktionsträgers vorbereitet: der *Grid Interprocess Communication Daemo*n (GIPCD), laut Dokumentation ein Hilfsprozess, der gegenwärtig angeblich noch funktionslos (sinnlos?) sein soll.

Bei so vielen Neuzugängen von Prozessen gibt es natürlich auch Abgänge: RACG, der Health Check Monitor, vertreten durch die Prozesse racgimon und racgmain, u.a. verantwortlich für das Ausführen von Callout-Skripten, wird abgelöst durch den CRS abhängigen Oracle-Agenten. Die Rolle von OPROCD wird vom neuen CSS-Agenten übernommen. Die init-Prozesse für CSS, EVM und CRS sind ersetzt durch den OHASD-Initprozess. Der Global Service Daemon (GSD) ist in 11.2 deaktiviert.

Der Überblick
Bei dem hier untersuchten System handelt es sich um einen 3 Knoten RAC 11gR2 auf Oracle Enterprise Linux 5.4 64bit in Oracle VM. Es wurde eine Enterprise Edition installiert und ein ACFS-Dateisystem eingerichtet. Auf diesem einfach gehaltenen RAC mit Basisinstallation ergibt eine Prozesszählung, dass auf jedem einzelnen Knoten weit über 100 Prozesse gestartet werden:

```
[oracle@node1~]$ps -ef | egrep -i  \
           'init.d|grid|acfs|oks|asm|ora_|orcl'  \
           | grep -v grep |wc -l
118
```

Um einen Überblick über diese Lawine zu gewinnen, müssen die Prozesse zunächst nach Architekturschichten gruppiert werden. Wie das egrep-Kommando nahelegt, ist es sinnvoll, die vorhandenen RAC-Prozesse in die folgenden vier Gruppen zu sortieren:

1) Prozesse der Clusterware `ps -ef | egrep -i 'init.d|grid'`
   ("grid" kommt auf dem vorliegenden System im Pfadnamen von $GRID_HOME vor)
2) Treiber und Prozesse für ACFS `ps -ef | egrep -i 'acfs|oks'`
3) Prozesse der ASM-Instanz `ps -ef | grep -i asm`
4) Prozesse der Datenbankinstanz `ps -ef | egrep -i 'ora_|orcl'`
   (orcl ist hier Bestandteil des Instanznamens)

Der Start der RAC-Prozesse wird von Oracle in vier verschiedene Phasen (Level 1-4) unterteilt (Abb 1.). Als erstes (Level 1) startet der OHASD, der neue *High Availabitilty Services Daemon* die Basisprozesse der Clusterware. Zu diesen zählen der OHASD-abhängige Oracle-Agent, der OHASD-abhängige Root-Agent, und zwei der drei Prozesse, die die *Cluster Synchonization Services* (CSS) bilden: der CSSD-Agent und der CSSD-Monitor.

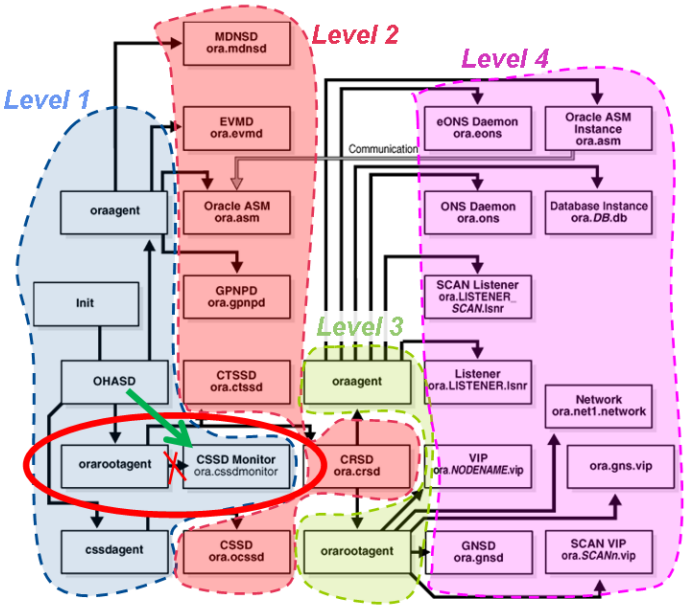

Abb. 1: Erläuternde Zeichnung aus dem Oracle Clusterware Administration and Deployment Guide 11.2 (E10717-03) mit Zuordnungsfehler (der CSSD Monitor wird direkt vom OHASD gestartet und nicht vom orarootagent). Die farblichen Ergänzungen zu den Startphasen (Level 1-4) stammen vom Autor.

In Startphase 2 werden die OHASD-abhängigen Ressourcen gestartet, zu denen der CRSD zählt. Auch die ASM-Instanz wird in dieser zweiten Phase hochgefahren. Die dritte Phase besteht aus dem Start der CRSD-abhängigen Agenten, einer unter Rootrechten, der u.a. für das Aktivieren und Deaktivieren von Netzwerkressourcen zuständig ist. Der zweite CRSD-abhängige Agent läuft mit den Rechten des Softwarebesitzers, z.B. "oracle". Dieser ist es, der schließlich die Datenbankinstanz startet. Alle Prozesse und Ressourcen die von den CRSD-abhängigen Agenten verwaltet werden, fallen in die letzte, die vierte Startphase der Clusterprozesse.

Zu den wichtigsten Quellen auf der Suche nach einem Überblick über die Clustewareprozesse zählt hierbei eine Skizze aus dem *Clusterware Administration and Deployment Guide 11.2* (Abb.1). Die Zeichnung wird gerne zur Anschauung verwendet, so in Metalink-Note 1053147.1 (*"11gR2 Clusterware and Grid Home - What You Need to Know"*) oder in der Vorstellung der 11gR2 Grid Infrastruktur in den DOAG News Q1 2010 (Seite 56, *"Grid Infrastructure 11gR2 – eine Grid-Infrastruktur für alle Fälle"*, Markus Michalewicz) oder auch im einzigen zu diesem Zeitpunkt (4.9.2010) verfügbaren Buch zum Thema RAC 11gR2 (Prusinski *et al.*, "*Oracle 11g R1/R2 Real Application Clusters Handbook*", Packt-Publishing 2010, Seite 624). Zu den wichtigsten Quellen zählt diese Zeichnung deshalb, weil sie zu einer der wenigen autoritativen Überblickszeichnungen gehört, in der die Prozessbeziehungen beschrieben werden. So kann man hier z.B. erkennen, dass der *Event Manager* vom OHASD-abhängigen Oracleagenten gestartet wird, der *Oracle Notification Service* dagegen vom CRSD-abhängigen Oracleagenten. Schön, dass es diese Skizze gibt. Schade, dass sie nicht fehlerlos ist: der CSSD-Monitor wurde fälschlich mit dem OHASD-abhängigen orarootagent verdrahtet, in Wirklichkeit wird er direkt vom OHASD gestartet. Dies zeigt ein Blick auf den Prozessbaum:

```
[root@node1 ~]# pstree -U -G -p -u -n
          ...
          +-init.ohasd(3197)
          +-ohasd.bin(3272)---{ohasd.bin}(3282)
          ...
          +-oraagent.bin(3395,oracle)---{oraagent.bin}(3397)
          ...
          +-gipcd.bin(3408,oracle)---{gipcd.bin}(3417)
          ...
          +-cssdmonitor(3438)---{cssdmonitor}(3440)
          ...
          +-orarootagent.bi(3460)---{orarootagent.bi}(3478)
          ...
```

Der orarootagent startet erst nach dem CSSD-Montitor (und nach GIPCD). In der Textbeschreibung zu dieser Abbildung http://download.oracle.com/docs/cd/E11882_01/rac.112/e10717/img_text/cwadd004.htm steht bis heute (Stand 4.9.2010) darüber hinaus fälschlich, dass auch der GIPCD vom orarootagent gestartet werden soll, was ebenfalls nicht möglich sein kann (in der Dokumentation, die man sich offline herunterlädt, ist zumindest der zweite Fehler bereinigt, der begleitende Text in Metalink-Note 1053147.1 ist korrekt).

Welche Abhängigkeiten bestehen zwischen diesen Prozessen, wann ist die Startreihenfolge kritisch, wann spielt sie eine untergeordnete Rolle? Mit Hilfe der Oracle Dokumentation, den Linux-Kommandos `ps` / `pstree`, dem Cluster Control Utility `crsctl` und der Ausgabe in Logdateien wurde versucht, alle Prozesse in eine Übersichtsskizze zu zwängen. Der zweifelhafte Erfolg eines dieser Versuche kann in Abb. 2 begutachtet werden. Klar wird zumindest, dass der Komplexitätsgrad eines RAC 11gR2 das Auffassungsvermögen eines menschlichen Durchschnittsgehirns überfordert.

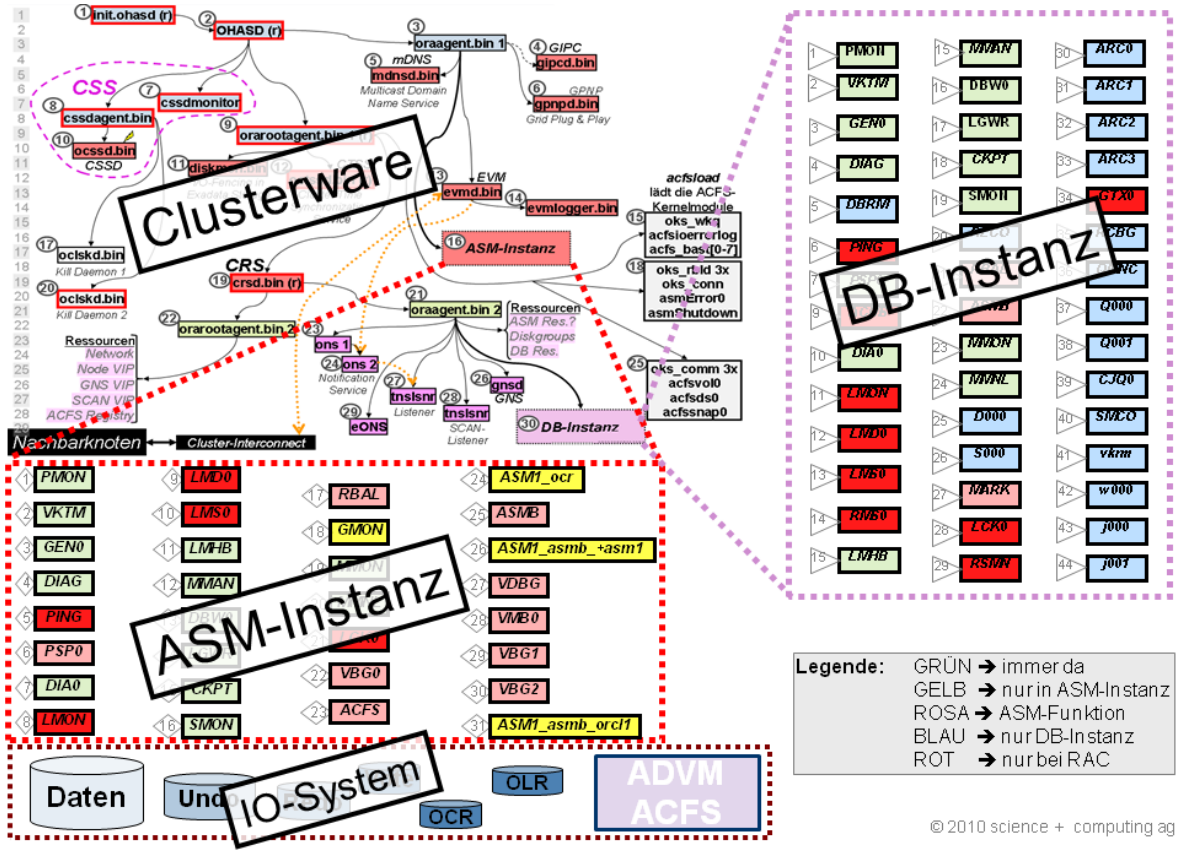

Abb. 2:RAC Prozessübersicht - einer von zahlreichen Versuchen, alle Prozesse eines RAC in eine Folie zu zwängen. Ohne Wechselwirkung zwischen den Prozessen. Der Komplexitätsgrad eines 11gR2 RAC überfordert das Auffassungsvermögen eines menschlichen Durchschnittsgehirns.

So unübersichtlich wie die Evolution der RAC-Prozesse gestaltet sich die Sammlung der aktuellen Werkzeuge: über 20 Administrationsprogramme stehen dem RAC-Administrator zur Verwaltung und Diagnose zur Verfügung, für den Alltag benötigt er jedoch in der Regel nur wenige. Zu den wichtigsten zählen `srvctl` und `crsctl`. Werkzeuge von einst wie `racgons` spielen keine Rolle mehr, während andere mächtige Werkzeuge wie `asmca` in 11.2 neu hinzugekommen sind.

Wem das hier zu kompliziert ist, dem empfiehlt sich *Grid Control*. Doch auch hier darf der DBA nicht verfrüht von der Leichtigkeit des Seins träumen – er sollte genügend Zeit und reichlich Hardware-Ressourcen einplanen. Glücklich darf sich derjenige schätzen, bei dem die *Grid Control* Installation schmerzfrei gelingt. Für alle anderen gibt es das Template für Oracle VM. Der Weg zur einfachen Leichtigkeit des RAC ist steinig. So oder so.

Kontaktadresse:
Dr. Thorsten W. Grebe
twg-it
Geisenheimer Straße 6
D-14197 Berlin

Telefon: +49 (0) 176 31403337
E-Mail thorsten.grebe@twg-it.de
Internet: http://twg-it.de